Volumetric analysis of HeLa cancer cells imaged with Serial Block Face Scanning Electron Microscopy

Cefa Karabağ

Supervisor: Constantino Carlos Reyes-Aldasoro

Department of Electrical and Electronic Engineering

School of Mathematics, Computer Science and Engineering

City, University of London

This dissertation is submitted for the degree of

Doctor of Philosophy

MMXX

Fig. 1 An illustration of semantic segmentation. (a) **Maya Karabağ** is playing with her soft toy on the carpet. (b) Semantic segmentation of the same image. All pixels were classified as one of the three possible outcomes - Maya, her toy, and the carpet.

I would like to dedicate this dissertation to my daughter

Maya

who has been giving me joy since her birth.

Lbmcjn!Nbzb(zb

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. I hereby grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Cefa Karabağ
MMXX

Acknowledgements

First and foremost, I am extremely grateful for the tremendous support from my supervisor, **Dr Constantino Carlos Reyes-Aldasoro**, who not only has incredible foresight and views of the big picture, but also guided me through numerous technical details throughout my research. I benefited a great deal from his expertise in image processing and computer programming, his positive approach, and constructive criticisms during my years at **City, University of London**. It has been a great pleasure and a privilege to work with him, to learn from him and to be inspired by him. I can confidently confirm that whatever I know about Image Analysis, I learned it from him.

Secondly, the collaboration with **The Francis Crick Institute** has given me the opportunity to interact with interdisciplinary scientists and I am especially thankful to **Martin L. Jones** and **Lucy M. Collinson** for the provision of the data sets and their invaluable advice and support in a biological context.

Next, I would like to thank **José Alonso Solís-Lemus** for his invaluable support throughout my years working at the centre. I would also like to acknowledge the support from the School of Mathematics, Computer Science and Engineering which enabled this research through the award of a tuition fee waive. It would not be possible to finish my work and this dissertation without it.

Finally, from the bottom of my heart, thanks to my wife, **Özgü**, and my daughter, **Maya**, who made the home a happy place to finish my work as their love and generosity have inspired me. Their daily encouragement and support in all matters make it all worthwhile.

Abstract

This dissertation investigates the volumetric analysis of a variety of cervical cancer cells called HeLa cells. HeLa cells were derived from cervical cancer cells taken from Henrietta Lacks at the Johns Hopkins Hospital and hence the name HeLa remains. The shape of cells is important as the regular or irregular shape of the cell and its structures can be related to some conditions of health or disease.

In this dissertation, a traditional image processing algorithm to segment the nuclear envelope of HeLa cells imaged with Serial Block Face Scanning Electron Microscopy is proposed. The algorithm is fast, robust and accurate and it was compared against different deep learning architectures. Three deep learning architectures were deployed through transfer learning and U-Net was trained from scratch for semantic segmentation of HeLa cells. The algorithm outperformed all four deep learning architectures and active contours (snakes) in both accuracy and time as suggested by the similarity metrics. The segmented nuclear envelope was further investigated through a visualisation technique to obtain a graphical model. This model provides volume and surface metrics which can be used to compare different cells. Wild-type of HeLa cells were compared against Chlamydia trachomatis-infected HeLa cells and geometric differences were revealed.

The open-source image processing algorithm, developed in programming environment of Matlab® (The Mathworks™, Natick, USA), provides cell segmentation in a fraction of manual segmentation time therefore it is an alternative to expensive commercial software and manual segmentation, which is still widely used despite the significant disadvantages of time and inter- and intra-user variability.

Publications

Journal publications

[**1**] **Karabağ, C.**, Jones, M.L., Peddie, C.J., Weston, A.E., Collinson, L.M., and Reyes-Aldasoro, C.C. (2020). Semantic segmentation of HeLa cells: An objective comparison between one traditional algorithm and four deep-learning architectures. PLoS ONE 15(10): e0230605; doi.org/10.1371/journal.pone.0230605.

[**2**] Ortega-Ruiz, M.A., **Karabağ, C.**, Garduño, V.G., and Reyes-Aldasoro, C.C. (2020). Morphological Estimation of Cellularity on Neo-Adjuvant Treated Breast Cancer Histological Images. Journal of Imaging 2020, 6(10), 101; doi.org/10.3390/jimaging6100101.

[**3**] **Karabağ, C.**, Jones, M.L., Peddie, C.J., Weston, A.E., Collinson, L.M., and Reyes-Aldasoro, C.C. (2019). Segmentation and Modelling of the Nuclear Envelope of HeLa Cells Imaged with Serial Block Face Scanning Electron Microscopy. Journal of Imaging 2019, 5(9), 75; doi:10.3390/jimaging5090075.

[**4**] **Karabağ, C.**, Verhoeven, J., Miller, N.R., and Reyes-Aldasoro, C.C. (2019). Texture Segmentation: An Objective Comparison between Five Traditional Algorithms and a Deep-Learning U-Net Architecture. Applied Sciences 2019, 9(18), 3900; doi:10.3390/app9183900.

Conference publications

[**1**] **Karabağ, C.**, Jones, M.L., Peddie, C.J., Weston, A.E., Collinson, L.M., and Reyes-Aldasoro, C.C. (2019). Segmentation and Modelling of HeLa Nuclear Envelope. In Carbayo, M.J.L. and Ballester, M.A.G., editors, IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), page 1510–1513. IEEE and IEEE Xplore.

[2] Karabağ, C., Jones, M.L., Peddie, C.J., Weston, A.E., Collinson, L.M., and Reyes-Aldasoro, C.C. (2018). Segmentation and Modelling of HeLa Nuclear Envelope. The British Association for Cancer Research (BACR) Conference at The Francis Crick Institute.

[3] Karabağ, C., Jones, M.L., Peddie, C.J., Weston, A.E., Collinson, L.M., and Reyes-Aldasoro, C.C. (2018). Automated Segmentation of HeLa Nuclear Envelope from Electron Microscopy Images. In Nixon, M., Mahmoodi, S., and Zwiggelaar, R., editors, Medical Image Understanding and Analysis (MIUA 2018), Communications in Computer and Information Science, page 241–250. Springer International Publishing.

bioRxiv publications

[1] Cefa Karabağ, Martin L. Jones, Christopher J. Peddie, Anne E. Weston, Lucy M. Collinson, Constantino Carlos Reyes-Aldasoro (2020). Semantic Segmentation of HeLa Cells: An Objective Comparison between a Traditional Algorithm and Three Deep-Learning Architectures. bioRxiv 2020.03.05.978478; doi:10.1101/2020.03.05.978478.

[2] Cefa Karabağ, Martin L. Jones, Christopher J. Peddie, Anne E. Weston, Lucy M. Collinson, Constantino Carlos Reyes-Aldasoro (2020). Geometric differences between nuclear envelopes of Wild-type and Chlamydia trachomatis-infected HeLa cells. bioRxiv 2020.04.02.021733; doi:10.1101/2020.04.02.021733.

[3] Cefa Karabağ, Martin L. Jones, Christopher J. Peddie, Anne E. Weston, Lucy M. Collinson, Constantino Carlos Reyes-Aldasoro (2018). Modelling the nuclear envelope of HeLa cells. bioRxiv 344986; doi:10.1101/344986.

[4] Mauricio Alberto Ortega-Ruiz, **Cefa Karabağ**, Victor García Garduño, Constantino Carlos Reyes-Aldasoro (2020). Morphological estimation of Cellularity on Neo-adjuvant treated breast cancer histological images. bioRxiv 2020.04.01.020719; doi:10.1101/2020.04.01.020719.

[5] Mauricio Alberto Ortega-Ruiz, **Cefa Karabağ**, Victor García Garduño, Constantino Carlos Reyes-Aldasoro (2020). Estimation of cellularity in tumours treated with Neoadjuvant therapy: A comparison of Machine Learning algorithms. bioRxiv 2020.04.09.034348; doi:10.1101/2020.04.09.034348.

Table of contents

List of figures

List of tables

List of acronyms

Other Symbols

$\circledast$	Convolution operator
Δ direction	Change of direction
$\langle \cdot, \cdot \rangle$	Scalar product operator
$\mathbb{R}$	Set of real numbers
$\mathbf{x}, \mathbf{v}, \mathbf{P}$	Vectors
$\mathcal{B}$	Set of points describing the boundary of an object
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution (Gaussian) with a mean μ and variance σ^2
$\mathbb{K}, \mathbb{S}$	Matrices representing kernels or 2D filters
∂_x, ∂_y	Partial derivatives
$c(s)$	Parametric curve with parameter s
N_e	Number of columns in an image
N_r	Number of rows in an image
px, qx	Representing the value of at image at a specific position
$R(\theta)$	Rotation matrix, rotating by angle θ
s, p	Parameters in 2D curves
t	Referring to time, iterations or time frame in image sequences
x, y	Discrete variables used to address positions in an image
$\mathbb{I}$	Matrix representing an image

Acronyms / Abbreviations

AI	Artificial Intelligence
ATCC	American Tissue Culture Collection
BMVC	British Machine Vision Conference
CNN	Convolutional Neural Network

DL	Deep Learning
DSMZ	Deutsche Sammlung von Mikroorganismen und Zellkulturen
ECACC	European Collection of Cell cultures
EMBL	European Molecular Biology Laboratory
EM	Electron Microscopy
EPFL	Ecole Polytechnique F´edrale de Lausanne
ER	Endoplasmic reticulum
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
GT	Ground Truth
GUI	Graphical User Interface
HD	Hausdorff Distance
IEEE	Institute of Electrical and Electronic Engineers
ISBI	IEEE Symposium on Biomedical Imaging
JCRB	Japanese Collection of Research Bioresources
JI	Jaccard Index
MATLAB	Matrix Laboratory - Numerical computing environment and programming language developed by MathWorks
ML	Machine Learning
NCMIR	National Centre for Microscopy and Imaging Research
NE	Nuclear Envelope
RGB	Red, Green and Blue images
ROI	Region of Interest
SBF SEM	Serial Block Face Scanning Electron Microscopy
SLIC	Simple Linear Iterative Clustering
TIFF	Tagged Image File Format
TN	True Negative
TP	True Positive

Chapter 1

Introduction and background

1.1   Overview

The Biomedical Image Analysis has its four quadrant namely physics, medical imaging, machine learning and image processing, and graphics [1] and it is a branch of Biomedical Engineering where computational algorithms are developed to analyse images of different biological nature and context [2]. Observation of biological processes through microscopy has allowed for "large-scale, systematic and automated studies", as stated by Rittscher et al. [3]. For example, experiments where the observation of cellular processes in genetically modified organisms produces several images per second. Manual assessment of such experiments is a difficult and time-consuming task, for which an automated or semi-automated technique could obtain useful information from the data at a faster rate and ultimately more reliable as the manual assessment is error-prone and inter- and intra-observer variability. To extract information from an image, a collection of computational, mathematical and statistical procedures can be performed on the images, and the information obtained can aid tasks such as cell detection, segmentation, and classification.

Cell segmentation and classification have been important, and challenging problems for many years [4–7]. They have attracted considerable attention, both in clinical practice and computing research [8], as the identification of individual cells, and the

shape of the cell and its parts, like the nuclear envelope (NE), may reveal some conditions of health or disease [9].

The study of cells helps us understand how cells work in healthy and diseased states. Cell biologists working in animal, plant and medical science will be able to develop new vaccines, more effective medicines, plants with improved qualities and through increased knowledge a better understanding of how all living things live. Eventually it will be possible to produce a 'health forecast' by analysing our database of genetic and cell information [10]. Using this database we will be able to take more control over our health in a preventive way.

The work documented in this dissertation contributes towards (i) segmenting the nuclear envelope (NE) of HeLa cells - Chapter 4, Chapter 6 [11] (ii) modelling the NE against a spheroid to provide statistical measurements of HeLa cells in relation to the shape of the cells Chapter 4, Chapter 6 [11] and (iii) automating the task through computer algorithms Chapter 4, Chapter 5, and Chapter 6 [11, 12]. The novel computational algorithms described in this dissertation extract information from the electron microscopy (EM) images and provide statistical analysis of its results.

1.2 Motivation

Cervical cancer is one of the leading causes of death in females [13], and together with breast cancer, it contributes to 4.2% of the global causes of death [14]. In the United Kingdom, around 3,000 women are diagnosed with cervical cancer every year, and around 1,000 of those die annually [15]. The substantial levels of ongoing morbidity and mortality have led to heightened interest in new methods to identify cell abnormalities. Consequently, clinicians are interested in early detection of cell abnormalities to effectively predict and control possible cancer events in the future. Cancer cell lines and tissue culture studies have been used extensively to understand tumour biology and facilitate drug discovery processes [16–18]. The availability of cell lines together with genetic advances has contributed to cancer research, which has

grown significantly as a proportion of all research in biomedical areas [19]. Perhaps the most important cell line has been the cervical cancer *HeLa* cells, derived in 1951 [20, 21] and widely used as model for thousands of biological experiments [22, 23].

The NE is the membrane that surrounds the chromosomes and partitions them from the rest of the cell contents, thereby forming a protected environment for the genetic material. Almost every cell (red blood cells being a notable exception) has a nucleus bounded by a nuclear envelope, and so it is a primary target for segmentation as a reference structure for visualising and spatial positioning of other cell organelles. The NE is an important structure in cervical cancer cells as they undergo mitosis, breaking down to allow the cell to duplicate and split before reforming around each daughter nucleus [24, 25]. This importance is also underlined in cancerous cells [21, 20, 22] where certain components of the nuclear envelope take central roles in cell functions that affect tumour development and progression. Visualising and investigating the NE requires its accurate segmentation [4, 7, 26, 27] from a set of serial images of the cell obtained through electron microscopy (EM).

Segmentation of cells and organelles observed with electron micrographs is in many cases still a manual process [28–30], despite the significant disadvantages of time and inter- and intra-user variability and intensifying efforts by the community to automate the process. Manual segmentation (Figure 2.9), which is still considered as the gold standard, requires a great amount of expert knowledge and training and it is very tedious and time consuming [31]. Hence, an automated system is required to help biomedical researchers in fast and reliable segmentation of the NE. Once the segmentation is achieved, a 3D reconstruction of the NE can be created and interpreted to study its biological characteristics and relationships to other structures in the cell.

Problem statement. The HeLa cells imaged with Serial Block Face Scanning Electron Microscopy (SBF SEM) present additional challenges for automated analysis that is not necessarily shared with other problems widely studied in computer vision. For example, (i) the nuclear envelope of HeLa cells might have several disjoint regions when the cells break down during mitosis, (ii) Visualising and investigating the NE

requires its accurate segmentation from a set of serial images of the cell obtained through electron microscopy (EM). The human vision can often overcome some of these challenges, however, a computational analysis would require complex methods to analyse each slice. The techniques developed for this purpose would provide robust and repeatable analysis to image sequences of HeLa cells and Chlamydia trachomatis-infected HeLa cells, aiding in the confirmation of the hypotheses made from visual inspection.

1.3 Aims and objectives

The aim of this work is the volumetric analysis of cervical cancer cells or HeLa cells observed with Electron Microscopy (EM) images. This is important in both biological and computing research, as the identification of individual cells, and the shape of the cell and its parts, like the nuclear envelope (NE), may reveal some conditions of health or disease. This work provides an accurate cell segmentation in a fraction of the manual segmentation time. The open-source algorithm developed in this work provides an alternative to expensive commercial software and manual segmentation.

For this purpose, firstly, a semantic segmentation of HeLa cells is performed. Initially, the NE of HeLa cells is segmented. To achieve this aim, we develop an automated and unsupervised image processing algorithm for the segmentation of the NE of Wild-type of HeLa cells imaged with imaged with Serial Block Face Scanning Electron Microscopy (SBF SEM).

Furthermore, we perform segmentation of different type of cell images such as Chlamydia trachomatis-infected HeLa cells. We apply the segmentation algorithm onto Chlamydia trachomatis-infected HeLa cells and compare the results with that of Wild-type of HeLa cells. Moreover, the geometrical comparison is presented so differences between nuclear envelopes of Chlamydia trachomatis-infected HeLa cells and Wild type of HeLa cells can be explained.

Finally, deployment of deep learning architectures for semantic segmentation of HeLa cells is explored. The objective comparison between traditional image processing algorithm and deep learning architectures is presented.

All algorithms are assessed by using different metrics i.e. Jaccard similarity index, Hausdorff distance, and accuracy.

The objectives for this dissertation can be defined as follows:

1. To perform semantic segmentation of HeLa cervical cancer cells imaged with Serial Block Face Scanning Electron Microscopy (SBF SEM). The segmentation was performed with a proposed algorithm, which was compared against four other algorithms through Transfer Learning.

2. To obtain a graphical model of the nuclear envelope (NE) of HeLa cells i.e. to model the NE against a spheroid in order to extract some metrics that may be relevant to characterise the biological nature of the cells.

3. To provide fast and accurate segmentation of different type of HeLa cells in a fraction of manual segmentation time so a comparison could be done between Wild-type of HeLa cells and Chlamydia trachomatis-infected HeLa cells to reveal their geometrical differences.

1.4   Contributions

The original contributions of this dissertation are:

1. An image processing algorithm for the segmentation of the nuclear envelope (NE) of cervical cancer cells or HeLa cells including disjoint regions. The algorithm could work for any electron microscopy (EM) images (Chapter 4 and Chapter 6) such as Chlamydia trachomatis-infected HeLa cells [32]. The tool can be used to create a 3D volume of the NE and segmentation results can be used as training data for Deep Learning architectures. Modelling the NE against a

spheroid provides a graphical model of the NE and some volume and surface metrics that can be used to compare different types of HeLa cells and to explain their geometrical differences.

2. Development and testing of an image processing algorithm distributed as a set of routines under an open source license. Such algorithm would directly address the aims and objectives described in the previous section.

3. Comparison of the algorithm against four Deep Learning architectures in order perform semantic segmentation of the nuclear envelope of HeLa cells.

1.5   Organisation of the dissertation

An overview of the dissertation is represented graphically in Figure 1.1, and subdivided in the chapters described below.
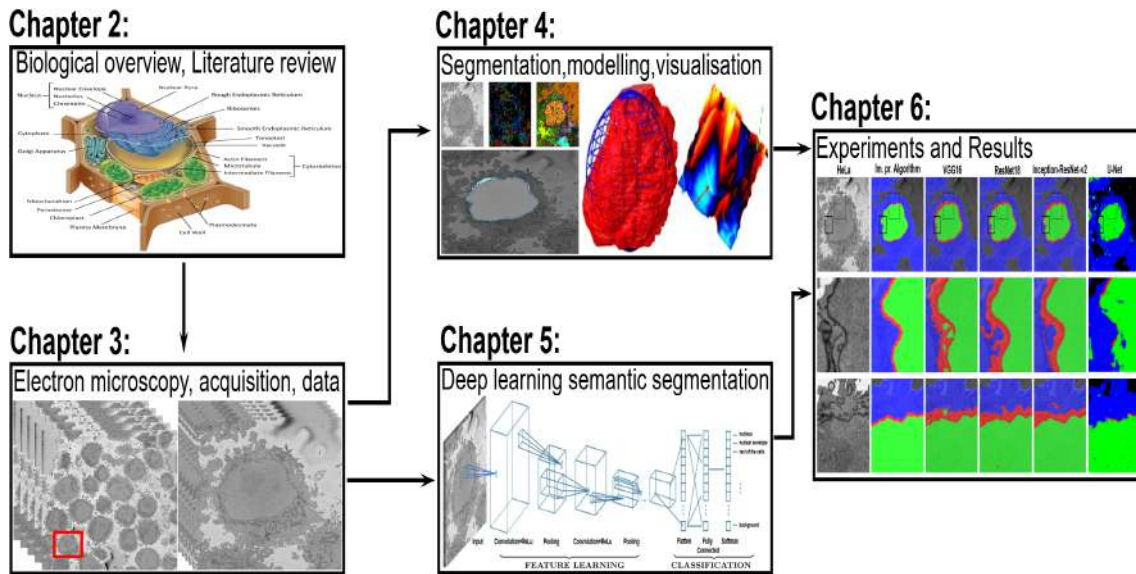
Fig. 1.1 Graphical outline of the dissertation. Chapter 2 provides the context for HeLa cells from a biological and mathematical point of view. Chapter 3 represents a thorough exploration of the data. Chapters 4 and 5 provide the major contributions of this work, the results are presented in Chapter 6.

The initial four chapters address the background and the relevant literature for this research, whereas Chapter 5 and Chapter 6 explain the implementation details and

relevant results for this work. Moreover, the limitations and possible future extensions for this work are discussed in the last chapter. For a greater appreciation of this dissertation, a brief description for individual chapters is presented below.

Chapter 1 introduces the basic theme of this work with a short introduction followed by the aims and objectives of the dissertation. Moreover, the importance of the research problem is highlighted with the help of disease incidence statistics in this chapter.

Chapter 2 subsequently defines the biological context of this work for the general audience. Starting with cell biology and mitosis, we explain the cancer diseases with a specific emphasis on cervical cancer and associated HeLa cells. This is followed by the importance of HeLa cells and an overview of cross-contamination of HeLa cells.

Chapter 3 deals with electron microscopy, its working principles, and medical imaging techniques. We conclude this chapter with a focus on Serial Block Face Scanning Electron Microscopy (SBF SEM) acquisition and Cryo-electron Microscopy. At the end of this chapter we explain the acquisition of both Wild-type and Chlamydia trachomatis-infected HeLa cell images through SBF SEM.

Chapter 4 addresses the basic problem of image processing and image segmentation. We start with an overview of simple edge and region-based methodologies and provide a literature based detailed review. The basic principles will be explained along with its mathematical derivation. In the next section we extend our discussion to active contours segmentation models.

Chapter 5 delves into Artificial Intelligence, Machine Learning and specifically Deep Learning and Transfer Learning. Convolutional Neural Networks (CNNs) will be investigated in detail and finally three most accurate pre-trained Deep Learning architectures will be discussed for the segmentation of the NE of HeLa cells used in this work.

Chapter 6 presents our results we have obtained so far and discusses the efficiency of the image processing segmentation algorithm as well as pre-trained Deep Learning architectures through an objective comparison. starts with the details of data used in this

work, followed by the ground truth construction of the NE of HeLa cells for comparative purposes. In the subsequent section, we explain the automatic segmentation algorithm developed for this work. In the next section, we present quantitative comparisons between segmentation of the algorithm and the manual segmentation. Application of active contours on HeLa cells is given in the following section. This application of three pre-trained Deep Learning architectures on HeLa cells and performance of semantic segmentation results are presented. We conclude this chapter with the nuclear envelope shape modelling of HeLa cells against a spheroid and present quantitative metrics extracted from this modelling. We present both visual and statistical results in context of the NE segmentation.

Chapter 7 concludes our work and explains some of the limitations of this work. Moreover, we discuss some of the future directions in which this work can be extended.

1.6   Summary

The study of cells and their organelles have interested scientists from the early days of Hooke and van Leeuwenhoek to the formulation of cell theory by Schleiden and Schwann [33]. Since then, cells and their environment; its presence or absence, shapes, inner components, interactions, regulation of processes, and many other characteristics have been thoroughly analysed, especially trying to relate these to conditions of health and disease [34–39]. To observe cells, it is necessary to use microscopy and one of its many different techniques like light, fluorescent or differential interference microscopy. Electron Microscopy (EM) can provide resolving power several orders of magnitude higher than conventional light and fluorescence microscopes and thus it is ideal to observe small structures of the cellular environment. Modern EM settings allow the acquisition of contiguous slices of a sample by slicing very thin sections of the top face with an ultramicrotome with a diamond knife [40]. Once the sample is sliced, the slice itself is discarded, the sample is raised and the scanning process continues for a given

number of slices, thus creating a three-dimensional data set of contiguous slices. This process is called *Serial blockface scanning EM* (SBF SEM) [41].

The nuclear envelope (NE) is a bi-layer membrane separates the nucleus with the chromosomes from the rest of the cellular compartments [42] and contains a large number of membrane proteins with sophisticated roles and functions [43, 42, 44, 45]. The structure and condition of the NE is of huge importance as it has been related to viral infections [46–50], Muscular dystrophy [51], Cancer [52–56], Osteoporosis [57], Cardiovascular diseases [58–60], other diseases [61–63], and ageing [64–66]. Therefore, algorithms for the segmentation, visualisation and analysis of the NE could provide parameters to understand the conditions of health and disease of a cell.

The segmentation and classification of cells and their environment through image-processing tasks have been important for many years and numerous algorithms have been proposed. Candia *et al.* summarised the importance of objective analysis emphatically in the following way: "we need unbiased, mathematically robust, scalable methods that allow us to identify key parameters that consistently characterise cell sub-populations ... to build signatures of health and disease" [35]. PubMed [67] contains more than $33,000$ entries with the words cell and classification or segmentation in the title and abstract ((classification[Title/Abstract] OR segmentation[Title/Abstract]) AND cell[Title/Abstract]). The number of entries drops considerably to less than 1000 when the keyword "electron" is added to the search. Segmentation and classification of images acquired with electron microscopy is difficult for several reasons. The considerable increase of size and resolution as compared with light and fluorescence microscopy provides complex morphological structures. Whilst fluorescence microscopy allows several channels that identify structures of interest, EM only provides a grey scale image and with a reduced contrast between the structures of interest and the background. Furthermore, when serial sections are obtained, the images are transformed into a volumetric data set.

Recently, advances in machine learning and artificial intelligence, especially those related to deep learning algorithms [68], have revolutionised image processing tasks

[69–74]. Several deep learning algorithms [75–77] have obtained outstanding results in difficult tasks such as those of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [78]. Not surprisingly, deep learning has become a popular tool for segmentation and classification. Convolution neural networks (CNN) [79], are versatile and have been shown to be very effective for a wide range of tasks including object detection [80, 81] and image classification [82–86] andsegmentation [87]. The U-Net architecture, proposed by Ronneberger [88] has become a widely used tool for segmentation and analysis. It recently became the most cited paper presented in the prestigious MIC-CAI conference. Ciresan [89] et al. applied Deep neural networks (DNN) to detect membrane neuronal and mitosis detection in breast cancer [90].

Numerous researches have deployed deep learning, that is trained end-to-end and does not rely on prior knowledge of the data, to study the structure and different organelles of cells imaged with electron microscopy [91–97].

Deep learning algorithms have two main limitations: 1) they require a large amount of training data and 2) they require significant computational power. As graphics processing units (GPUs) become more popular, the main limitation is thus the scarcity of training data [98, 99, 83, 100, 101].

Deep learning architectures were used to validate or improve analytical results of segmentation provided by the image processing algorithm. Moreover, in recent times in computer science community, Artificial Intelligence (AI) inspired technology has been used to to tackle some of difficult problems in many areas, among them those related to healthcare and medical imaging [102, 103]. Thus, in this dissertation we investigate the performance of four different Convolutional Neural Network (CNN) models to assess the classification of HeLa cells into four classes: nuclear envelope, nucleus, rest of the cell, and background.

In this work we trained the Deeplab v3+ network [104], one type of convolutional neural network (CNN) designed for semantic image segmentation, to perform semantic segmentation of HeLa cells imaged with SBF SEM. U-Net [88] was trained from scratch perform the semantic segmentation of HeLa cells.

The algorithms were pre-trained with ImageNet and then fine-tuned with training data prepared for this work. These were then compared with a traditional image processing algorithm [11]. The image processing algorithm follows a pipeline of traditional tasks: low-pass filtering, edge detection, dilation, generation of super-pixels, distance transforms, mathematical morphology and post-processing to segment automatically the nuclear envelope and background of HeLa cells. For each architecture, the results were compared against the ground truth with four classes and then the following metrics were calculated: accuracy (labels correctly classified) and Jaccard similarity index.

The results of the four algorithms were objectively compared against a ground truth that was formed by a manually segmented nuclear envelope and an automatically segmented background. All the programming was performed in Matlab® (The Mathworks™, Natick, USA) and the codes are freely available through GitHub and the data sets through EMPIAR and Zenodo repositories:

https://github.com/reyesaldasoro/Hela-Cell-Segmentation,

https://github.com/reyesaldasoro/HeLaSegmentationUNET,

https://github.com/karabagcefa/Hela-Cell-Semantic-Segmentation,

EMPIAR:http://dx.doi.org/10.6019/EMPIAR-10094,

Zenodo:https://doi.org/10.5281/zenodo.3834608.

The key research presented in this work is the investigations done on the HeLa cell images acquired by the **Electron Microscopy Science Technology Platform (EM STP) at The Francis Crick Institute** in London [40], where various results were found. Later investigations were focused on Chlamydia trachomatis-infected HeLa cells obtained from **The Cell Image Library - CIL**50051 **and CIL**50061.

Chapter 2

Cell and Cancer Biology

2.1 Introduction

The aim of this dissertation is to develop an automatic and unsupervised image-processing algorithm to segment the nuclear envelope (NE) of cancerous HeLa cells imaged with Serial Block Face Scanning Electron Microscopy (SBF SEM) and to compare against deep learning architectures.

This chapter gives the necessary knowledge to understand the biological content of the project. It starts with cell and its structure, explains cell division called mitosis and finally presents cancer cells in general and narrows it down to cancerous HeLa cells. The common usage of HeLa cells around the world and cross-contamination of HeLa cells will also be discussed. The chapter ends with a summary.

2.2 Cell

All living creatures are made of cells, which are the structural, functional and fundamental units of life with the extraordinary ability to create copies of themselves by subdivision [25, 24]. Most human cells fall within a size range of 2-120 microns [105] and they show signs of life: particles move around inside them, cells slowly change their shapes and interact with their environment, and some cells divide into
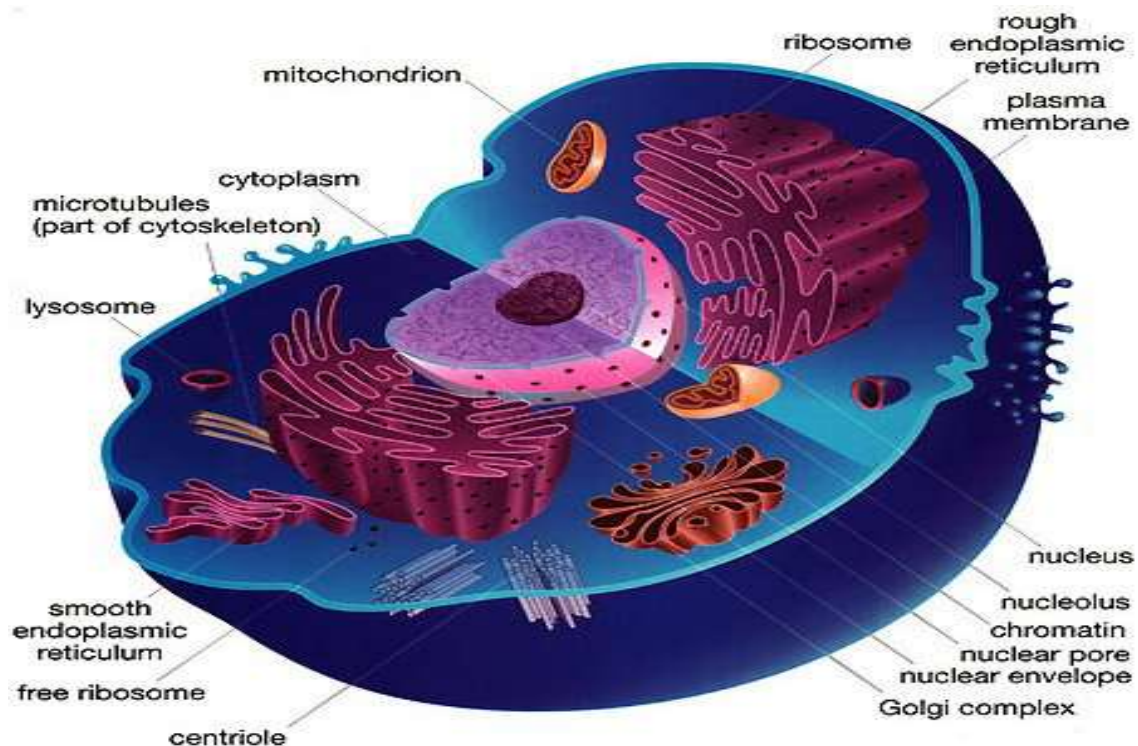
Fig. 2.1 Schematic diagram of internal membranes and the cytosol. Fine structure of a cell illustrates internal membranes such as nucleus, nuclear envelope, mitochondrion, endoplasmic reticulum, Golgi apparatus or complex, ribosome, and lysosomes. Each organelle is specialised to perform a different function. The rest of the cell, excluding all organelles, is called the cytosol, and is the site of many vital cellular activities. Images adapted from [24].

two. Figure 2.1 illustrates the schematic diagram of fine structure of a cell showing internal membranes such as nucleus, nuclear envelope, mitochondrion, endoplasmic reticulum, Golgi apparatus or complex, ribosome, and lyso-somes. Each organelle is specialised to perform a different function. The rest of the cell, excluding all organelles, is called the cytosol, and is the site of many vital cellular activities. The presence or absence of a nucleus is used as the basis for a simple but fundamental classification of all living organisms. Those whose cells have a nucleus are called eukaryotes and those whose cells do not have a nucleus are called prokaryotes. The nucleus and nuclear envelope (NE) of an eukaryotic cell imaged with electron microscope are shown in Figure 2.3. The magnified version of this structure is illustrated in Figure 2.2.
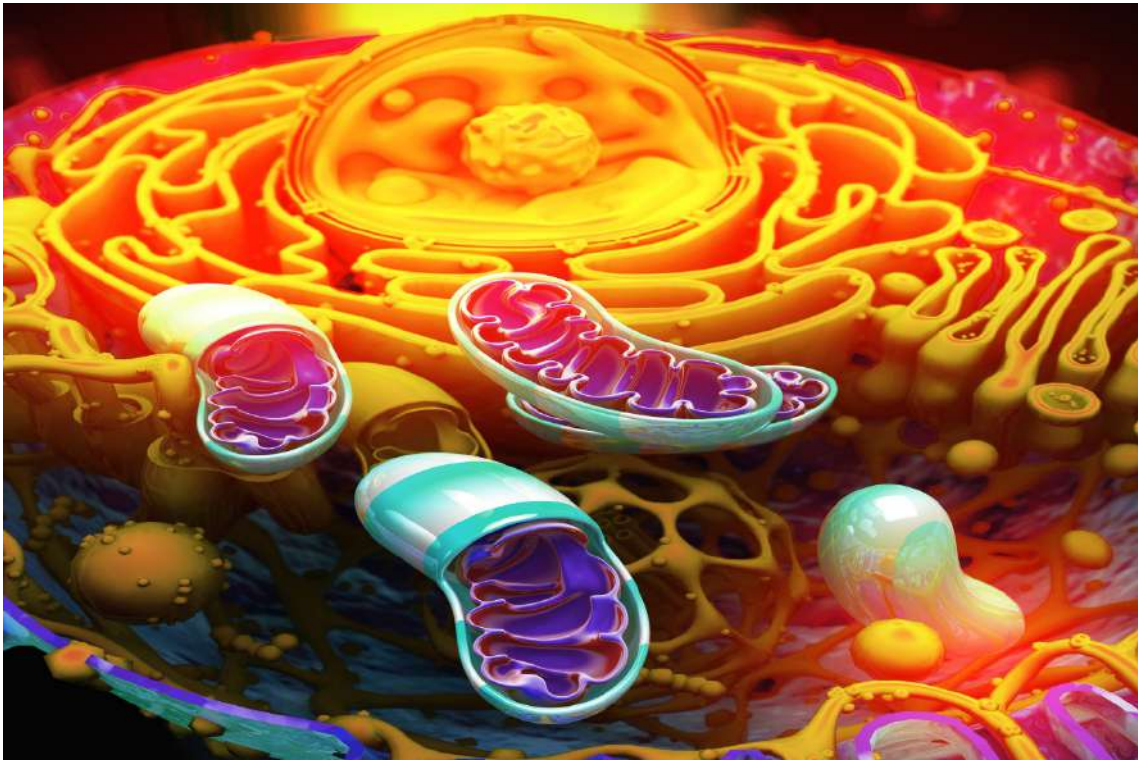
Fig. 2.2 Illustration of the nucleus, a large round body in the middle, and enclosed within two concentric membranes that form the nuclear envelope (NE), contains most of the DNA of the eukaryotic cell and other organelles. Image credit to Ella Marushchenko [106].

The cell has a sharply defined boundary, formed by an enclosing plasma membrane - a fatty film so thin and transparent that it cannot be seen in the light microscope. The membrane is semi-permeable and it has channels and pumps through which molecules can cross the membrane. All cell membranes are composed of lipids and proteins. Millions of lipid molecules are arranged in two closely opposed sheets, forming a lipid bilayer which has been firmly established as the universal basis of cell-membrane structure. A large round body called the nucleus, shown in Figure 2.1, Figure 2.2, Figure 2.3, andFigure 2.4 is the most prominent organelle in a cell. The nucleus is enclosed within two concentric membranes that form the nuclear envelope (NE), and it contains molecules of DNA-extremely long polymers that encode the genetic specification of the organism [24].

(a)   (b)
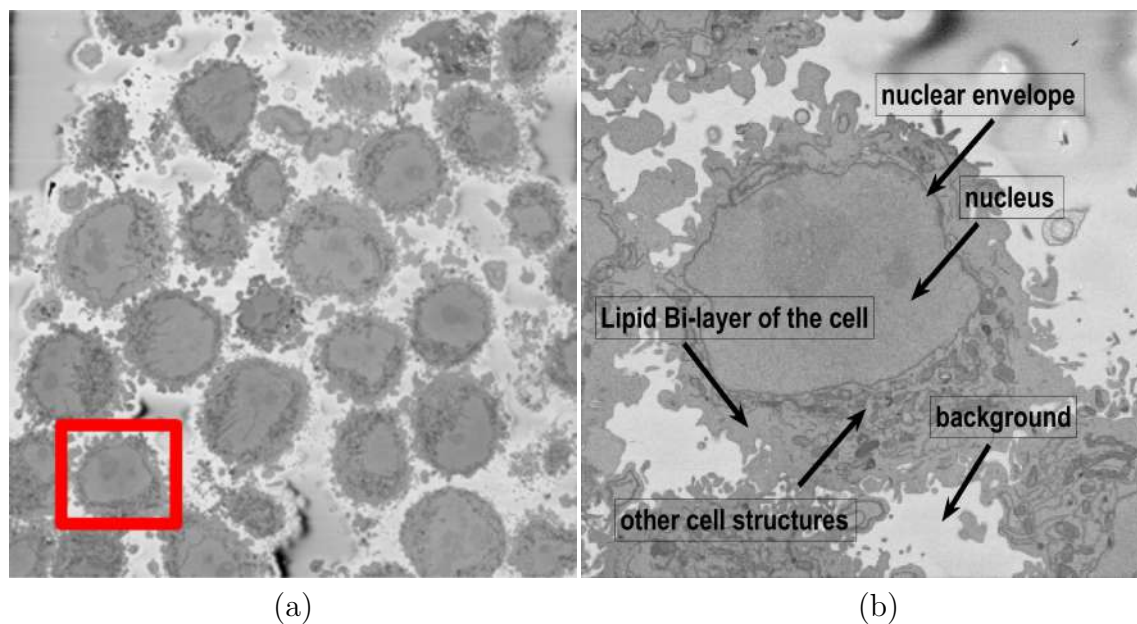
Fig. 2.3 The electron microscopy (EM) images of HeLa cells. (a) A representative 8192×8192 images arranged as 3D stack. The HeLa cells are the darker regions and the background is a brighter shade of grey. The red box indicates a region of interest (ROI), that is magnified on the right. (b) An example of EM image of a HeLa cell fixed in gel in three dimension and region of interest (ROI) with one HeLa cell centred is shown. At this resolution, it is easy to distinguish the nuclear envelope, borders of the cell and other structures. The nucleus, a large round body in the middle and enclosed within two concentric membranes that form the nuclear envelope (NE), contains most of the DNA of the eukaryotic cell and is seen here a thin section of a mammalian cell examined in the electron microscope (EM). This HeLa cell image is used throughout this work and obtained from The Francis Crick Institute [107].

(a)   (b)
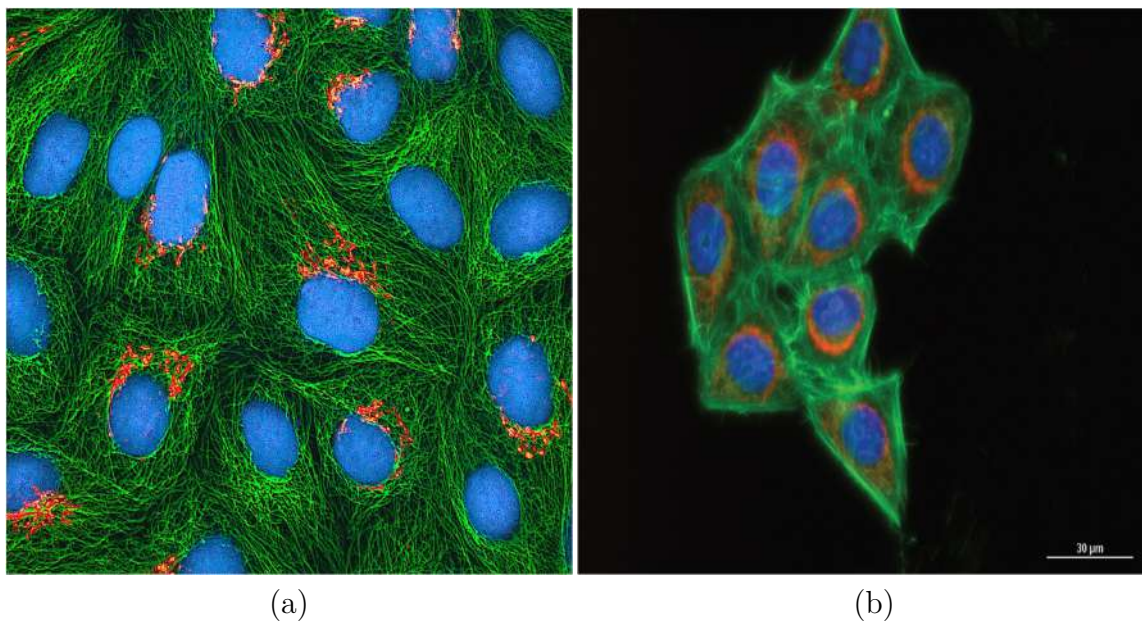
Fig. 2.4 Fluorescence Microscopy images of HeLa cells. (a) Multiphoton fluorescence image of cultured HeLa cells with a fluorescent protein targeted to the Golgi apparatus (orange), microtubules (green) and counterstained for DNA (cyan). (b) Magnified version of (a). Fixed and three colour fixed and stained HeLa cells. Mitochondria identified by a mouse anti-mitofilin primary antibody followed with a Rabbit antimouse IgG monoclonal antibody labeled with Texas Red (Red). Nuclei and actin filaments are identified by DAPI (blue) and AlexaFluor® - 488-phalloidin (green) counterstaining respectively. Scale bar indicates 30 μm. Images adapted from [108].

Different cellular components or organelles provide different functions for cells [25] and some of the sub-cellular structures that can be observed with Electron microscope (EM) in a cell are explained below.

The Nucleus is usually the most prominent organelle in a eukaryotic cell [25, 24] and contains the genetic material (DNA) of the cell as well as other structures that have important roles in mitosis or nuclear division. The nucleus is easily identified visually due to a fairly regular texture and low variance of grey value (Figure 2.3).

The Nuclear envelope (NE) contains the nucleus and characterised by two dark edges at a regular distance apart, a double lipid bi-layer structure, with the gap between the edges called the perinuclear space. A lipid bi-layer is a thin membrane made of two layers of lipids, a general category of molecules which function as signalling molecules, for passing information between cells, as well as modulators of membrane morphology. The NE is visually identified by marking the boundary between the different textures inside and outside the nucleus. It is typically 40 nm thick and encloses the nucleus separating it from the cytoplasm. The NE is a sub-cellular structure that surrounds the chromosomes of cells, separating them from the cytoplasm and other structures. During mitosis, the dividing of a cell into two daughter cells, the NE is broken down to allow the genetic information to duplicate and separate, and is reformed around the daughter chromosomes to form the nuclei of the daughter cells [109, 110].

Mitochondria are present in essentially all eukaryotic cells and they have a very distinctive structure. Each mitochondrion is sausage or worm shaped; it is from one to many micrometers long; and it is enclosed in two separate membranes. Mitochondria generate energy from food to power the cell.

Nuclear pores enable the transfer and exchange of materials between the nucleus and the cytoplasm and are typically 100-125nm in diameter, being roughly circular.

The Endoplasmic reticulum (ER) is similar to the NE, however it is not a container for the nucleus and is therefore identified as being surrounded by cytoplasm. The endoplasmic reticulum is one structure with the outer lipid bi-layer of the NE and often expands to the membrane separating the cell from its outer environment. It is

also another important compartment that plays a large part in the reformation of the NE during mitosis.

Golgi apparatus or complex receives and often modifies chemically the molecules made in the endoplasmic reticulum, and then directs them to the exterior of the cell or to various other locations.

Lysosomes are small, irregular shaped organelles in which intracellular digestion occurs, releasing nutrients from food particles and breaking down unwanted molecules for recycling or excretion.

Cytoplasm is the total content within the cell membrane other than the contents of the nucleus of the cell and it is a concentrated aqueous gel of large and small molecules.

Vesicles are small dark circles in the cytoplasm, with a very regular size and intensity profile.

2.3   Mitosis

Mitosis, or cell division, is the process of division of a nucleus into two daughter nuclei following the duplication of the genetic material in the parent nucleus [24]. Before mitosis begins each chromosome is replicated and consists of two identical chromatids called sister chromatids [24]. Mitosis is traditionally divided into five stages or phases after inter-phase (Figure 2.5).

Interphase is the state in which the cell is most stable and the NE is completely formed, during which the cell replicates all its genetic material and structures that are essential for the daughter cells to subsequently undergo mitosis (Figure 2.5a). During prophase chromatin, genetic material combining DNA and proteins, begins to condense to form chromosomes which can then be observed through a microscope (Figure 2.5b). Prometaphase in which the NE breaks down, the chromosomes are fully formed in pairs and begin moving towards the middle of the nucleus (Figure 2.5c). In metaphase, shown in Figure 2.5d, the chromosomes align along the middle of the
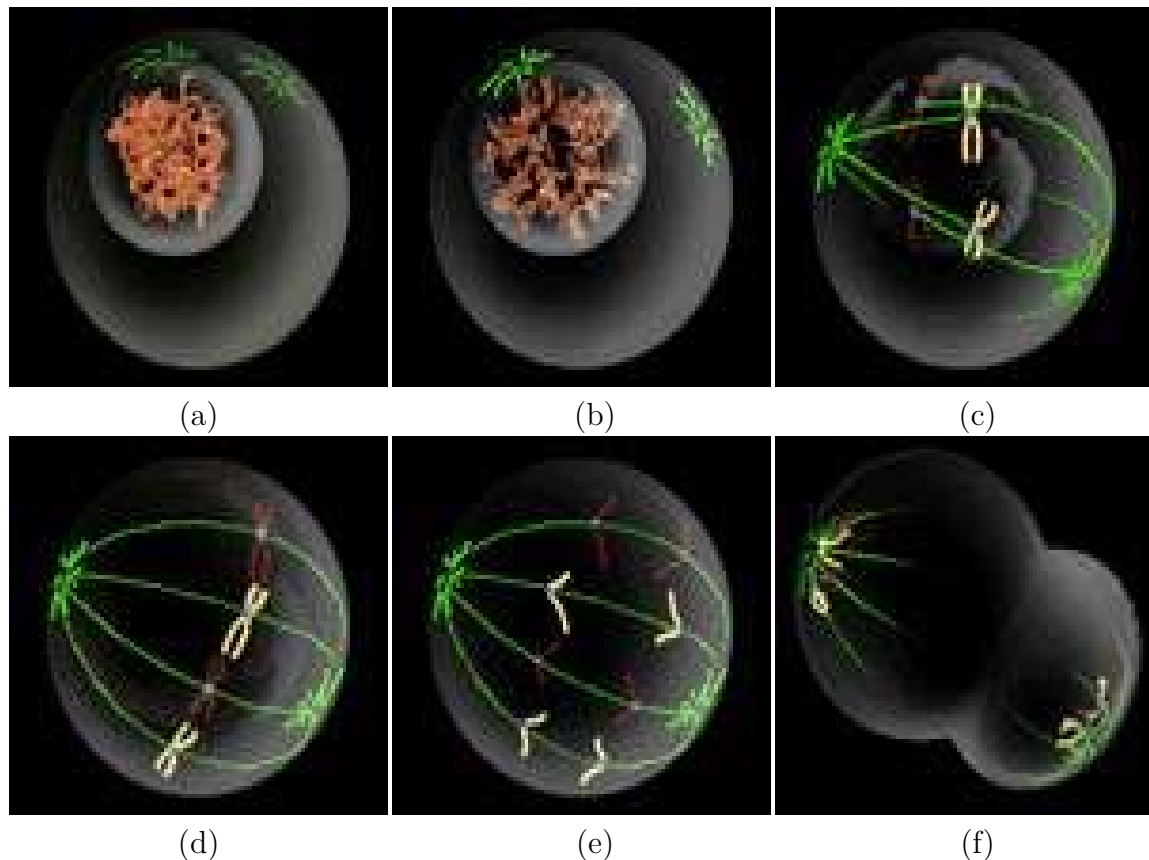
Fig. 2.5 Mitosis stages or phases. (a) Interphase is often included in discussions of mitosis, but interphase is technically not part of mitosis. Mitosis is nuclear division plus cytokinesis, and produces two identical daughter cells during (b) Prophase; Chromatin, genetic material combining DNA and proteins, begins to condense to form chromosomes which can then be observed through a microscope. (c) Prometaphase; The nuclear envelope breaks down, the chromosomes are fully formed in pairs and begin moving towards the middle of the nucleus. (d) Metaphase; The chromosomes align along the middle of the nucleus, ensuring that in the next phase each daughter receives a single copy of each chromosome. (e) Anaphase; Paired chromosomes separate and move to opposite sides of the cell. (f) Telophase; New membranes form around the daughter nuclei as their nuclear envelopes, the chromosomes also disperse and are no longer visible under a microscope. Images adapted from [24].
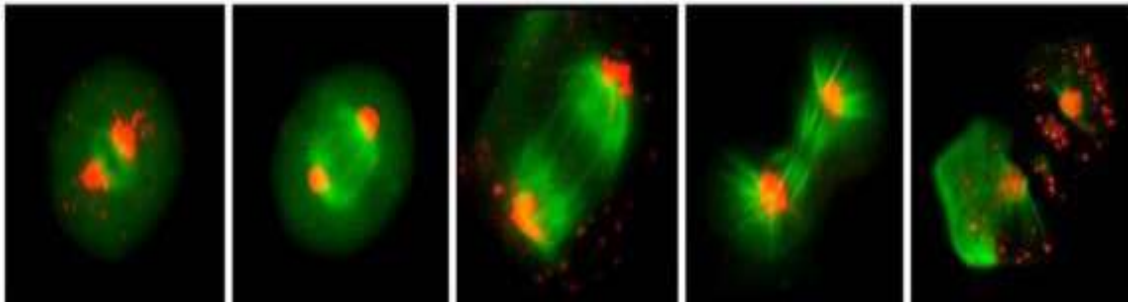
Fig. 2.6 Mitosis stages of a HeLa cell. From left to right: Metaphase, eary anaphase, late anaphase, telophase, and late cytokinesis. Images adapted from [111].

nucleus, ensuring that in the next phase each daughter receives a single copy of each chromosome. In anaphase paired chromosomes separate and move to opposite sides of the cell (Figure 2.5e). In telophase, shown in Figure 2.5f, new membranes form around the daughter nuclei as their nuclear envelopes, the chromosomes also disperse and are no longer visible under a microscope.

Figure 2.6 shows some of the stages of mitosis of a HeLa cancer cell [111]. DNA molecules become individually visible as chromosomes when they become more compact as a cell prepares to divide into two daughter cells.

2.4   Cancer

Cancer is a condition in which cells do not subdivide correctly during mitosis. There exist a mutation in a proliferation gene that causes the protein produced by the gene to be over-expressed or hyperactive results in excessive cell multiplication [110, 25]. Cancer is the product of mutations that set cells free from the usual controls on cell proliferation and survival [25]. A cell in the body mutates through a series of random accidents and acquires the ability to proliferate without the normal restrains. Its progeny inherit the mutations and give rise to a tumour that can grow without limit. Fault control of cell proliferation is not the only defect in a cancer cell, but it is a central and essential feature. The mutations that make cancer cells defective in this respect affect two broad categories of genes: *proliferation genes*, which encode proteins

that normally help to promote cell division, and *antiproliferation genes*, which encode proteins that normally help to apply the brakes that halt the cell cycle.

A mutation in a proliferation gene that causes the protein produced by the gene to be over expressed or hyperactive results in excessive cell multiplication. The mutant gene is then classified as an oncogene or a cancer-promoting gene, while the normal gene is known as a proto-oncogene. Oncogene is known as a cancer promoting gene [25, 110, 109], which can lead to the development of cancer. Correct reformation of the NE is critical to all cells that go through mitosis [24]. Defects in the formation of the NE have been linked to cancers as well as other human diseases; for example, increased malleability of the NE is observed in small-cell lung carcinomas. How the formation of the NE is regulated may give clues as to why it differs in cancerous cells. The change from a normal cell to a malignant one has multiple characteristics, sometimes summarised as six *hallmarks*: self-sufficiency in growth signals, evasion of apoptosis, insensitivity to anti-growth signals, sustained angiogenesis, tissue invasion and metastasis and limitless replicative potential [110, 112].

Figure 2.7 illustrates the differences between healthy and cancerous cells and their nuclear structures. Healthy cells have large cytoplasm, single nucleus, single nucleolus and fine chromatin while cancerous cells have small cytoplasm, multiple nuclei, multiple and large nucleoli and coarse chromatin. Nuclei can become irregular but it is important to note that not all of these changes are observed simultaneously in nuclei of actual cancer cells - different abnormalities are associated with different cancer types.

2.5   HeLa cells

The HeLa cell line was developed in the 1950s from a particularly aggressive strain of cervical cancer cells taken during a routine biopsy from a 30-year-old mother of five, Henrietta Lacks [115]. She was treated for the disease by Dr. George Gey in the coloured ward of The Johns Hopkins Hospital. As the head of tissue research, Dr. George Gey was at that time attempting to establish immortal cell lines that
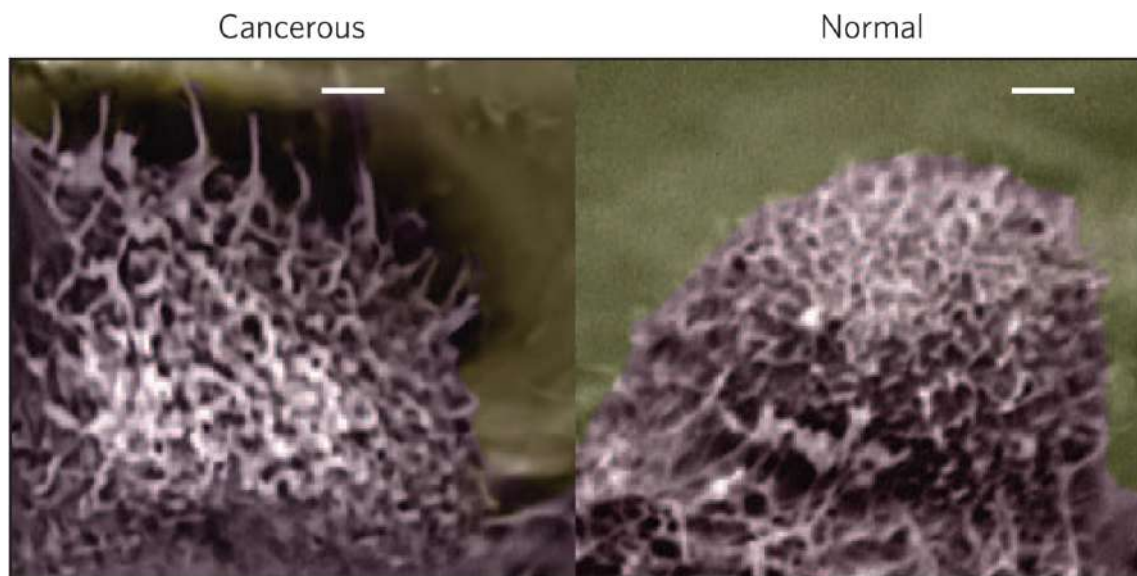
Fig. 2.7 Scanning electron microscope images of a cancerous (left) and normal cell, showing the differences in cell "brush". Cells' surface features, including microridges and hair-like microvilii, which, perhaps acting like sensors, are one key way that the cells interact with their environment. Together, these features form a cell's "brush." Normal cervical cells tend to have a brush layer consisting of a single average length - 2.4 μm - while the cancerous cells have mostly two typical lengths - 2.6 and 0.45 μm. Additionally, the long cancer-cell brush is about half as dense as that of the normal-cell brush while the short cancer-cell brush is more than twice as dense. Scale bars 5 μm. Image adapted from [113].
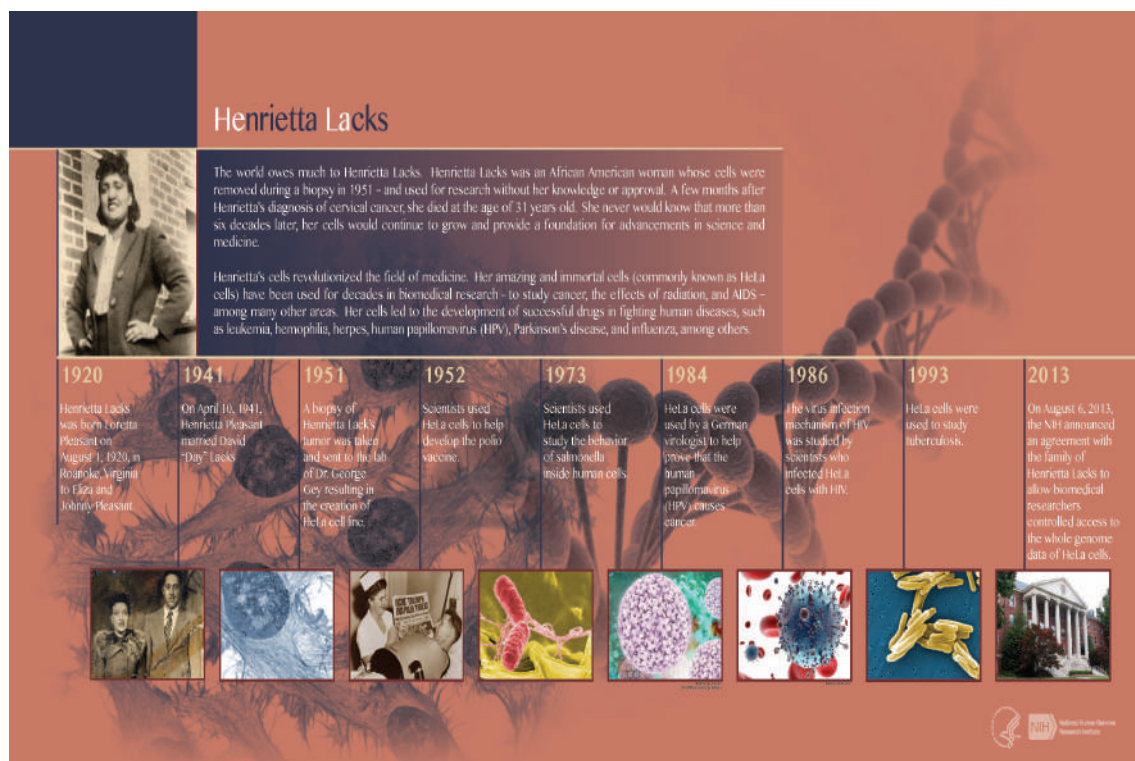
Fig. 2.8 A poster that describes the history of HeLa cells - Henrietta Lacks and HeLa cells in modern medicine. HeLa cells have been a vital tool in biomedical research and have been used for decades, leading to an increased understanding of the fundamentals of human health and disease. Some of the research involving HeLa cells also served as the underpinning of several Nobel Prize winning discoveries. To honour Ms. Lacks' and her family's continued support of biomedical research, NIH analyzed and evaluated the scientific literature involving HeLa cells and found over 110000 publications that cited the use of HeLa cells between 1953 to 2018. This analysis further highlights the persistent impact of HeLa cells in science and medicine, proving that they have been a consistent, essential tool that has allowed researchers to expand the knowledge base in fields such as cancer biology, infectious disease, and many others. While Henrietta Lacks' story has been known in the research community for some time, it raised further awareness after the publication of the best-selling book "The Immortal Life of Henrietta Lacks" [114]. Image adapted from [108].

could be used in medical research. Taking tissue samples from poor and/or African American cancer patients being treated in the coloured ward without informed written consent was not unusual and neither doctors nor their patients were aware of any of the ethical issues involved. Henrietta Lacks was diagnosed in 1950, and died within a year at the age of 31. But her cells lived on and became the first human cell line to be established. The reason why these cells were immortal while several others failed to grow in culture is not entirely understood. In addition to cervical cancer caused the Human Papilloma Virus (HPV) (multiple copies of HPV genome were later found in the HeLa cell line), Henrietta Lacks also had syphilis which probably suppressed her immune system. In any case, her cells pioneered research that led to a better understanding of the causes and treatment of human cancers. In 1952, the Tuskegee Institute set up a laboratory to supply the cell line to other researchers and laboratories which soon spawned a company named Microbiological Associates that supplied HeLa cells for profit. The cells were used in laboratories around the world and have been crucial for the development of vaccines, for instance, the polio vaccine. Although initially the cells were known by the pseudonym Helen Lane, as a tribute to George Gey who died of pancreatic cancer in 1970, the cells were correctly identified as having been sourced from Henrietta Lacks. The Lacks family learnt for the first time that her cells were still alive when scientists at the Johns Hopkins Hospital approached them for tissue samples for a genetic analysis. They soon learnt that her cells had also been commercialised. Henrietta's children found this upsetting—for one, they wondered how their mother's soul could rest in peace if her cells were still around; for another, the family was poor and could ill afford health insurance, and yet their mother's cells were part of a multi-billion-dollar industry. Henrietta Lacks story caught public attention in 2010 with the publication of an award-winning book, which stayed on the New York Times best-seller list for 2 years, "The Immortal Life of Henrietta Lacks" by Rebecca Skloot. The book highlighted the fact that the development of the cell-line, the genetic and molecular analysis, and the commercial applications developed from the cell line had all been done without the knowledge or consent of Henrietta Lacks or her family,

nor had they been the beneficiaries of the developments. The family felt their privacy had been violated, so much so that her grandchildren contacted the European Molecular Biology Laboratory and asked them to withdraw a paper that they had published on the genetic make-up of HeLa cells. The scientists withdrew the paper with apologies to the family. Meanwhile, Rebecca Skloot set up The Henrietta Lacks Foundation in 2010 by donating a portion of the book's proceeds and donations from the reading public and scientists who would like to do something in return for the family. The foundation provides scholarship funds and health coverage to Henrietta's descendants (Skloot, 2011). Also, in 2013, Skloot and the Lacks family held meetings with the NIH Director Francis Collins, and NIH Deputy Director for Science, Outreach, and Policy Kathy Hudson, along with scientists and ethicists from Johns Hopkins to discuss modalities for publishing genetic information and future applications using the cells. What was agreed upon was that Lacks' genome data will be accessible only to those who apply for and are granted permission; two representatives of the Lacks family would serve on the NIH group responsible for reviewing applications for controlled access to the HeLa cells. Also, researchers who use the data would include an acknowledgement to the Lacks family in their publication. There was however no legal provision for monetary compensation to the family. Nevertheless, the agreement is viewed as "a moral and ethical victory for a family long excluded from any acknowledgement and involvement in genetic research their matriarch made possible" (Caplan, 2013).

2.5.1   Genome of HeLa cells sequenced for the first time

HeLa is the most widely used model cell line for studying human cellular and molecular biology. To date, no genomics reference for this cell line has been released, and experiments have relied on the human reference genome. Effective design and interpretation of molecular genetic studies done using HeLa cells requires accurate genomics information. Here we present a detailed genomics and transcriptomic characterisation of a HeLa cell line [116].

HeLa cells originate from the cervical cancer tumour of a patient named Henrietta Lacks (hence their name), who later died of her cancer in 1951. They were the first type of human cells that were successfully cultivated in the laboratory, and have since been the most widely used human cell line in biology. Cancer cells by definition continuously replicate themselves, thus making good candidates for experimental biology; HeLa cells were particularly aggressive at doing so and thus became very popular. There are lots of different sub types of HeLa cells that are now used – the study by EMBL researchers has sequenced the genome of one sub-type of HeLa cell.

Scientists deliver the first high-resolution sequence of HeLa cells, a key research tool for human disease and general biology. Sequence analysis reveals the full extent to which HeLa cells are different to the Human Genome Project reference. Resource could enhance the quality of research using HeLa cells [116].

HeLa cells are the world's most commonly used human cell lines, and have served as a standard for understanding many fundamental biological processes. In a study published today in G3: Genes, Genomes and Genetics online, scientists at the European Molecular Biology Laboratory (EMBL) in Heidelberg, announce they have successfully sequenced the genome of a HeLa cell line. It provides a high-resolution genomics reference that reveals the striking differences between the HeLa genome and that of normal human cells. The study could improve the way HeLa cells are used to model human biology.

The scientists' analysis of the HeLa genome revealed widespread abnormalities in both the number and structure of chromosomes, as well as factors commonly associated with cancer cells like losing healthy copies of genes. In particular, the researchers found that countless regions of the chromosomes in each cell were arranged in the wrong order and had extra or fewer copies of genes. This is a telltale sign of chromosome shattering, a recently discovered phenomenon associated with $2-3\%$ of all cancers. Knowledge of the genetic landscape of these cells can inform the design of future studies using HeLa cells, and strengthen the biological conclusions that can be made from them.

"The results provide the first detailed sequence of a HeLa genome," explain Jonathan Landry and Paul Pyl from European Molecular Biology Laboratory (EMBL), who carried out the research. "It demonstrates how genetically complex HeLa is compared to normal human tissue. Yet, possibly because of this complexity, no one had systematically sequenced the genome, until now."

"Our study underscores the importance of accounting for the abnormal characteristics of HeLa cells in experimental design and analysis, and has the potential to refine the use of HeLa cells as a model of human biology," adds Lars Steinmetz from EMBL, who led the project.

For decades HeLa cells have provided effective and easily usable biological models for researching human biology and disease. They are widely regarded as the 'industry standard' tool for studying human biology. Studies using them have led to two Nobel prizes and a host of advancements in many areas, including cancer, HIV/AIDS and the development of the polio vaccine. The HeLa genome had never been sequenced before, and modern molecular genetic studies using HeLa cells are typically designed and analysed using the Human Genome Project reference. This, however, misrepresents the sequence chaos that characterises HeLa cells, since they were derived from a cervical tumour and have since been adapting in laboratories for decades.

The study provides a high-resolution genetic picture of a key research tool for human biology. It highlights the extensive differences that cell lines can have from the human reference, indicating that such characterisation is important for all research involving cell lines and could improve the insights they deliver into human biology.

2.5.2   How have HeLa cells been used in science?

The use of HeLa cells has contributed to scientists' understanding of a wide range of fundamental biological processes and more than 60,000 research publications. One of the earliest uses of HeLa cells was to develop the vaccine against the polio virus. Recently, two Nobel prizes have been awarded for discoveries where HeLa cells played a central role: the link between human papillomavirus and cervical cancer (2008, Harald

zur Hausen) and the role of telomerase in preventing chromosome degradation (2011, Elizabeth Blackburn, Carol Greider, and Jack Szostak). During the past decade, HeLa cells have been used to develop large-scale technologies like RNA sequencing, RNA interference screens, and other 'omics' approaches. Studies using these approaches on HeLa cells have furthered our understanding of important cellular processes, such as cell division. HeLa cells have also been used in studies of nuclear toxins, and have even been into space.

2.5.3 Why are HeLa cells so useful?

Biologists tend to use HeLa cells as the 'default' human cell line because they are so easy to work with – they reproduce rapidly, easily and cheaply. Despite being cancerous, HeLa cells still share many basic characteristics with normal cells – they produce proteins, express and regulate genes, communicate with one another, and are susceptible to infections. It is thus possible for scientists to use HeLa cells to study not only cancer, but also basic functions carried out by all human cells. The genome sequence should make genetic studies using HeLa even more effective. The HeLa cell line sequenced in this study has spent decades in labs, dividing and thus undergoing mutations and changes – it is very different from the original cells that started growing in 1951 [21]. Furthermore, the original cells were taken from Mrs. Lacks' cervical cancer – as cancer is a disease of the genome, the DNA of cancer cells is different to that of the patient. Therefore, the genome we sequenced contains a combination of genetic variants originating from the donor's genome, variants that arose during the tumour's development, and variants that occurred during the many years of in-lab evolution of the cell line. Comparisons to common genetic variants in human populations today allow one to infer variants likely to have been present in the donor's genome, although such inferences can only be made with a certain likelihood. Measurements of common variants in the HeLa genome have been published before (e.g. SNP arrays) and further variants can also be inferred from HeLa DNA sequencing data already deposited in the public domain (e.g. ChIP-Seq and RNA-Seq). The data does not change the fact

that it is possible to make predictions about Mrs. Lacks' genome, or those of her descendants. The value of our study is in creating a comprehensive resource of all variants (common and non-common) present in a cell line that has been of such high value for biomedical research.

2.5.4 Cross-contamination of HeLa cells

HeLa cells proliferated in cultures around the world and evidently as the years passed they contaminated other cell lines [23, 117]. Cell lines are expected to provide an unlimited source of material, free of contaminating cells and often easily cultured. Unfortunately, there have been problems with misidentification and cross-contamination and they compromised research seriously. The most frequent contributor to cross-contamination of cell lines is the HeLa cell isolated from an aggressive cervical adenocarcinoma [118]. Cross-contamination of cell lines have regularly been brought to light but have not received many audiences until cell banks such as American Tissue Culture Collection (ATCC), Deutsche Sammlung von Mikroorganismen und Zellkulturen (DSMZ), European Collection of Cell cultures (ECACC), Japanese Collection of Research Bioresources (JCRB) decided to act by informing their clients or even by withdrawing the false cell lines from their catalogue [23]. Researchers do not want to use false cell lines leading to misleading publications, which can potentially have a very high cost in terms of invalid hypotheses and paradigms and patient treatments [23]. However, most new cell lines are freely exchanged between laboratories, rarely having their identities checked. In order to avoid cross-contamination of these lines, periodic re-authentication of cell lines is advisable. Today all reputable cell banks employ methods to confirm the identity and origin of the cell lines they distribute. This is very important since distribution of misidentified or cross-contaminated cell lines may later be the subject of costly and embarrassing actions. Most cell banks may also test cell lines provided by their users or originators [23].

2.6   Literature review

Cervical cancer is one of the leading causes of death in females [13], and together with breast cancer, it contributes to 4.2% of the global causes of death [14]. In the United Kingdom, around 3,000 women are diagnosed with cervical cancer every year, and around 1,000 of those die annually [15]. Cancer cell lines and tissue culture studies have been used extensively to understand tumour biology and facilitate drug discovery processes [16–18]. The availability of cell lines together with genetic advances has contributed to cancer research, which has grown significantly as a proportion of all research in biomedical areas [19]. Perhaps the most important cell line has been the cervical cancer *HeLa* cells, derived in 1951 [20, 21] and widely used as model for thousands of biological experiments [22, 23]

The NE is the membrane that surrounds the chromosomes and partitions them from the rest of the cell contents, thereby forming a protected environment for the genetic material. Almost every cell (red blood cells being a notable exception) has a nucleus bounded by a nuclear envelope, and so it is a primary target for automated segmentation as a reference structure for visualising and spatial positioning of other cell organelles, and as a feature in its own right as its morphology is known to alter in cancer, infection and in rare genetic disorders such as the laminopathies. As there may be a single cell of interest, understanding the nuclear formation [119], the arrangement of the chromosomes [120], the breaching of the nuclear envelope [121] or the molecular sociology [122, 123] become relevant. The NE is an important structure in cervical cancer cells as they undergo mitosis, breaking down to allow the cell to duplicate and split before reforming around each daughter nucleus [24, 25]. This importance is also underlined in cancerous cells [21, 20, 22] where certain components of the nuclear envelope take central roles in cell functions that affect tumour development and progression. Visualising and investigating the NE requires its accurate segmentation [4, 7, 26, 27] from a set of serial images of the cell obtained through EM. Manual segmentation (Figure 2.9) requires a great amount of expert knowledge and training and it is very tedious and time consuming [31] as it takes on average a couple of weeks
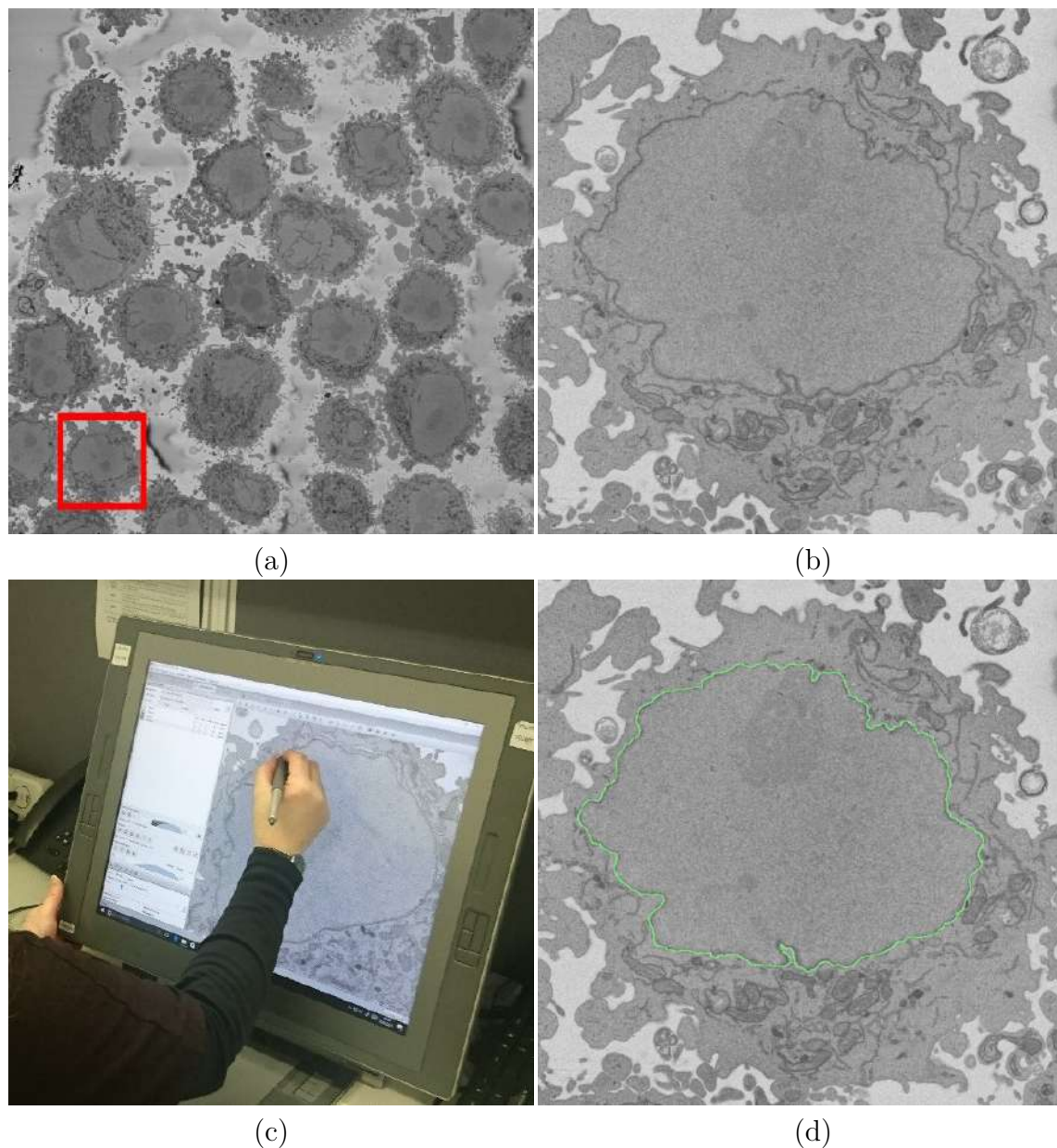
Fig. 2.9 HeLa cell imaged with electron microscope and manual segmentation of HeLa cell. (a) Sample of the Serial Blockface Scanning Electron Microscopy (SBF SEM) image showing several HeLa cells. Red box denotes a region of interest (ROI), an individual cell, which is magnified in (b). (b) Individual cells were cropped and hand located so the algorithm developed for this work can be applied to the image to segment the nuclear envelope (NE). (c) An expert segments the NE of a HeLa cell manually. Manual segmentation is very tedious and time consuming but remains widely used and is still considered as the gold standard despite the significant disadvantages of time and inter- and intra-user variability. (d) The manual segmentation or the ground truth (GT) of the nuclear envelope of this particular slice is shown in green.

to segment all the structures in a full data set. Once the segmentation is achieved a 3D reconstruction of the nuclear envelope can be created and interpreted to study its biological characteristics and relationships to other structures in the cell.

Segmentation is often described as a foreground/background separation in an image [124]. Probably one of the most common shared tasks in image analysis systems is segmentation. Segmentation aims to partition the image plane into meaningful regions [125]. The simplest way of thinking of an image is that it is a grid of points or elements called "matrix" and each of them will contain a certain value called a pixel or a "picture element" [126]. Segmentation subdivides an image into its constituent regions or objects and it should stop when the objects or regions of interest in an application have been detected [127–129]. The aim of segmentation is to find a way how to extract useful information from the images in automatic or semi-automatic methods [130] and their algorithms on computer processing.

Cell segmentation and classification has been an important problem for many years [6] and has attracted considerable attention in clinical practice and research, as the presence or absence of cells or their characteristics like size or shape, could be important indicators for presence or severity of a disease [9]. Segmentation of cells and particularly of nuclei and the NE is highly dependent on the contrast, signal to noise ratio, complex morphological structures, and the resolution of the imaging. Thus, techniques that work at lower resolutions (such as those use for immunohistochemistry), like watersheds, are not immediately applicable at higher resolutions such as those provided by EM. Furthermore, structures observed with EM display far more complex morphological structures, and many times with lower contrast than those observed in light and fluorescence microscopy. Serial section EM provides contiguous images of a sample [131] that correspond to the volumetric sample, thus the data is 3D and complicates both storing and transferring data as well as processing and interpreting it [40].

Segmentation of cells [132] and organelles in electron micrographs is in many cases still a manual process and considered as the gold standard [28–30], despite the

significant disadvantages of time and inter- and intra-user variability and intensifying efforts by the community to automate the process. This is largely due to complexity and diversity of features within a single image, many of which have similar grey values, making a clean histogram-based feature extraction unlikely. The success of algorithms designed for feature extraction is highly dependent on the contrast and signal-to-noise ratio of the image, as well as the resolution, the crowding and diversity of features, and the heterogeneity of the appearance of the feature in the image, which is linked to its morphology and orientation in the 2D slice. These, and other properties of the sample and image, mean that techniques that work well for imaging techniques like immunohistochemistry and light microscopy (e.g. watersheds) [133–136] do not usually port well to EM.

Shallow [137, 138] and deep learning methodologies [71, 69] are becoming popular for segmentation and classification of image data, however, they require significant computational power as well as very large training data sets [98, 99], which are rarely available in biological electron microscopy. There are some commercial environments with segmentation tools, like Amira™(Thermo Fisher Scientific, Waltham, MA, USA), IMARIS and, AIVIA [139, 140]. However, the cost of these tools may prevent scientist to use them [141].

An alternative approach is being developed for cell segmentation by taking advantage of *citizen science* where an *army of non-experts* [142] are recruited to provide non-expert human annotation, segmentation or classification through web-based interfaces. Whilst these results are valuable, they take considerable time as they depend on volunteers (or paid workers like Amazon's Mechanical Turk) and they do not guarantee a correct answer.

A project called Etch a Cell [107] was created to provide manual segmentation of the NE of HeLa cells. Thousands of volunteers segment the NE of HeLa cells and this approach is producing promising results, but further development is required to quality control, aggregate and train with these non-expert contributions. Thus, specific processing algorithm for automated analysis and visualisation of cell features in large

microscopy data sets are still important [143–147], both for direct analysis of data and
for production of additional training data.

2.7   Summary

In this chapter the structure of cells, different organelles of the eukoryatic cells, and
cell division or Mitosis have been explained. Cancer cells in general and specifically
cervical cancer cells or HeLa cells were explained in detail including its history. Finally,
how HeLa cells have been used in science, why they are useful and cross-contamination
of HeLa cells were explained in the next section of this chapter.

Scientists tried to keep the human cells alive outside the human body, in a test tube
in a laboratory, for very long time in the early twentieth century. Most of the time they
managed to maintain cells alive for a maximum of several weeks [21]. The breakthrough
came on 8th February 1951 when an African-American woman called Henrietta Lacks
(Figure 2.8) was admitted to John Hopkins Hospital in Baltimore, Maryland. She died
on 4th October 1951 and the reason of her death was the cervical cancer. Researchers
led by George Otto Gey, a scientist from Tissue Culture Laboratory at Johns Hopkins
Hospital, took some samples of her cancer cells from her cervix and they developed
the first human cancer continuous cell line in a test tube. Henrietta's cancer cells were
kept alive in a test tube. Cervical cancer is normally slow growing but this was not
an ordinary cancer as it did not respond to radiotherapy and the cells were thriving
[21]. The abbreviation, HeLa, we use today was devised by using the first two letters
of Henrietta Lacks' first and last names to keep her real name in a secret. In order to
give a measure of confidentiality the donor was said to be Helen Lane or Helen Larson
[21]. This immortal cell line, derived from the Henrietta Lacks' cervical cancer cells,
was thus called HeLa.

It is important to highlight that doctors who treated Henrietta did not request any
consent from Henrietta Lacks or her relatives to harvest the cells. However, the ethical
procedures were different at that time in the USA and it was not a requirement. There

were not any laws to enforce doctors to ask for written permission to get samples for a research and Henrietta Lacks was not an exception.

HeLa cells became a robust, immortal cell line and they were easily propagated over generations in culture. George Otto Gey supplied samples very generously to scientists who were interested in studying the first established human cancer cell line in the Unites States and across the world. HeLa cells have survived up to the now and are still used in laboratories as a model for human cells in thousands of biological experiments, contributing to the understanding of disease processes [22, 23]. Knowledge of almost every process that occurs in human cells has been obtained using HeLa cells and many other cell lines that have since been isolated [21]. Figure 2.3b shows a HeLa cell image taken by serial blockface electron microscope and it was used in this work. Figure 2.3c shows a fluorescence staining of HeLa cells for cytokeratin-8, (red) cytoskeletal actin (green), and nuclei (blue).

Chapter 3

Electron Microscopy and Materials

3.1 Introduction

The following chapter delves into microscopy and its fundamentals and then describes the importance of Electron Microscopy (EM) in cell research. The chapter explains main types of electron microscopy techniques actively used in current research. Basic structure and fundamentals of EM will also be presented.

In the last part of this chapter, we also describe the HeLa images analysed in this work from the point of view of the EM images. It is a continuation of the biological context described in Chapter 2 and this chapter concludes with the acquisition of images of cells from Serial Block Face Scanning Electron Microscop (SBF SEM). Different data sets to be analysed in this work, where a thorough description of the images acquired is provided and the challenges to analyse the images are presented.

Microscopy

Microscopy refers to the use of microscope to observe objects and areas of objects that cannot be seen with the naked eye as these objects and areas are not within the resolution range of the normal eye [148, 149] (Figure 3.1).

Microscopic techniques allow for the visualisation of objects which go beyond the resolution limit of the human eye and the visible range of light wavelengths [3], which involve colours ranging from red (700 nm, 428 THz) to violet (420 nm, 714 THz). The
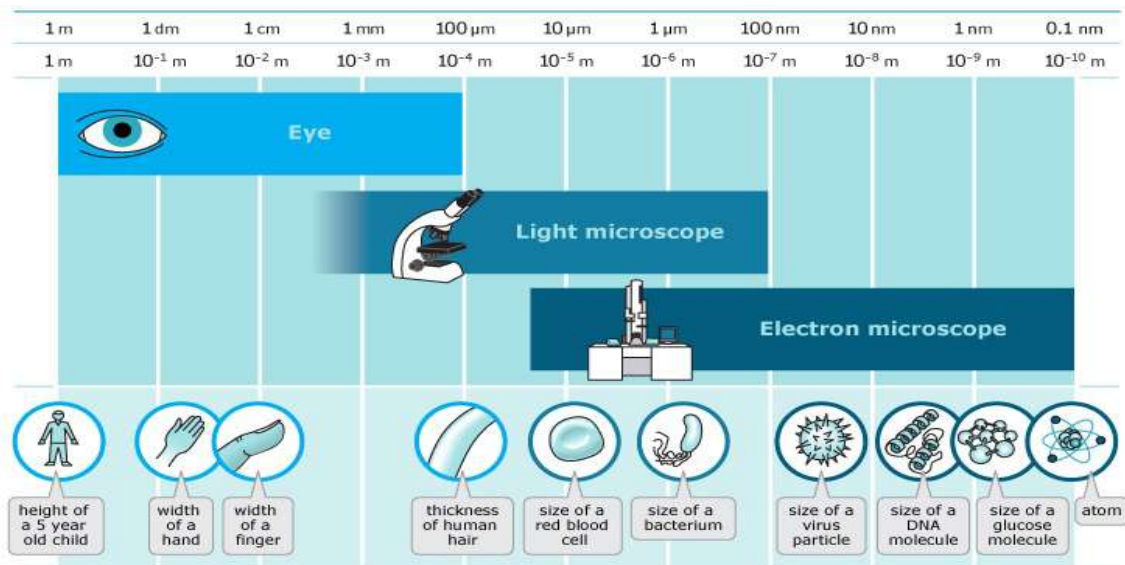
Fig. 3.1 Resolving power of naked eye, light microscope and the electron microscope (EM). Different techniques are used to see finer details in sub nanometre resolution. EM provides images with several hundred times the resolving power of the best light microscopes. Image adapted from [150].

human eye requires contrast to perceive details of objects [151] and cells are generally transparent.

Cells are small and the first practical problem in cell biology is how to see them [24]. For this purpose, several methods have been proposed: colour staining for transparent cells, in which a chemical is added to the sample which absorbs light at a certain wavelength, thus improving contrast; dark field, which exploits the scattering of light on small particles that differ from their environment; phase contrast, where light incident on the sample is split in two spaced beams of light; fluorescence microscopy (Figure 3.3), in which the reflected light passes through the objective where it is focused onto the fluorescent specimen. The emissions from the specimen are in turn, passed back up through the objective – where magnification of the image occurs –and now through the dichroic mirror. The underlying process of fluorescence involves the absorption of light energy (a photon) by an indicator followed by the emission of some of this light energy (as another photon) a few nanoseconds later [152]. Because some energy is lost in this process, the emitted photon has less energy than the absorbed photon.

Light with a short wavelength (toward the blue) has higher energy than light with a long wavelength (toward the red). Therefore, light emitted from an indicator usually has a longer wavelength than that of the absorbed (excitation) light. This change is called the Stokes shift. Finally, electron microscopy (EM), which will be described in detail as it is relevant to this work. The basic principle of the different techniques to acquire images from a microscope involves an energy source which acts upon a sample by scanning or transmitting through it. The images in this work were acquired through Serial Block Face Scanning Electron Microscopy (SBF SEM) which utilises an ultra-microtome with a diamond knife to cut and discard very thin slices (The knife can cut 2000 layers of a human hair) or sections from the top face of the sample. The sample is then raised to the focal plane to be scanned, and this process is repeated sequentially, resulting in a stack of images through the sample volume.

Before the development of microscopy and its application, the fine structures of cells and viruses were not accessible to biologists and the invention of the microscope in the 17th century made cells visible for the first time. Traditionally, cell biology has relied on light and fluorescence microscopy in order to analyse cells and tissues. Although light microscopes incorporate many sophisticated improvements, the properties of light itself set a limit to the fineness of detail they can reveal as they allow us to magnify cells up to a thousand times and to resolve details as small as 0.2μm - a limitation imposed by the wavelike nature of light [148]. In order to overcome this limitation of the wavelength, it is possible to use electron beams instead of light and this lead to the development of electron microscopy (EM).

Equation 3.1 gives the relationship between the wavelength and the frequency of the electromagnetic waves.

$$\lambda = v/f \tag{3.1}$$

where λ is the wavelength in m (metre), v is the velocity of the electromagnetic waves in m/s^{-1}, and f is the frequency in Hz or s^{1}.

Equation 3.2 shows the inverse proportion between the energy and the wavelength. As electrons have a much shorter wavelength than photons, they have very high energy therefore high spatial resolution [149].

$$E = hc/\lambda \tag{3.2}$$

where h is the Planck's constant, c is the speed of light in vacuum and λ is the electron wavelength.

In 1945 Albert Claude, Keith Porter and Ernest Fullam published the first picture of an intact cell taken with an EM [148]. The EM image magnified the specimen 1600 times and their first electron micrograph of a cell revealed the mitochondria, the Golgi apparatus and the endoplasmic reticulum [149].

Figure 3.2 illustrates the resolving power of naked eye, light microscope and the EM as well as some of the natural and man made materials with their dimensions. Figure 3.4 shows the EM used to get the images for his work and it is located at the Francis Crick Institute. The main difference between light microscopy and EM is light (photon) is used in a traditional microscope to illuminate a specimen, whereas an EM uses a beam of electrons to see finer details in sub nanometre resolution [148]. High energy electrons have short wavelengths that allow us to observe nanoscale features in samples. Nowadays the electron microscope provides images with several hundred times the resolving power of the best light microscopes.

The main reasons why we use electrons as a probe are;

1. It is easy to produce high brightness electron beams.

2. Electrons are easily manipulated.

3. High energy electrons have a short wavelength and shorter wavelengths means higher spatial resolution (Rayleigh Criterion).

4. Electrons interact strongly with matter.
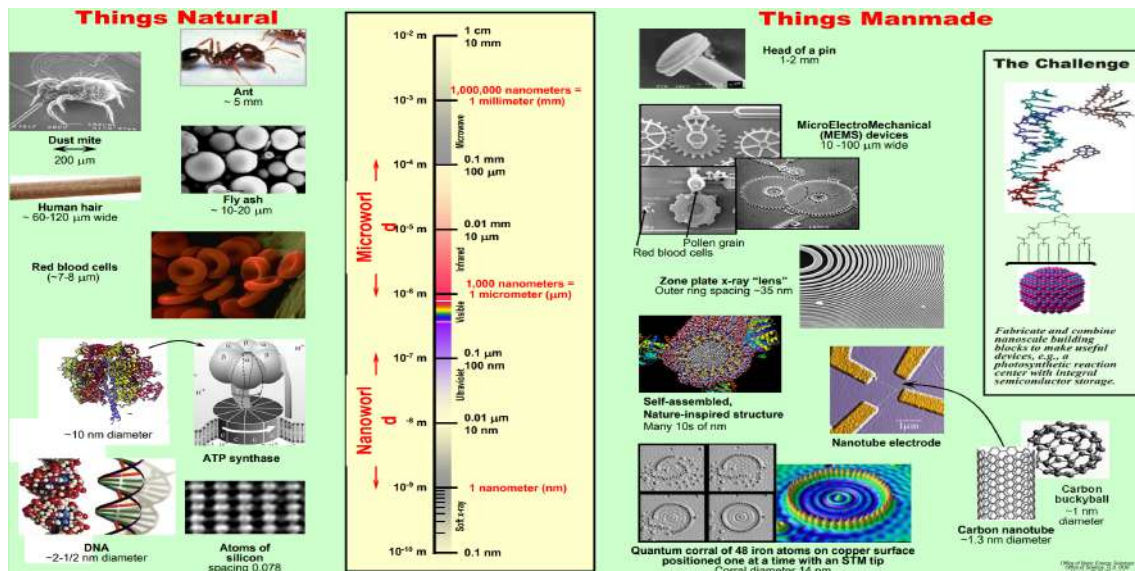
Fig. 3.2 Some of the natural and man made materials with their dimensions. Image adapted from [150].

<div align="center">(a)   (b)</div>
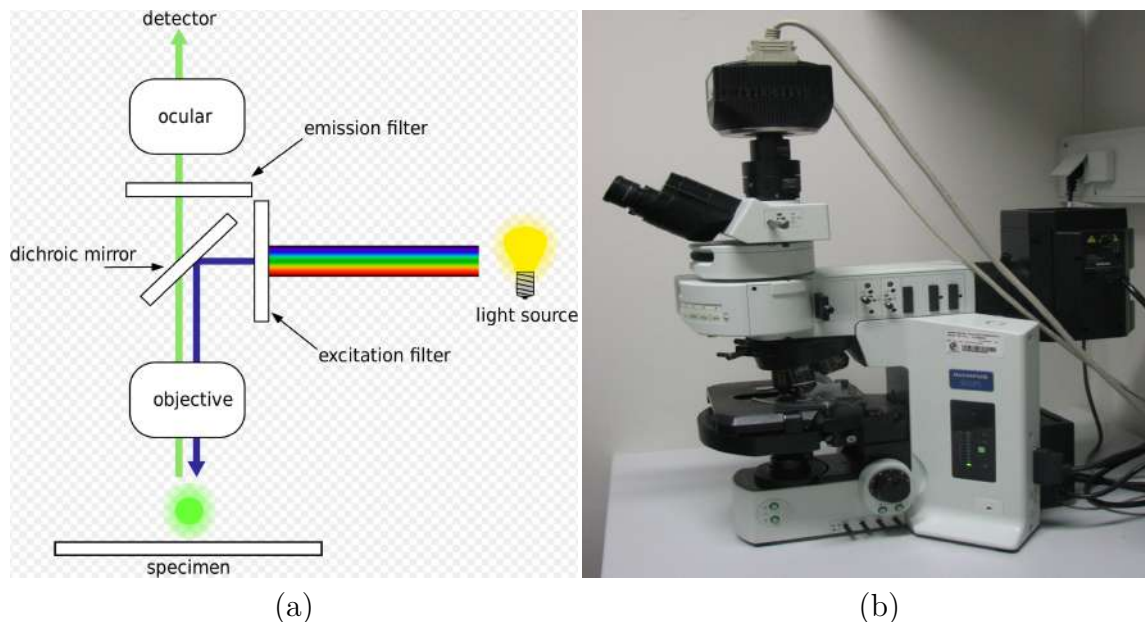
Fig. 3.3 (a) Schematic of a fluorescence microscope and (b) An upright fluorescence microscope (Olympus BX61) with the fluorescence filter cube turret above the objective lenses, coupled with a digital camera. Images credit to Masur.
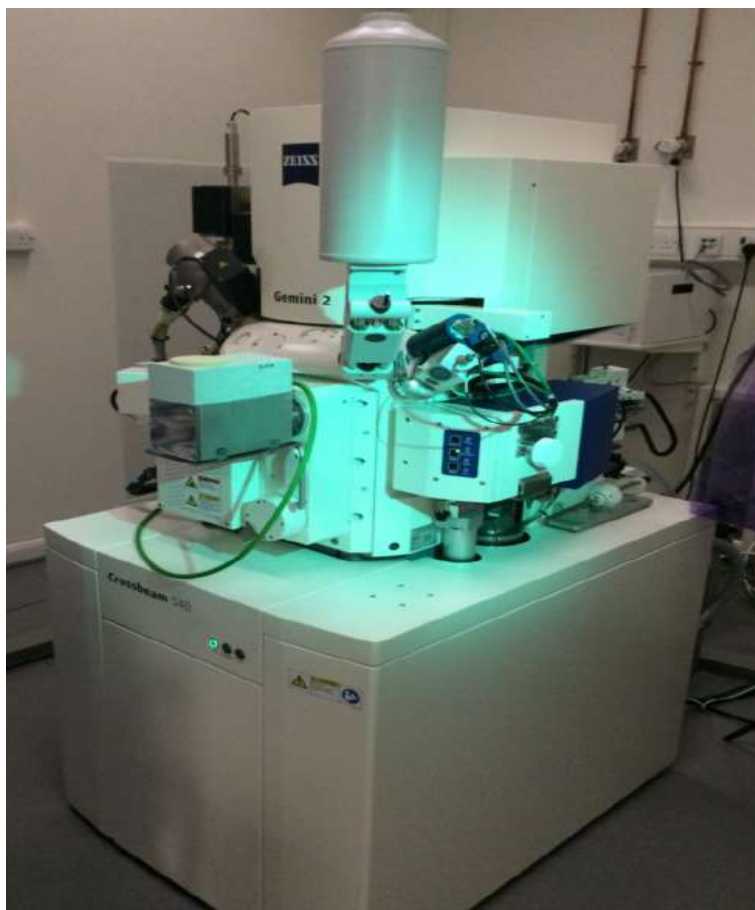
Fig. 3.4 The Serial Block Face Scanning Electron Microscopy (SBF SEM) from which the data sets of this work were obtained is located at the lower ground floor of the Francis Crick Institute. Faraday cage was employed in order to protect the electron microscope from the magnetic fields and to absorb vibrations the EM was placed on a 20 tonne concrete block. SBF SEM utilises an ultra-microtome with a diamond knife to cut and discard very thin slices or sections from the top face of the sample. The sample is then raised to the focal plane to be scanned, and this process is repeated sequentially, resulting in a stack of images through the sample volume.

As a beam of electrons is used as the source of illumination and as electrons interact strongly with matter, EM overcomes the limitation of light microscopes and greatly extending our ability to see the fine details of cells. Electron microscopy allows imaging of the cell and the structures relevant to the NE throughout the different stages of mitosis. It also allows biologists to analyse sub-cellular structures such as mitochondria and nuclei. For the past half a century cell biologists have scrutinised all types of cells and tissues with the aid of the electron microscope. It is today impossible to imagine what biology would be, if this instrument had not been invented.

There are two varieties of electron microscopes: transmission electron microscope (TEM) and scanning electron microscope (SEM) and they will be explained in the following sections. It is important to understand how images are formed in each type of microscope as well as their unique limitations and applications [149]. Both microscopes operate in a high vacuum in order that molecules in the air do not scatter the beam of electrons.

3.2 Transmission electron microscopy (TEM)

The transmission electron microscope (TEM) is in principle similar to an inverted light microscope, and the specimen is placed in vacuum and it must be very thin [149]. The TEM has a useful magnification of up to a million-fold and a resolution, with biological specimens, of about 2 nm. In TEM, a beam of electrons is incident upon an ultra thin sections. The behaviour of the electrons as they hit the sections is dependent upon its properties; some electrons are absorbed, others are scattered or pass through. The design of the TEM consists of an electron gun, a series of lenses and a charge-coupled device (CCD) camera. A condenser lens focuses the electron beam on the specimen. Three other lenses work to focus the transmitted electrons onto the camera to record the image. The overall quality of the image is dependent upon the lenses and their configurations. Magnifications of up to $1000000\times$ are possible with TEM. The ultra thin sections must withstand a vacuum in the electron microscope. To do so they are

fixed to an epoxy resin after staining with heavy metals and dehydrating with ethanol. Sections around 70nm are taken from the specimen such that they are thin enough for the electron beam to pass through, however for large structures this requires many serial sections to be taken and imaged. Therefore, to create a full 3D reconstruction images for each section must be aligned and segmented.

The transmission electron microscope is a very powerful tool and is commonly used [153]. A high energy beam of electrons is shone through a very thin sample, and the interactions between the electrons and the atoms can be used to observe features such as the crystal structure and features in the structure like dislocations and grain boundaries. Chemical analysis can also be performed. TEM can be used to study the growth of layers, their composition and defects in semiconductors. High resolution can be used to analyse the quality, shape, size and density of quantum wells, wires and dots.

The TEM operates on the same basic principles as the light microscope but uses electrons instead of light. Because the wavelength of electrons is much smaller than that of light, the optimal resolution attainable for TEM images is many orders of magnitude better than that from a light microscope. Thus, TEMs can reveal the finest details of internal structure - in some cases as small as individual atoms.

Imaging

The beam of electrons from the electron gun is focused into a small, thin, coherent beam by the use of the condenser lens [154]. This beam is restricted by the condenser aperture, which excludes high angle electrons. The beam then strikes the specimen and parts of it are transmitted depending upon the thickness and electron transparency of the specimen. This transmitted portion is focused by the objective lens into an image on phosphor screen or charge coupled device (CCD) camera. Optional objective apertures can be used to enhance the contrast by blocking out high-angle diffracted electrons. The image then passed down the column through the intermediate and projector lenses, is enlarged all the way.
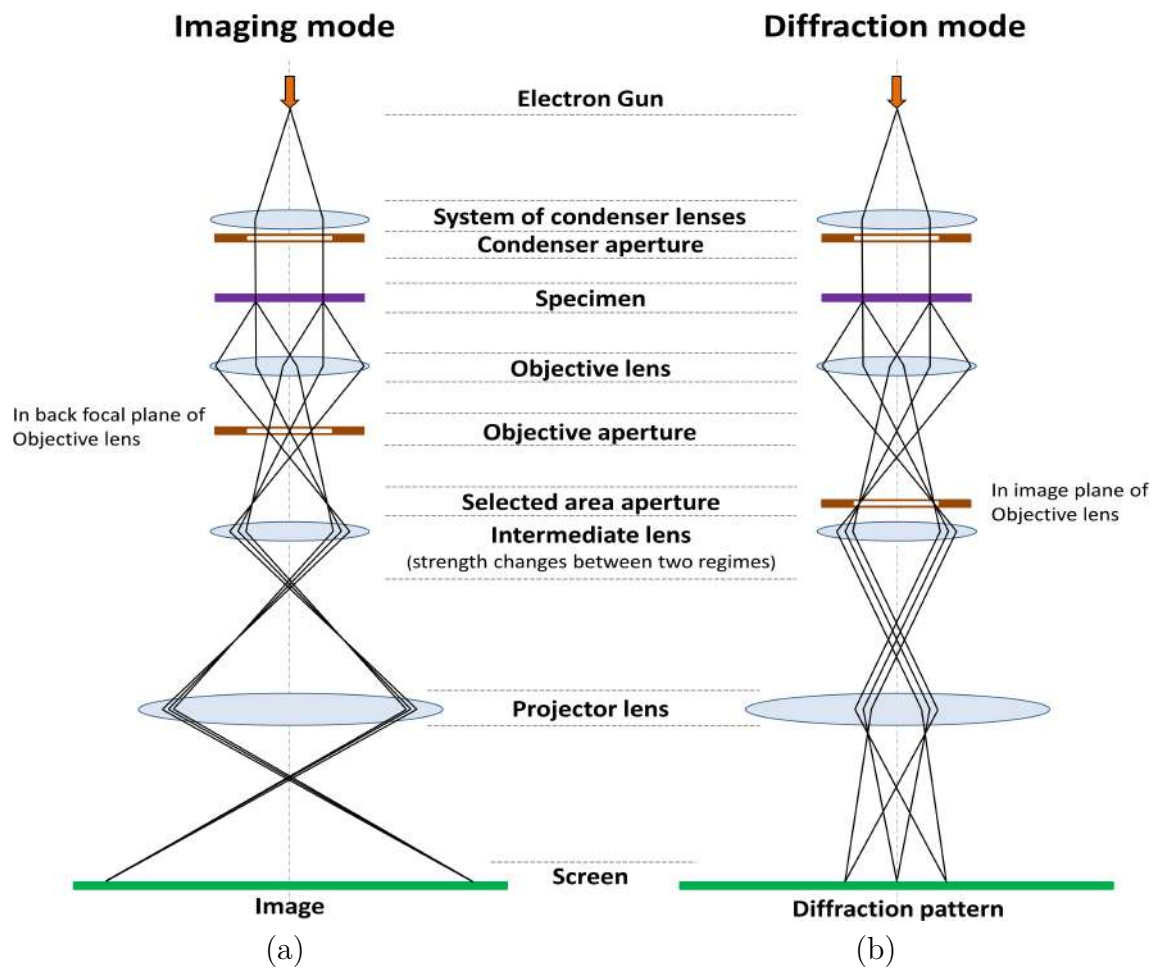
Fig. 3.5 Schematic view of imaging and diffraction modes in TEM. (a) General layout of a TEM describing the path of electron beam in a TEM (Taken from JEOL 2000FX Handbook). (b) A ray diagram for the diffraction mechanism in TEM. Image credit to Eric Kvaalen.

The image strikes the phosphor screen and light is generated, allowing the user to see the image. The darker areas of the image represent those areas of the sample that fewer electrons are transmitted through while the lighter areas of the image represent those areas of the sample that more electrons were transmitted through.

Diffraction

Fig 3.5b shows a simple sketch of the path of a beam of electrons in a TEM from just above the specimen and down the column to the phosphor screen. As the electrons pass through the sample, they are scattered by the electrostatic potential set up by the constituent elements in the specimen. After passing through the specimen they pass through the electromagnetic objective lens which focuses all the electrons scattered from one point of the specimen into one point in the image plane. Also, shown in Fig 3.5 Right is a dotted line where the electrons scattered in the same direction by the sample are collected into a single point. This is the back focal plane of the objective lens and is where the diffraction pattern is formed.

Specimen preparation

A TEM specimen must be thin enough to transmit sufficient electrons to form an image with minimum energy loss. Therefore specimen preparation is an important aspect of the TEM analysis. For most electronic materials, a common sequence of preparation techniques is ultrasonic disk cutting, dimpling, and ion-milling. Dimpling is a preparation technique that produces a specimen with a thinned central area and an outer rim of sufficient thickness to permit ease of handling. Ion milling is traditionally the final form of specimen preparation. In this process, charged argon ions are accelerated to the specimen surface by the application of high voltage. The ion impingement upon the specimen surface removes material as a result of momentum transfer

3.3   Scanning electron microscopy (SEM)

The scanning electron microscope (SEM) developed by Max Knoll in 1935 [155] scanned a focused beam of electrons on a sample surface by magnetic deflection. The contrast of the image was due to the scattered electrons in various parts of the sample. Manfred von Ardenne developed an SEM with submicroscopic resolution in 1938 [156]. In the United States, Vladimir K. Zworykin and his research team developed the first working SEM . In the SEM, when the primary electron beam is incident on the specimen several processes can occur: there can be backscattered electron [149], secondary electrons, Auger electrons, X-rays, unscattered electrons, elastically scattered electrons and inelastically scattered electrons. SEMs can produce a spot size of the electron beam on the sample of 5 to 10 nm. The spatial resolution of the SEM is a function of the size of the electron spot, and that depends on both the wavelength of the electrons and the optical components that form the electron beam. The magnification in an SEM can range from $25 \times$ to $250\,000 \times$. The primary concept of SEM is that of a focused electron beam being scanned across a specimen in a raster approach (Figure 3.6 and Figure 3.7) , that is from top left to bottom right in horizontal lines [157, 149, 158]. Image contrast can be obtained from the backscattered electrons, high energy electrons, from the incident electron beam as heavier atoms (those with higher atomic number) will backscatter more strongly that lighter atoms, and this provides the contrast in the image. They are backscattered out of the specimen interaction volume by elastic scattering due to interaction with the atoms of the specimen [149].

The microscope creates images of a three dimensional objects with a great depth of focus and a resolution between 3 nm and 20 nm [148]. A couple of variants of SEM exist [157], namely focused ion beam SEM (FIB/SEM) and serial block face SEM (SBF/SEM) [131]. The particular advantage to these methods is that the complete sample is in situ within the chamber of the microscope, as opposed to slices of the specimen being placed individually into the microscope to be imaged. From a 3D reconstruction perspective, this means that there are no problems with registration of
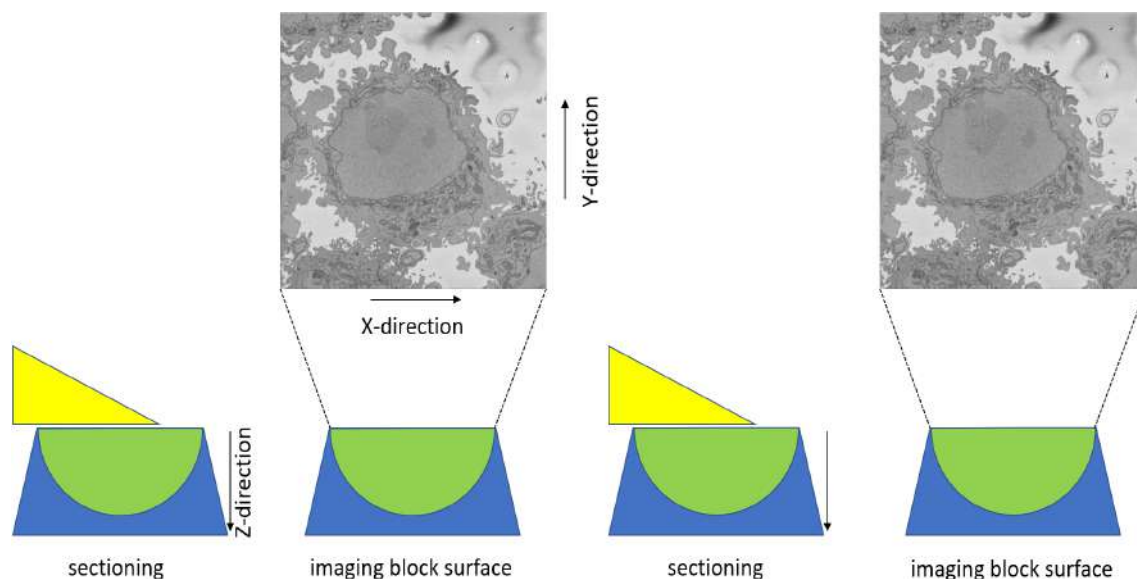
Fig. 3.6 Schematics of Serial Block Face Scanning Electron Microscopy (SBF SEM). In these microscopes, the cell surface is imaged, and then a thin section is cut away, and then the cell is imaged again. A diamond knife is used to cut the sections away in the SBF SEM. These microscopes can cut sections so thin that it would take $2000 - 20000$ sections to cut through the width of a single human hair. This process of cutting and imaging creates a series of 2D images [107].

multiple 2D slices as the specimen is sliced within the microscope itself and is stationary at all times.

The two methods vary only upon how they incrementally remove slices of the specimen. In FIB/SEM, a gallium ion beam is used to 'mill' away material, whereas SBF/SEM, developed and implemented by Denk and Horstmann [41], enables the automation of serial imaging of relatively large volumes with nanometer resolution. SBF SEM utilises an ultramicrotome with a diamond knife to cut and discard very thin slices or sections from the top face of the sample. The sample is then raised to the focal plane to be scanned, and this process is repeated sequentially, resulting in a stack of images through the sample volume [40]. With the advent of SBF SEM and other volume electron microscopy techniques [40, 159, 160], three-dimensional visualisation of specimens with unprecedented detail became possible.

Figure 3.4 shows an electron microscope located at the Francis Crick Institute and this EM acquired HeLa cell images used in this work.
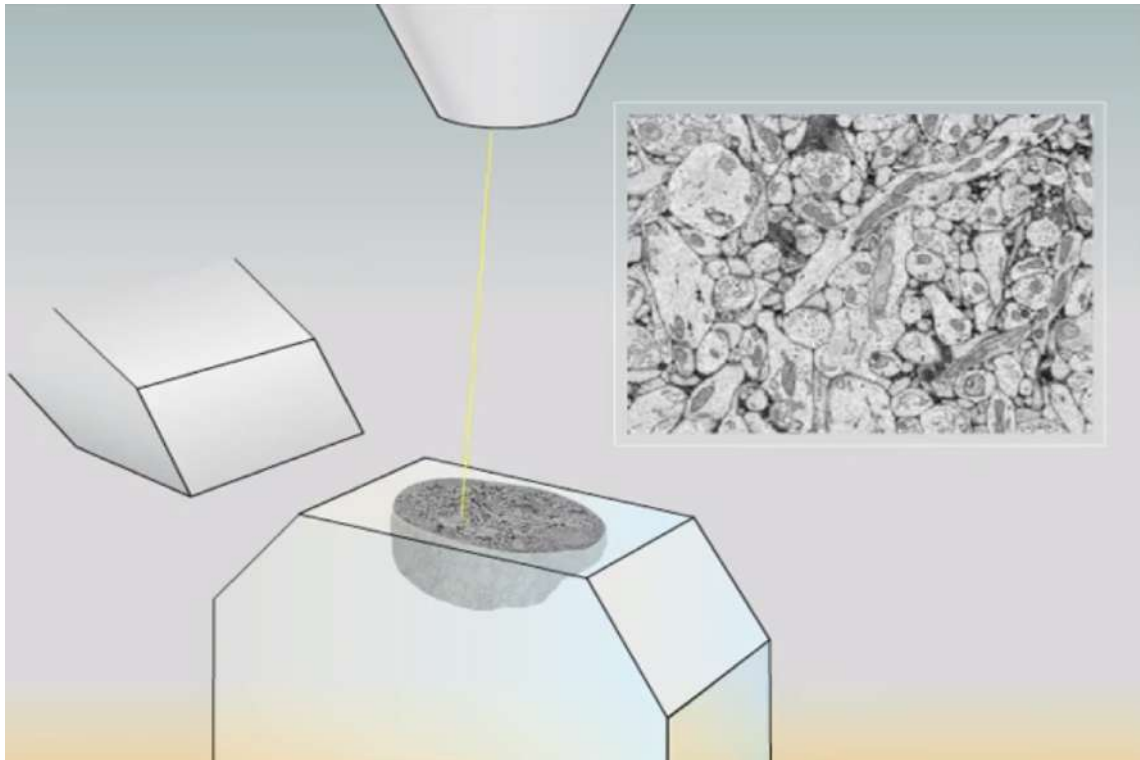
Fig. 3.7 Illustration of Serial Block Face Scanning Electron Microscopy (SBF SEM). SBF SEM utilises an ultramicrotome with a diamond knife to cut and discard very thin slices or sections from the top face of the sample. The sample is then raised to the focal plane to be scanned, and this process is repeated sequentially, resulting in a stack of images through the sample volume [40].
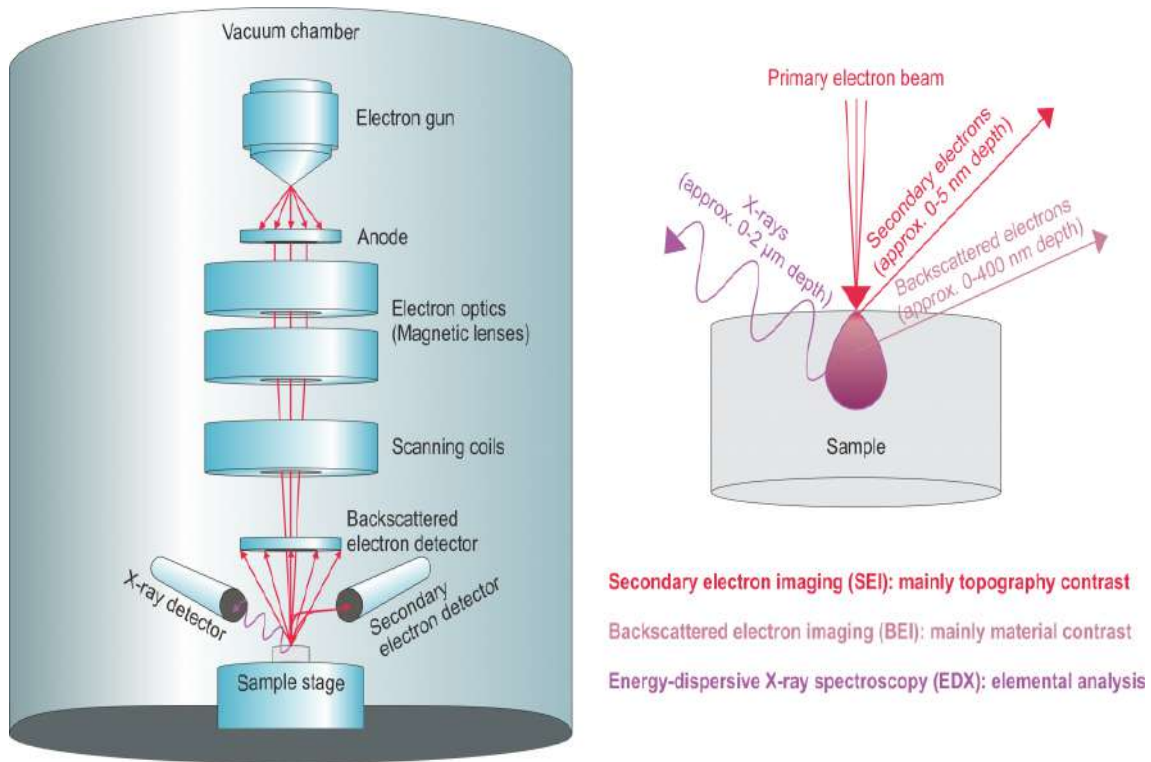
Fig. 3.8 Schematics of a scanning electron microscopy (SEM). In order to increase the mean free path of electrons, the electron optics and sample are placed into a vacuum chamber. Thus, samples have to be vacuum-stable. Depending on the employed detector, SEM provides different image contrasts, which are based on different penetration depths and sample properties. [107].

In scanning-electron microscopy (SEM), images are generated by raster-scanning a focused electron beam over a sample and detecting the secondary electrons emitted or the electrons backscattered by the sample [161]. Because electrons can be focused more tightly than light, SEM provides access to spatial resolutions in the nanometer range.

In order to increase the mean free path of electrons, the electron optics and sample are placed into a vacuum chamber. Thus, samples have to be vacuum-stable. Depending on the employed detector, SEM provides different image contrasts, which are based on different penetration depths and sample properties.

Secondary electron imaging (SEI) In secondary electron imaging (SEI) the secondary electrons emitted from the atoms of the sample are detected. As a part of the excitation energy is consumed for the extraction of electrons from the electron

clouds (work function), with their relatively low kinetic energy secondary electrons can only escape from the top few nanometers of the sample to reach the detector. The image contrast mainly reflects the surface topography of the sample.

Backscattered electron imaging (BEI) The backscattered primary electrons reaching the detector typically represent the top few 100 nm of a sample. Not primarily influenced by surface corrugation, BEI mainly visualizes the distribution of different chemical elements, which is also referred to as material contrast. High-order-number elements scatter the incoming electrons efficiently and therefore appear bright in BE images.

Energy-dispersive X-ray spectroscopy (EDX) SEM instruments often also provide X-ray emission spectroscopy with an energy-dispersive spectrometer (EDX). Sample atoms that absorb incoming electrons can emit X-ray spectra containing element-specific emission lines. Typically, X-rays generated within the top 2μm of a sample can reach the detector. Either single spectra from specific spots or EDX maps representing spatial distributions of chemical elements can be acquired.

3.4   Cryo-electron microscopy

A major problem of using electron microscopy with biological specimens is their sensitivity to radiation from the electron beam [162]. This critical problem is counterbalanced by the high resolution attainable with the electron microscope. In 1981 Dubochet and McDowall introduced the cryotechnique to improve specimen preservation. In this technique cryo-electron microscope (cryo-EM) fires beams of electrons at proteins that have been frozen in solution to deduce the biomolecules' structure. Jacques Dubochet, Joachim Frank and Richard Henderson were awarded the Nobel prize on 4th October 2017 for their work in developing cryo-EM [163]. The biological specimen is embedded in water or a buffer solution by rapidly freezing to the temperature of liquid nitrogen (90 K). The liquid water is converted to an amorphous state and avoids cellular damage due to ice crystal formation. This technique permitted biological

samples to be investigated in their native state without the introduction of artifacts, thereby simplifying image interpretation [149].

3.5   Acquisition of images of HeLa cells

In this work, imaging will be understood as the process of acquiring images [164], which will be analysed at a later stage. The acquired data can be single frames (photos) or time sequences (videos). Imaging allows the acquisition of high spatial resolution and high temporal resolution, allowing high detail and several frames per second to be obtained. This section provides an overview of HeLa cell imaging as a way to visualise the nuclear envelope (NE) of HeLa cells that can produce data in the form of 3D volumes.

Being so widely used, there are many ways in which HeLa cells have been imaged; fluorescence, Phase Contrast, Electron Microscope (EM), however HeLa cells which were used in this research were acquired by Serial Blockface Scanning Electron Microscope (SBF SEM) at the Francis Crick Institute (Figure 3.2b). Schematic diagram of SBF SEM is shown in (Figure 3.6).

HeLa cells were prepared and embedded in Durcupan resin for SBf SEM following the method of the National Centre for Microscopy and Imaging Research (NCMIR)[165]. SBf SEM data was collected using a 3View2XP (Gatan, Pleasanton, CA) attached to a Sigma VP SEM (Zeiss, Cambridge). In total, 517 images of $8,192 \times 8,192$ pixels were acquired. Voxel size was $10 \times 10 \times 50$ nm with intensity $[0-255]$ (Figure 3.9). Initially, the data was acquired at higher bit-depth (32 bit or 16 bit) and after contrast/histogram adjustment it was reduced to 8 bit. For this work, seven individual cells were manually cropped as volumes of interest. For each cell, the centroid was manually selected as the centre of a sub-volume of 300 slices with dimensions $(n_h, n_w, n_d) = (2000, 2000, 1)$ and were saved as single channel *TIff* files. Figure 2.9a shows one representative slice of a 3D stack acquired with serial blockface electron microscope containing numerous HeLa cells fixed in gel in three dimension. In Figure 2.9b region of interest (ROI) with
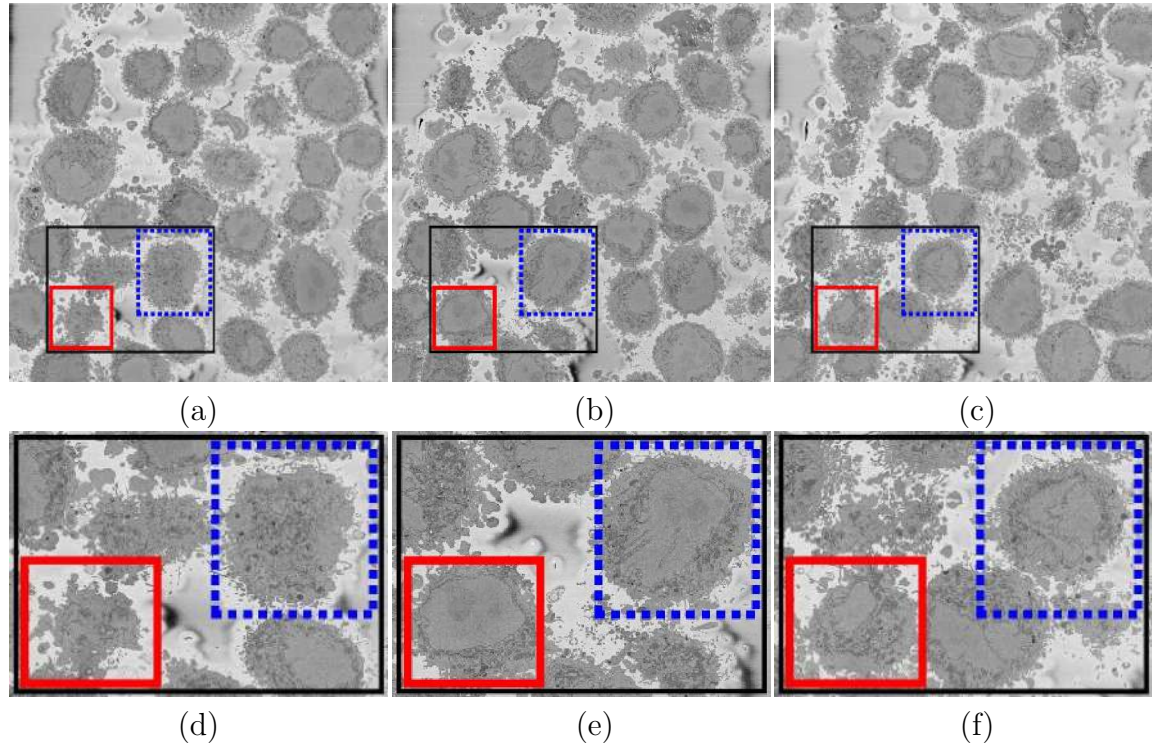
Fig. 3.9 Three representative slices of a 3D image stack acquired by Serial Block Face Scanning Electron Microscope (SBF SEM) containing numerous HeLa cells. Boxes indicate two regions of interest (ROIs), which contain two of the cells that will be posteriorly segmented. (a) A slice on the lower section of the stack; slice 43 of 300. (b) A slice on the central section; 118/300. (c) A slice on the higher section; 241/300. Black box denotes the ROI that is magnified in (d-f). Notice the differences in sizes of cell and nuclei in the images. In particular, the nuclei are hardly visible in (a), largest in (b), and in (c) the nucleus in the blue dotted box appears as several disjoint regions surrounded by a darker nuclear envelope (NE).

one HeLa cell centred is shown. At this resolution, it is easy to distinguish the nuclear envelope, borders of the cell and other structures.

3.6 Chlamydia trachomatis-infected HeLa cells

Preparation and acquisition

Chlamydia trachomatis is the most common cause of bacterial sexually transmitted infection. It produces an unusual intracellular infection: within a membrane-bound compartment called the chlamydial inclusion, the elementary body (EB) converts

into the larger, metabolically-active reticulate body (RB). This RB replicates and then converts into an EB, which is the infectious form. We used quantitative three-dimensional electron microscopy to show that C. trachomatis RBs divide by binary fission and undergo a six-fold reduction in size as the population expands. Conversion only occurs after at least six rounds of replication, and correlates with smaller RB size. These results suggest that RBs only convert into EBs below a size threshold, reached by repeatedly dividing before doubling in size. Our findings support a model in which RB size controls the timing of RB-to-EB conversion without the need for an external signal [166].

Description and Technical details

Serial Block-Face Scanning Electron Microscopy (SBF SEM) was performed on monolayers of C. trachomatis-infected HeLa cells. Stacks of consecutive 60-nm-thick sections were acquired and subsequently digitally aligned, which allowed individual bacteria to be observed and analysed in multiple successive sections. Then all the EM sections were combined computationally into a 3D reconstruction of the inclusion. This analysis provided detailed quantitative information about the C. trachomatis inclusion and its developmental forms.

The preparation of the cell has been published previously [166], but briefly, HeLa cells were grown in Advanced DMEM supplemented with 2% fetal bovine serum and 2 mM GlutaMAX-I in 5% CO_2 at 37^oC. The cell monolayers were infected with Chlamydia trachomatis serovar L2, strain L2/434/Bu at a multiplicity of infection of 3 in sucrose-phosphate-glutamic acid (SPG). Infections were carried out by centrifugation at $700 \times g$ in a Sorvall Legend Mach 1.6 R centrifuge for 1 hour at room temperature. After centrifugation, the inoculum was replaced by fresh cell culture medium and monolayers were incubated at 37^oC and 5% CO_2.

Chlamydia-infected monolayers were fixed in a solution of 2% paraformaldehyde and 2.5% glutaraldehyde in 0.1 M cacodylate buffer, pH 7.4 for 1 hour. Cells were washed 5X in cold 0.1 M cacodylate buffer then incubated in solution containing 1.5% potassium ferrocyanide and 2% osmium tetroxide supplemented with 2 mM calcium
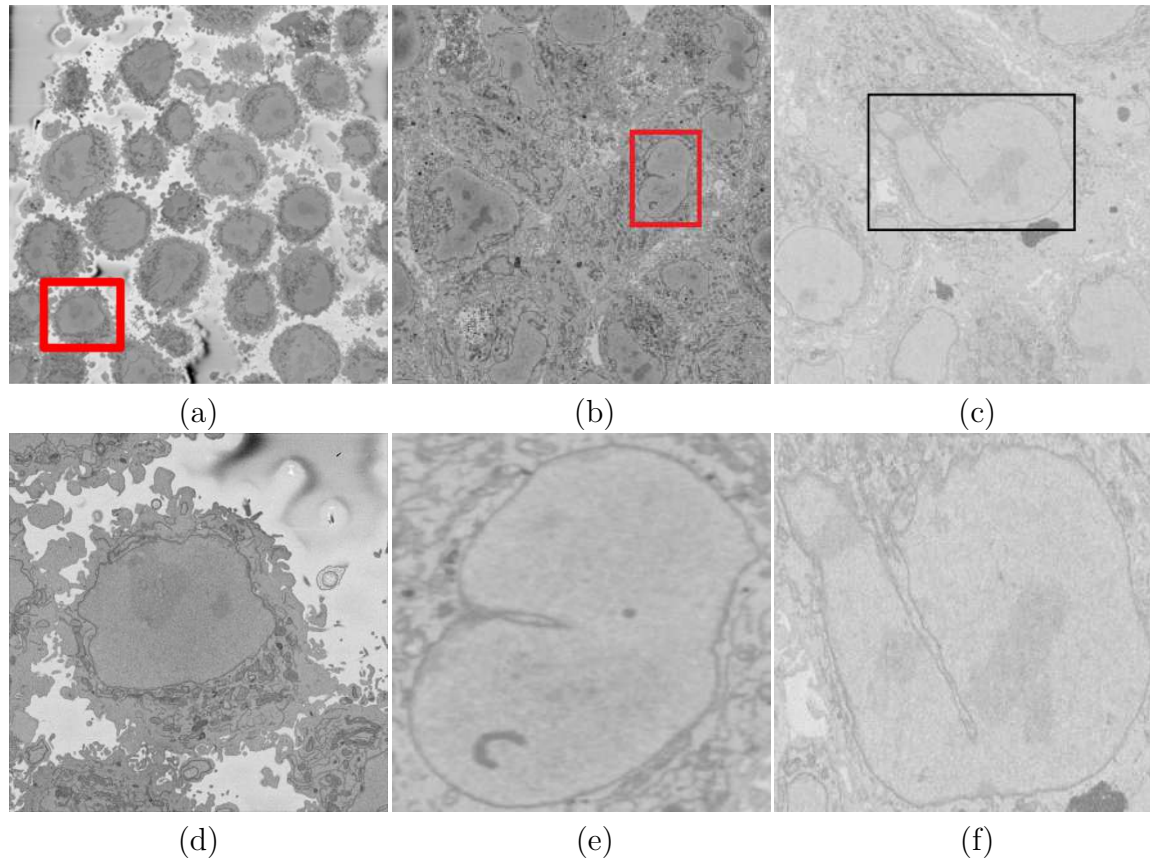
Fig. 3.10 Illustration of the Serial Block Face Scanning Electron Microscope (SBF SEM) images containing Wild-type and monolayers of Chlamydia trachomatis-infected HeLa cells. (a) A representative 8192×8192 image from a 3D stack. The Wild-type HeLa cells are the darker regions and the background is a brighter shade of grey. The red box indicates a region of interest (ROI), that is magnified in (d). (b) A representative 3200×3200 image from The Cell Image Library ($CIL50051$) arranged as 3D stack (413 slices). Cells have 28 hours post infection (hpi) and voxel size $3.6 \times 3.6 \times 60$ nm. The red box indicates a region of interest (ROI), that is part of a cell which was segmented and modelled against a spheroid in this work. (c) A representative 2435×2489 image of the Chlamydia trachomatis-infected HeLa cells from a different data set (CIL50061). Cells in this data set have 12 hpi and voxel size $8.6 \times 8.6 \times 60$ nm. The black box denotes a cell in this slice from this stack (406 slices) that will be posteriorly segmented. (d-f) Detail of the ROIs with a single cell in the centre. The size of magnified images are 2000×2000 and 530×1000. The nucleus is the large and fairly uniform region in the centre and it is surrounded by the nuclear envelope (NE) which is darker than the nucleus.

chloride in 0.1 M cacodylate buffer for 30 min on ice. After 5×2-min washes in doubled distilled water, cells were incubated in 1% thiocarbohydrazide for 10 min at room

temperature. Following 5×2-min washes in double distilled water at room temperature, cells were placed in 2% osmium tetroxide in double distilled water for 10 min at room temperature. The cells were rinsed 5×2 min with double distilled water at room temperature and subsequently incubated in 2% uranyl acetate at $4^{o}C$ overnight. The next day, cells were washed 5×2 min in double distilled water at room temperature and en bloc Walton's lead aspartate staining was performed for 10 min at $60^{o}C$. Following 5×2-min washes in double distilled water at room temperature, cells were dehydrated using a series of ice-cold graded ethanol solutions and then embedded in Durcupan ACM resin. The resin was allowed to polymerize in a vacuum oven at $60^{o}C$ for 48 hours. SBF SEM imaging was completed using a Gatan automated 3View system (Gatan Inc.) and images were recorded at 60 nm cutting intervals. Two representatives of Chlamydia trachomatis-infected HeLa cell images showing several cells and two in boxes, are shown in Figs. 3.10b, c and they are magnified in Figs. 3.10e, f.

3.7 Summary

The observation of the nucleus is a key element in the study of cancerous cells [119] however, the organisation of the nucleus itself remains an area largely unexplored [123]. One of the reasons behind this lack of research is due to the high resolution required to reveal cells' fine structures [148, 149], which in turn requires complex algorithms and difficult to obtain ground truth segmentation.

Electron Microscopy (EM) is an imaging technique that provides resolving power several orders of magnitude higher than conventional light and fluorescence microscopes. Recent developments in semi-automated image acquisition have prompted a revival in the use of EM, transforming biomedical imaging experiments. Serial blockface scanning EM (SBF SEM) was developed and implemented by Denk and Horstmann [41] to enable the automation of serial imaging of relatively large volumes with nanometer resolution. With the advent of SBF SEM and other volume electron microscopy techniques [40, 159, 160], three-dimensional visualisation of specimens with unprecedented detail

became possible. SBF SEM utilises an ultramicrotome with a diamond knife to cut and discard very thin slices or sections from the top face of the sample. The sample is then raised to the focal plane to be scanned, and this process is repeated sequentially, resulting in a stack of images through the sample volume [40].

In the last two sections of this chapter the acquisition of Wild-type of HeLa cells and Chlamydia trachomatis-infected HeLa cells was explained in detail. These two type of cells will be segmented and modelled against a spheroid by using the automatic image processing segmentation algorithm developed in this work and they will be compared statistically in the following chapters.

All SBF SEM Chlamydia trachomatis-infected HeLa cells image data sets can be accessed by downloading from the Cell Centered Database and Cell Image Library under project ID 20099 [32].

The Cell Image Library: *http://cellimagelibrary.org/images/50051*,

The Cell Image Library: *http://cellimagelibrary.org/images/50061*.

The Wild-type of HeLa cell images are available from the following repositories:

EMPIAR: *http://dx.doi.org/10.6019/EMPIAR-10094*.

In the following chapter, an overview of image analysis techniques applied to the HeLa data will be presented. The information empirically observed from the intensities of images will be explored and quantified with the objective of finding the nuclear envelope of HeLa cells to create a 3D volume of each cell. Several algorithms will be tested with the data sets.

Chapter 4

Exploration of the data through image analysis

This chapter presents an exploration of the data through image analysis, showing the processes of image alteration and modification, commonly grouped as pre-processing, segmentation and post-processing. The mathematical foundations for the representation of images will be discussed in detail.

The objective of this chapter is to provide an analysis on segmentation methods through various data sets and evaluate their performance to develop tools for an automatic analysis of data described in Chapter 2.

The exploration of the data touches on disjoint regions of the nuclear envelope of HeLa cells, in which traditional segmentation techniques produce good results. All methods presented in this chapter are evaluated on the HeLa data sets to provide context for the reader in terms of the design choices when developing the tools to analyse the data (Chapter 6); same techniques are also applied in some of the Chlamydia trachomatis-infected HeLa cells data sets.

General notation.

Throughout this work, column vectors will be represented with lowercase bold letters, e.g. $\mathbf{x}, \mathbf{v}, \mathbf{p}$, and matrices will be represented either in uppercase letters, like A, B, C or in blackboard uppercase letters when an image is being referred to, for

example $\mathbb{I}, \mathbb{K}$; where each element can be represented symbolically as $A = ((a_{i.j}))$ where i represents the *ith* row and j represents the j th column in the matrix. Values in images at a certain location, $\mathbb{I}(x, y)$, are referred to as pixels, and are represented by a lowercase p with a lowercase subscript, e.g p_s, p_r. The plane containing the pixels in the image will be called the spatial domain.

In the following section, an overview of image processing techniques is presented, centred in particular around image segmentation. Processes performed prior and after segmentation are called pre- and post- processing techniques. The section is thus separated in all three categories.

4.1   Classical image analysis

Image processing is an essential field in many applications, including medical imaging, astronomy, astrophysics, surveillance, video, image compression and transmission, just to name a few [127]. In one dimension, images are called signals. In two dimensions we work with planar images, while in three dimensions we have volumetric images (such as MR images). These can be grey scale images (single-valued functions), or colour images (vector-valued functions). Noise, blur and other types of imperfections often degrade acquired images. Thus such images have to be first pre-processed before any further analysis and feature extraction.

In this section we will formulate in mathematical terms several image processing tasks: image de-noising, image de-blurring, image enhancement, image segmentation, edge detection. We will discuss techniques for image filtering in the spatial domain (using first- and second order partial derivatives, the gradient, Laplacian, and their discrete approximations by finite differences, averaging filters, order statistics filters, convolution), and in the frequency domain (the Fourier transform, low-pass and high-pass filters), zero-crossings of the Laplacian.

In the last part of the chapter, we will present more advanced methods that can be formulated as a weighted Laplace equation for image restoration, and curve evolution

techniques called snakes for image segmentation (including total variation minimisation and active contours without edges).

Three main stages in image processing can be recognised, which will be analysed in the following sections. The stages are pre-processing, which refer to the intensity based operations performed on an image to aid in the segmentation techniques perform better. Then, the core process of image segmentation will be discussed; such process involves the classification of each pixel into one of two categories: background and foreground. Finally, a post-processing stage follows in which small errors in segmentation are addressed, this stage usually involves a type of binary operation.

Following sections present the relevant techniques in each category, applied to the data presented in sections 3.5, and 3.6.

Figure 4.1 displays the test frame which will be used to illustrate many of the techniques described herein. As the objective is comparison, a region of interest of the image will be presented, as it showcases the data set's principal problems.

As the objective is comparison, a region of interest of the image will be presented, as it showcases the data set's principal problems.

4.1.1   Representation of images

The objective of this section is to provide a homogeneous representation of digital images and the operations that can be performed in them; the overview includes the notation that will be utilised throughout the dissertation.

An image can be seen as a digitalisation of a continuous function in two variables [127] corresponding to the spatial coordinates in an image. Let $f : \mathcal{I}_X \times \mathcal{I}_Y \to \mathbb{R}, f(X, Y)$ represent a function in two continuous variables, (X, Y), over two intervals, represented by $\mathcal{I} \subset \mathbb{R}$, in the Cartesian plane. The intervals can represent a rectangular section of the plane. For simplicity, and without loss of generality, let $\mathcal{I}_X = [0, a]$ and $\mathcal{I}_Y = [0, b]$ where a and b are numbers such that $a, b > 0$, positioning the rectangle $\mathcal{I}_X \times \mathcal{I}_Y$ in the first quadrant of the plane. The range of f is assumed to be the range of values $[0, g]$.

(a) (b)
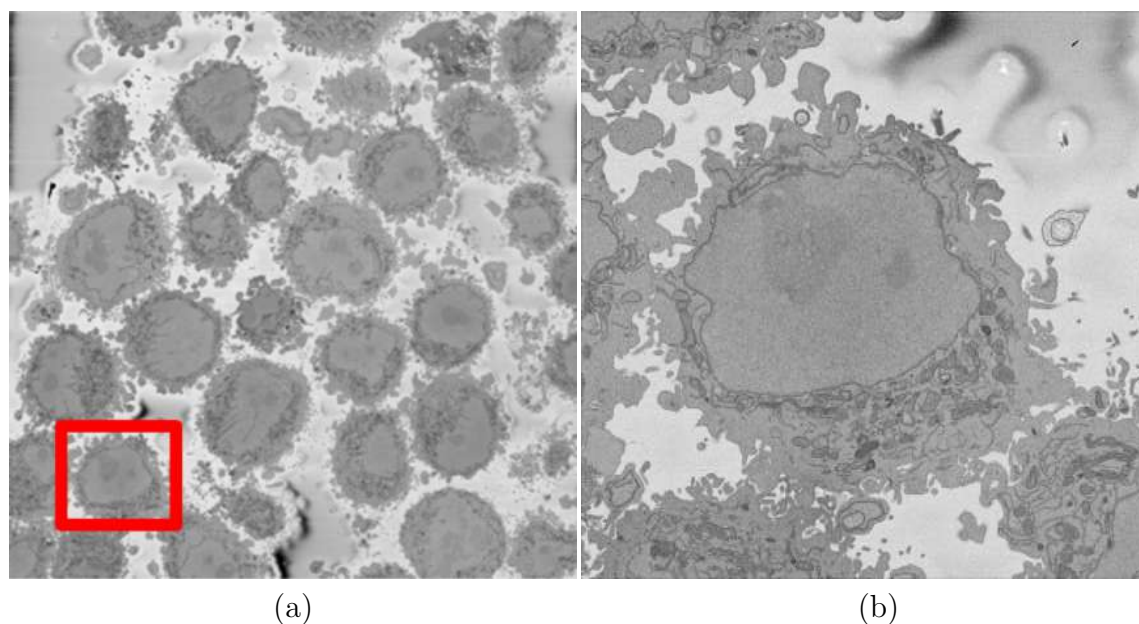
Fig. 4.1 Presentation of test frame and the cropped images used in testing throughout this chapter. The frame will be referred to throughout this work. (a) A representative slice of a 3D image stack acquired by Serial Block Face Scanning Electron Microscope (SBF SEM) containing numerous HeLa cells. This image represents the full size frame from Wild-type of HeLa data set. Red box represents the region of interest (ROI) on which different techniques will be presented and this ROI will be magnified in (b). (b) Detail of the ROI with a single cell in the centre. The nucleus is the large and fairly uniform region in the centre and it is surrounded by the nuclear envelope (NE) which is darker than the nucleus.

(a)   (b)
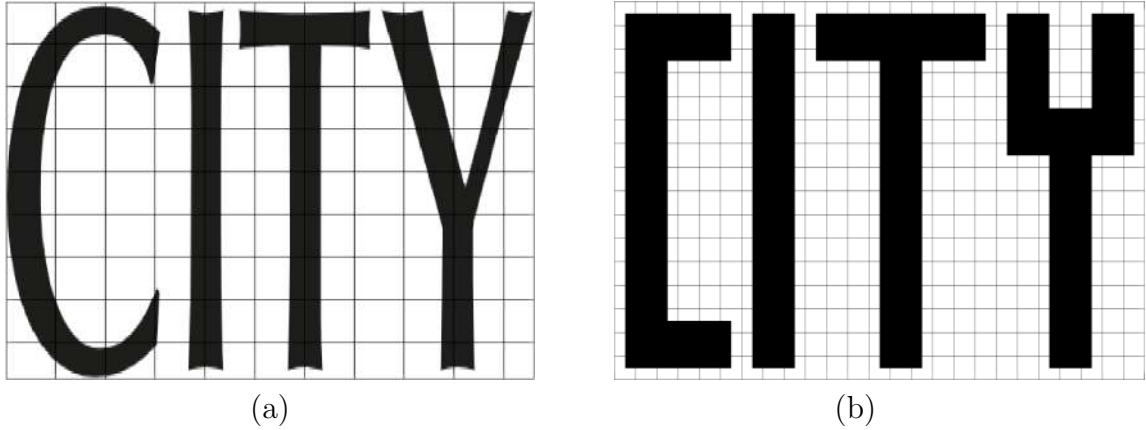
Fig. 4.2 Representation of the creation of a digital and binary image. (a) A representation of the continuous function *f*, the grid represents a sensor array in which the image will be projected. Continuous function $f(X, Y)$. (b) A digital image in which a shape can be observed and others can be appreciated as background. The size of this image is 25×26 meaning there are 650 pixels. A binary image in which the pixels corresponding to the object have been set to 0 (black pixel) and the background pixels have been set to 1 (white pixel). Digitalised image pixels $\mathbb{I}(x_i, y_i)$ show quantised intensities - 0 and 1 as a binary image in this example. In this image the function *f* is digitalised as a matrix $\mathbb{I}(x, y)$, given the sensor array.

The digitalisation of *f* consists of (i) sampling, (ii) discretisation in the (X, Y) variables; and (iii) quantisation, which is a discretisation of the range of *f*. Let the discrete variables (x, y) correspond to the spatial positions in $\mathcal{I}_X \times \mathcal{I}_Y$ where the samples of *f* are taken. For simplicity, $x = 1, 2, \cdots, N_r$, where N_r is the number of rows and $y = 1, 2, \cdots, N_c$, where N_c is the number of columns. The intensities of the image will be quantised into the finite set $I = 1, 2 \cdots, L$. For example, in 8 -bit images, the quantisation results in the interval $I = 1, 2, \cdots, 255$. Throughout this work, all digitised images will be referred to with the symbol $\mathbb{I} = \mathbb{I}(x, y)$.

Figure 4.2 shows a representation of the digitalisation of an image. In the figure, the arrangement of the digital coordinates (x, y) would represent the image as a matrix $\mathbb{I}$ of size $N_r \times N_c$

Digital images are generated when an array of light sensitive sensors capture the light entering them. The number of sensors in the array corresponds to the number of positions in the resulting image. Each position in the image is called a pixel, and it is

represented by the symbol $p_x = \mathbb{I}(x, y)$. The number of pixels in an image depends on the resolution of the sensor array.

Colour images

Colour is perceived by humans by the white sun light which is reflected from the objects it hits [127]. The spectrum of colours forms the different colours which blend smoothly into each other, as the frequency of the light wave increases; however, seven colours can be broadly separated in the visible spectrum: red, orange, yellow, green, blue and violet.

A colour space is the specification of a coordinate system where each colour is represented by a single point. The RGB colour space is based on a three dimensional Cartesian space in which each coordinate represents the intensity of a primary value, R for red, G for green and B for blue. Each position (pixel) (x, y) on an image will have three values to it, each one to represent the colour. This implies that an image will be a 3D matrix $\mathbb{I}(x, y, c)$ of size $N_r \times N_c \times 3$, each of the three $N_r \times N_c$ will be referred to as layers.

Binary images Levels of intensities in images are created in terms of a given set $I = l_1, l_2, \ldots, l_L$, in general either each $l_i = i - 1$ or they are bounded between zero and one, $l_i = (i - 1)/L$. This section presents the special case where $L = 2$, without loss of generality, $I = \{0, 1\}$. Binary images can be used to provide context to scenes in images, where regions of interest can be defined as foreground and the rest can be defined as background. In Figure 4.2 a representation of a binarised image where some some pixels are created as foreground and others as background is illustrated.

The connected region in the binarised image will be referred to as detection or object, and it can be modelled as a set in a 2D plane with a size equal to the number of pixels in it. Building a binary image can be done either by manually selecting which pixels will be assigned as 1 or done automatically by a computer algorithm; which incorporates context to the image by classifying some pixels as positive, or "1", or negatives, or "0".

A simple image formation model

By image, we understand the usual intuitive meaning - an example might be the image on the human eye retina or the image captured by a camera [128]. The image can be modelled by a continuous function of two or three variables; in the simple case arguments are the co-ordinates *(x,y)* in a plane, while if images change in time a third variable *t* might be added.

In the continuous case, a planar image is represented by a two-dimensional function $(x,y) \rightarrow f(x,y)$. The value of *f* at the spatial coordinates *(x,y)* is positive and it is determined by the source of the image. If the image is generated from a physical process, its intensity values are proportional to energy radiated by a physical source. Therefore, *f(x,y)* must be nonzero and finite:

$0 < f(x,y) < \infty$

The image-function *f* may be characterised by two components:

1. the amount of source illumination incident on the scene (illumination) *i(x, y)*

2. the amount of illumination reflected by the objects (reflectance) *r(x, y)*

We have $f(x,y) = i(x, y)r(x, y)$ where $0 < i(x, y) < \infty$ and $0 < r(x, y) < 1$.

Reflectance is bounded below by 0 (total absorption) and above by 1 (total reflectance). *i(x, y)* depends on the illumination source *r(x, y)* depends on the characteristics of the imaged objects. The same expressions are applicable to images formed via transmission of the illumination through a medium (chest X-ray, etc). Then we deal with transmissivity instead of reflectivity. From the above construction, we have

$$L_{\min} \leq l = f(x,y) \leq L_{\max}$$

where $l = f(x, y)$ is the gray-level at coordinates *(x, y)*. It is common to shift the gray-scale (or intensity scale) from the interval $[L_{\min}, L_{\max}]$ to the interval $[0, L-1]$. Then $l = 0$ is considered black and $l = L - 1$ is considered white on the gray scale. The intermediate values are shades varying from black to white.

An image is a grid of points or elements and each of them will contain a certain value [126]. The value of the element is related to the colour or intensity of the image

itself. The grid is called a "matrix", and each point of the grid is called a "pixel", which is the combination of the words "picture element".

In digital images, the size of the grid will define the resolution of the image. For example, it may be a square of 128×128, 256×256 or 512×512 pixels or it may depend on the resolution of the cameras used to capture the image. When data sets contain volumetric information, like those acquired from magnetic resonance images or multiphoton microscopes, the elements are related to a volume and thus the "volume elements" are called "voxels". In the general case an image can be of size m-by-n, 'm' pixels in the rows or vertical direction and 'n' pixels in the columns or horizontal direction.

The simplest possible image representation is a binary image, where each pixel has a value of either 0 or 1 [126–128, 167]. Binary images require very little storage space but are usually only the result of some image-processing techniques that aim to identify salient features.

A grey scale image has values in the range 0 to 255. For a moderate amount of detail each value can be stored in a byte and can take any integer value between 0 and 255; such an image is an 8-bit grey scale image. More detail is obtained by increasing each value to be represented by 16-bits, with the drawback of increased image storage space required. Grey scale images are more commonly found in applications than binary images and often require more complex processing techniques to aid their understanding.

Colour images contain important information about the perceptual phenomenon of colour related to the different wavelengths of visible electromagnetic spectrum [126]. In many cases, the information is divided into three primary components, Red, Green and Blue (RGB), or psychological qualities such as hue, saturation and intensity (HSI). Full colour images in the RGB model have three components of colour; red, green and blue. Each colour component is in the range 0 to 255, giving each pixel value a total of 3 bytes to store. Having colour images available can often simplify the application of computer vision, however more complex processing techniques may be required.

An alternative representation of an image is a histogram of pixel intensities. For a grey scale image, the histogram is a plot of intensity (in the range $0 - 255$) against the frequency of that particular intensity value in the image [128]. The intensity histogram can be useful as a simple representation of global information in the image, from which particular values can be extracted, and a visualisation of the effects of image transformations.

4.2   Ground truth (GT) and labelled images.

The ground truth (GT) is a binary image, which contains the correct labelling of objects of interest. Normally, an image is labelled manually by an expert in the field. A ground truth for one cell was provided for the Wild-type of HeLa data set (shown by the red box in Figure 4.3). During the work, a GT for another cell (shown by the blue box in Figure 4.3) was segmented by the author of this dissertation with a Wacom pen-and-tablet. In total, 300 slices were segmented in around 47 hours. The GT software, which is based on MATLAB® *roipoly* function, allows the user to manually label images of cells. Figure 4.3c, d show masks (GT or GT masks) of two different cells shown in the black box in Figure 4.3a, b.

In order to provide training data for deep learning architectures, another GT was obtained by the author this time in MATLAB® *Image Labeler*. In this GT the cell was labelled with which four classes (nuclear envelope, nucleus, rest of the cell, background) (Figure 7.1d). The GT was replicated to create an image with three channels to be consistent with the RGB images commonly used with pre-trained neural networks.

An example of both manually segmented image and labelled image for deep learning architectures can be seen in Figure 4.4. It is important to notice that the ground truth was generated from a non-expert's perspective. in which the shapes of the cells was easily recognisable. This part of the dissertation development some of the difficulties of dealing with the data set.

Comparison of binary images.

(a)   (b)

(c)   (d)
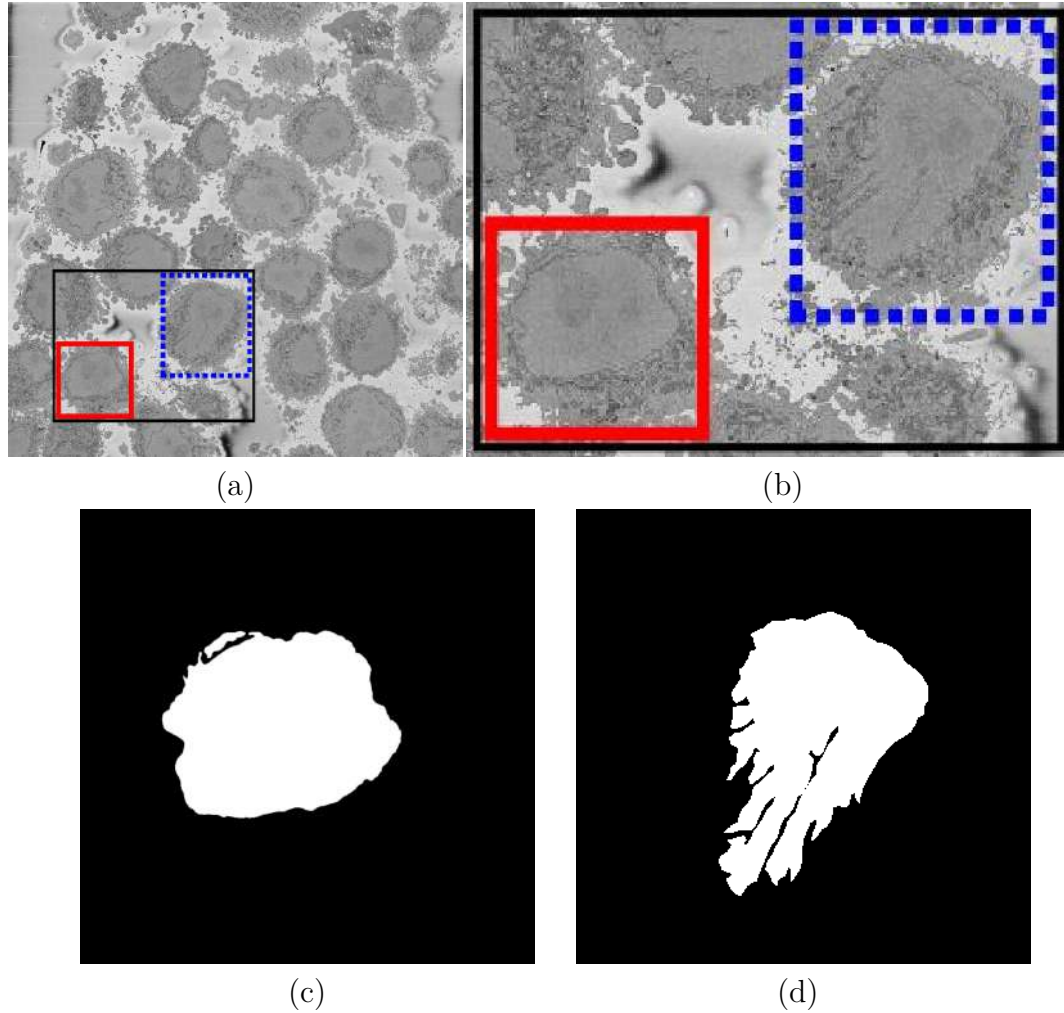
Fig. 4.3 Manual segmentation or ground truth (GT) of a HeLa cell for two different slices. A representative slice of a three-dimensional (3D) image stack acquired by Serial Block Face Scanning Electron Microscope (SBF SEM) containing numerous HeLa cells. Boxes indicate two regions of interest (ROIs), which contain two of the cells that were manually segmented by Anne E. Weston at the Francis Crick Institute and the author of this dissertation in this work. (a) A slice on the central section; 118/300. Black box denotes the ROI that is magnified in (b) Notice the differences in sizes of cell and nuclei in the images. In particular slice, the nuclei are the largest. The nucleus in the blue dotted box has several disjoint regions surrounded by a darker nuclear envelope (NE). (c) GT for this particular slice showing the nucleus in red box.(d) GT for this particular slice showing the nucleus in blue dotted box. The NE was obtained as the boundary of the nucleus.
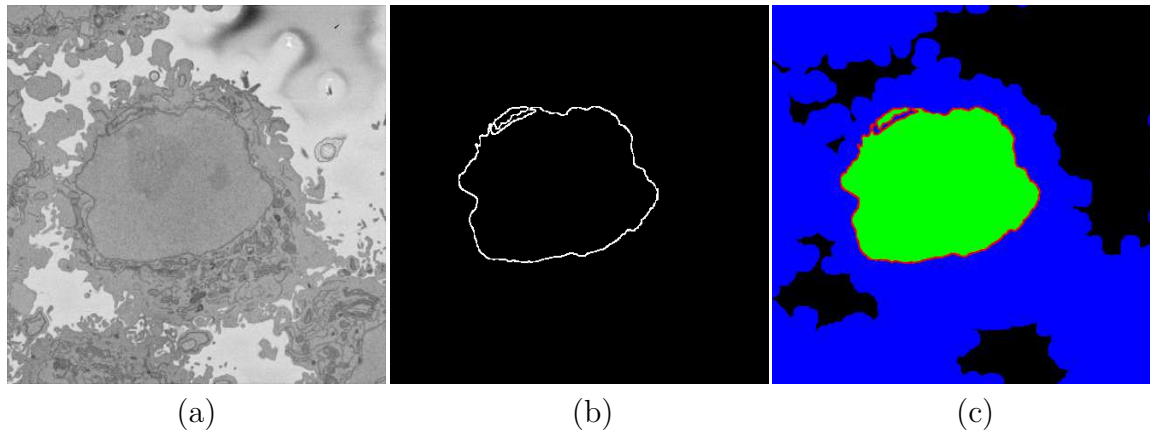
Fig. 4.4 Illustration of the process followed to generate the ground truth. (a) Filtered HeLa image - slice 118/300. (b) Manually delineated nuclear envelope (NE). (c) Labelled image, generated in MATLAB® *Image Labeler*, indicating the four different classes - nuclear envelope (red), nucleus (green), rest of the cell (blue), and background (black).

In Section 2.4, the definition of ground truth (GT) was provided both in general and in the context of the data used in this work. In practical terms, given an image which contains regions to be automatically segmented, its ground truth will be a binary image of the same dimensions where all the regions were selected manually by a human, in some cases, an expert. The output of a detection software is called a segmentation, the methodologies will be explained in Section 3.1.

Manual annotations are then compared with the output of segmentation algorithms. There are various ways of comparing the annotated images with segmentation outputs, which involve looking at images like sets. In simple terms, the analysis can be done comparing each pixel in both the ground truth and the algorithm's output image. Four possibilities, depending on the comparisons: (i) True positive (TP) as the pixels where both the output and ground truth show a positive or 1; (ii) True negative, (TN) in pixels where both images show a negative or 0; (iii) False positive, (FP) where the segmentation shows a 1 but the ground truth shows a 0, and (iv) False negative (FN) where the segmentation shows a 0, but the ground truth shows a 1.

Counting the number of pixels that correspond to each category can produce metrics for the performance of a segmentation, or how close the automatic output is from a given manual annotation.

The Jaccard similarity index [168], or simply Jaccard index, is defined in terms of sets and their sizes, as the ratio of the intersection by the size of the union. On the other hand, in the work by Fawcett [169], two measurements are presented: precision and recall. Precision measures how many detected pixels are relevant, by computing the ratio of true positives and the overall number of detected pixels, TP + FP. Recall computes the ratio of true positives with the sum of the true positives and the false negatives — or points that should have been detected, but were missed —, this measurement can be seen as a ratio of the relevant pixels that were detected. Finally, Randen and Husoy [170] present accuracy which compares the number of correct detections, whether positives or negatives against the overall number of pixels. All measurements can be understood from the diagram in Figure 7.6, the mathematical formulation is given in the following section.

An important part of this work involves the automatic detection of cell structures in images where the cells have disjoint regions. Therefore detections must be compared carefully to the ground truth, as these regions could lead to confusions and lower the similarity indices. The notion of the performance measurements will be extended in the following sections to address such problem.

4.3   Segmentation performance metrics

Given two input curves that are perimeters of segmentation result and manual segmentation or GT, it is natural to ask how similar they are to each other. In order to assess the accuracy of the segmentation, two methodologies were followed. first, the Jaccard similarity index (JI) (Figure 4.5a) of intersection over union was calculated to assess the area inside the NE and second, the Hausdorff distance (HD) (Figure 4.5b) , a well known similarity measure, of the maximum of the set of shortest distances between

two lines was calculated to assess how far the real boundary of the NE was from the calculated one. The smaller the Hausdorff distance between two shapes the greater is their degree of resemblance [171].

The JI [168] is defined by Equation (4.1):

$$\text{Jaccard Similarity Index} = \frac{TP}{TP + FP + FN} \tag{4.1}$$

where *TP, FP, FN* stand for true positive, false positive and false negative (Figure 4.5a) . HD [172, 173] is defined as the maximum distance between a point on one curve and its nearest neighbour on the other curve [174, 175] (Figure 4.5a). It is such a metric that enables calculation of how far two subsets of a metric space are from each other and defined by Equation (4.2):

$$d_H(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right\} \tag{4.2}$$

Figure 7.6 shows the possible characterisation of the output of a segmentation algorithm, given its output and the value of the same pixel in the ground truth. Four possibilities, depending on the comparisons: (i) True positive (TP) as the pixels where both the output and ground truth show a positive or 1; (ii) True negative, (TN) in pixels where both images show a negative or 0; (iii) False positive, (FP) where the segmentation shows a 1 but the ground truth shows a 0, and (iv) False negative (FN) where the segmentation shows a 0, but the ground truth shows a 1.

Counting the number of pixels that correspond to each category can produce metrics for the performance of a segmentation, or how close the automatic output is from a given manual annotation. In the works by Jaccard [168], Fawcett [169] and Randen and Husoy [170], such measures are provided. First, the Jaccard similarity index, or simply Jaccard index, is defined in terms of sets and their sizes, as the ratio of the intersection by the size of the union. In terms of the statistics described before it can be computed by the number of true positives divided by the sum of true positives, false positives and false negatives.

<center>(a)   (b)</center>
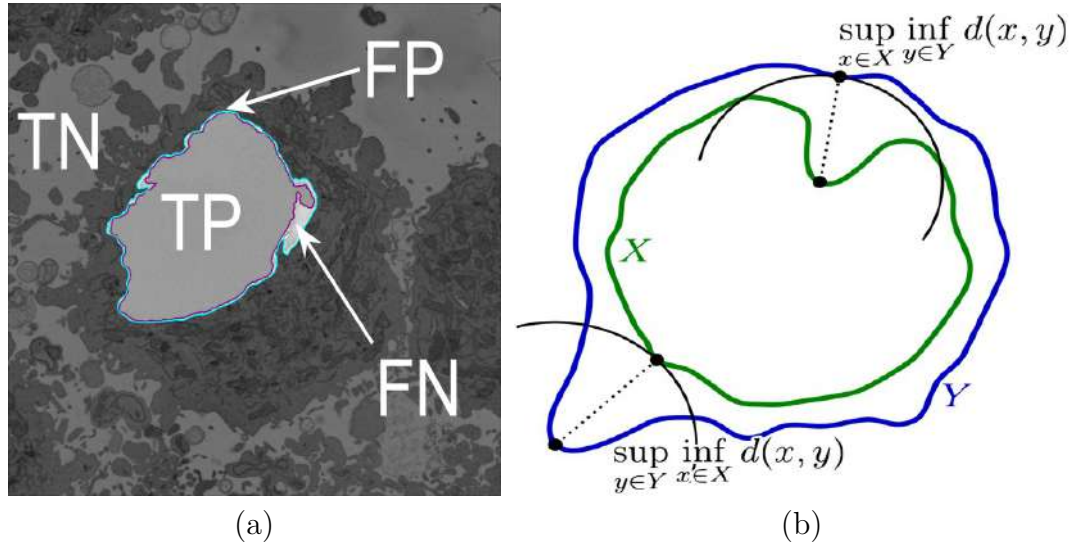
Fig. 4.5 Illustration of the pixel-based metrics. Schematics of Jaccard similarity index [176] and Hausdorff distance [177] used in this work to asses the algorithms. Jaccard similarity index, or intersection over union in (a) Segmentation overlaid on a slice (slice 184/300 - on a filtered image): ground truth (GT) in cyan, automated segmentation in purple. True Positives (TP, nuclear pixels segmented as nucleus), true negatives (TN, background pixels segmented as background), false positives (FP, background pixels segmented as nucleus) and false negatives (FN, nuclear pixels segmented as background). These quantities were used to compute accuracy and Jaccard similarity index for the image-processing algorithm and three pre-trained deep neural networks. (b) Components of the calculation of the Hausdorff distance (HD) between the green line X and the blue line Y. The HD is the greatest of all the distances from a point in one set to the closest point in the other set. Image credit to Rocchini.
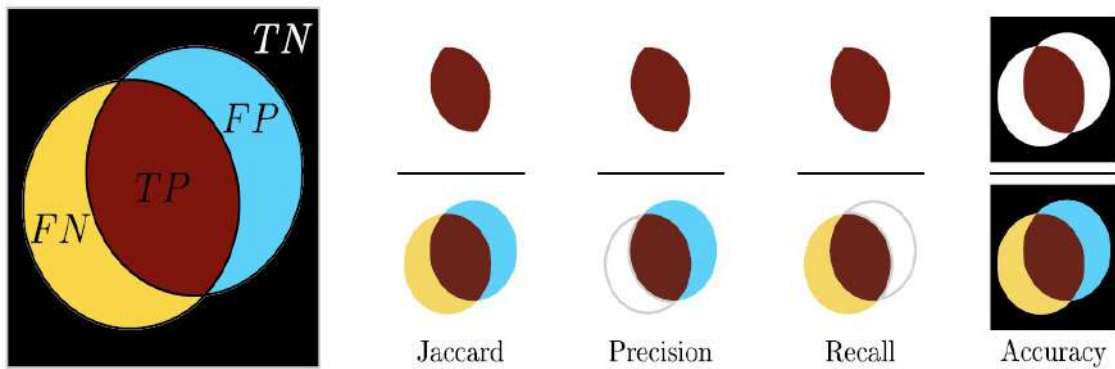
Fig. 4.6 Graphical representation of the Jaccard index, Precision and Recall measurements. The diagram shows an abstract representation of the sets of pixels in an image, and how they can be classified. The ratios described in Equation (4.1), Equation (4.3), Equation (4.4), and Equation (4.5) is shown, based on this graphical representation. Images credit to [2].

On the other hand, in the work by Fawcett, two measurements are presented. First of all, precision, which measures how many detected pixels are relevant, by computing the ratio of true positives and the overall number of detected pixels, TP + FP. Second of all, the work presents the recall measurement, which computes the ratio of true positives with the sum of the true positives and the false negatives — or points that should have been detected, but were missed—, this measurement can be seen as a ratio of the relevant pixels that were detected.

Finally, Randen and Husoy present accuracy or mis-classification which compares the number of correct detection, whether positives or negatives against the overall number of pixels. Equation (4.1), Equation (4.3), Equation (4.4), and Equation (4.5) describe all measurements previously described, and can be observed in Figure 7.6.

$$\text{Overall Accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \tag{4.3}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.5}$$

In the first dataset there were 300 images of a HeLa cell. To train the image processing algorithm, the manual ground truth of the NE were overlaid on the original images so each ground truth has two classes - NE and background for binary classification (Figure 4.4b. In order to train deep learning architectures original HeLa images were labelled in MATLAB® *Image Labeler* with four different classes - NE, nucleus, rest of the cell and background (Figure 4.4c. As Equation (4.3) is overall accuracy and there was only one positive class (NE) for the image processing algorithm segmentation. On the other hand, if there's more than one positive class ('object'), the presented accuracy metrics do not reflect it, as they are used for binary classification only, i.e. if the model outputs a multiclass prediction (softmax), then the false positive is not just low IoU, but also false class prediction with high IoU.

4.4 Summary

This chapter described image analysis techniques to segment objects from background. The chapter was utilised as a thorough exploration of the data presented in Chapters 2 and 3, in particular Wild type of HeLa cell images. As shown, HeLa data sets involve several challenges. Most of them are due to the imaging technique used, which caused an uneven distribution of the intensities. However, such also allow to observe cellular structures from a different perspective.

Challenges observed included: disjoint regions due to cell division, uneven segmentation and some holes or incomplete segmentation of the cells and inconsistencies in the imaging towards the upper right and lower left corners of the images.

A traditional image analysis approach, including pre-processing, thresholding, and post-processing, was described thoroughly. Various techniques per stage were discussed. This segmentation approach could overcome some of the challenges in the data set, such as noise before and after applying a threshold. However, it is not enough to solve the problem of disjoint regions, since a threshold will categorise intensities into specific classes. Adaptive thresholds did not prove more useful than hysteresis thresholds and had the downside of being time consuming.

This chapter presents the materials used in this project. The chapter includes of four main topics: (i) the mathematical representation of images, (ii) the main data sets of HeLa cells analysed in this work, and the secondary data sets used (Chlamydia trachomatis-infected HeLa cells); (iii) the representation of cell boundaries and (iv) the generation of ground truth.

Understanding the context of acquisition and cellular processes in the previous chapter enhanced the familiarisation of the analysis with the data. Furthermore, the presentation of the creation of digital images was of useful to understand the broad context of digitalisation and introduce some key concepts and notation, such as size of images or colour space. Segmentation, which will be addressed in the following chapter, can be defined as the creation of a binary image by classifying certain pixels from others.

Two collections of data sets were presented, corresponding to preliminary work and main focus. Data sets includes the Crick EM Wild type HeLa cell images and the Cell Image Library mono layers of Chlamydia trachomatis-infected HeLa cells.

In the following chapter, an overview of Deep Learning architectures applied to the data set will be presented. The information empirically observed from the intensities of HeLa cells will be explored and quantified with the objective of finding the overall shape of the cell. Several pre-trained deep learning architectures will be tested with the data sets.

Chapter 5

Image-processing Segmentation Algorithm

In this chapter, the image-processing algorithm developed in this work will be described in detail. The image-processing algorithm followed a pipeline of several traditional steps like edge detection, dilation and morphological operators and some of these are explained in Chapter 4 and the rest in appendix B.

The flow chart of the image-processing segmentation algorithm developed in this work is shown in Figure 5.1.

The chapter is subdivided into two major sections related with (i) the segmentation of nuclear envelope of HeLa cells through traditional image processing algorithm, (ii) the modelling of segmented nuclear envelope of HeLa cell against a spheroid and its analysis and the foremost conclusions in such work.

5.1   Image processing algorithm

The details of the data sets are displayed in Table 5.1. Images were provided by two different EM sources and were used to segment the nuclear envelope of different type of HeLa cells.
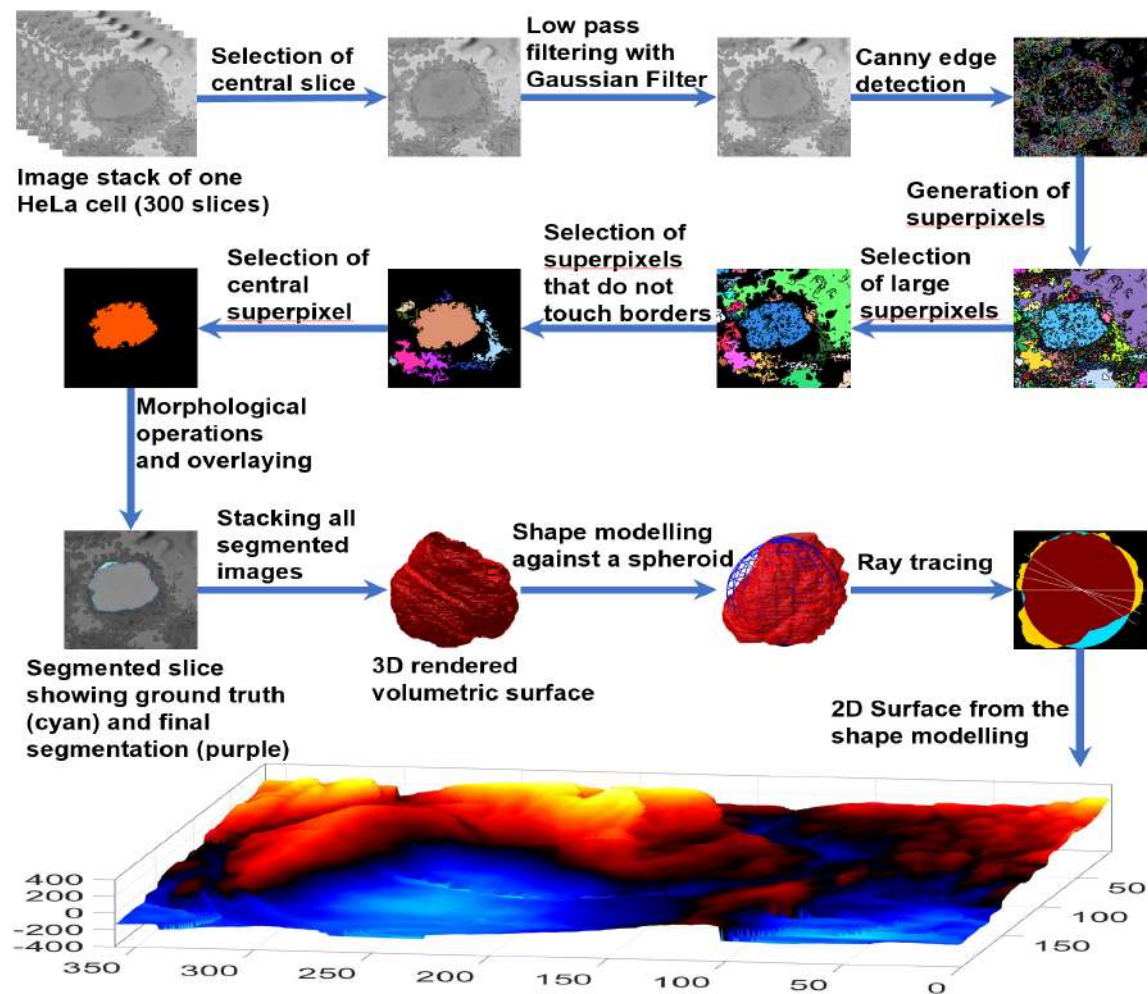
Fig. 5.1 Flow chart for the image-processing algorithm developed in this work.

Table 5.1 Summary of the data sets that were used in this work.

Image source	No. of images	Voxel size	No. of classes
Francis Crick Institute	300	$10 \times 10 \times 50$ nm	2 and 4
Cell Image Library (CIL50051)	406-28 hours post infection	$3.6 \times 3.6 \times 60$ nm	2
Cell Image Library (CIL50061)	413-12 hours post infection	$8.6 \times 8.6 \times 60$ nm	2

(a) (b)
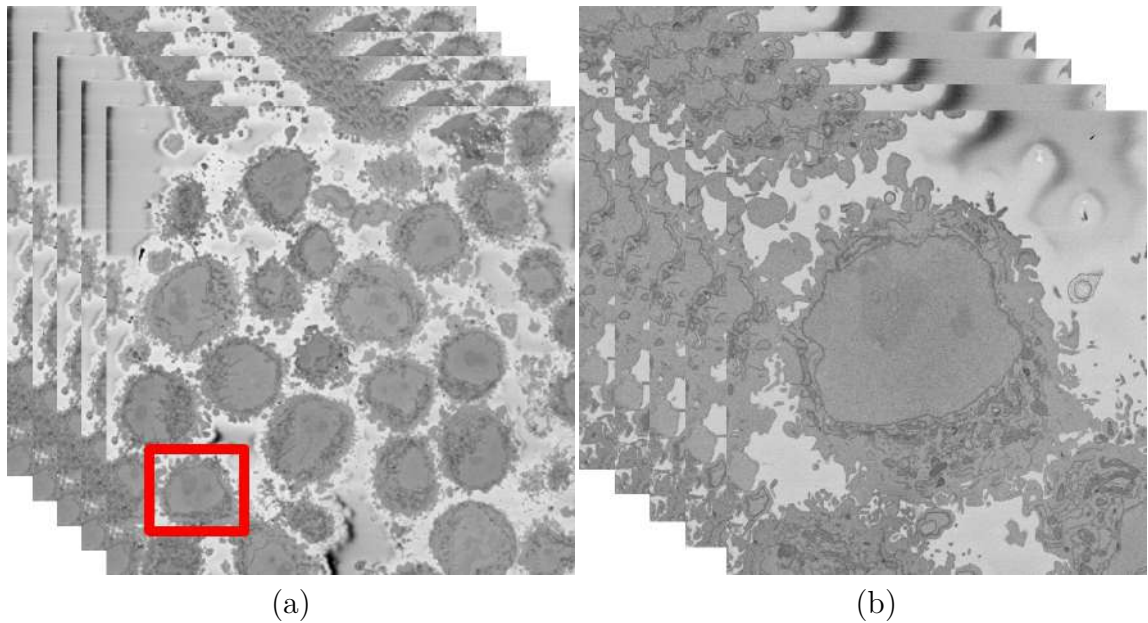
Fig. 5.2 Illustration of the Serial Block face Scanning Electron Microscope (SBF SEM) images containing cancerous HeLa cells. (a) Five representative 8192×8192 images arranged as 3D stack. The HeLa cells are the darker regions and the background is a brighter shade of grey. The red box indicates a region of interest (ROI), that is magnified on the right. (b) Detail of the ROI with a single cell in the centre. This is slice 118 of 300. The nucleus is the large and fairly uniform region in the centre and it is surrounded by the nuclear envelope (NE) which is darker than the nucleus.
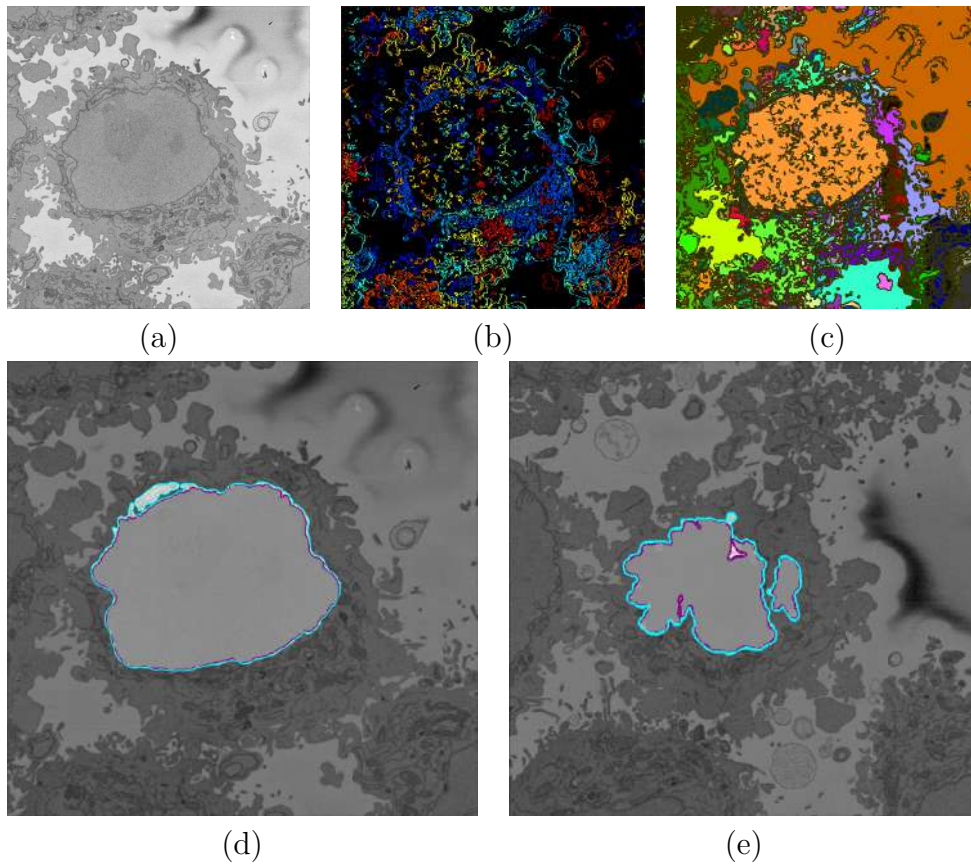
Fig. 5.3 Illustration of intermediate steps of the proposed segmentation algorithm. (a) Cropped region around one HeLa cell (red box in Figure 5.2a and Figure 5.2b), surrounded by resin (background) and edges of other cells. This image was low-pass filtered. (b) Edges detected by Canny algorithm. The edges were further dilated to connect those edges that may belong to the nuclear envelope (NE) but were disjoint due to the variations of the intensity of the envelope itself. (c) Superpixels obtained with the image-processing algorithm and they were generated by removing dilated edges. Small superpixels and those in contact with the image boundary were discarded and the remaining superpixels were smoothed and filled, before discarding those by size that did not belong to the nucleus. (d) final segmentation of the NE overlaid on the filtered image shown in purple. The manual segmentation or the ground truth (GT) is also shown in cyan. (e) A different slice showing final segmentation and GT overlaid on the filtered image. By using neighbouring segmentation as input parameter to the current segmentation and taking the regions into account, the segmentation was considerably improved and was able to identify disjoint regions as part of a single nucleus. Details of differences can be appreciated and the nuclear area covered by the GT and segmentation was brightened up for visualisation purposes.

Images were initially low-pass filtered with a Gaussian kernel with size $h = 7$ and standard deviation $\sigma = 2$ to remove high frequency noise and to enhance the larger scale edge features. This was required as the images presented a grainy texture, which would impact in subsequent steps (Figure 5.4a,b), which relied on the intensity of the classes.

(a)   (b)
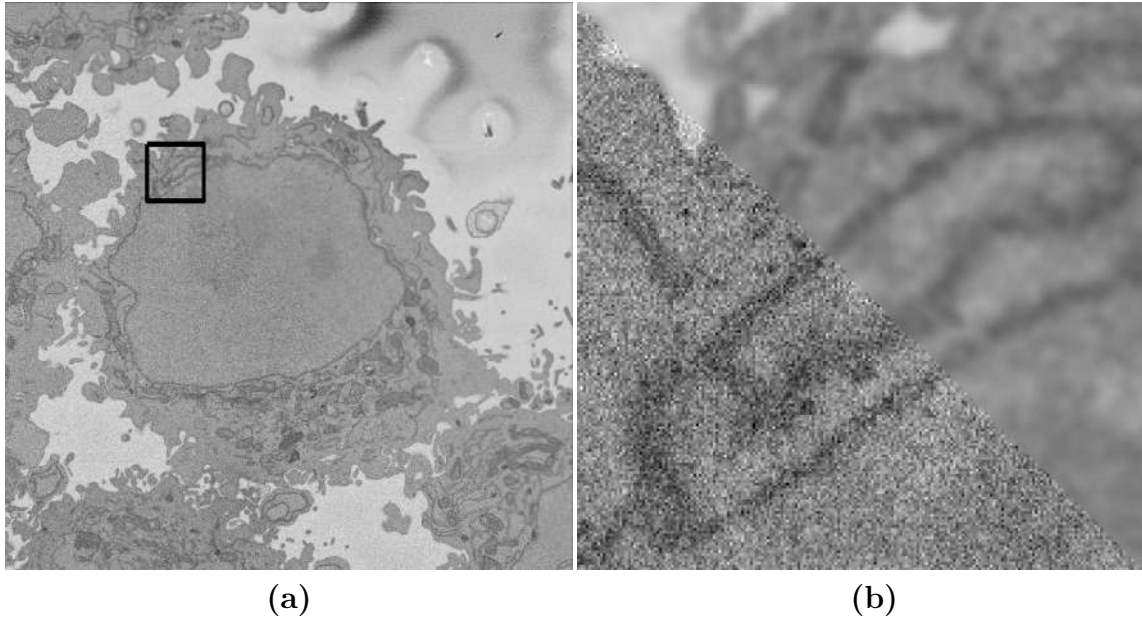
Fig. 5.4 Illustration of the effects of low-pass filtering on one representative image of a HeLa cell. (a) Composite image where the top is a region of interest of the original (Figure 5.2b) and the bottom is filtered. (b) Closer view of the composite image.

The algorithm exploited the abrupt change in intensity at the NE compared with the neighbouring cytoplasm and nucleoplasm by Canny edge detection [178]. The edges were dilated to connect disjoint edges by using the structural element of size 5 or greater than 5 depending on the standard deviation of the Canny edge detector. These disjoint edges were part of the NE and were initially missed due to intensity variations in the envelope itself (Figure 5.3b). The connected pixels not covered by the dilated edges were labelled by the standard 8-connected objects found in the image to create a series of superpixels (Figure 5.3c). The superpixel size was not restricted so that large superpixels covered the background and nucleoplasm. Morphological operators were
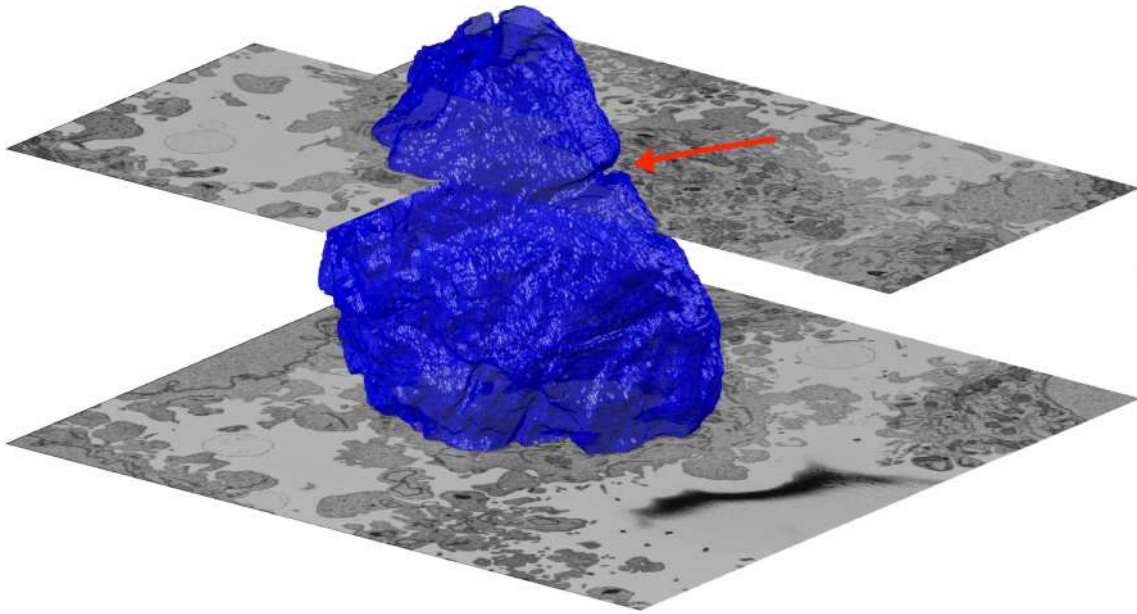
Fig. 5.5 Surface of the nuclear envelope (NE) and partial electron microscopy (EM) slices to give context. Illustration of the 3D surface of the nuclear envelope (NE) of a HeLa cell and different slices as shown 2D cross-sections. Planes could be interpreted as the microtome knife used in serial blockface electron microscope (SBF SEM). The central slice of the surface of HeLa cell was assumed to be the one in which the nuclear region would be centrally positioned and have the largest diameter therefore the image processing algorithm segments this central slice and propagates up and down. Notice the notch (arrow) on the upper right side of the NE.

used to: remove regions in contact with the borders of the image, remove small regions, fill holes inside larger regions and close the jagged edges (Figure 5.3d,e).

A preliminary work approach described in [179] assumed that there was only one nuclear region in each image, which was designated as the nucleus, and the nuclear islands caused by the 2D plane were ignored.

That approach worked well for cells with a smooth near-spherical nuclear morphology. However, in cells with an irregular NE, islands were missed. To improve this approach, the new algorithm exploited the 3D nature of the data by using adjacent images to check for connectivity of islands to the main nuclear region, as a human operator would. As illustrated in Figure 5.5a when the segmentation is performed from top to bottom the algorithm segments only one of the disjoint regions (Figure. 5.5b) in the image. If the central slice (Figure. 5.5c) of the cell is segmented prior to top or bottom slices then
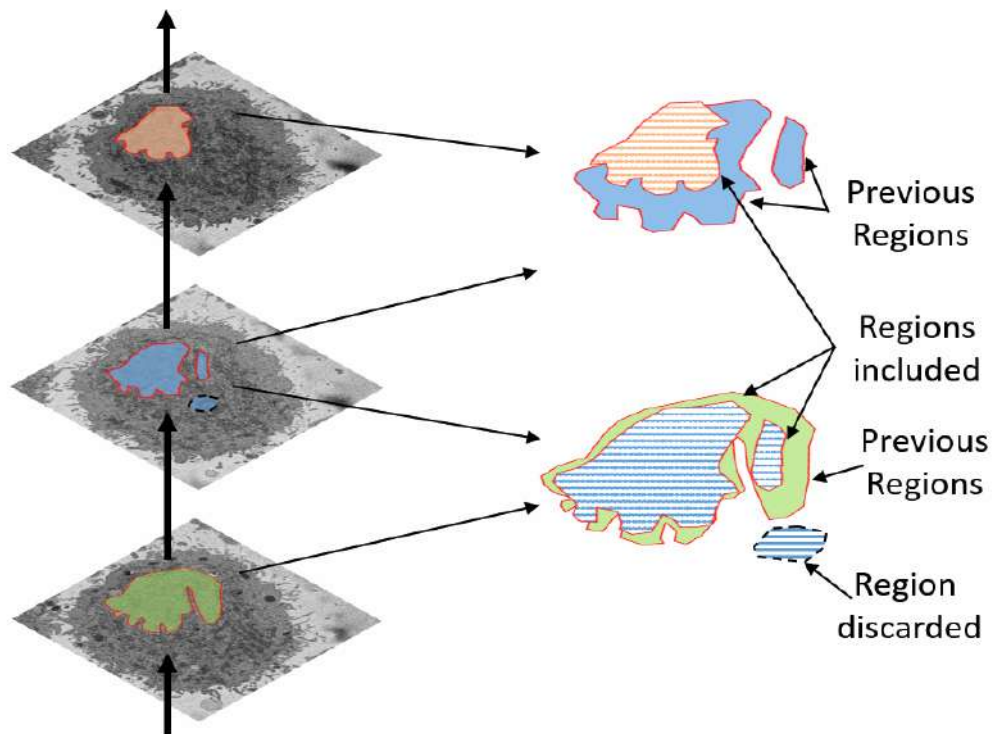
Fig. 5.6 Illustration of the propagation regions. Regions of a slice that overlap with the regions of a previous segmentation are maintained, whilst regions that do not are discarded. The algorithm propagates up and down starting from central slice (slice number 150 for all seven cells which were segmented in this work) in the same way as a human expert to include or discard regions.

this segmentation can be used as a reference. The algorithm thus began at the central slice of the cell, which was assumed to be the one in which the nuclear region would be centrally positioned and have the largest diameter. The algorithm then proceeded in both directions (up and down through the serial images) and propagated the initial result to decide if a disjoint nuclear region was connected above or below the current slice of analysis. By using neighbouring segmentations as input parameters to the current segmentation and taking the regions into account, the algorithm was able to identify disjoint nuclear regions as part of a single nucleus (Figure 5.3e). Finally, the NE was obtained as the boundary of the nucleus. This new approach worked well for cells with disjoint regions.

To find the boundary, a distance transform from the region to every pixel of the image was performed. Figure 5.7a illustrates the distance map from the central superpixel where the black pixels are for the nucleus of the HeLa cell. The average intensity of every loci with constant distance was calculated (Figure 5.7b), where a valley clearly indicated the location of the nuclear envelope (where the intensity is making a dip). It should be noticed how the intensities inside and outside the nuclear envelope were brighter than the envelope itself.

In order to test the image processing algorithm developed in this work, it was tested on a different data set obtained from The Cell Image Library [32]. Each cell was manually cropped from 2489×2435 electron microscopy (EM) image before segmentation algorithm was applied. The algorithm segments each cell (6 cells were segmented so far) in less than 1 minute (It was 40 minutes for EMPIAR data set) with good accuracy as it was visually confirmed. There are 222 slices in the cell shown in (m) and it is important to understand biology of this particular cell as it shows some interesting characteristics. The cells shown in (h) to (k) are from CIL:50061 and the cell in (m) is from CIL:50051. There were 413 and 406 slices in data sets respectively. Serial Block-Face Scanning Electron Microscopy (SBF SEM) provides a comprehensive quantitative analysis of the intracellular chlamydial infection over time. SBF SEM was performed on monolayers of Chlamydia trachomatis-infected HeLa cells.
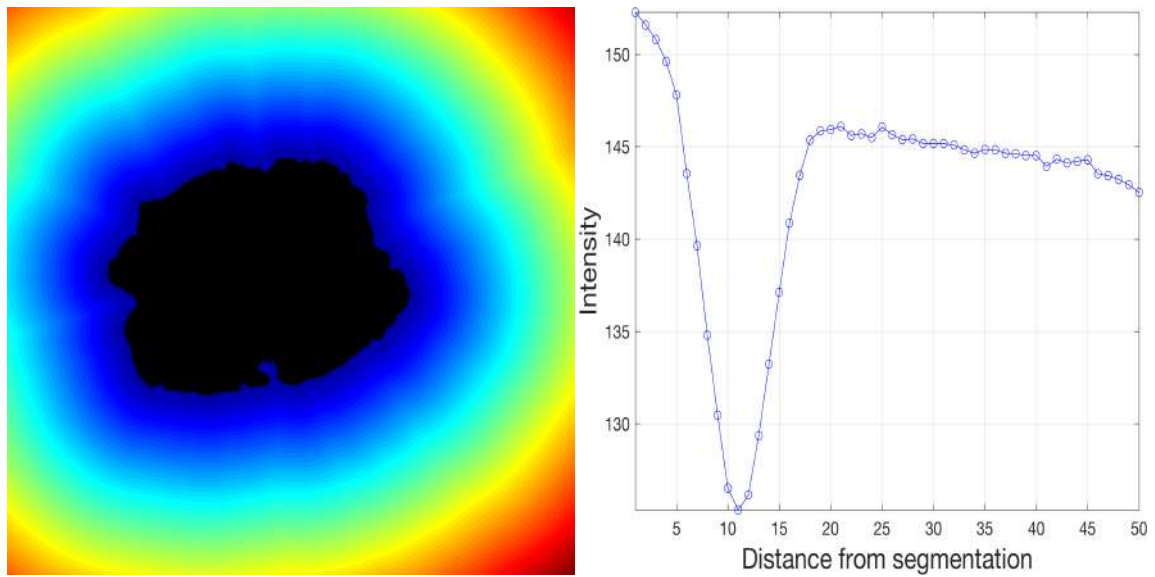
Fig. 5.7 (a) Distance map from the central superpixel. (b) Average intensity from the central superpixel, the minimum corresponds to the nuclear envelope that appears dark as compared with the nuclei and its surrounding structures.
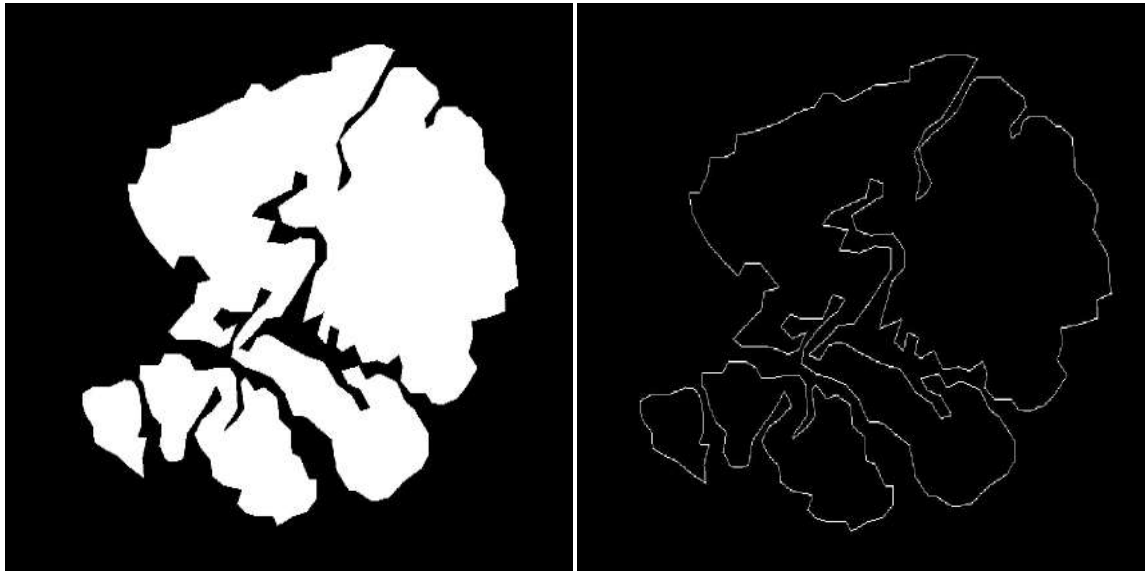
Fig. 5.8 (a) Manual segmentation of a different cell showing the nucleus of the cell as an area and (b) its nuclear envelope showing distinctive disjoint regions. Manual segmentation was performed with a Wacom pen-and-tablet within the MATLAB® *roipoly* function by the author of this dissertation in 47 hours. This image was magnified so the reader can appreciate the boundary and several disjoint regions.
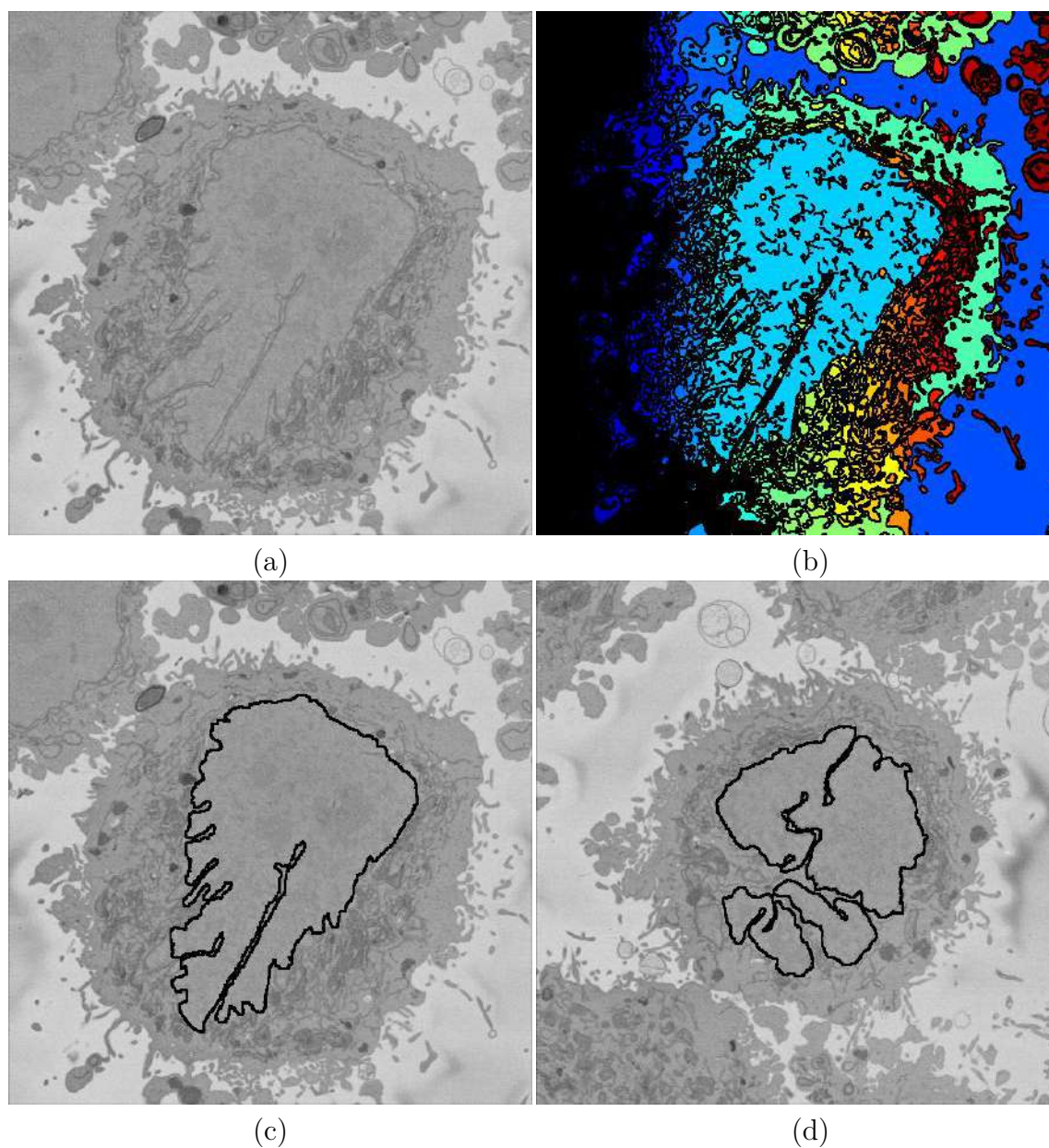
Fig. 5.9 (a) Region of Interest (ROI) with one HeLa cell centred from a 3D stack imaged with an Electron Microscopy(EM). The nucleus is the large and fairly uniform region in the centre and it is surrounded by the nuclear envelope which is darker than the nucleus. (b) Superpixels obtained with the algorithm. (c) Final segmentation of the NE shown by the thick black line. (d) Segmentation of a different slice showing several disjoint nuclear regions.

EMPIAR data set	CIL50061 data set	CIL50051 data
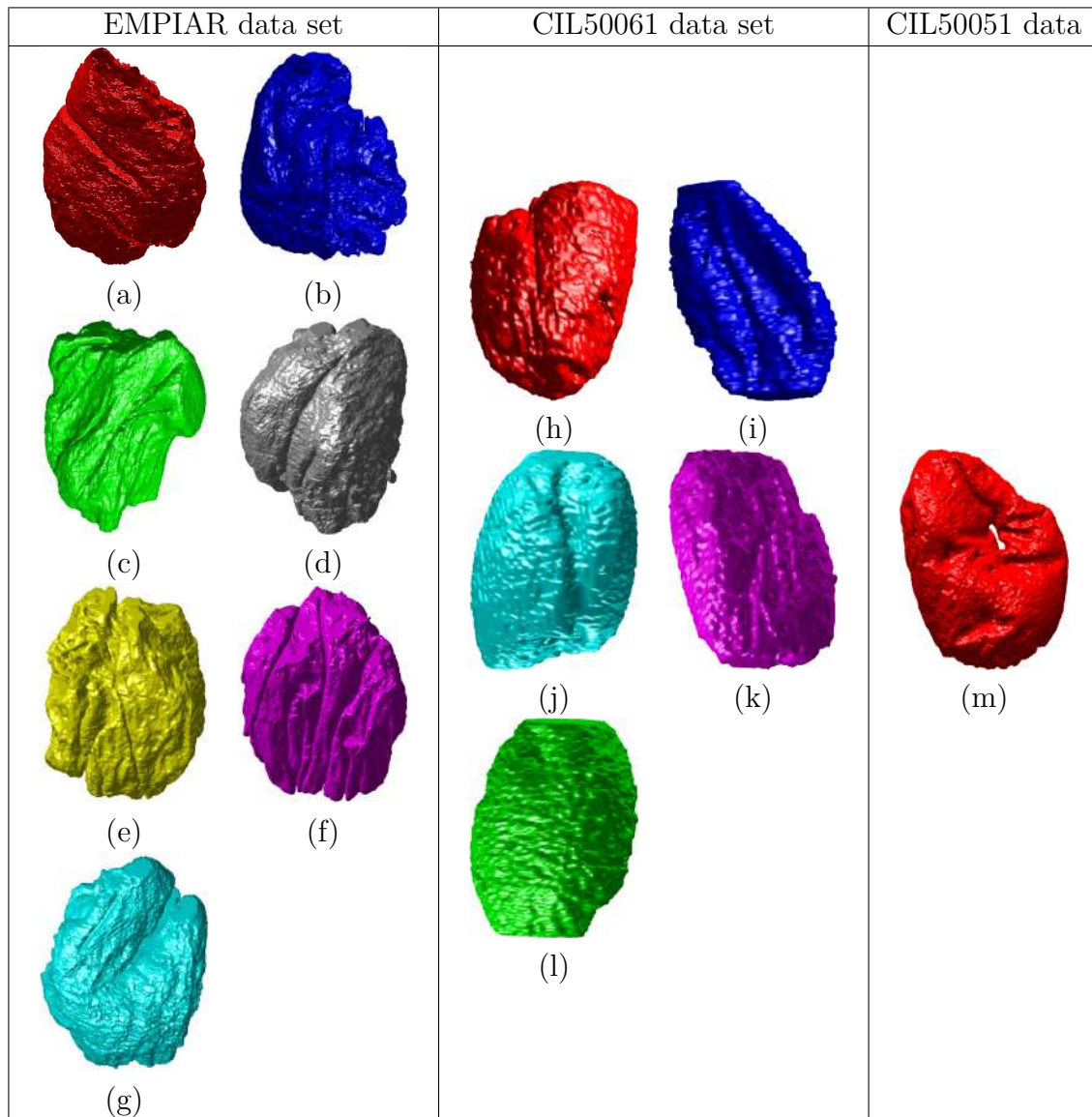
Fig. 5.10 Final results of the automated segmentation of thirteen Nuclear Envelopes (NEs) displayed as rendered volumetric surfaces for Wild type and Chlamydia trachomatis-infected HeLa cells. Surface rendering of thirteen Nuclear Envelopes (NEs), different colours are used for visualisation purposes. Seven Wild-type HeLa cells (EMPIAR data set - left column) and six Chlamydia trachomatis-infected HeLa cells from two data sets (centre and right column). (a-g) Wild-type HeLa cells. In each cell, notice the notches that travel up-down along the nuclei (grey (d), yellow (e), purple (f), and cyan (g) cell) and invaginations. Voxel size of Wild-type HeLa cells is $10 \times 10 \times 50$ nm. (h-l) Segmented cells from CIL50061 data set. Cells in this data set are 12 hours post infection (hpi) and voxel size $8.6 \times 8.6 \times 60$ nm. (m) A cell from CIL50051 with 28 hpi and voxel size is $3.6 \times 3.6 \times 60$ nm. Notice the hole of this particular cell.

<div align="center">(a)   (b)</div>
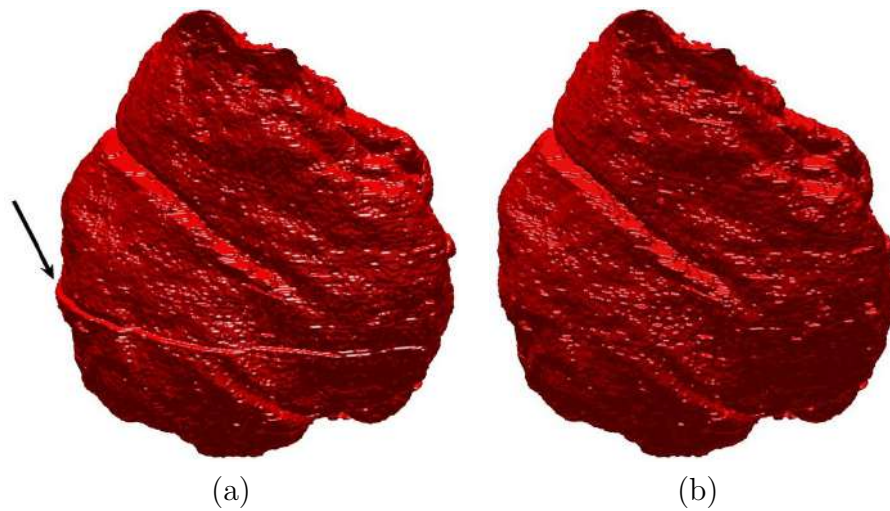
Fig. 5.11 Image registration result of the nuclear envelope (NE) of a HeLa cell. (a) An artefact, shown by the arrow, due to a slight shift (vibration) in the acquisition of Electron Microscope was observed. An image registration algorithm was developed to measure the shift between two consecutive slices (slice 121 and 122). The latter slice was shifted by 11 rows and 20 columns. (b) This artefact was resolved (corrected) and final 3D rendered NE is shown.

Stacks of consecutive 60-nm-thick sections were acquired and subsequently digitally aligned, which allowed individual bacteria to be observed and analyzed in multiple successive sections. All the EM sections were then combined computationally into a 3D reconstruction of the inclusion.

In total, the NE of seven different Wild-type and six Chlamydia trachomatis-infected HeLa cells were analysed. The algorithm was developed and trained on one cell from which the parameters were derived and for which 300 slices of GT existed, (Figure 7.8b) and tested on the NEs of the other twelve cells (Figures 7.8d, 7.10) (*Left*)) of which only one had a corresponding GT with 300 slices (Figure 7.8d) with accurate results and good visual assessment of the remaining.

The shapes of the final segmentations show the complexity of the NE with rather convoluted notches and invaginations. It is speculated that these shapes may have biological significance which is beyond the scope of this work. Whilst segmenting the first cell, a displacement artifact, which is assumed to have been caused by an external vibration to the microscope, was detected (arrow in Figure 5.11). The slices

were carefully observed by scrolling up and down, and a displacement was found between slices 121 and 122. A rigid registration algorithm was applied to measure the shift between slices, which reported a displacement of 11 rows and 20 columns. This displacement was corrected by shifting slices 122 to 300 and since the displacement was present on both images and GT, it had no impact on the JI or HD.

The automated segmentations were compared to manual GT segmentations in order to assess the accuracy of the algorithm. Two different similarity metrics, JI and HD, were computed. For the cell shown in Figure 7.8b (red cell), all 300 slices were visually checked and it was noticed that the bottom 26 and top 40 slices did not contain any cell therefore they were not included in similarity metrics calculations. The algorithm detected cells in all slices between 42 and 254 and JI and HD were computed as 93% and 9 pixels respectively. For the slices between 50 and 250, the mean JI is 95% and the mean HD 8 and for slices between 75 and 225 (interquartile range - IQR) the mean JI is 98% with the mean HD 4 pixels (Figure 7.8a). Similarly the bottom 40 of 300 slices of the cell shown in Figure 7.8d showed no cell and the algorithm detected the cell for all slices between 47 and 289 and JI and HD were computed as 90% and 17 pixels respectively. The mean JI is 93% and the mean HD 13 pixels for slices between 50 and 250 and 94% with the mean HD 13 pixels for slices between 75 and 225 (Figure 7.8c). JI decreased and HD increased towards the top and bottom of the cells as the structure was considerably more complex and the areas become much smaller (Figures 7.8a,c).

5.2   Nuclear envelope shape modelling

In order to further study the shape of the segmented NE, this was modelled against a spheroid. The spheroid was created with the same volume as the nucleus and the position adjusted to fill the NE as closely as possible, as illustrated in Figure 5.12a, where NE is displayed as a red rendered volumetric surface and the spheroid as blue mesh.

(a)

(b)

(c)

(d)
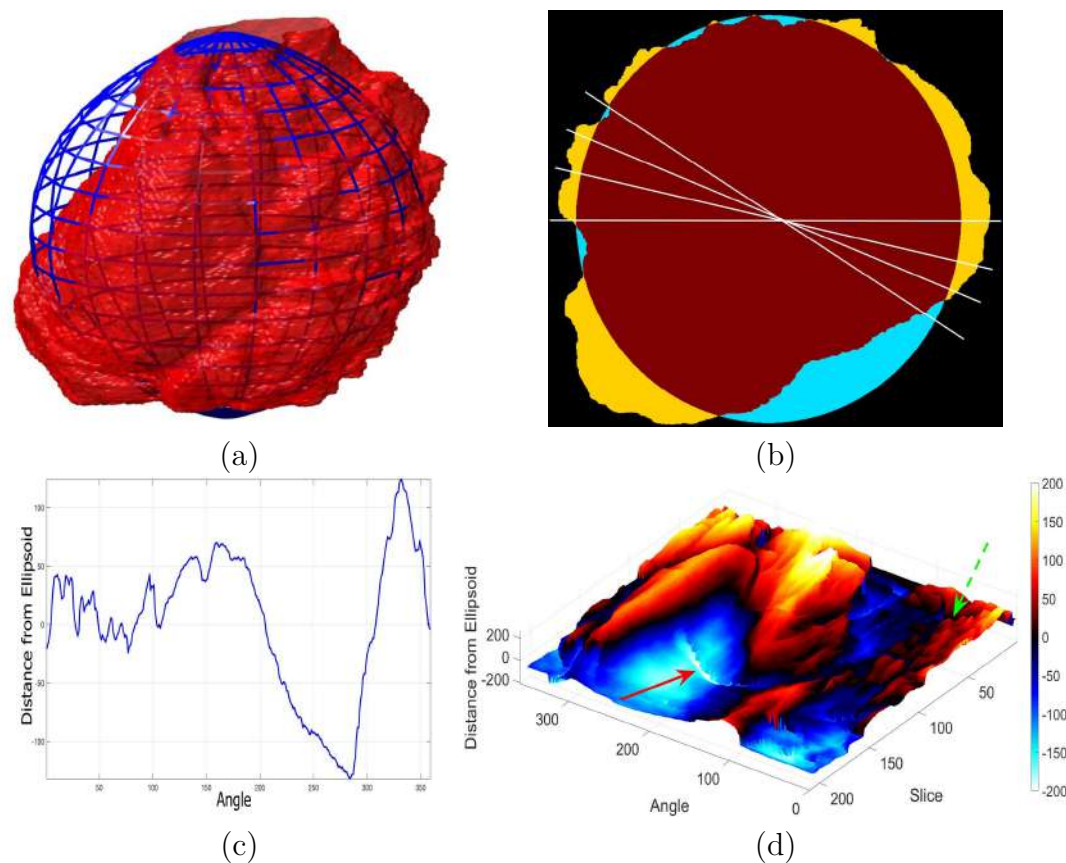
Fig. 5.12 Nuclear envelope (NE) surface modelling against a spheroid. (a) Rendering of the nuclear envelope (NE) (red surface) against the model spheroid (blue mesh). (b) Illustration of distance calculations by ray tracing in one slice. Yellow regions correspond to the nucleus outside the spheroid, cyan regions where nucleus inside the spheroid. (c) Measurements obtained along the boundary. (d) Surface corresponding to the distance from the NE to a model spheroid. Solid red arrow indicates a notch, dashed green arrow shows rugged region.

The surfaces of the spheroid and the nucleus were subsequently compared by tracing rays (Figure 5.12b) from the centre of the spheroid and the distance between the surfaces for each ray was calculated (Figure 5.12c). It was designated that when the NE was further away from the centre, the difference was positive. Figure 5.12d shows the surface corresponding to the distance from the NE of the first cell (Figure 7.8b) to a model spheroid. The notch that travels along the cell between slices is visible (red arrow) and the dashed green arrow highlights a particularly rugged region (Figure 5.12d).

Numerous metrics that characterise the NE can now be extracted, either directly from the NE: *nuclear volume, Jaccard Index (JI) to the spheroid,* or from the altitudes of the modelled surface: *(mean value (μ) and standard deviation (σ)), range of altitudes.* Other derived metrics can also be extracted, for instance, the ratio of the number above or below a certain threshold like the $\mu \pm \sigma$. finally, the correlation between JI and statistical values were calculated so that some conclusions could be drawn about biological characteristics of cells that are beyond the scope of this work.

Mercator map projection was used in this modelling against a spheroid and this could be a limitation of the algorithm. Whilst many interesting characteristics such a large notch on the NE of the third (grey) cell or large protuberance on the second (green) cell, at this moment it is only possible to speculate the biological correlation between the surfaces and the nature of the cell itself.

Chapter 6

Artificial Intelligence, Machine Learning and Deep Learning

In this chapter artificial intelligence, machine learning and specifically principles of Deep learning will be explained. The Convolutional Neural Networks (CNNs) and their structures will be explored and three most accurate CNNs, VGG16, ResNet18 and InceptionResnetv2 will be investigated. They will be used to segment the nuclear envelope of HeLa cells and the segmentation results will be compared with that of our classical segmentation algorithm developed in this work.

6.1   Artificial intelligence (AI)

Artificial intelligence (AI) is a thriving field with many practical applications and active research topics [68]. We look to intelligent software to automate routine labour, understand speech or images, make diagnoses in medicine and support basic scientific research.

6.2   Machine learning (ML)

Computers need to capture this same knowledge in order to behave in an intelligent way [68]. One of the key challenges in AI is how to get this informal knowledge into a computer. Several AI projects have sought to hard-code knowledge about the world in formal languages. A computer can reason automatically about statements in these formal languages using logical inference rules. This is known as the knowledge base approach to AI.

The difficulties faced by systems relying on hard-coded knowledge suggest that AI systems need the ability to acquire their own knowledge, by extracting patterns from raw data. This capability is known as Machine Learning (ML).

Many AI tasks can be solved by designing the right set of features to extract for that task, then providing these features to a simple ML algorithm.

For many tasks, however, it is difficult to know what features should be extracted.

Figure 6.1 illustrates a Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to Artificial Intelligence (AI).

6.3   Deep learning (DL)

Deep learning (DL) has had a long and rich history, but has gone by many names, reflecting different philosophical view points, and has waxed and waned unpopularity [68]. It has become more useful as the amount of available training data has increased. DL models have grown in size over time as computer infrastructure (both hardware and software) for deep learning has improved and have solved increasingly complicated applications with increasing accuracy over time.

DL models are deep artificial neural networks [70]. Each neural network consists of an input layer, an output layer, and multiple hidden layers.
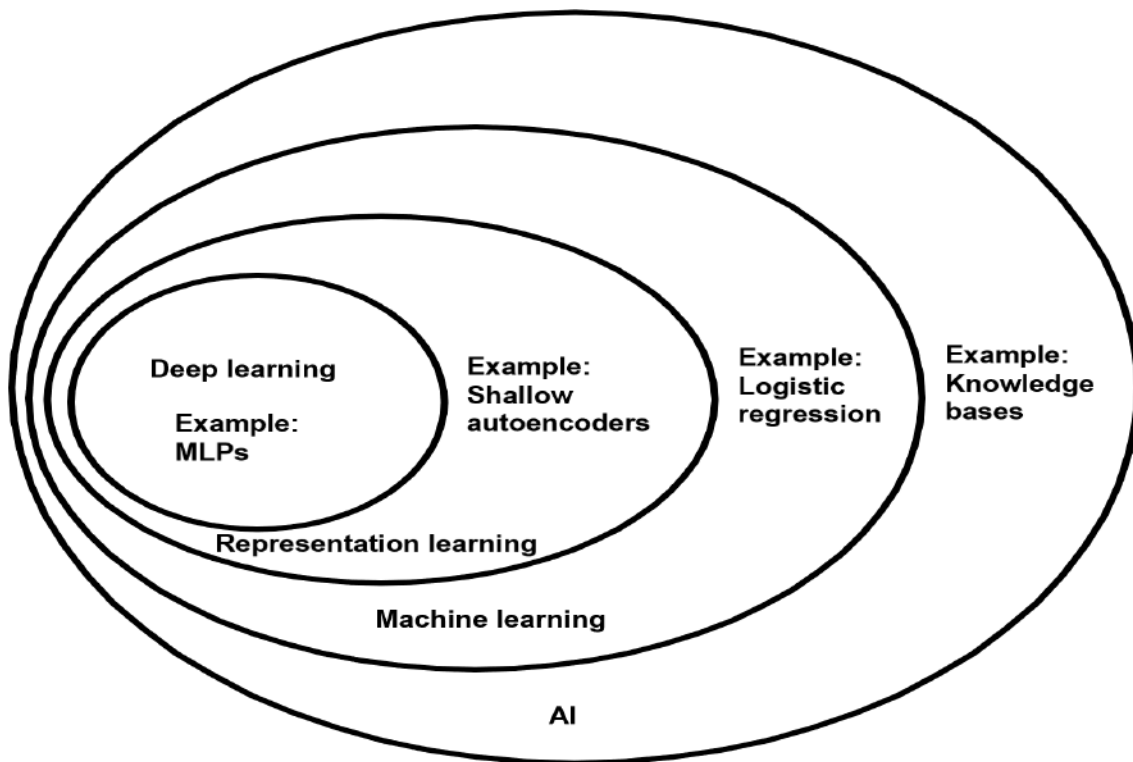
Fig. 6.1 A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to Artificial Intelligence (AI). Each section of the Venn diagram includes an example of an AI technology. A multilayer perceptron (MLP) is a class of feed forward artificial neural network. A MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilises a supervised learning technique called back-propagation for training. MLP was used to apply in computer vision, now succeeded by Convolutional Neural Network (CNN). MLP is now deemed insufficient for modern advanced computer vision tasks. Image adapted from [68].

A DL system can represent the concept of an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges [68].

DL is an approach to AI. Specifically, it is a type of machine learning, a technique that enables computer systems to improve with experience and data. We contend that machine learning is the only viable approach to building AI systems that can operate in complicated real-world environments. DL is a particular kind of machine learning that achieves great power and flexibility by representing the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.

We first introduce basic neural network architectures and then briefly introduce building blocks which are commonly used to boost the ability of the networks to learn features that are useful for image segmentation.

6.4   Fully convolutional neural networks (FCNs)

The idea of FCN was first introduced by Long et al. [180] for image segmentation [70]. FCNs are a special type of CNNs that do not have any fully connected layers. In general, FCNs are designed to have an encoder-decoder structure such that they can take input of arbitrary size and produce the output with the same size. Given an input image, the encoder first transforms the input into high-level feature representation whereas the decoder interprets the feature maps and recovers spatial details back to the image space for pixel-wise prediction through a series of transposed convolution and convolution operations. Here, transposed convolutions are used for up-scaling the feature maps, typically by a factor of 2. These transposed convolutions can also be replaced by unpooling layers and up-sampling layers. Compared to a patch-based CNN for segmentation, FCN is trained and applied to the entire images, removing the need for patch selection (Shelhamer et al., 2017). FCN with the simple encoder-decoder structure in Figure 6.2 may be limited to capture detailed context information in an

(a)

(b)
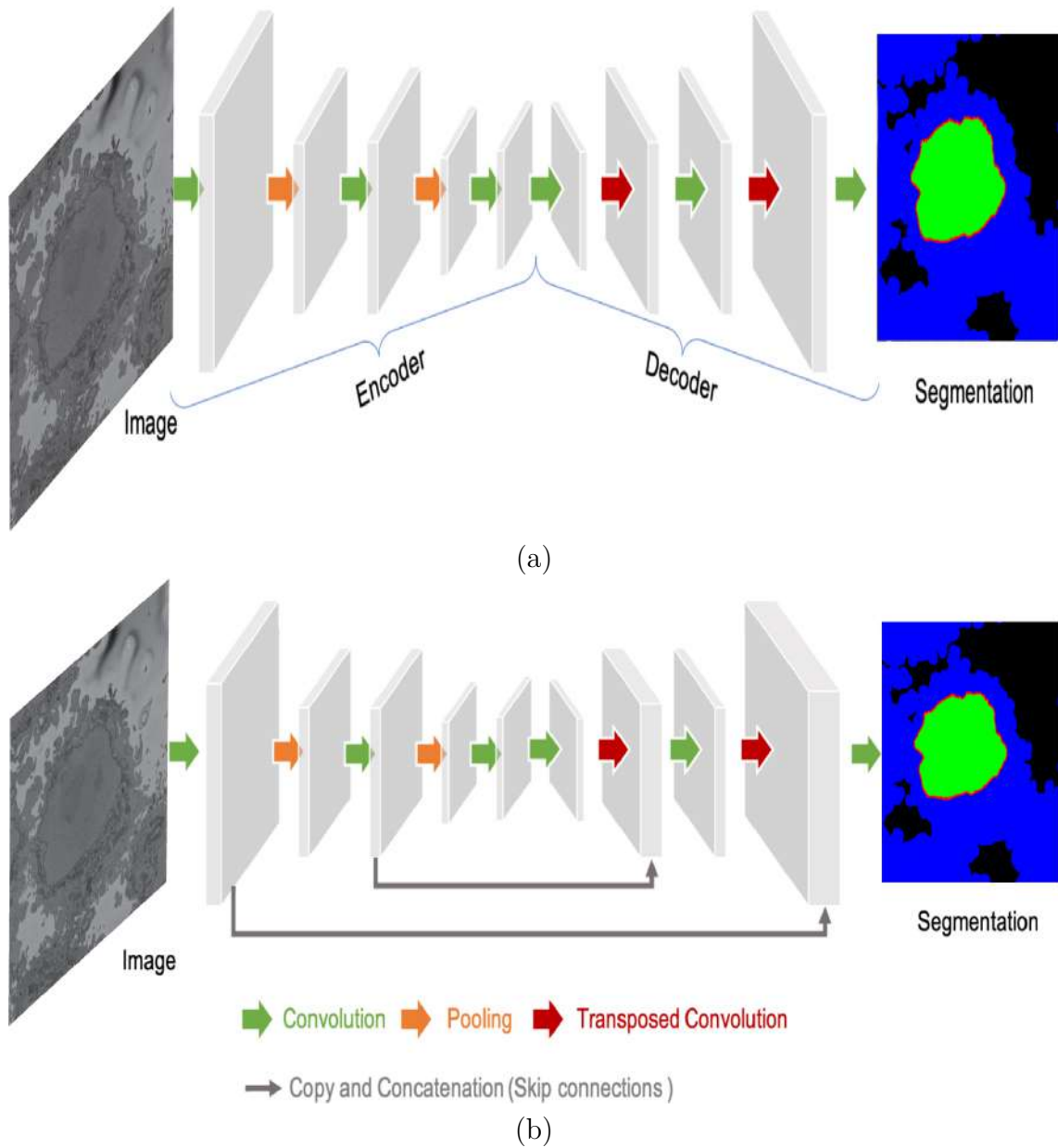
Fig. 6.2 The FCN first takes the whole image as input, learns deep image features though the encoder, gradually recovers the spatial dimension by a series of transposed convolution layers in the decoder and finally predicts a pixel-wise image segmentation for the left ventricle cavity (the blue region), the left ventricular myocardium (the green region) and the right ventricle (the red region). One use case of this FCN-based cardiac segmentation can be found in Tran (2016). (b) A schematic drawing of U-net. On the basis of the basic structure of FCN, U-net employs 'skip connections' (the grey arrows) to aggregate feature maps from coarse to fine. Of note, for simplicity, we reduce the number of down-sampling and up-sampling blocks. For detailed information, see the original paper [88].

image for precise segmentation as some features may be eliminated by the pooling layers in the encoder. Several variants of FCNs [181] have been proposed to propagate features from the encoder to the decoder, in order to boost the segmentation accuracy.

The most well-known and most popular variant of FCNs for biomedical image segmentation is the U-net [88]. On the basis of the vanilla FCN ([180]), the U-net employs skip connections between the encoder and decoder to recover spatial context loss in the down-sampling path, yielding more precise segmentation (Figure 6.2b).

Several state-of-the-art image segmentation methods have adopted the U-net or its 3D variants, the 3D Unet [182] and the 3D V-net [183], as their backbone networks, achieving promising segmentation accuracy for a number of segmentation tasks.

6.4.1   Reduce over-fitting

The biggest challenge of training deep networks for medical image analysis is over-fitting, due to the fact that there is often a limited number of training images in comparison with the number of learnable parameters in a deep network. A number of techniques have been developed to alleviate this problem. Some of the techniques are the following ones:

i) Weight initialization ([184]) and weight regularisation (i.e. L1/L2 regularisation), ii) Dropout ([85]), iii) Ensemble learning ([185]), iv) Data augmentation by artificially generating training samples via affine transformations, v) Transfer learning with a model pre-trained on existing large data sets. In the next sections we will discuss these techniques in detail.

6.5   Data set augmentation

Data augmentation helps prevent the network from overfitting [85] and memorizing the exact details of the training images. Data augmentation is used during training to provide more examples to the network because it helps improve the accuracy of the network. The best way to make a machine learning model generalise better is to train

it on more data [68]. Of course, in practice, the amount of data we have is limited. One way to get around this problem is to create fake data and add it to the training set. An example for data augmentation for the HeLa dataset is given in Figure6.3top where a HeLa cell image is flipped along the y-axis to be fed into a CNN. At the bottom of the same figure: Three different transformations on a HeLa cell image are shown. (a) Original HeLa image. (b) Original image is flipped along the x-axis. (c) Rotation after flipping. (d) Rotated image in (c) is translated by (shifting the image) by 50 pixels in the x-direction and 75 pixels in the y-direction.

For some machine learning tasks, it is reasonably straight forward to create new fake data. This approach is easiest for classification. A classifier needs to take a complicated, high-dimensional input x and summarise it with a single category identity y. This means that the main task facing a classifier is to be invariant to a wide variety of transformations. We can generate new (x, y) pairs easily just by transforming the x inputs in our training set. Data set augmentation has been a particularly effective technique for a specific classification problem: object recognition. Images are high dimensional and include n enormous range of factors of variation, many of which can be easily simulated. Operations like translating the training images a few pixels in each direction can often greatly improve generalisation, even if the model has already been designed to be partially translation invariant by using the convolution and pooling techniques. Many other operations, such as rotating the image or scaling the image, have also proved quite effective. One must be careful not to apply transformations that would change the correct class. For example, optical character recognition tasks require recognising the difference between "b" and "d" and the difference between 6 and 9, so horizontal flips and 180° rotations are not appropriate ways of augmenting data sets for these tasks.

Data augmentation is essential to teach the network the desired in-variance and robustness properties, when only few training samples are available [70]. It is easy to improve the generalisation of a classifier by increasing the size of the training set by adding extra copies of the training examples that have been modified with

transformations that do not change the class. Object recognition is a classification task that is especially amenable to this form of data set augmentation because the class is invariant to so many transformations and the input can be easily transformed with many geometric operations. Classifiers can benefit from random translations, rotations, and in some cases, flips of the input to augment the data set. The value of data augmentation for learning in-variance has been shown in [186].

When comparing machine learning benchmark results, taking the effect of data set augmentation into account is important. Often, hand-designed data set augmentation schemes can dramatically reduce the generalisation error of a machine learning technique. To compare the performance of one machine learning algorithm to another, it is necessary to perform controlled experiments. When comparing machine learning algorithm A and machine learning algorithm B, make sure that both algorithms are evaluated using the same hand-designed data set augmentation schemes. Suppose that algorithm A performs poorly with no data set augmentation, and algorithm B performs well when combined with numerous synthetic transformations of the input. In such a case the synthetic transformations likely caused the improved performance, rather than the use of machine learning algorithm B. Sometimes deciding whether an experiment has been properly controlled requires subjective judgement. For example, machine learning algorithms that inject noise into the input are performing a form of data set augmentation. Usually, operations that are generally applicable (such as adding Gaussian noise to the input) are considered part of the machine learning algorithm, while operations that are specific to one application domain (such as randomly cropping an image) are considered to be separate pre-processing steps.

In the following paragraphs, we summarise several major challenges in the field of cell segmentation and some recently proposed approaches that attempt to address them. These challenges and related works also provide potential research directions for future work in this field.

Weakly and semi-supervised learning Weakly and semi-supervised learning methods aim at improving the learning accuracy by making use of both labelled and
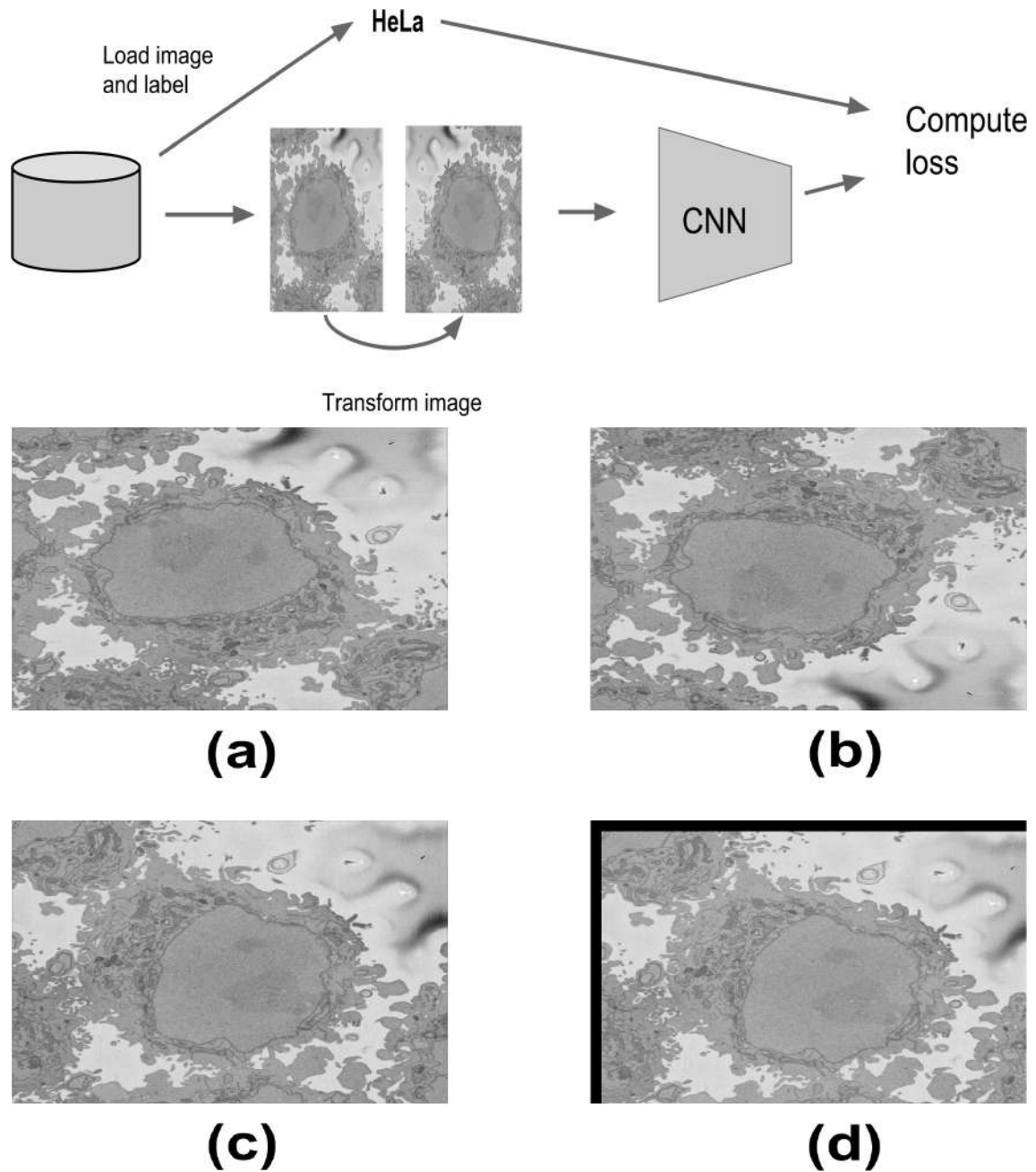
Fig. 6.3 Illustration of data augmentation with a HeLa cell image for deep neural networks training. Convolutional Neural Networks (CNNs) can benefit from random translations, rotations, and in some cases, flips of the input to augment the data set. Top: A HeLa cell image is flipped along the y-axis to be fed into a CNN. Image adapted from [187]. Bottom: Three different transformations on a HeLa cell image. (a) Original HeLa image. (b) Original image is flipped along the x-axis. (c) Rotation after flipping. (d) Rotated image in (c) is translated by (shifting the image) by 50 pixels in the x-direction and 75 pixels in the y-direction.

unlabelled or weakly labelled data (e.g annotations in forms of scribbles or bounding boxes).

Self-supervised learning. Another approach is self-supervised learning which aims at utilising labels that are generated automatically without human intervention. These labels, designed to encode some properties or semantics of the object, can provide strong supervisory signals to pre-train a network before fine-tuning for a given task. Compared to a network trained from scratch, networks pre-trained on the self-supervised task performed better, especially when the training data was extremely limited.

Unsupervised learning. Unsupervised learning aims at learning without paired labelled data. Compared to the former three classes, there is limited literature about unsupervised learning methods for image segmentation, perhaps because of the difficulty of the task.

Apart from utilising unlabelled images for training neural networks, another interesting direction is active learning, which tries to select the most representative images from a large-scale data set, reducing redundant labelling workload and training cost. This technique is also related to incremental learning, which aims to improve the model performance with new classes added incrementally while avoiding a dramatic decrease in overall performance. Given the increasing size of the available medical data sets, and the practical challenges of labelling and storing large amounts of images from various sources, it is of great interest to develop algorithms capable of distilling a large scale data set into a small one containing the most representative cases for labelling and training.

Scarcity of Labels One of the biggest challenges for deep learning approaches is the scarcity of annotated data [70]. The majority of studies uses a fully supervised approach to train their networks, which requires a large number of annotated images. In fact, annotating images is time consuming and often requires significant amounts of expertise. While data augmentation techniques such as cropping, padding, and geometric transformations (e.g. affine transformations) can be used to increase the size

of training samples, their diversity may still be limited, failing to reflect the spectrum of real world data distributions. Several methods have been proposed to overcome this challenge. These methods can be categorised into four classes: transfer learning with fine-tuning, weakly and semi-supervised learning, self-supervised learning, and unsupervised learning.

6.6   Transfer learning with fine-tuning

In some cases the deep learning models can be trained from scratch on new applications/data sets (assuming a sufficient quantity of labelled training data), but in many cases there are not enough labeled data available to train a model from scratch and one can use transfer learning to tackle this problem [188]. In transfer learning, a model trained on one task is re-purposed on another (related) task, usually by some adaptation process toward the new task. For example, one can imagine adapting an image classification model trained on ImageNet to a different task, such as texture classification, or face recognition. In image segmentation case, many people use a model trained on ImageNet (a larger data set than most of image segmentation data sets), as the encoder part of the network, and re-train their model from those initial weights. The assumption here is that those pre-trained models should be able to capture the semantic information of the image required for segmentation, and therefore enabling them to train the model with less labelled samples.

By using transfer learning, we could replace the last layer(s) of the network by something that is more dedicated to our use case (different than the 1000 *ImageNet labels*). For instance, if we are interested in finding whether our image contains either nuclear envelope (NE) or a nucleus, we would have only two different labels and our last layer could be made of only two neurons (compared to the 1000 of the VGG16). If we assume that the rest of the network should remain unmodified, we would have to learn only 8194 free parameters (corresponding to 2 bias value and all weight connections between the 4096 neurons of the one to the last layer with the 2 neurons of the last

layer). This is a very small number of parameters to learn for such a complex task. We would, therefore, need a relatively small training data set to train our specialised VGG16 algorithm.

Transfer learning has become a sizeable sub-field in machine learning [68]. It has ideological benefits, because it is seen as an important aspect of human learning, and also practical benefits, because it can make machine learning more efficient. As computing power increases and researchers apply machine learning to more complex problems, knowledge transfer can only become more desirable. Transfer learning is machine learning with an additional source of information apart from the standard training data: knowledge from one or more related tasks.

6.6.1   Training Neural Networks

Before being able to perform inference, neural networks must be trained [70]. This training process requires a data set that contains paired images and labels x, y for training and testing, an optimiser (e.g. stochastic gradient descent, Adam) and a loss function to update the model parameters. This function accounts for the error of the network prediction in each iteration during training, providing signals for the optimiser to update the network parameters through back-propagation. The goal of training is to find proper values of the network parameters to minimise the loss function.

6.6.2   Dropout

Dropout layers at the end of the contracting path perform further implicit data augmentation [70]. Dropout, a powerful regularisation strategy, can be seen as a process of constructing new inputs by multiplying by noise [68].

Dropout [85] provides a computationally inexpensive but powerful method of regularising a broad family of models. To a first approximation, dropout can be thought of as a method of making bagging practical for ensembles of very many large neural networks. Bagging involves training multiple models and evaluating multiple
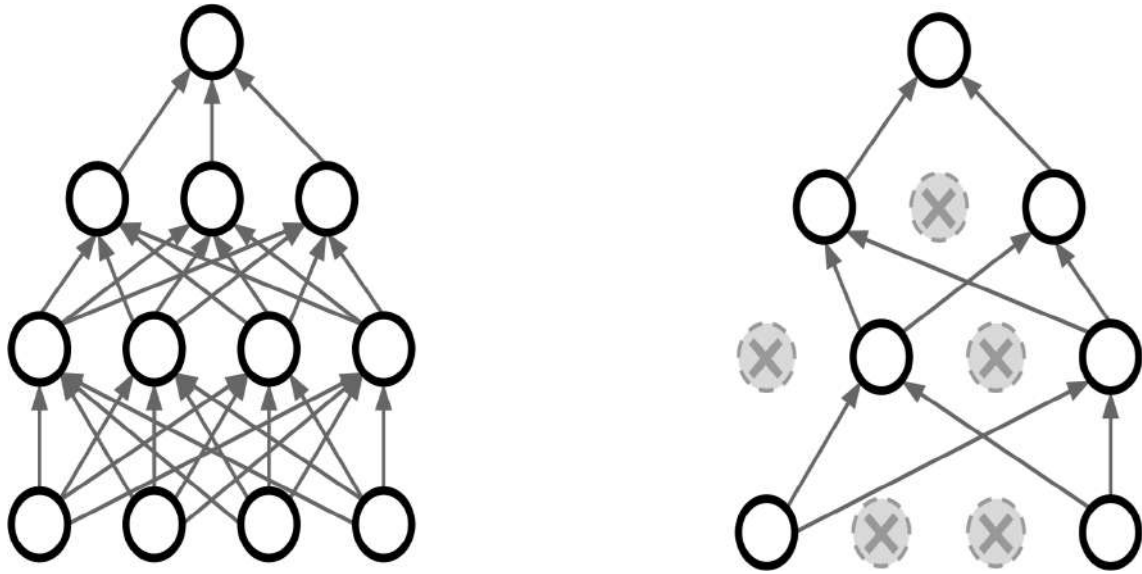
Fig. 6.4 Schematic diagram of Dropout used in Convolutional neural networks (CNNs). Dropout is training a large ensemble of models that share parameters. In each forward pass, randomly set some neurons to zero. Probability of dropping is a hyper-parameter; 0.5 is common. Dropout forces the network to have a redundant representation and prevents co-adaptation of features. Image adapted from [85].

models on each test example. This seems impractical when each model is a large neural network, since training and evaluating such networks is costly in terms of run time and memory. It is common to use ensembles of five to ten neural networks — [189] used six to win the [78] — but more than this rapidly becomes unwieldy. Dropout provides an inexpensive approximation to training and evaluating a bagged ensemble of exponentially many neural networks. Specifically, dropout trains the ensemble consisting of all sub-networks that can be formed by removing non output units from an underlying base network, as illustrated in Figure 6.4.

In most modern neural networks, based on a series of affine transformations and non-linearities, we can effectively remove a unit from a network by multiplying its output value by zero.

Here, we present the dropout algorithm in terms of multiplication by zero for simplicity, but it can be trivially modified to work with other operations that remove a unit from the network. Recall that to learn with bagging, we define k different models, construct k different data sets by sampling from the training set with replacement,

and then train model i on data set i. Dropout aims to approximate this process, but with an exponentially large number of neural networks. Specifically, to train with dropout, we use a mini-batch-based learning algorithm that makes small steps, such as stochastic gradient descent (SGD). Each time we load an example into a mini-batch, we randomly sample a different binary mask to apply to all the input and hidden units in the network. The mask for each unit is sampled independently from all the others. The probability of sampling a mask value of one (causing a unit to be included) is a hyper-parameter fixed before training begins. It is not a function of the current value of the model parameters or the input example. Typically, an input unit is included with probability 0.8, and a hidden unit is included with probability 0.5. We then run forward propagation, back-propagation, and the learning update as usual.

Dropout training is not quite the same as bagging training. In the case of bagging, the models are all independent. In the case of dropout, the models share parameters, with each model inheriting a different subset of parameters from the parent neural network. This parameter sharing makes it possible to represent an exponential number of models with a tractable amount of memory. In the case of bagging, each model is trained to convergence on its respective training set. In the case of dropout, typically most models are not explicitly trained at all — usually, the model is large enough that it would be infeasible to sample all possible sub-networks within the lifetime of the universe. Instead, a tiny fraction of the possible sub-networks are each trained for a single step, and the parameter sharing causes the remaining sub-networks to arrive at good settings of the parameters. These are the only differences. Beyond these, dropout follows the bagging algorithm.

6.6.3 Optimisation for training deep models

This section presents optimisation techniques for neural network training. Deep learning algorithms involve optimisation in many contexts [68]. For example, performing inference in models involves solving an optimisation problem. We often use analytical optimisation to write proofs or design algorithms. Of all the many optimisation

problems involved in deep learning, the most difficult is neural network training. It is quite common to invest days to months of time on hundreds of machines to solve even a single instance of the neural network training problem. Because this problem is so important and so expensive, a specialised set of optimisation techniques have been developed for solving it.

This section focuses on one particular case of optimisation: finding the parameters θ of a neural network that significantly reduce a cost function $J(\theta)$, which typically includes a performance measure evaluated on the entire training set as well as additional regularisation terms.

6.6.4   Batch and mini batch algorithms

We have previously introduced the gradient descent algorithm that follows the gradient of an entire training set downhill (Figure6.5. This may be accelerated considerably by using stochastic gradient descent to follow the gradient of randomly selected mini-batches downhill.

Stochastic Gradient Descent

Stochastic gradient descent (SGD) and its variants are probably the most used optimisation algorithms for machine learning in general and for deep learning in particular. It is possible to obtain an unbiased estimate of the gradient by taking the average gradient on a mini-batch of m examples drawn i.i.d from the data-generating distribution.

Figure6.5 shows how to follow this estimate of the gradient downhill. A crucial parameter for the SGD algorithm is the learning rate. Previously, we have described SGD as using a fixed learning rate. In practice, it is necessary to gradually decrease the learning rate over time. This is because the SGD gradient estimator introduces a source of noise (the random sampling of m training examples) that does not vanish even when we arrive at a minimum. By comparison, the true gradient of the total cost function becomes small and then 0 when we approach and reach a minimum using batch gradient descent, so batch gradient descent can use a fixed learning rate. The

(a)

(b)
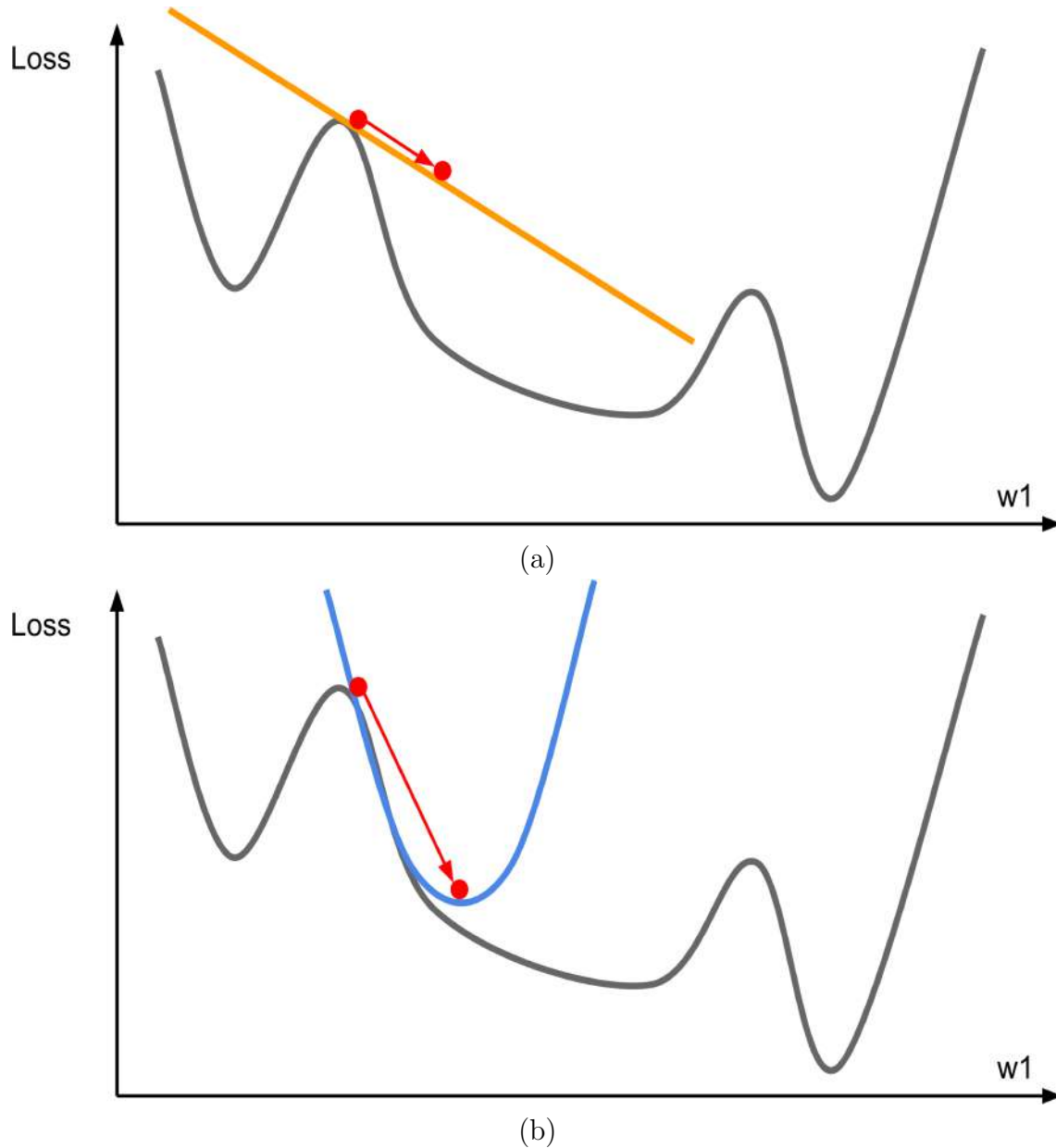
Fig. 6.5 First and Second order optimisation in Convolutional neural networks (CNNs). (a) First-Order Optimisation uses gradient form linear approximation and step to minimise the approximation. (b) Second-Order Optimisation uses gradient and Hessian to form quadratic approximation and step to the minima of the approximation. Image credit to [187].

learning rate may be chosen by trial and error, but it is usually best to choose it by monitoring learning curves that plot the objective function as a function of time. This is more of an art than a science, and most guidance on this subject should be regarded with some scepticism.

If learning rate is too large, the learning curve will show violent oscillations, with the cost function often increasing significantly. Gentle oscillations are fine, especially if training with a stochastic cost function, such as the cost function arising from the use of dropout. If the learning rate is too low, learning proceeds slowly, and if the initial learning rate is too low, learning may become stuck with a high cost value. Typically, the optimal initial learning rate, in terms of total training time and the final cost value, is higher than the learning rate that yields the best performance after the first 100 iterations or so. Therefore, it is usually best to monitor the first several iterations and use a learning rate that is higher than the best-performing learning rate at this time, but not so high that it causes severe instability. The most important property of SGD and related mini-batch or online gradient-based optimisation is that computation time per update does not grow with the number of training examples. This allows convergence even when the number of training examples becomes very large. For a large enough data set, SGD may converge to within some fixed tolerance of its final test set error before it has processed the entire training set. Batch gradient descent enjoys better convergence rates than stochastic gradient descent in theory. One can also trade off the benefits of both batch and stochastic gradient descent by gradually increasing the mini-batch size during the course of learning.

6.6.5   AdaGrad

The AdaGrad algorithm individually adapts the learning rates of all model parameters by scaling them inversely proportional to the square root of the sum of all the historical squared values of the gradient [190]. The parameters with the largest partial derivative of the loss have a correspondingly rapid decrease in their learning rate, while parameters with small partial derivatives have a relatively small decrease in their learning rate.

The net effect is greater progress in the more gently sloped directions of parameter space. In the context of convex optimisation, the AdaGrad algorithm enjoys some desirable theoretical properties. Empirically, however, for training deep neural network models, the accumulation of squared gradients from the beginning of training can result in a premature and excessive decrease in the effective learning rate. AdaGrad performs well for some but not all deep learning models.

6.6.6   RMSProp

The RMSProp algorithm [75] modifies AdaGrad to perform better in the non-convex setting by changing the gradient accumulation into an exponentially weighted moving average. AdaGrad is designed to converge rapidly when applied to a convex function [68]. When applied to a non-convex function to train a neural network the learning trajectory may pass through many different structures and eventually arrive at a region that is a locally convex bowl. AdaGrad shrinks the learning rate according to the entire history of the squared gradient and may have made the learning rate too small before arriving at such a convex structure. RMSProp uses an exponentially decaying average to discard history from the extreme past so that it can converge rapidly after finding a convex bowl, as if it were an instance of the AdaGrad algorithm initialised within that bowl. Compared to AdaGrad, the use of the moving average introduces a new hyper-parameter that controls the length scale of the moving average. Empirically, RMSProp has been shown to be an effective and practical optimisation algorithm for deep neural networks. It is currently one of the go-to optimisation methods being employed routinely by deep learning practitioners.

6.6.7   Adam

Adam [191] is yet another adaptive learning rate optimisation algorithm. The name "Adam" derives from the phrase "adaptive moments" [68]. In the context of the earlier algorithms, it is perhaps best seen as a variant on the combination of RMSProp

and momentum with a few important distinctions. First, in Adam, momentum is incorporated directly as an estimate of the first-order moment (with exponential weighting) of the gradient. The most straight forward way to add momentum to RMSProp is to apply momentum to the re-scaled gradients. The use of momentum in combination with re-scaling does not have a clear theoretical motivation. Second, Adam includes bias corrections to the estimates of both the first-order moments (the momentum term) and the (un-centred) second-order moments to account for their initialisation at the origin. RMSProp also incorporates an estimate of the (un-centred) second-order moment; however, it lacks the correction factor. Thus, unlike in Adam, the RMSProp second-order moment estimate may have high bias early in training. Adam is generally regarded as being fairly robust to the choice of hyper-parameters, though the learning rate sometimes needs to be changed from the suggested default.

6.6.8   Choosing the right optimisation algorithm

We have discussed a series of related algorithms that each seek to address the challenge of optimising deep models by adapting the learning rate for each model parameter. At this point, a natural question is: which algorithm should one choose? Unfortunately, there is currently no consensus on this point.

[192] (2014) presented a valuable comparison of a large number of optimisation algorithms across a wide range of learning tasks. While the results suggest that the family of algorithms with adaptive learning rates (represented by RMSProp and AdaDelta performed fairly robustly, no single best algorithm has emerged. Currently, the most popular optimisation algorithms actively in use include SGD, SGD with momentum, RMSProp, RMSProp with momentum, Ada Delta, and Adam. The choice of which algorithm to use, at this point, seems to depend largely on the user's familiarity with the algorithm (for ease of hyper-parameter tuning). SGD, SGD+Momentum, Adagrad, RMSProp, Adam all have learning rate as a hyper-parameter.

6.6.9   Data types

The data used with a convolutional network usually consist of several channels, each channel being the observation of a different quantity at some point in space or time [68]. So far we have discussed only the case where every example in the train and test data has the same spatial dimensions. One advantage to convolutional networks is that they can also process inputs with varying spatial extents. These kinds of input simply cannot be represented by traditional, matrix multiplication-based neural networks. This provides a compelling reason to use convolutional networks even when computational cost and over-fitting are not significant issues. For example, consider a collection of images in which each image has a different width and height. It is unclear how to model such inputs with a weight matrix of fixed size. Convolution is straightforward to apply; the kernel is simply applied a different number of times depending on the size of the input, and the output of the convolution operation scales accordingly. Convolution may be viewed as matrix multiplication; the same convolution kernel induces a different size of doubly block circulant matrix for each size of input. Sometimes the output of the network as well as the input is allowed to have variable size, for example, if we want to assign a class label to each pixel of the input. In this case, no further design work is necessary. In other cases, the network must produce some fixed-size output, for example, if we want to assign a single class label to the entire image. In this case, we must make some additional design steps, like inserting a pooling layer whose pooling regions scale in size proportional to the size of the input, to maintain a fixed number of pooled outputs. Note that the use of convolution for processing variably sized inputs makes sense only for inputs that have variable size because they contain varying amounts of observation of the same kind of thing-different lengths of recordings over time, different widths of observations over space, and so forth. Convolution does not make sense if the input has variable size because it can optionally include different kinds of observations. For example, if we are processing college applications, and our features consist of both grades and standardised test scores, but not every applicant took the standardised test, then it does not make sense

to convolve the same weights over features corresponding to the grades as well as the features corresponding to the test scores.

6.7   U-Net segmentation algorithm

U-Net is convolutional network architecture with broad application and has become a broadly used tool for semantic segmentation. U-Net architecture is a type of fully convolutional network [194] in which, after the downsampling steps obtained by convolutions and downsampling, there is a series of upsampling steps through which the classification is propagated towards higher resolution layers and finally returns to the original resolution of the input. The shape of the architecture is more or less symmetric with the shape of the letter "U", hence the name. The U-Net can be trained end-to-end from relatively few pairs or patches of images and their corresponding classes. Applications of U-Nets include cell counting, detection, and morphometry, [195], automatic brain tumour detection and segmentation [196], and texture segmentation [197].

As thousands of training images are usually beyond reach in biomedical tasks a new architecture was presented. [88] build upon a more elegant architecture, the so-called "fully convolutional network" [180] and modified and extended this architecture such that it works with very few training images and yields more precise segmentation (Figure6.6). The main idea is in [180] to supplement a usual contracting network by successive layers, where pooling operators are replaced by up-sampling operators. Hence, these layers increase the resolution of the output. In order to localise, high resolution features from the contracting path are combined with the up-sampled output. A successive convolution layer can then learn to assemble a more precise output based on this information. One important modification in this architecture is that in the up-sampling part we have also a large number of feature channels, which allow the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting path, and yields a
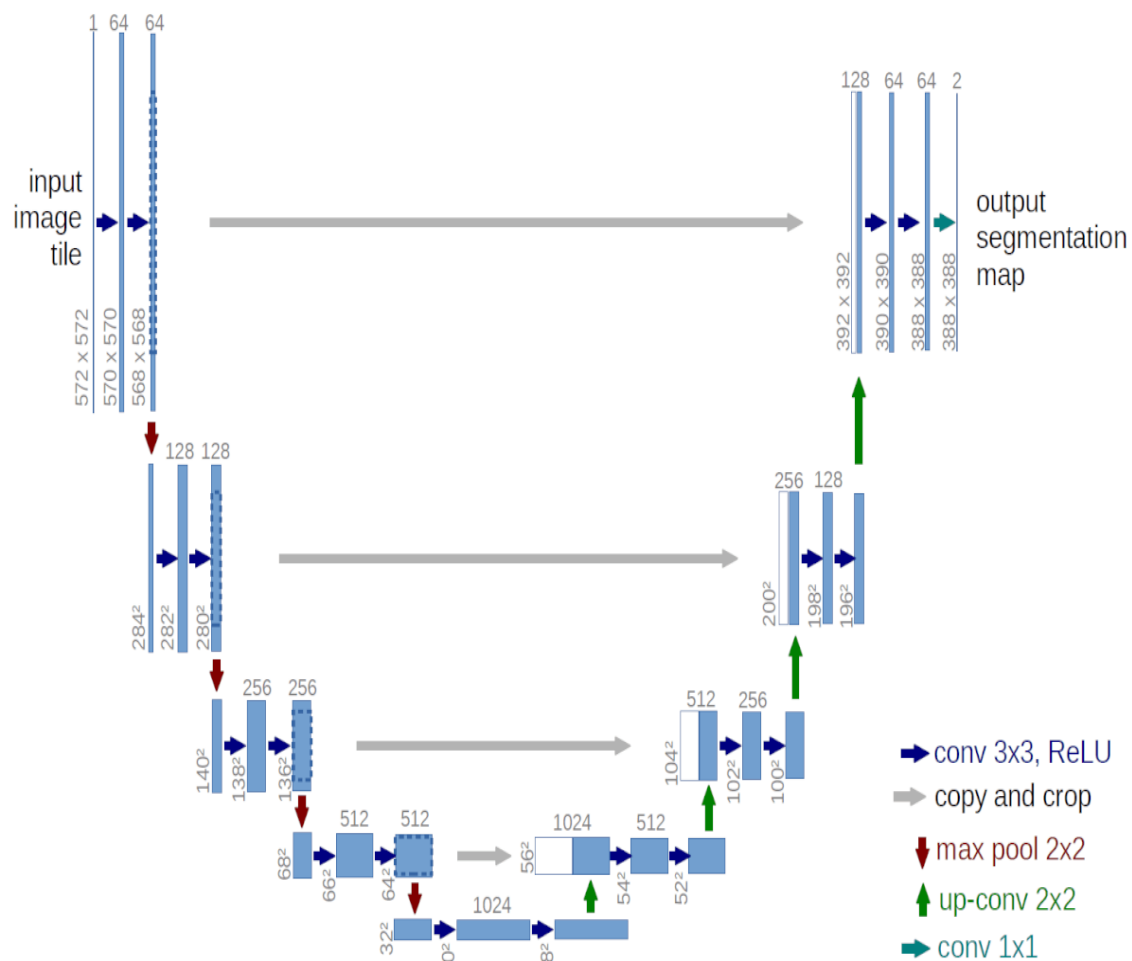
Fig. 6.6 Illustration of U-net architecture (example for 32×32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. Image adapted from [193].

U-shaped architecture. The network does not have any fully connected layers and only uses the valid part of each convolution, i.e., the segmentation map only contains the pixels, for which the full context is available in the input image. This strategy allows the seamless segmentation of arbitrarily large images by an overlap-tile strategy. To predict the pixels in the border region of the image, the missing context is extrapolated by mirroring the input image. This tiling strategy is important to apply the network to large images, since otherwise the resolution would be limited by the GPU memory. As there is very little training data available, we use excessive data augmentation by applying elastic deformations to the available training images. This allows the network to learn in-variance to such deformations, without the need to see these transformations in the annotated image corpus. This is particularly important in biomedical segmentation, since deformation used to be the most common variation in tissue and realistic deformations can be simulated efficiently. Another challenge in many cell segmentation tasks is the separation of touching objects of the same class. To this end, the use of a weighted loss is proposed, where the separating background labels between touching cells obtain a large weight in the loss function.

Network architecture

The network architecture is illustrated in Figure 6.6. It consists of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3×3 convolutions (un-padded convolutions), each followed by a rectified linear unit (ReLU) and a 2×2 max pooling operation with stride 2 for down-sampling. At each down-sampling step, the number of feature channels is doubled. Every step in the expansive path consists of an up-sampling of the feature map followed by a 2×2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1×1 convolution is used to map each 64-component feature vector to the desired number of classes.

In total the network has 23 convolutional layers. To allow a seamless tiling of the output segmentation map, it is important to select the input tile size such that all 2×2 max-pooling operations are applied to a layer with an even x- and y-size.

The Need for Up-sampling

The output of semantic segmentation is not just a class label or some bounding box parameters. In-fact the output is a complete high resolution image in which all the pixels are classified. Thus if we use a regular convolutional network with pooling layers and dense layers, we will lose the "where" information and only retain the "what" information which is not what we want.

In case of segmentation we need both "what" as well as "where" information. Hence there is a need to up sample the image, i.e. convert a low resolution image to a high resolution image to recover the "where" information.

There are various methods to conduct up-sampling operation: Nearest neighbor interpolation, Bi-linear interpolation, and Bi-cubic interpolation. All these methods involve some interpolation method which we need to chose when deciding a network architecture. It is like a manual feature engineering and there is nothing that the network can learn about.

In the literature, there are many techniques to up sample an image. Some of them are bi-linear interpolation, cubic interpolation, nearest neighbor interpolation, unpooling, transposed convolution, etc. However in most state of the art networks, transposed convolution is the preferred choice for up sampling an image. If we want our network to learn how to up-sample optimally, we can use the transposed convolution. It does not use a predefined interpolation method. It has learnable parameters.

Transposed Convolution

Transposed convolution (sometimes also called as deconvolution or fractionally strided convolution) is a technique to perform up sampling of an image with learnable parameters.

Transposed Convolution is the most preferred choice to perform up sampling, which basically learns parameters through back propagation to convert a low resolution image

to a high resolution image. The transposed convolution operation forms the same connectivity as the normal convolution but in the backward direction. We can use it to conduct up-sampling. Moreover, the weights in the transposed convolution are learnable. So we do not need a predefined interpolation method.

Even though it is called the transposed convolution, it does not mean that we take some existing convolution matrix and use the transposed version. The main point is that the association between the input and the output is handled in the backward fashion compared with a standard convolution matrix (one-to-many rather than many-to-one association). As such, the transposed convolution is not a convolution. But we can emulate the transposed convolution using a convolution. We up-sample the input by adding zeros between the values in the input matrix in a way that the direct convolution produces the same effect as the transposed convolution.

6.8 Pre-trained deep learning architectures

To illustrate the training procedure, this example trains Deeplab v3+ [1], one type of convolutional neural network (CNN) designed for semantic image segmentation.

In this work we trained the Deeplab v3+ network, one type of convolutional neural network (CNN) designed for semantic image segmentation, to perform semantic segmentation of HeLa cells imaged with SBF SEM.

The basic structures and fundamentals of deep learning architectures were widely explained in literature, but for completeness, a brief description of each of these three pre-trained deep learning architecture is given below.

VGG16: VGG16 is a convolution neural network (CNN) [77, 79] and it was created by Simonyan and Zisserman of the University of Oxford with a basic idea to keep the architectural design, simple and deeper. The model was runner-up at ILSVRC [198] 2014 with 7.3% error rate. They have created networks with layer weights varying between 11 to 19. The best result was obtained by 16 weight layers - hence the name VGG16.
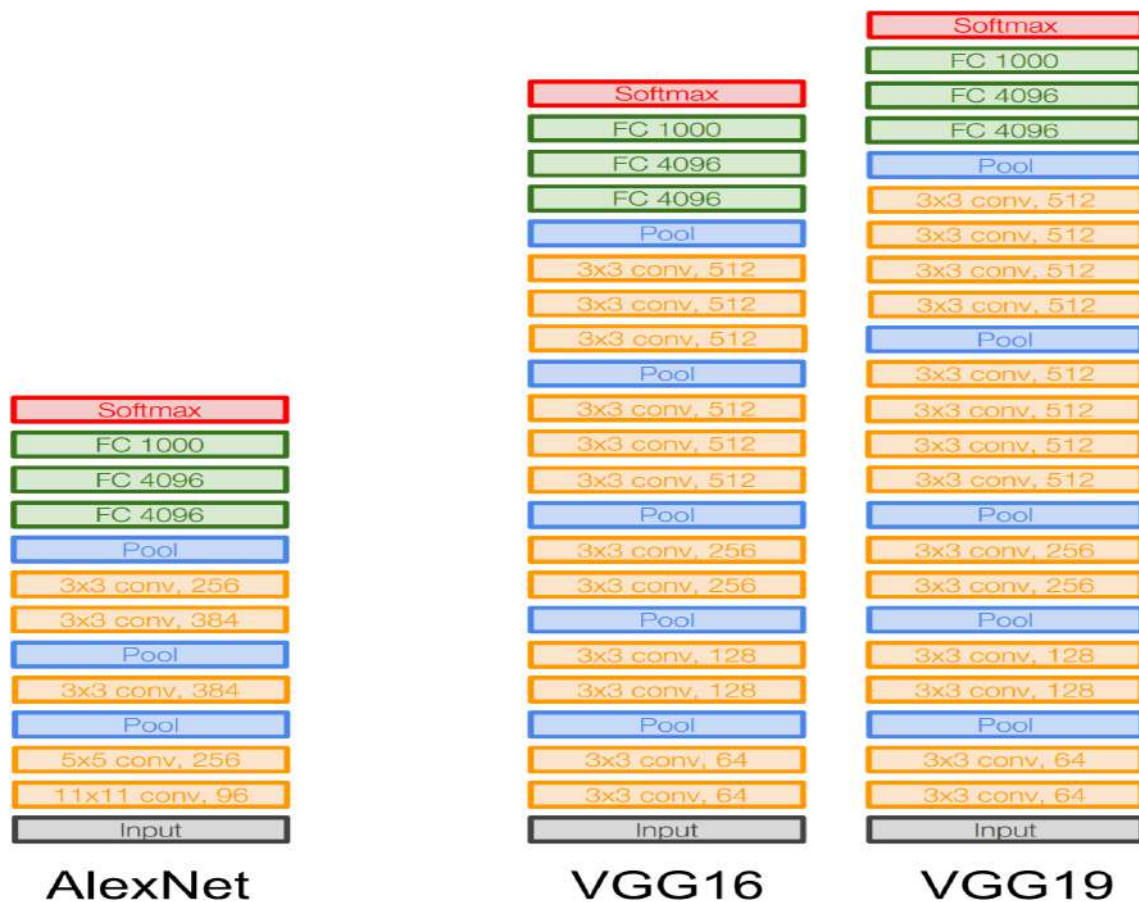
Fig. 6.7 Schematics of AlexNet, VGG16, and VGG19 layers. AlexNet has five convolution layers, and two fully connected layers before the final fully connected layer going to the output classes. It was trained on ImageNet, with inputs at a size $227 \times 227 \times 3$ images. The first layer of AlexNet is a convolution layer and it is 11×11 filters, 96 of these applied at stride 4. The second layer is a pooling layer with 3 filters of 3×3 applied at stride 2. There are 11×11 filters at the beginning, then five by five and some three by three filters. In the end, there is a couple of fully connected layers of size 4096 and finally, the last layer is going to the soft-max, which is going to the 1000 ImageNet classes. This architecture is the first use of the ReLU non-linearity. AlexNet was the winner of the ImageNet Large Scale Visual Recognition Challenge(ILSVRC) classification the benchmark in 2012. VGG16 is 16 layer architecture with a pair of convolution layers, poolings layer and at the end fully connected layer. VGG network is the idea of much deeper networks and with much smaller filters. VGGNet increased the number of layers from eight layers in AlexNet. It had models with 16 to 19 layers variant of VGGNet. One key thing is that these models kept very small filters with 3×3 convolution all the way, which is basically the smallest convolution filter size that is looking at a little bit of the neighbouring pixels. And they just kept this very simple structure of 3×3 convolutions with the periodic pooling all the way through the network. Image credit to [187].

Figure 6.7 illustrates the basic network architectures of AlexNet, VGG16, and VGG19. The 16 in VGG16 refers to it has 16 layers that have weights - hence the name VGG16 [199]. Figure 6.8 shows different layers in VGG16 architecture used for semantic segmentation of HeLa cells.

ResNet18: ResNet18 is mainly inspired by the philosophy of VGG16 [200], its total number of weighted layers is 18 and has an image input size of 224-by-224. The convolutional layers mostly have 3×3 filters and follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer. Shortcut connections which turn the network into its counterpart residual version are inserted. The identity shortcuts can be directly used when the input and output are of the same dimensions.

Inception-ResNet-v2:

Inception-ResNet-v2 is the combination of two of residual connections and the latest revised version of the Inception architecture [202] and it has an image input size of 299-by-299 and 164 layers deep. The basic network architecture of Inception-ResNet-v2 is given in Figure 6.10. In the Inception-ResNet block, multiple sized convolutional filters are combined by residual connections. The usage of residual connections not only avoids the degradation problem caused by deep structures but also reduces the training time [201]. The 35×35, 17×17 and 8×8 grid modules, known as Inception-A, Inception-B and Inception-C blocks, are used in the Inception-ResNet-v2 network. Similar to ResNet18, Inception-ResNet-v2 [203] is also a convolutional neural network that is trained on more than a million images from the ImageNet [198] database. This network has about 55.9 million (approx) parameters.

The details of the architectures are displayed in Table 6.1.
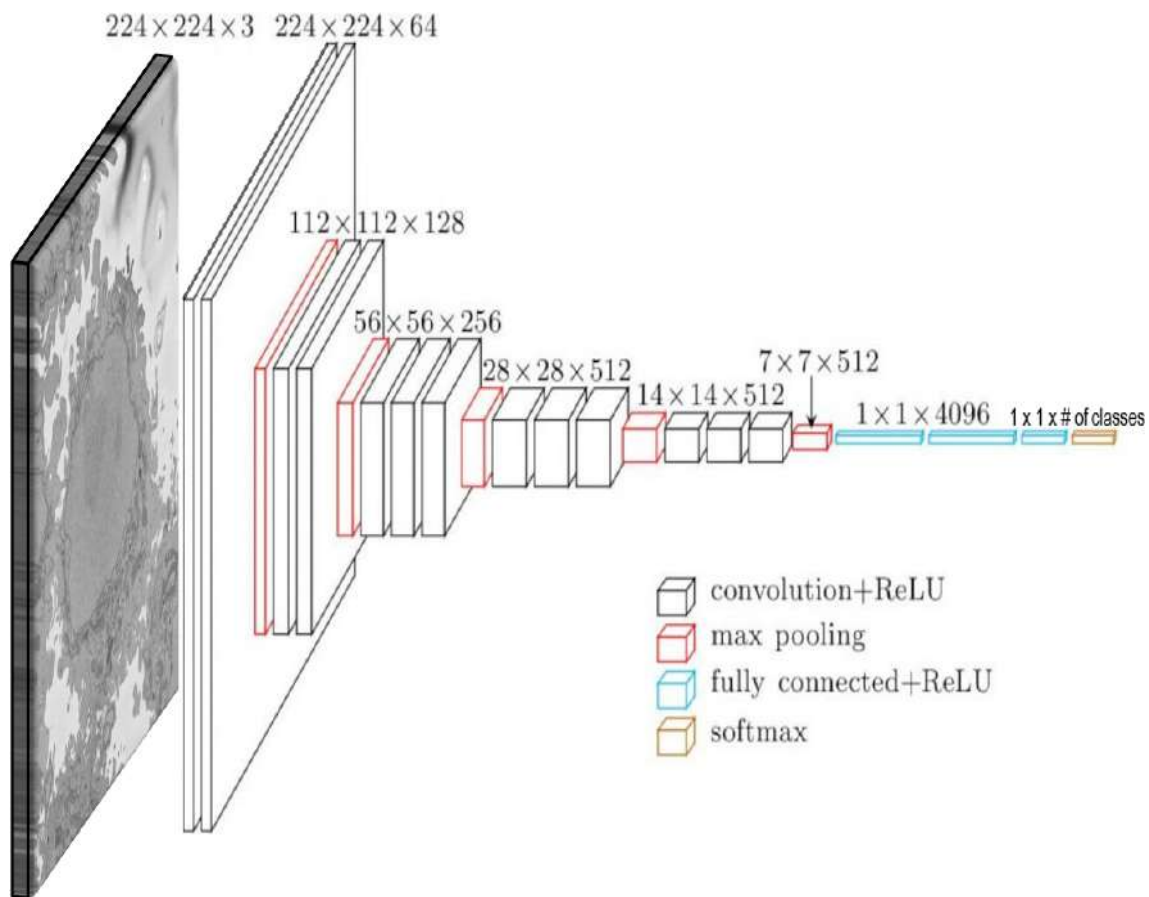
Fig. 6.8 VGG16 architecture layers for semantic segmentation of HeLa cell images. Training took 21.25 hours on a single CPU and the accuracy and loss during VGG16 training was plotted. The optimisation algorithm used for training is stochastic gradient descent with momentum (SGDM). The hyperparameters used for SGDM were specified. Data augmentation is used during training to provide more examples to the network because it helps improve the accuracy of the network. Here, random left/right reflection and random X/Y translation of $+/-10$ pixels is used for data augmentation. The training data and data augmentation selections were combined. The deep network reads batches of training data, applies data augmentation, and sends the augmented data to the training algorithm. The VGG16 deep network training had 100 epochs with learning rate 0.001. Iterations per epoch was 45 therefore total number of iterations for the whole training was 4500.
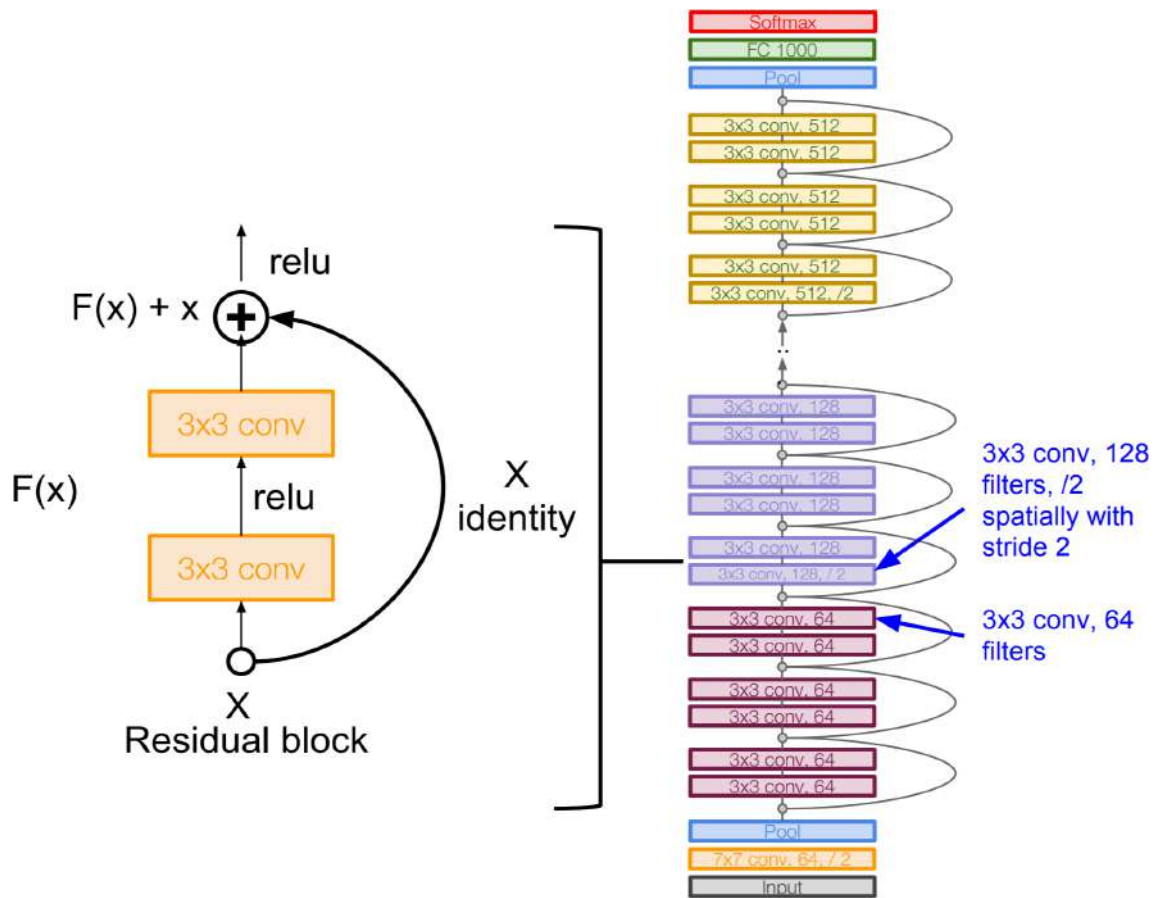
Fig. 6.9 Illustration the basic architecture of Residual networks (ResNet18) layers. ResNet [200] were proposed as a family of multiple deep neural networks with similar structures but different depths. ResNet introduces a structure called residual learning unit to alleviate the degradation of deep neural networks. This unit's structure is a feed forward network with a shortcut connection which adds new inputs into the network and generates new outputs. The main merit of this unit [201] is that it produces better classification accuracy without increasing the complexity of the model. Image credit to [187].

Table 6.1 Summary of convolutional neural networks (CNNs) that were used in this work.

No.	Network	Depth	Image Input Size	Reference
1	ResNet18	18	224-by-224	[200]
2	VGG16	16	224-by-224	[77]
3	Inception-ResNet-v2	164	299-by-299	[202]
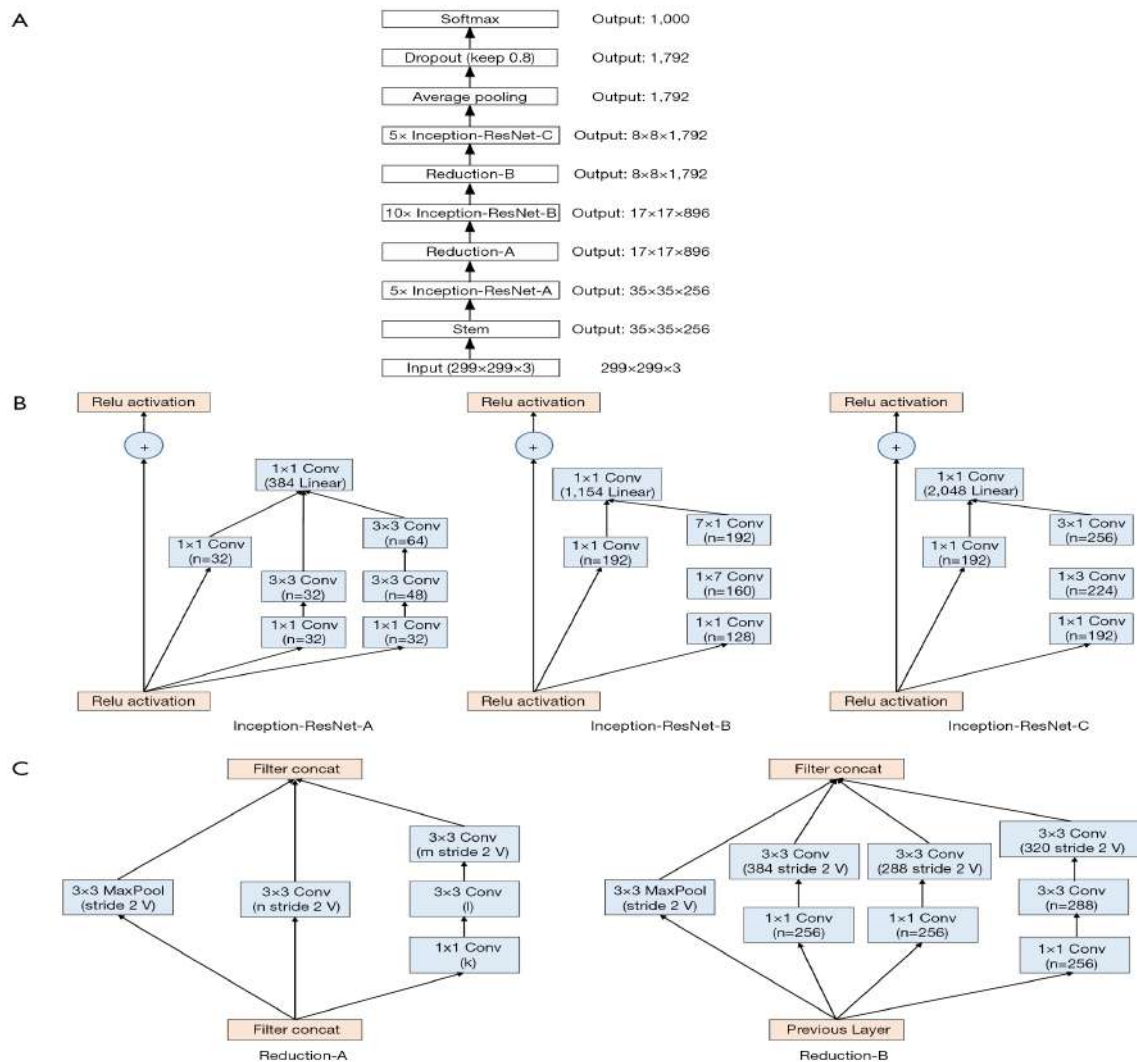
Fig. 6.10 Basic network architecture of Inception-ResNet-v2 layers. Inception-Resnet-v2 [202] is formulated based on a combination of the Inception structure and the Residual connection. In the Inception-Resnet block, multiple sized convolutional filters are combined by residual connections [201]. The usage of residual connections not only avoids the degradation problem caused by deep structures but also reduces the training time. Image credit to [203].
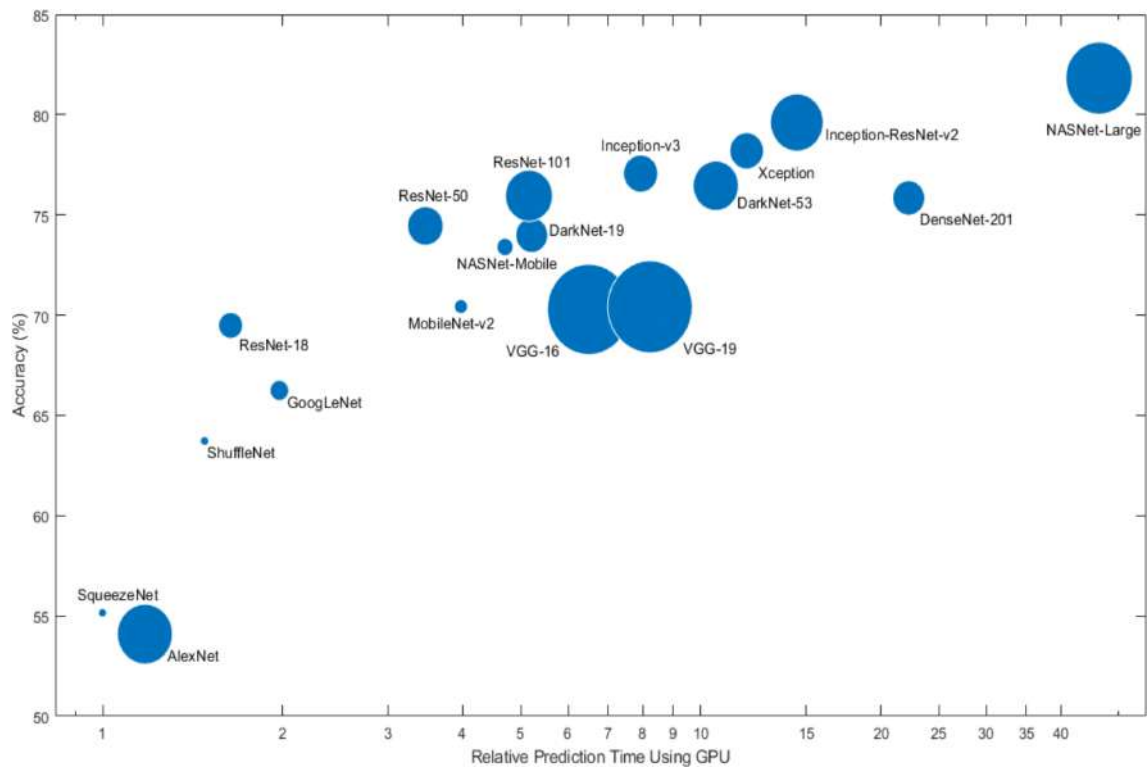
Fig. 6.11 Pre trained deep neural networks comparison. VGG16, ResNet18, and Inception-ResNet-v2 were selected due to their good balance between accuracy and computational complexity, especially ResNet and Inception-ResNet-v2, which outperform other common configurations and are at the Pareto frontier considering accuracy and complexity [204–206]. Image credit to [207].
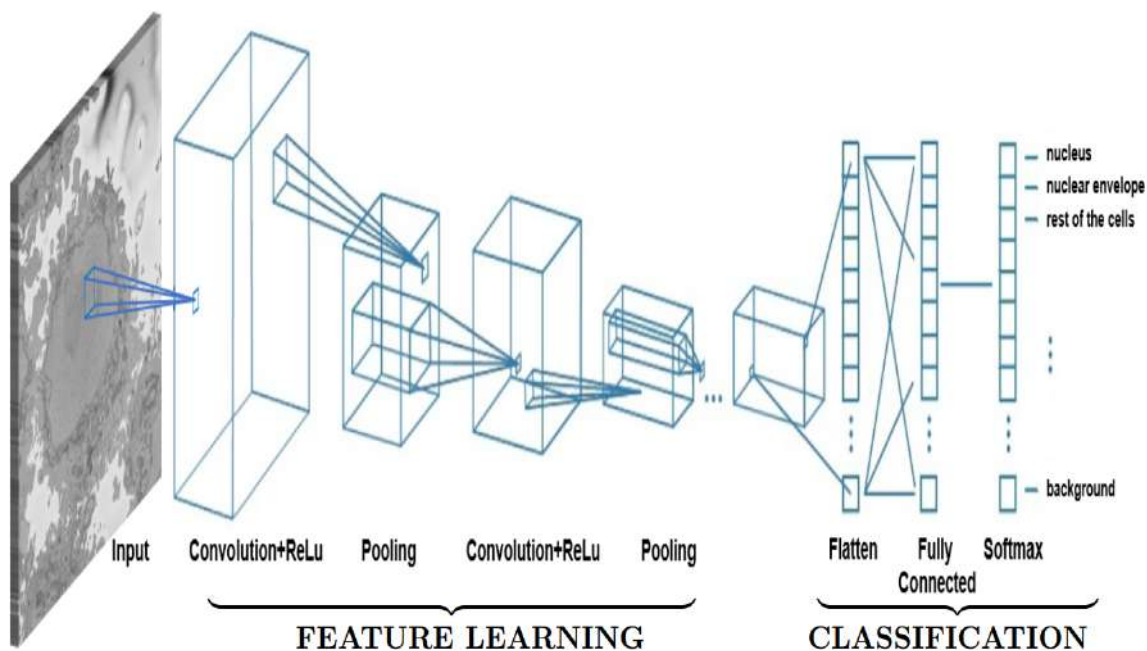
Fig. 6.12 Illustration of the typical architecture of a Convolutional Neural Network (CNN) for the analysis of HeLa images imaged with electron microscopy (EM). A CNN is composed of an input layer, an output layer, and many hidden layers in between. These layers perform operations that alter the data with the intent of learning features specific to the data. Three of the most common layers are: convolution, activation or Rectified linear unit (ReLU), and pooling. The input image is passed through a stack of convolutional layers, where filters are used with a very small receptive field: 3×3. Every convolutional layer transforms its input to a three-dimensional output volume of neuron activations. The convolution stride is fixed to 1 pixel; the spatial padding of convolution layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1 pixel for 3×3 convolution layers. ReLU allows for faster and more effective training by mapping negative values to zero and maintaining positive values. This is sometimes referred to as activation, because only the activated features are carried forward into the next layer. Pooling simplifies the output by performing nonlinear down sampling, reducing the number of parameters that the network needs to learn. The pooling layer down samples the volume spatially, independently in each feature map of its input volume. Spatial pooling is carried out by five max-pooling layers, which follow some of the convolution layers. Max-pooling is performed over a 2×2 pixel window, with stride 2. A stack of convolutional layers is followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third, the soft-max layer, performs 4-way classification and thus contains 4 channels (one for each class - nucleus, nuclear envelope, rest of the cell, and background). These operations are repeated over tens or hundreds of layers (VGG16 has approximately 138 million parameters), with each layer learning to identify different features.

6.9 Summary

In this chapter, history of Artificial Intelligence, Machine learning and Deep learning were presented and their fundamentals were explained in detail. Convolutional neural network (CNN) and its structure were presented so readers could be guided to the following chapter where CNNs are used for semantic segmentation of HeLa cells. CNNs are quite large neural networks with hundreds of millions of learnable parameters that are generally hard to train by individuals with limited computing resources. The importance of transfer learning was also emphasised here and different techniques for training and optimisation of deep learning were explained. Transfer learning allows individuals to recycle pre-trained models for other purposes with minimal (re)training or even without retraining at all for the case of reverse image search. We have demonstrated that the convolutional neural networks are capable of extracting the features associated with an image.

The main property of CNNs is that the weights of the neurons in the convolutional layers are learnable. They can be seen as a set of learnable filters. Each filter is small spatially, and extends through the full depth of the input volume. During the forward pass, the input volume is convolved with each filter producing 2D activation maps, one per filter. The activation maps are then stacked together to form one output volume whose depth is equal to the number of convolution filters. During the convolution, the stride s specifies the number of pixels with which we slide the convolution filter. A stride of one, i.e. $s = 1$, means that the filters slide one pixel at a time. $s = 2$ means that the filters move two pixels at a time as we slide them around. If $s > 1$ then this procedure will produce spatially smaller output volumes.

VGG16 has been widely used in a variety of image classification problems. ResNet solves the problem of vanishing/exploding gradients and was the winner of ILSVRC 2015 [78]. Inception-ResNet-v2 employs dropout to avoid overfitting and is seen as the successor of GoogLeNet [202].

These networks were selected due to their good balance between accuracy and computational complexity, especially ResNet and Inception-ResNet-v2, which outperform

other common configurations and are at the Pareto frontier (Figure 6.11 considering accuracy and complexity [204–206].

In the last section of this chapter, U-Net, which is considered one of the standard CNN architectures for image classification tasks [197], was presented. In these tasks we need not only to define the whole image by its class but also to segment areas of an image by class, i.e. produce a mask that will separate an image into several classes. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localisation.

The network is trained in end-to-end fashion from very few images [88] and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images, U-Net won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512×512 image takes less than a second on a modern GPU.

The U-net architecture achieves outstanding performance on very different biomedical segmentation applications. It only needs very few annotated images and has a very reasonable training time of just 10 hours on NVidia Titan GPU (6 GB).

Chapter 7

Experiments and Results

In this chapter, the principal experiments performed in the data sets are explained, and its corresponding results presented. The experiments presented include all the contributions by the author, spanning the different analyses made in the different data sets explained in Chapter 2 and 3.

The chapter is subdivided into three major sections related with (i) the segmentation of nuclear envelope of HeLa cells through traditional image processing algorithm, (ii) the modelling of segmented nuclear envelope of HeLa cell against a spheroid and its analysis and the foremost conclusions in such work and (iii) research in semantic segmentation by using three pre-trained deep learning architectures. The key research presented in this chapter is the investigations done on the HeLa cell images acquired by the **Electron Microscopy Science Technology Platform (EM STP) at The Francis Crick Institute** in London [40], where various results were found. Later investigations were focused on Chlamydia trachomatis-infected HeLa cells obtained from **The Cell Image Library - CIL50051 and CIL50061**.

These include detection of background and segmentation the disjoint regions of nuclear envelope of HeLa cells because of cell division over-viewed in Chapter 4. From the algorithms described in Chapter 4 and 5, the classification and semantic segmentation of single HeLa cell is presented; finally, the comparison, based on accuracy

and Jaccard similarity index, between the traditional image processing algorithm and three pre-trained deep learning architectures is given.

All contributions, provided both technical and biological insights, were found from the data sets. All sections describe the experiments referring to the algorithms and techniques specified in previous chapters. Then, the results are presented, making the distinction between results previously published and extensions made for this dissertation or that were not published in due time. Finally, a brief discussion on the specific experiments is provided.

7.1   Ground truth (GT) and training data

Two cells (i.e. 600 slices) were manually segmented to provide a ground truth to assess the accuracy of the image processing algorithm. Each cell was segmented by different persons without knowledge of each other and with different acquisition conditions. The first cell (Figure 7.8b) was segmented using Amira and a Wacom Cintiq 24HD interactive pen display by one of the collaborators from [208] and took around 30 hours. The second cell (Figure 7.8d) was segmented by the author of this dissertation with a Wacom pen-and-tablet using the MATLAB® *roipoly* function and took around 47 hours. In order to determine whether disjoint regions belong to the nucleus, the user scrolled up and down through neighbouring slices to check connectivity of the regions. In a few cases, there were discontinuities in the line of the NE, and thus to morphological dilation was used to ensure a closed contour.

To provide GT for pre-trained deep learning architectures, we also obtained a separate ground truth. The next part presents the process followed to generate it.

As explained in the previous section, three hundred slices, corresponding to one cell (red box in Figure 5.2a), were segmented with a combination of manual and algorithmic steps to provide the first part of a ground truth (GT) for training of deep learning architectures. The NE was delineated using Amira and a Wacom Cintiq
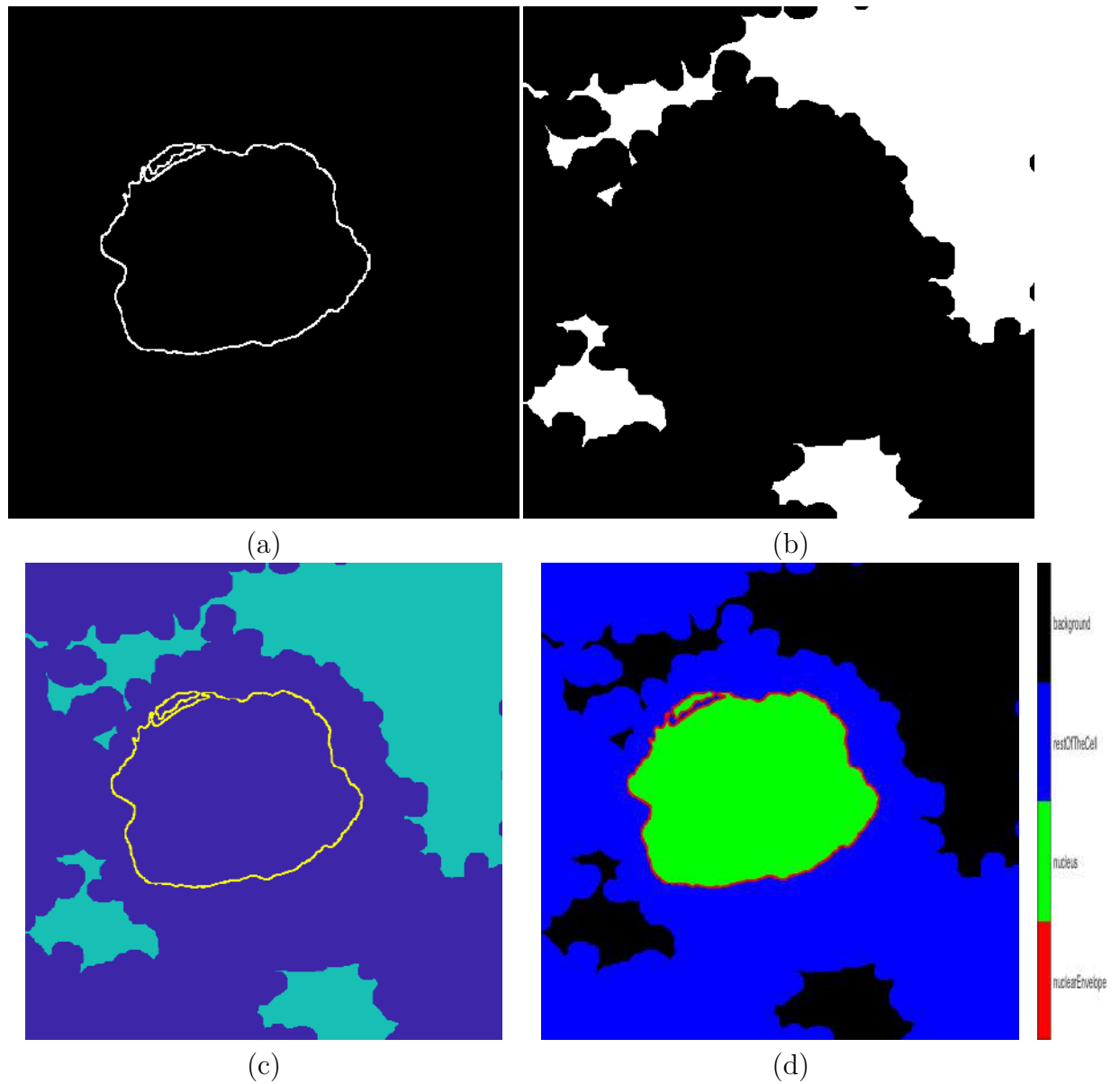
Fig. 7.1 Illustration of the process followed to generate the ground truth. (a) Manual delineation of the nuclear envelope (NE). (b) Background segmentation. (c) Combination of the NE and the background. (d) Labelled image with a colour bar indicating the four different classes.

24HD interactive pen display by Anne E. Weston at the Francis Crick Institute in approximately 30 hours (Figure 7.1(a)).

The background of HeLa cell image was segmented automatically in the following way. Image-processing algorithm was deployed and HeLa images were low-pass filtered with a Gaussian kernel with size $h = 7$ and standard deviation $\sigma = 2$ to remove high frequency noise. Canny edge detection was used to detect abrupt changes of intensity - edges. In order to to connect disjoint edges, they were further dilated. A blurrier edge detection was used to create large superpixels and morphological operators discarded small regions, filled holes inside larger regions to detect the background (Figure 7.1(b)).

Next, the NE and the background were combined (Figure 7.1(c)), exported to MATLAB® *Image Labeler* and then four classes - nuclear envelope, nucleus, rest of the cell, and background were labelled (Figure 7.1(d)). Since the pre-trained neural networks expect colour images, the GT was replicated to create an RGB image with three channels.

7.2 Deep learning semantic segmentation of HeLa cells

A semantic segmentation network classifies every pixel in an image, resulting in an image that is segmented by class. Applications for semantic segmentation include road segmentation for autonomous driving and cancer cell segmentation for medical diagnosis.

7.2.1 Description of Network Training

Data processing

The HeLa Pixel-Labeled Images data set, shown in Figure 7.1d, provides pixel-level labels for four semantic classes including nucleus, nuclear envelope, rest of the cell and background. These classes were specified before the training. In order to see the

<center>(a)   (b)</center>
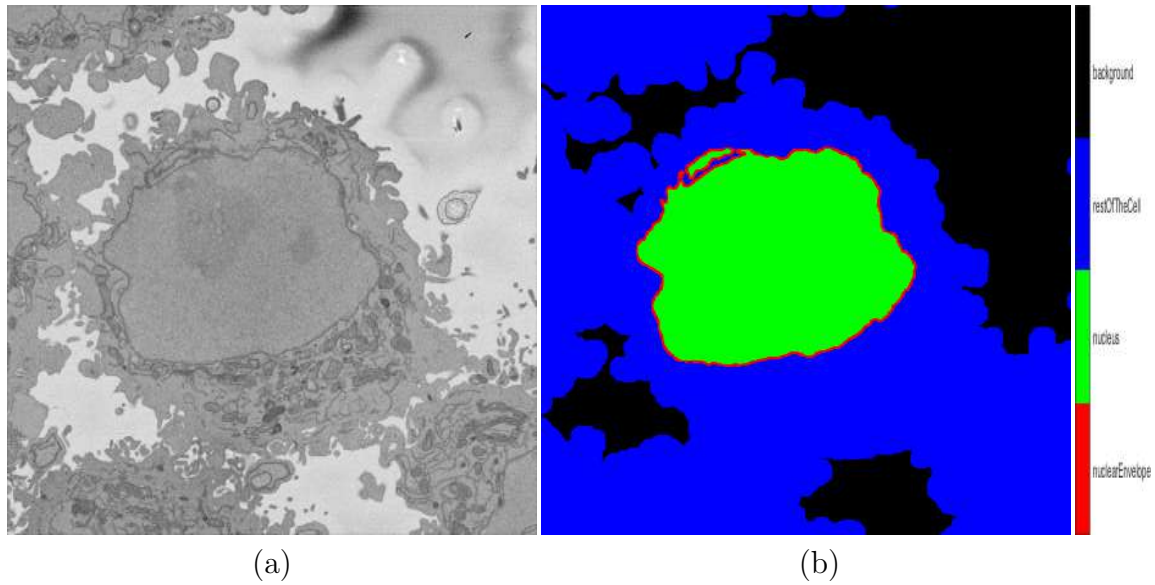
Fig. 7.2 An illustration of semantic segmentation of a HeLa cell. A semantic segmentation network classifies every pixel in an image, resulting in an image that is segmented by class. (a) A filtered HeLa image (slice 118/300). (b) Semantic segmentation of the same image. All pixels were classified as one of the four possible outcomes - nuclear envelope (NE), nucleus, rest of the cell, and background.

distribution of class labels in the data set, the number of pixels by class label were counted. Ideally, all classes would have an equal number of observations. However, the classes in the data set are imbalanced, which is a common issue in biomedical imaging. HeLa images have more nucleus, rest of the cell and, background pixels than the nuclear envelope pixels because nucleus, rest of the cell and background cover more area in the image. If not handled correctly, this imbalance can be detrimental to the learning process because the learning is biased in favour of the dominant classes. To improve training, class weighting was used to balance the classes. The pixel label counts computed earlier was used in order to calculate the median frequency class weights.

The images and labelled training data in the HeLa data set are $2000 \times 2000 \times 1$. In order to reduce training time and memory usage, all images and pixel label images were resized to $360 \times 480 \times 3$. The network was trained using 60% of the images from the data set. The rest of the images (40%) were used to test the network after training.

The network randomly splits the image and pixel label data into a training and test set. The whole data set for each pattern has been divided into two. Sixty percent is kept for training and its number has been increased artificially by using image augmentation techniques such as translation and reflections. EM images do not contain any colour information therefore they have the dimensions $(n_h, n_w, n_d) = (2000, 2000, 1)$.

An image data augmenter in neural network configures a set of pre-processing options for image augmentation, such as resizing, rotation, and reflection and generates batches of augmented images. Data augmentation is used during training to provide more examples to the network because it helps improve the accuracy of the network, prevent the network from over fitting [85] and memorising the exact details of the training images.

Training of the networks

The training took 21.25 hours on a single CPU and the training plot was obtained to check the accuracy and loss during training of the three pre-trained deep neural networks (VGG16, ResNet18 and Inception-ResNet-v2). The optimisation algorithm used for training is stochastic gradient descent with momentum (sgdm) and this was specified in training options. The sgdm algorithm can oscillate along the path of steepest descent towards the optimum. Adding a momentum term to the parameter update is one way to reduce this oscillation [209].

Three hundred images were used for training. These were split 60%/40% for training and testing. Data augmentation was applied with random translations of ± 10 pixels on the horizontal and vertical axes and random reflection on the horizontal axis. The training data and data augmentation selections were combined. The deep network reads batches of training data, applies data augmentation, and sends the augmented data to the training algorithm.

The VGG16 deep network training had 100 epochs with learning rate 0.001. Iterations per epoch was 45 therefore total number of iterations for the whole training was 4500. Similarly, training took approximately 1.5 hours for ResNet18 and 4.7 hours for

Inception-ResNet-v2 with 30 epochs and 660 iterations and learning rate 0.00009 on a single CPU.

The U-Net architecture was trained from scratch with $36,000$ training images and labels constructed from a subset of the data and the ground truth with size 128×128.

U-Net: The U-Net [88] architecture was constructed with the following layers: *Input, Convolutional, ReLu, Max Pooling, Transposed Convolutional, Convolutional, Softmax* and *Pixel Classification.* Several training configurations were tested, number of layers $(10, 15, 20)$, number of epochs $(5, 10, 15)$, classifiers stochastic gradient descent (sgdm), Adam (Adam) [191] and Root Mean Square Propagation (RMSprop) and size of training images $(64 \times 64, 128 \times 128)$ and the best results were obtained with 36000 training pairs, 15 epochs, 20 layers and RMSprop with 128×128 training images. The downsampling and upsampling blocks had the following configuration:

- Input,

- Convolutional, ReLu, Max Pooling,

- Convolutional, ReLu, Max Pooling,

- Convolutional, ReLu, Max Pooling,

- Convolutional, ReLu,

- Transposed Convolutional, Convolutional,

- Transposed Convolutional, Convolutional,

- Transposed Convolutional, Convolutional,

- Softmax,

- Pixel Classification.

(a)   **(b)**
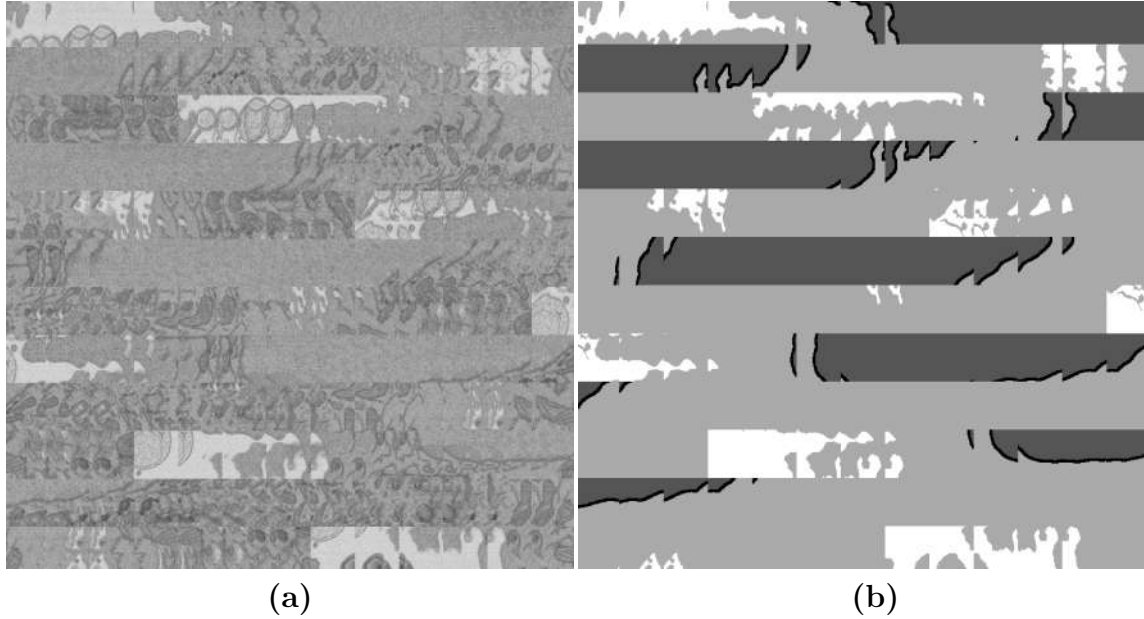
Fig. 7.3 Illustration of the image-label pairs created to train the U-Net architecture. (a) A sample of regions, each 128×128 pixels placed next to each other as a montage. (b) Montage of the labels corresponding to the regions of (a). The labels contain four classes, from dark to bright: Nuclear Envelope, Nucleus, Cell, Background.

To train the U-Net, the image input layer was configured for the 128×128 patches. The patches were formed from a subset of the images, specifically the odd slices between slice 101 and slice 180. The images were cropped to regions of 128×128 pixels with an overlap of 50%. An illustration of the data and the labels is shown in Figure 7.3.

Comparison of segmentation results

The semantic segmentation results from image-processing algorithm, and Deeplab v3+ networks were obtained.

These results were compared with the labelled data shown in (Figure 7.1d) and accuracy and Jaccard similarity index were calculated to asses the accuracy of the network. In order to measure accuracy for the data set, deep neural networks were run on the entire test set.

Visually, the semantic segmentation results overlap well for classes such as nucleus, rest of the cell, and background. However, smaller objects like the nuclear envelope are not as accurate. The amount of overlap per class can be measured using the
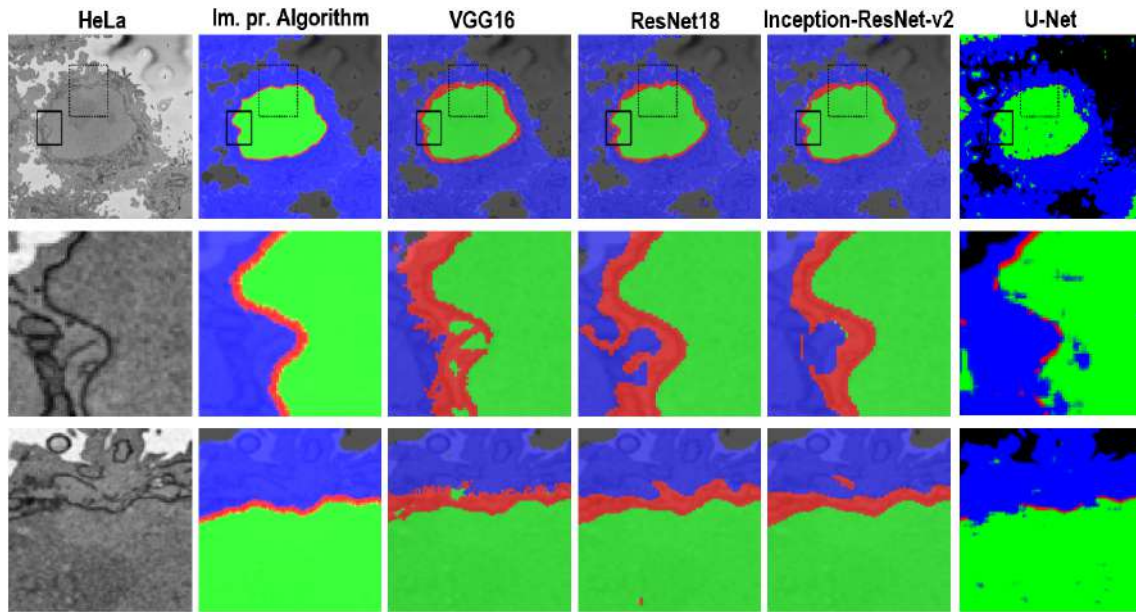
Fig. 7.4 A representative HeLa image and semantic segmentation results overlaid on the filtered image. Top row: Boxes denote two Region of Interests (ROIs) on filtered image as well as semantic segmentation results. Results were obtained from all algorithms - a traditional and four deep learning architectures. The ROIs are magnified and shown in the middle and bottom rows. Middle row: ROI 1 corresponding black box with solid line. Bottom row: ROI 2 - box with dashed line. Differences in segmentation results can be seen from these ROIs - the red dot obtained from ResNet18 semantic segmentation on bottom row and 4th column. Notice the U-Net segmentation in the last column. The result mixes the nucleus with cell, and background with other things but it also detects as nucleus the nucleus of the other cells that are in the periphery, which scores down with the ground truth (GT) but in reality is more precise. Even the nuclear envelope (NE) is detected in some sections.

intersection-over-union (IoU) metric, also known as the Jaccard index. Accuracy also was measured for all slices and compared with our algorithm. Although the overall data set performance is quite high, the class metrics show that underrepresented classes such as nuclear envelope is not segmented as well as classes such as nucleus, rest of the cell, and background. Additional data that includes more samples of the underrepresented classes might help improve the results.

The results provided by the U-Net semantic segmentation were very interesting. The training of the U-Net provided sufficient samples for the network to distinguish the nuclei of cells, that is, not only of the cell that is located in the centre, but also of
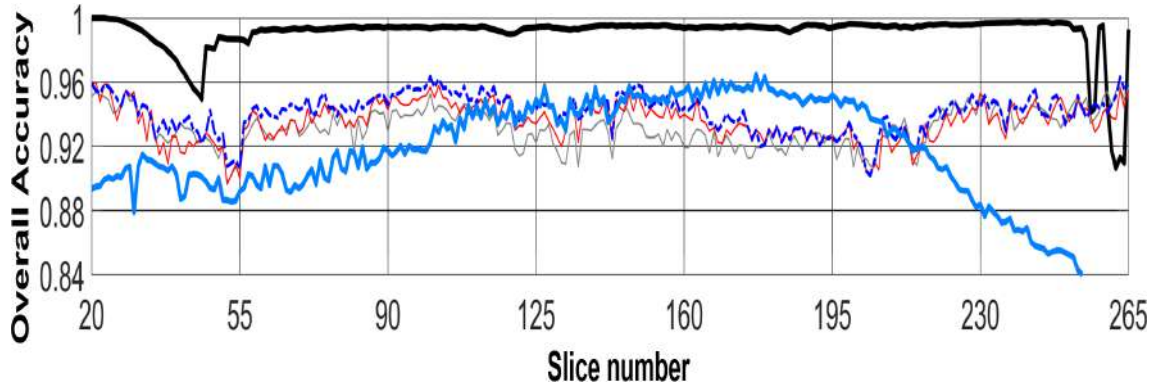
Fig. 7.5 Numerical comparison between the image-processing algorithm and deep learning architectures by using the ratio of true positive and true negative over the total number of pixels (overall accuracy). The training for the image-processing algorithm was performed on one cell and tested on all the other six cells. With U-Net, training was performed on every other slice and three pre-trained deep learning architectures (VGG16, ResNet18, Inception-ResNet-v2) were trained with 60 % of the data and tested with the rest (40 %). Equation 4.3 (overall accuracy) was used to obtain this plot.

other nuclei visible within the slices (Figure 7.7). It should be remembered that one assumption of the segmentation task was that there was a single cell in the centre of the volume of interest. This assumption will impact on the results of the U-Net as the ground truth was constructed with a single nucleus in the region. Therefore, for slices closer to the centre the accuracy and Jaccard Index were higher, and as the central nucleus became smaller towards the edges, and other nuclei appeared, these metrics decreased. The construction of a separate ground truth for U-Net, which would reveal a more accurate comparison for U-Net, is beyond the scope of this paper.

7.3 Quantitative comparisons

Given two input curves that are perimeters of segmentation result and manual segmentation or GT, it is natural to ask how similar they are to each other. In order to assess the accuracy of the segmentation, two methodologies were followed. first, the Jaccard similarity index (JI) of intersection over union was calculated to assess the area inside the NE and second, the Hausdorff distance (HD) , a well known similarity measure, of
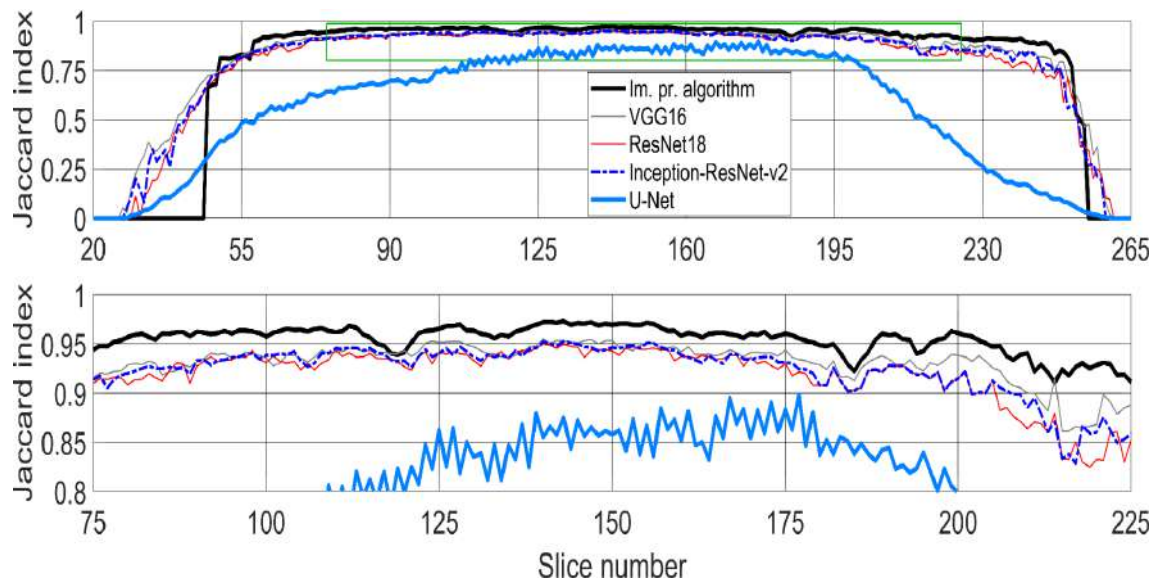
Fig. 7.6 Numerical comparison between the image-processing algorithm and deep learning architectures by using Jaccard similarity index as performance metric.
(Top row) Jaccard similarity index, also known as intersection over union, shows the comparison between the image-processing algorithm and deep learning architectures. Green box denotes the central slices and corresponding Jaccard similarity index that is magnified below. (Bottom row) Jaccard similarity index for central slices (slices between 75/300 and 225/300 - interquartile range (IQR)), for easier comparison.
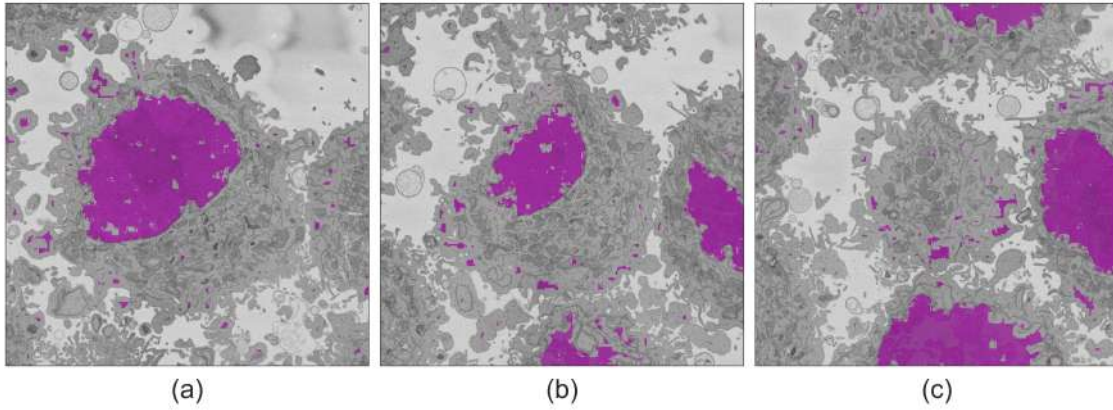
Fig. 7.7 Three slices semantically segmented slices with U-Net. (a) Slice 170, towards the centre of the cell, accuracy = 0.954 , Jaccard Index = 0.854. (b) Slice 220, towards the top of the cell, accuracy = 0.906, Jaccard Index = 0.435. (c) Slice 260, on the very top of the cell, accuracy = 0.818 , Jaccard Index = 0.0007. The area of the nucleus has been highlighted in pink. It should be noticed that in (a) the nuclear area is well segmented but there are other artefactual segmentations. In (b,c) the nuclei of contiguous cells are also segmented. These detections will drive down both accuracy and Jaccard Index.

the maximum of the set of shortest distances between two lines was calculated to assess how far the real boundary of the NE was from the calculated one. The smaller the Hausdorff distance between two shapes the greater is their degree of resemblance [171].

HD [172, 173] is defined as the maximum distance between a point on one curve and its nearest neighbour on the other curve [174, 175]. It is such a metric that enables calculation of how far two subsets of a metric space are from each other.

Segmentation results were compared against the GT using both metrics. To compute JI, the manual segmentation for each slice was morphologically closed to generate a region rather than a line. The HD between the GT and NE was calculated for each slice by comparing the boundaries of the curves.

In this work, images of HeLa cells observed with SBF SEM were semantically segmented with an image-processing algorithm and three pre-trained deep learning architectures (convolutional neural networks - CNNs), VGG16, ResNet18 and Inception-ResNet-v2. The accuracy and Jaccard similarity index of the segmentations against a ground truth were calculated. The image-processing algorithm is fully automatic
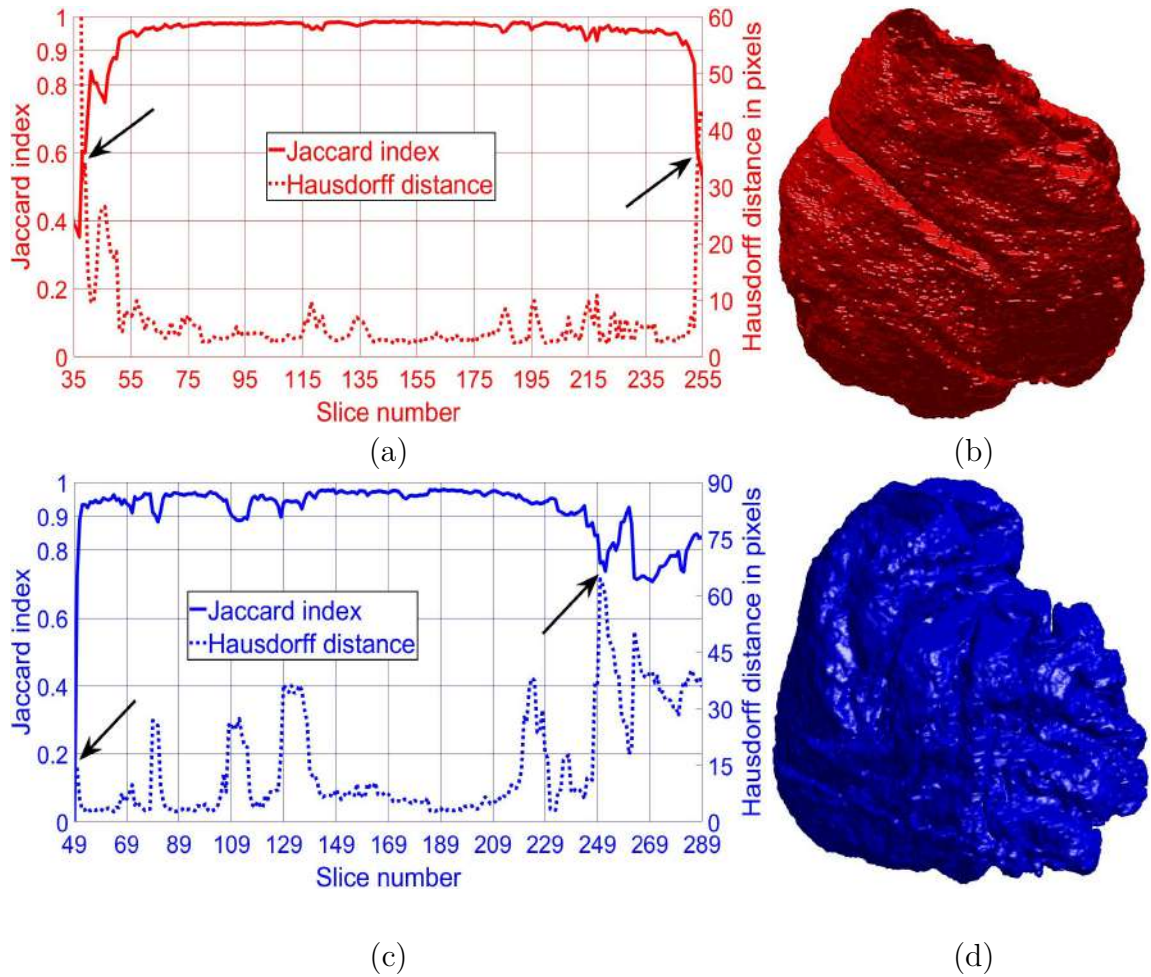
Fig. 7.8 Similarity metrics and final results of the automated segmentation displayed as rendered volumetric surfaces. (a) Jaccard Index (JI) (solid line) and Hausdorff distance (HD) (dotted line) for the segmented nuclear envelope (NE) shown in (b). Notice that the mean JI is 98% and the mean HD is 4 pixels for central slices. (b) final result of the segmentation displayed as a rendered volumetric surface. The algorithm was trained with this cell. (c) JI and HD for the second NE shown in (d). Note the decrease in JI and increase in HD (arrows) towards the upper and lower edges of the NE where the structure tends to become more fragmented in 2D cross-sections and complex regions that may have been missed by the algorithm.

and processed each slice of the HeLa cells in approximately 8 seconds and a whole cell containing 300 slices in approximately 40 minutes. On the other hand, the deep neural network architectures were fine-tuned and trained in 21.25 hours (VGG16), 1.5 hours (ResNet18), and 4.7 hours (Inception-ResNet-v2) to perform semantic segmentation on the whole cell with relatively good accuracy and differences can be appreciated in Figure 7.4 with two ROIs.

Segmentation of the nuclear envelope by all three deep neural networks was outperformed by the image-processing algorithm as shown in Figure 7.4, Figure 7.5, and Figure 7.6 . Visually, the semantic segmentation results overlap well for classes such as nucleus, rest of the cell, and background. However, smaller objects like the nuclear envelope are not as accurate. Although the overall data set performance is quite high, the class metrics show that under represented classes such as nuclear envelope is not segmented as well as classes such as nucleus, rest of the cell, and background.

In this work, a classical and unsupervised image processing algorithm was used to perform semantic segmentation of cancerous HeLa cell images from SBF SEM and compared with three pre-trained deep neural network architectures. The pre-trained deep neural network architectures, VGG16, ResNet18 and Inception-ResNet-v2 were trained in ImageNet and fine-tuned for semantic segmentation of the HeLa cells. Four different classes, nucleus, nuclear envelope, rest of the cell, and background were used in labelling training data for deep neural networks. Two similarity metrics, accuracy and Jaccard index, were calculated so the image-processing algorithm was compared with deep neural networks. As the image-processing algorithm was trained and tested only for the nuclear envelope and nucleus and as the bottom 26 and the top 40 slices did not contain any cell, the difference in Jaccard index, which can be observed away from central slices, is in favour of the deep neural networks towards the top and bottom of the cells (Figure 7.5 and Figure 7.6). Additional data that includes more samples of the under represented classes, nuclear envelope in this case, might help improve the results as the sizes of the classes in the HeLa data set were imbalanced. For the image-processing algorithm this would be expected, as the NE is more irregular on the

top and bottom slices than on the central ones. The segmentation of each cropped cell is fully automatic and unsupervised and the image-processing algorithm segments one slice in approximately 8 seconds and one whole cell in approximately 40 minutes with good accuracy.

As JI does not count true negative (TN), the values decrease towards the top and bottom slices of the cells as the structure was considerably more complex and the areas become much smaller (Figure 7.6 (Middle row) and (Bottom row)). On the other hand, accuracy includes the TN in both numerator and denominator and this, especially in cases where the objects of interest are small and there are large areas of background (e.g. the top and bottom slices of the cell) would render very high accuracy. Therefore, in contrast to JI, accuracy increases in slices towards both top and bottom ends. Overall, the best results were obtained by the image processing segmentation algorithm especially for central slices - slices between 75/300 and 225/300 (Figure 7.6 (Bottom row)).

Deploying deep learning architectures [68] and training to learn patterns and features directly from EM images and to automatically segment the NE and other parts of a cell is indeed necessary. Segmentation effort and assessment variability will be reduced and fast and accurate segmentation will provide a second opinion to support biomedical researchers' decisions as it shortens the time required to segment the cell.

The main contributions of this work are: (a) Generating 300 labelled images of the HeLa cell, shown in (Figure 5.2), defined by four different classes - nuclear envelope, nucleus, rest of the cell, and background. These images were used to train deep learning architectures. (b) an objective comparison, supported by accuracy and Jaccard similarity index, between the image-processing algorithm and three pre-trained deep neural networks - VGG16, ResNet18 and Inception-ResNet-v2 to perform semantic segmentation of HeLa cells defined by four different classes - nuclear envelope, nucleus, rest of the cell, and background.

7.4   Active contours

In order to compare the algorithm segmentation with an alternative approach, the NE was segmented with Chan-Vese active contours methodology [210, 211]. The function changes its parameters based on one of three states: Shrink, Grow, or Normal. One of the three states and its parameters were chosen empirically through numerous tests. A small circle was placed in the middle of the HeLa nucleus in one of the 300 images (Red box in Figure 3.9b) with the help of MATLAB® *roipoly* function and allowed to grow until it segmented the nucleus with the highest JI. The active contours were run once, with a set of parameters, then the Jaccard index was calculated. The parameters were adjusted (contraction bias $= -0.4$, smooth factor $= 1.5$, iterations $= 5000$) and the active contours was run again to obtain foreground and background. The number of iterations were increased in steps of 100 with the objective of determining the optimum number, i.e. to obtain the highest JI, and to continue until the algorithm had segmented not just the NE but the whole cell. Practically this consisted of running the algorithm between 0 and 5,000 iterations.

Shrink and Normal, the other two states of active contours, were also implemented but the best result was obtained by the state of Grow.

Active contours, on the other hand, provided much lower values of JI with a highest value of 75% (illustrated on slice 118/300-Figures 7.9c,d), and took considerably longer as 2200 iterations required 27 minutes for one slice only. On the other hand, the algorithm described segmented each slice in ∼8 seconds and the whole cell in approximately 40 minutes, that is, the segmentation of 2 slices with active contours would take longer than the algorithm for 300 slices.
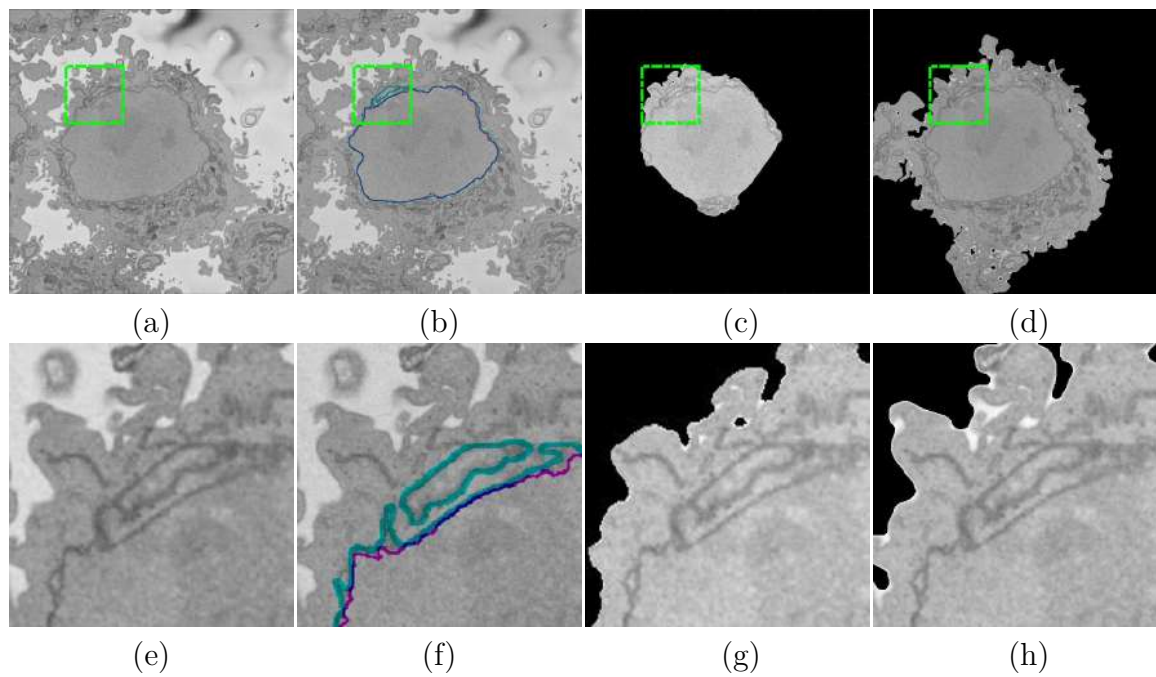
Fig. 7.9 Comparison of the algorithm against ground truth (GT) and Active Contours. (a) One representative slice from one HeLa cell (red box in Figure 3.9b). Green dashed-dotted box denotes the region of interest (ROI) that is magnified in (e-h). (b) Automated segmentation of the nuclear envelope (NE) (purple). For comparison, the hand segmented GT is shown (cyan). (c) Active contours segmentation with 2200 iterations provided the highest Jaccard index (JI) 75%. (d) Active contours segmentation with 5000 iterations included large sections of the cell showing the influence of the iterations on the result.
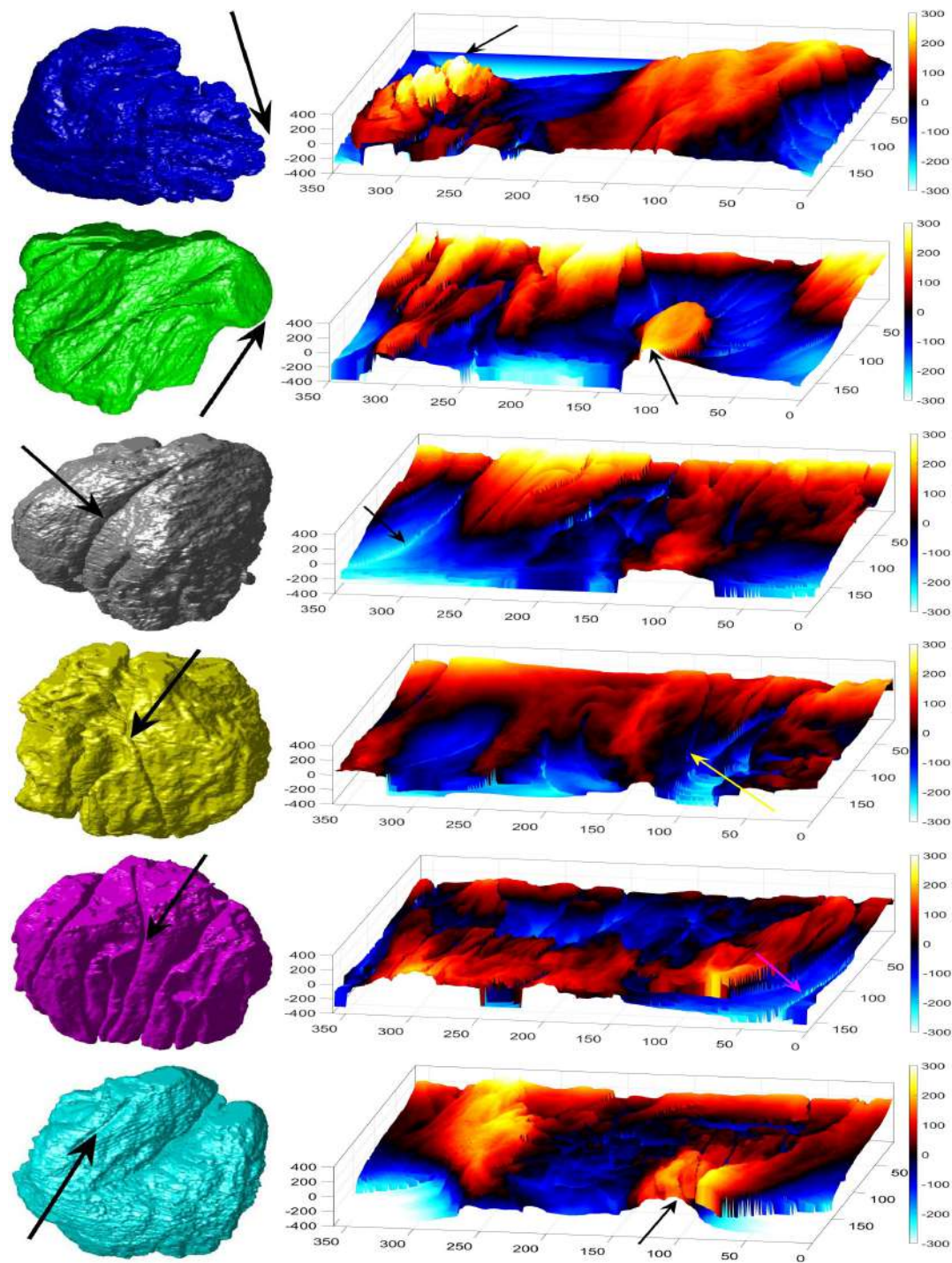
Fig. 7.10 Final results of the automated segmentation displayed as rendered volumetric surfaces and modelling for six cells. *Left* : Segmentation results displayed as rendered volumetric surfaces for six different cells. In each cell, note the notches and invaginations, which may be relevant biological characteristics of the cells. *Right*: Surfaces corresponding to the distance from the nuclear envelope (NE) to a model spheroid. Arrows, in the last four rows, show notches that travel along the nuclei (grey, yellow, purple, and cyan cell).

(a)   (b)
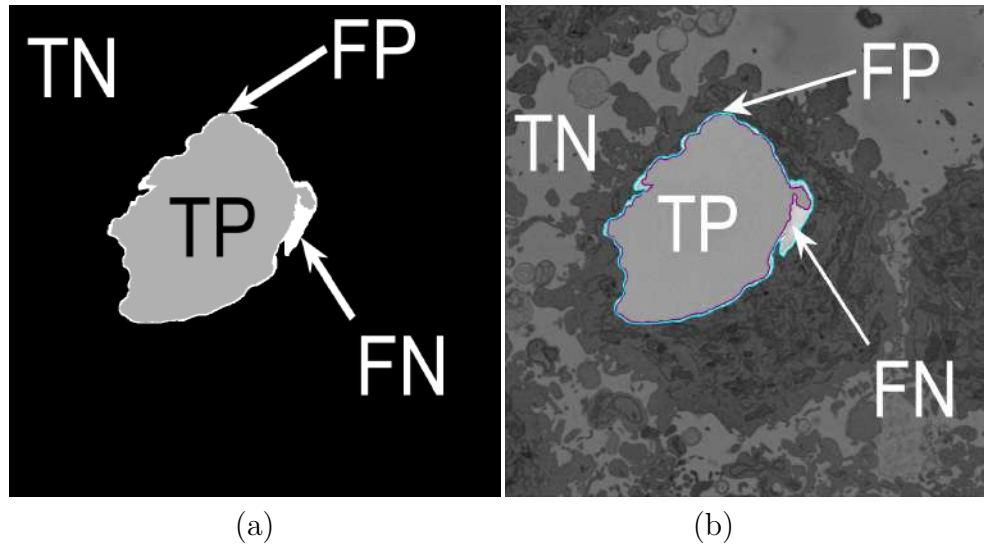
Fig. 7.11 Illustration of the pixel-based metrics. (a) True Positives (TP, nuclear pixels segmented as nucleus), true negatives (TN, background pixels segmented as background), false positives (FP, background pixels segmented as nucleus) and false negatives (FN, nuclear pixels segmented as background). These quantities were used to compute accuracy and Jaccard similarity index for the image-processing algorithm and three pre-trained deep neural networks. (b) Segmentation overlaid on the same slice as (a) (slice 184/300 - on a filtered image): ground truth (GT) in cyan, automated segmentation in purple.

7.5 Nuclear envelope shape modelling experiment result

In order to evaluate the accuracy of the proposed segmentation algorithm and deep neural networks, two different pixel-based metrics, accuracy and Jaccard similarity index (JI) [168], were calculated. All algorithms were compared with each other by using these metrics. In order to calculate JI, also known as intersection over union ,the manual segmentation for each slice was morphologically closed to generate a region rather than a line. This metric is based on the determination of the true positives (TP, nuclear pixels segmented as nucleus), true negatives (TN, background pixels segmented as background), false positives (FP, background pixels segmented as nucleus) and false negatives (FN, nuclear pixels segmented as background) as illustrated in Figure 7.11 and defined by the following equations:

The result of the segmentation of the nuclear envelope is a volumetric surface, with particular shape characteristics, such as notches or invaginations (Figure 5.10). To assess the particular geometrical characteristics of these surfaces, a model against a spheroid can be performed. The spheroid was created with the same volume as the nucleus and the position adjusted to fill the NE as closely as possible, as illustrated in Figure 7.12a, for one slice and in Figure 7.12b for the whole volume where NE is displayed as a red rendered volumetric surface and the spheroid as blue mesh. In order to position the spheroid, the centroid of the segmented cell was calculated, and the coordinates were used as the centre of the spheroid.

(a)   (b)
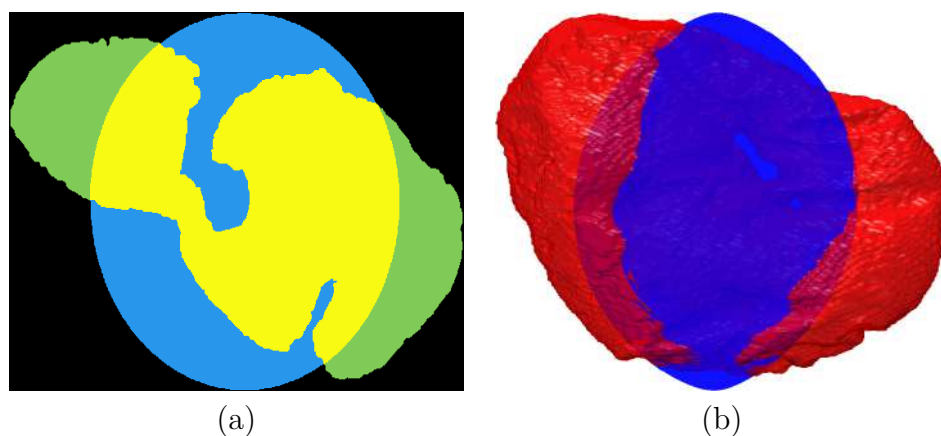
Fig. 7.12 Nuclear envelope (NE) surface modelling against a spheroid of Chlamydia trachomatis-infected HeLa cells. (a) One slice of the modelling where the spheroid is denoted in cyan and the NE in green and the overlap in yellow. (b) Rendering of the NE (red surface) against the model spheroid (blue mesh).

The surfaces of the spheroid and the nucleus were subsequently compared by tracing rays from the centre of the spheroid and the distance between the surfaces for each ray was calculated (Figure 7.13a). It was designated that when the NE was further away from the centre, the difference was positive. (Figure 7.13b) shows the surface corresponding to the distance from the NE of the first cell to a model spheroid.

Several metrics that characterise the NE can be extracted. From the volume itself, the volume of a cell is the first metric to be extracted. To measure the degree similarity between the inner volume of the spheroid and the nucleus, the Jaccard index can be calculated in the following way: the regions where the spheroid and the cell overlap
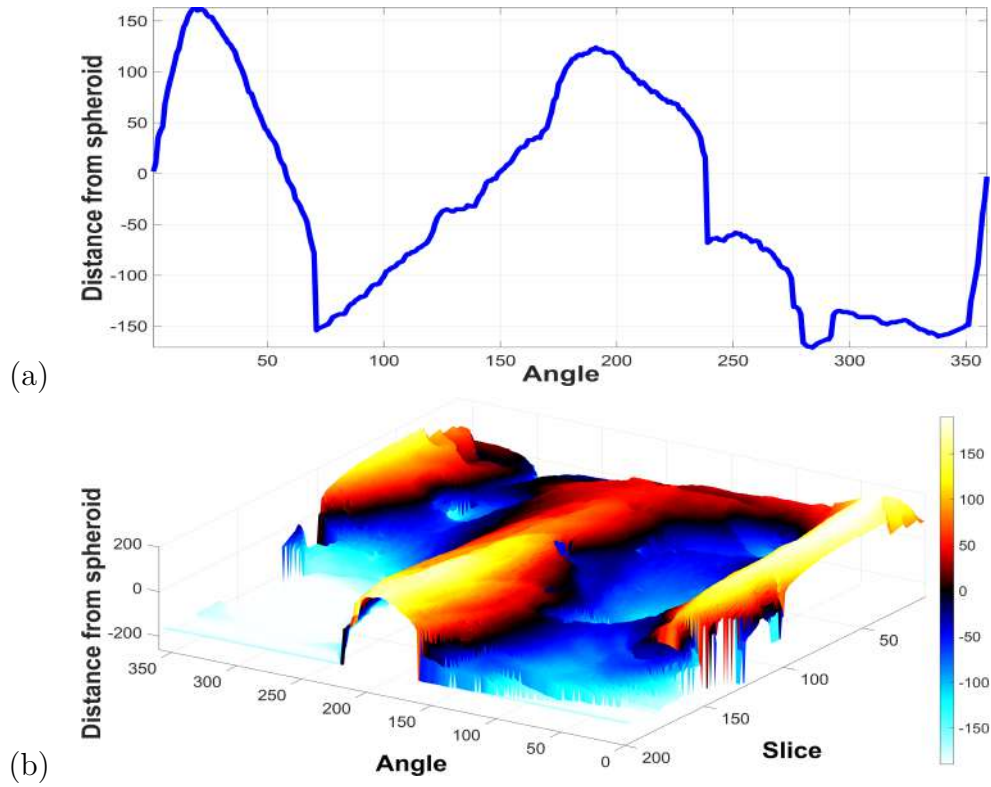
Fig. 7.13 Distances of the NE to the spheroid of Chlamydia trachomatis-infected HeLa cells. (a) Measurements obtained along the boundary of one slice of the NE. (d) Surface corresponding to the distance from the NE to a model spheroid. The surface is formed by placing the lines of each slice as shown in (a).

(yellow in Figure 7.12a) is considered as True Positive, the region of the spheroid not covered by the cell (blue) is a False Positive and the area of the cell not covered (green) is False Negative.

From the surface the following measurements can be extracted from the altitudes of the modelled surface: (1) standard deviation (σ) and (2) range of the altitude (distance of the highest peak and deepest valleys from the spheroid).

Finally, the statistical distributions of all volume and surface metrics of Wild-type HeLa cells and Chlamydia trachomatis-infected HeLa cells were combined for comparison.

Volume metrics comparison between Wild-type HeLa cells and Chlamydia trachomatis-infected HeLa cells from nuclear envelope (NE) shape modelling reveal that mean volume value of HeLa cells is much higher than that of Chlamydia trachomatis-infected

HeLa cells and this might indicate a shrink in HeLa cells when they are infected. Surface metrics comparison between Wild-type HeLa cells and Chlamydia trachomatis-infected HeLa cells from nuclear envelope (NE) shape modelling indicate that the Wild-type cells are far more rugged than the Chlamydia-infected cells.

One further analysis was performed on the cell from the CIL50051 dataset, the one with hole. Besides the clear hole, the NE has other deep crevices that nearly connect two opposite sides of the NE. This is shown in Figure7.14 with the NE rendered with different parameters (no face colour, edges in black and with transparency) and four different view points. Axis are added for reference. Although this level of invaginations and holes were found only in one cell, it is interesting to discover this as it may have significant biological meaning.

The metrics were calculated on a per-slice basis for all algorithms. JI is a strict measurement as compared with accuracy as it does not include the true negatives. On the other hand, accuracy includes the TN in both numerator and denominator and this, especially in cases where the objects of interest are small and there are large areas of background (e.g. the top and bottom slices of the cell) would render very high accuracy.

7.6   Summary

Results for all developments described in this dissertation were presented in this chapter. The chapter was subdivided in three main projects:

- Automatic segmentation by the proposed image processing algorithm,

- Deep learning architectures for semantic segmentation of HeLa cells,

- New EM data sets for testing the proposed image processing algorithm.

In this work, an automated image processing algorithm was compared with four pre-trained deep learning architectures, VGG16, ResNet18, Inception-ResNet-v2, and U-Net

(a)   (b)

(c)   (d)
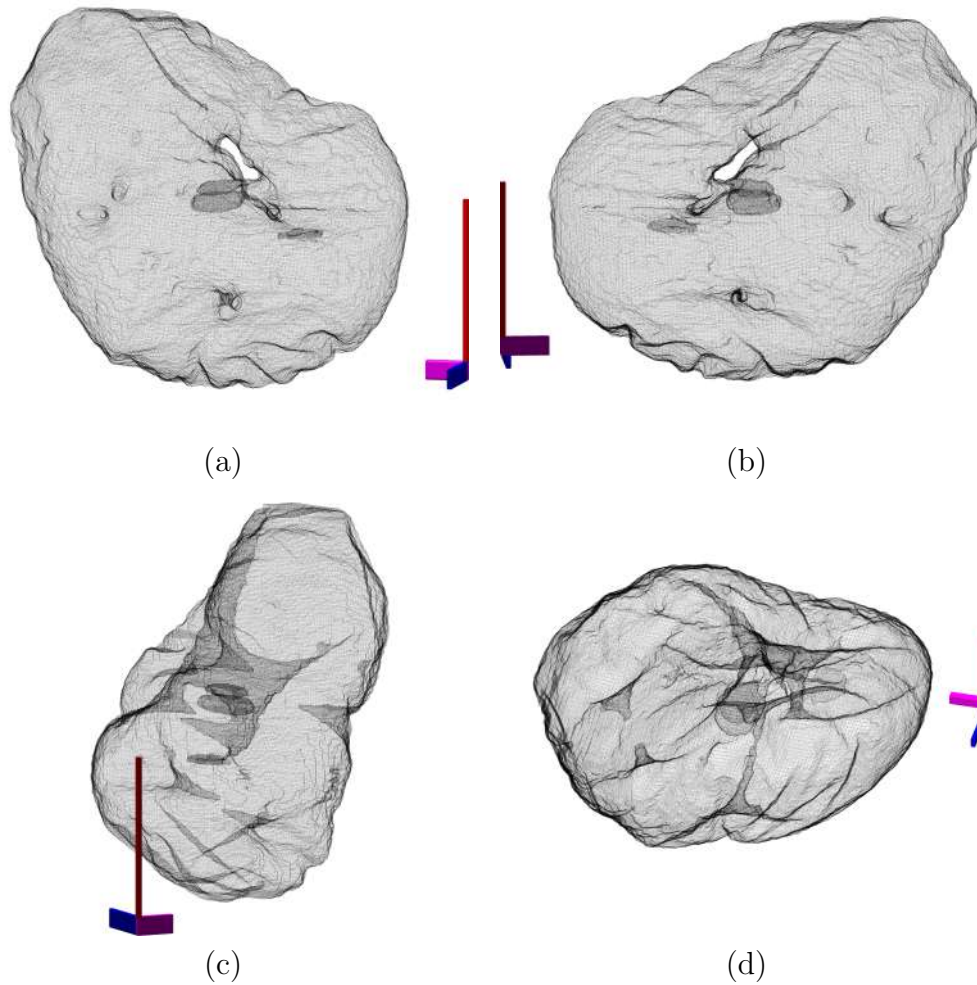
Fig. 7.14 Illustration of the cell from dataset CIL50051. The surface is displayed as a mesh with transparency to show the hole of the nuclear envelope (NE) and the crevices that go deep inside the nucleus. Notice in (c,d) how these invaginations nearly connect separate sides of the NE.

for semantic segmentation of HeLa cells imaged with EM. Segmentations results were also compared with the manual segmentations/GT and training data, in order to assess the accuracy of all the algorithms. For comparison, accuracy and JI were measured. The image-processing algorithm exploited the 3D nature of the data by using adjacent images to check for connectivity of islands to the main nuclear region. The algorithm thus began at the central slice of the cell, which was assumed to be the one in which the nuclear region would be centrally positioned and have the largest diameter. The algorithm then proceeded in both directions (up and down through the serial images)

and propagated the initial result to decide if a disjoint nuclear region was connected above or below the current slice of analysis. By using neighbouring segmentations as input parameters to the current segmentation and taking the regions into account, the segmentation was considerably improved. An example for this improvement is shown for one particular slice (slice 118/300) in (Figure 5.3d) with Jaccard similarity index and accuracy are 0.94 and 0.99, respectively. When a segmented nuclear region overlapped with the previous nuclear segmentations, it was maintained; when there was no overlap, it was discarded. finally, the NE was obtained as the boundary of the nucleus. The image-processing algorithm segments the NE of HeLa cervical cancer cells in approximately 8 seconds and a whole cell contained 300 slices in approximately 40 minutes. Except for the manual selection of the centroid to crop each cell from bigger EM images, the image processing algorithm is fully automatic.

The deep learning architectures perform semantic segmentation of the whole cell with relatively good accuracy (Table 7.1 (Top row)) but segmentation of the nuclear envelope by all three deep neural networks was outperformed by the image processing algorithm as shown in Figure 5.3d, and Table 7.1. Visually, the semantic segmentation results overlap well for classes such as nucleus, rest of the cell, and background. However, smaller objects like the nuclear envelope are not as accurate. Although the overall data set performance is quite high, the class metrics show that under represented classes such as nuclear envelope is not segmented as well as classes such as nucleus, rest of the cell, and background. Additional data that includes more samples of the under represented classes might help improve the results. The limitations of the deep learning strategies were the training data. In the case of U-Net, 36000 pairs of images and labels were used, perhaps with a larger number the results would improve. Similarly, different configuration of the network, like number of epochs, could impact on the results.
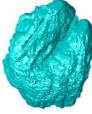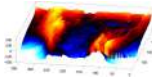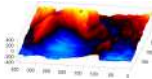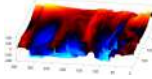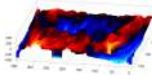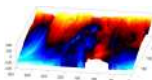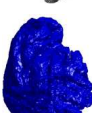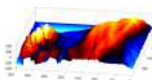
As JI does not count TN, the values decrease towards the top and bottom slices of the cells as the structure was considerably more complex and the areas become much smaller. On the other hand, accuracy includes the TN in both numerator and denominator and this, especially in cases where the objects of interest are small and

there are large areas of background (e.g. the top and bottom slices of the cell) would render very high accuracy. Therefore, in contrast to JI, accuracy increases in slices towards both top and bottom ends. As the image-processing algorithm was trained and tested only for the nuclear envelope and nucleus and as the bottom 26 and the top 40 slices did not contain any cell, the difference in Jaccard index is in favour of the deep learning architectures towards the top and bottom of the cells. Overall, the best results were obtained by the proposed image processing segmentation algorithm especially for central slices - slices between 75/300 and 225/300.

The comparison between the model spheroid and the whole nucleus (Figure 7.8b) reported a JI of 66% (Figure 5.12). In this case JI is measuring how spherical the NE is, not the accuracy of the segmentation. This value indicates a relative departure from a spheroid and it is speculated that the JI could be related to biological characteristics of cells. In addition, the measurements of distance from the nucleus to the spheroid showed rougher and smoother regions (Figures 5.12a,b,d). The surface corresponding to the distance from the nuclear envelope to a model spheroid (Figure 5.12d) showed graphically the hollow and prominent regions of the cell, but more important, elements such as a notch (solid red arrow) or ruggedness (dashed green arrow) can be an indication of NE breaking down or remodelling. An advantage of this modelling is that visually it is easier to assess a single 2D image than a 3D volumetric surface as can be seen in Figure 7.10 where six cells and their corresponding surfaces are shown. Mercator map projection was used in this modelling against a spheroid and this could be a limitation of the algorithm. Whilst many interesting characteristics such a large notch on the NE of the third (grey) cell or large protuberance on the second (green) cell, at this moment it is only possible to speculate the biological correlation between the surfaces and the nature of the cell itself. Table 7.1 illustrates the previously described metrics that can be extracted from the NE and the surface. Mean (μ), standard deviation (σ), and range of values, which are related to the height of peaks and depth of valleys could be related to some biological state of the cells, however, this has not yet been verified. A strong negative correlation between JI and σ, (correlation coefficient

$= -0.9139$) is due to the similarity between the NE and the spheroid as a NE which is closer to an spheroid (higher JI) will have a smoother surface (lower σ). This is also indicated by a weaker negative correlation between JI and range, (correlation coefficient $= -0.7388$), and a positive correlation between JI and μ with correlation coefficient (0.7142).

Table 7.1 Quantitative metrics extracted from surface modelling of a HeLa cell against a spheroid. Examples of metrics that can be automatically extracted from the nuclear envelope (NE) segmentation, including nuclear volume and Jaccard Index (JI) against the spheroid, and the mean value (μ), standard deviation (σ), range of values for the NE (distance of the highest peaks and deepest valleys from the spheroid). The last column shows the ratio of the number of pixels to the total number of pixels in one standard deviation above and below the mean value as height/depth of a peak/valley may not indicate if this is a thin spike or more of a plateau. In addition to a strong negative correlation between JI and σ, with correlation coefficient (-0.9139) which explains the similarity between the NE and the spheroid, and a weaker negative correlation between JI and range, with correlation coefficient (-0.7388), a comparatively weaker positive correlation between JI and μ with correlation coefficient (0.7142) was observed. These values could be used to draw some conclusions about biological characteristics of cells and more metrics can be extracted from the algorithm developed for this work.

Cell	Volume Metrics		Surface Metrics				
3D Shape	Volume (μm^3)	Jaccard Index	Surface modelling against a spheroid	Mean (μ)	Standard dev. (σ)	Range	Pixel ratio for $\mu\pm\sigma$
	393	0.5538		-23.424	142.47	681	16%&19%
	442	0.6610		-17.018	105.11	517	13%&18%
	454	0.6989		-11.013	96.968	553	12%&17%
	487	0.7084		-16.467	98.528	577	15%&15%
	502	0.6643		-27.290	116.11	544	13%&19%
	580	0.5991		-27.882	135.57	703	19%&18%
	600	0.5801		-29.894	163.80	894	17%&15%

Chapter 8

Conclusions and Future Work

An automated image processing algorithm for the volumetric analysis of different cervical cancer cells called HeLa cells, with special emphasis on segmentation of nuclear envelope of HeLa cells has been described. Through a visualisation technique, a graphical model of the nuclear envelope of HeLa cells, which could be related to biological characteristics of cells, has been explained. Geometric comparison between Wild-type and Chlamydia trachomatis-infected HeLa cells has been presented. Semantic segmentation comparison between the algorithm and deep learning architectures has been presented.

This chapter presents a summary of the dissertation, which highlights the key findings; then, the major contributions and conclusions; finally, limitations of the algorithms are presented, leading to suggestions for further research.

8.1   Summary

The NE of 13 Wild type and Chlamydia trachomatis-infected HeLa cells were successfully segmented, with good accuracy for the two cells for which GT was available. When the NE is modelled against a spheroid, it is possible to extract several quantitative metrics which may be related to the biological characteristics of the cells. The 2D maps of the NE surface can provide an easier way to assess the characteristics of a

3D structure. The metrics obtained from the algorithm provides a statistical and geometrical comparison between different type of HeLa cells.

The algorithm outperformed active contours as well as four pre-trained deep learning architectures in both accuracy and time. The segmentation of each cropped cell is fully automatic and unsupervised and segments one slice in approximately 8 seconds, one whole cell in approximately 40 minutes. On the other hand, the deep neural network architectures were fine-tuned and trained in 21.25 hours (VGG16), 1.5 hours (ResNet18), and 4.7 hours (Inception-ResNet-v2) to perform semantic segmentation on the whole cell with relatively good accuracy.

Deep learning methods have matched or surpassed the previous state of the art in a various segmentation applications, mainly benefiting from the increased size of public data sets, the amount of information scientists acquire is vast, and the emergence of advanced network architectures as well as powerful hardware for computing. Given this rapid process, one may wonder if deep learning methods can be directly deployed to real-world applications to reduce the workload of clinicians as an automated analysis would highlight patterns in the data not seen before. The current literature suggests that there is still a long way to go and the results obtained in this dissertation show that there is still room for traditional algorithms.

8.2   Major contributions

The original contributions of this dissertation are:

- An image processing algorithm for the segmentation of the nuclear envelope of cervical cancer cells or HeLa cells. The algorithm, specific for but not limited to the analysis of HeLa cells, could work for any electron microscopy images as shown in this dissertation (Chlamydia trachomatis-infected HeLa cells). The segmentation algorithm, which is fast and accurate compared to deep learning architectures, can be used to create a 3D volume of the nuclear envelope and biomedical researchers may benefit from it.

- A graphical model of the nuclear envelope was obtained by modelling the nuclear envelope against a spheroid, and volume and surface metrics, which could be related to biological characteristics of HeLa cells, were computed.

- Development and testing of an algorithm which is distributed as a set of routines under an open source license. The code developed for the segmentation strategies has been made available through GitHub. The open-source algorithm described in this paper provides an alternative to expensive commercial software and manual segmentation, which is still widely used despite the significant disadvantages of time and inter- and intra-user variability.

- 300 labelled images of the HeLa cell defined by four different classes - nuclear envelope, nucleus, rest of the cell, and background - were obtained, shown in (Figure 5.2). These images were used to train deep learning architectures. The four-class ground truth for 300 slices made available through Zenodo [212].

- An objective comparison, supported by accuracy and Jaccard similarity index, between the image processing algorithm and four pre-trained deep learning architectures - VGG16, ResNet18, Inception-ResNet-v2, and U-Net to perform semantic segmentation of HeLa cells.

8.3   Major conclusions

The following conclusions are considered:

- A classical and unsupervised image processing algorithm has been presented to perform semantic segmentation of cancerous HeLa cell images from SBF SEM and the segmentation results have been compared with four pre-trained deep neural network architectures. The pre-trained deep neural network architectures, VGG16, ResNet18 and Inception-ResNet-v2 were trained in ImageNet and fine-tuned for semantic segmentation of the HeLa cells. Two similarity metrics, accuracy and Jaccard index, were calculated so the image-processing algorithm

was compared with the deep learning architectures. For central slices, slices between 75/300 and 225/300, the image-processing algorithm outperformed active contours (snakes) as proved by the similarity metrics.

- The graphical model of the nuclear envelope against a spheroid has provided volume and surface metrics, which could be related to biological characteristics of HeLa cells.

- The open-source algorithm described in this paper provides an alternative to expensive commercial software and manual segmentation, which is still widely used despite the significant disadvantages of time and inter- and intra-user variability.

8.4   Limitations

The limitations presented include all algorithms: the segmentation and semantic segmentation from Chapter 4 and 5, as well as the modelling of the NE against a spheroid. These limitations may introduce possible routes moving the research forward.

- The decrease in Jaccard similarity index towards the bottom and top of the cell can be understood as the shape becomes less regular and more complex and the nucleus is formed by more than one region. Two sources of error were thus identified. First, the detected boundary did not match that of the manual observer. Further examination is required to notice the exact contribution of these errors to the total index. It should also be noticed that errors of the manual expert are possible.

- In some slices there were more than one region corresponding to the nucleus and this was not detected by the segmentation algorithm although for majority of slices that is not the case. This is still a limitation of the algorithm and further work can rely on the segmentation of contiguous slices.

- For the traditional image processing algorithm, it was assumed that there was a single HeLa cell of interest, which may be surrounded by fragments of other cells, but the centre of the cell of interest is located at centre of a three-dimensional (3D) stack of images. In addition, it was assumed that the nuclear envelope was darker than the nuclei or its surroundings and that the background was brighter than any cellular structure.

- The limitations of the deep learning strategies were the training data. In the case of U-Net, 36000 pairs of images and labels were used, perhaps with a larger number the results would improve. Similarly, different configuration of the network, like number of epochs, could impact on the results.

8.5 Suggestions for further research

The following suggestions for research do not necessarily address the problems stated in the previous section, but provide further extensions of the work.

- Future work will consider the analysis of the whole cell and segmentation of other organelles of HeLa cells such as mitochondria, Golgi apparatus, and endoplasmic reticulum (ER).

- Finding different data sets acquired by electron microscopy so the traditional image processing algorithm can be tested with it.

- The results of the deep learning approaches, for all cases, could have been improved by applying post-processing, e.g. to remove small regions of one class that were inside a large region of a different class, or by thinning or dilating the nuclear envelope. However, as the objective was to compare the image processing algorithm with the deep learning architectures, it was preferred not to post-process the latter ones. In addition, for the case of U-Net, the ground truth was restricted to a single nucleus in the volume. A ground truth with several nuclei could provide better results.

Closing statement.

Cell segmentation and classification were presented in this work as complex biological problems involving two complementary approaches to understanding it: image analysis or mathematical modelling. This work analysed the problem from the perspective of image analysis especially when it comes to segmentation.

The work presented in this dissertation lays the foundation for automated image analysis of HeLa cells. Several studies of cell segmentation are still analysed manually, which is accepted as "gold standard" despite the significant disadvantages of time and inter- and intra-user variability.

The work presented in this dissertation provides fast and accurate segmentation of cervical cancer cells called HeLa cells which would allow biomedical scientists to have consistent measurements of HeLa cells and its nuclear envelope so they could analyse the cells, thus allowing for re-producibility in studies with more complex questions. The algorithm could be extended to incorporate a more detailed analysis of HeLa cells or any other cells imaged with electron microscopy. The project presented in this dissertation follows a very specific scope, however it provides clear implications in several fields of research.

References

[1] Pravin R Lokhande, S Balaguru, G Deenadayalan, and Ratnakar R Ghorpade. A review of contemporary researches on biomedical image analysis. In *International Conference on Recent Trends in Image Processing and Pattern Recognition*, pages 84–96. Springer, 2018.

[2] J. A. Solis-Lemus. *Segmentation and shape tracking of overlapping macrophages in fluorescent microscopy images.* PhD thesis, City, University of London, October 2018.

[3] Jens Rittscher, Raghu Machiraju, and Stephen TC Wong. *Microscopic image analysis for life science applications.* Artech House, 2008.

[4] Florence Cloppet and Arnaud Boucher. Segmentation of overlapping/aggregating nuclei cells in biological images. *In Proceedings IEEE International Conference Pattern Recognition*, pages 1–4, 2008.

[5] Peter Bajcsy, Antonio Cardone, Joe Chalfoun, Michael Hlater, Derek Juba, Marcin Kociolek, Michael Majurski, Adele Peskin, Carl Simon, Mylene Simon, Antoine Vandecreme, and Mary Brady. Survey statistics of automated segmentations applied to optical imaging of mammalian cells. *BMC Bioinformatics*, 16(330), 2015.

[6] Zhenzhou Wang and Haixing Li. Generalizing cell segmentation and quantification. *BMC bioinformatics*, 18(1):189, Mar 2017. PMID: 28335722 PMCID: PMC5364575.

[7] Jose Alanso Solis-Lemus, Brian Stramer, Greg Slabaugh, and Constantino Carlos Reyes-Aldasoro. Segmentation and Shape Analysis of Macrophages Using Anglegram Analysis. *Journal of Imaging*, 4(2), 2017.

[8] Erlend Hodneland, Tanja Kogel, Dominik Michael Frei, Hans-Herman Gerdes, and Arvid Lundervold. CellSegm - a MATLAB toolbox for high-throughput 3D cell segmentation. *Source Code for Biology and Medicine*, 8(16), 2013.

[9] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener. Histopathological Image Analysis: A Review. *IEEE Reviews in Biomedical Engineering*, 2:147–171, 2009.

[10] British society for cell biology, 1965. Last accessed 13 Feb 2019.

[11] C. Karabağ, M. L. Jones, C. J. Peddie, A. E. Weston, L. M. Collinson, and C. C. Reyes-Aldasoro. Segmentation and modelling of the nuclear envelope of hela cells imaged with serial block face scanning electron microscopy. *Journal of Imaging*, 5(9):75, 2019.

[12] Cefa Karabag, Martin L Jones, Christopher J Peddie, Anne E Weston, Lucy M Collinson, and Constantino Carlos Reyes-Aldasoro. Semantic segmentation of hela cells: An objective comparison between one traditional algorithm and three deep-learning architectures. *bioRxiv*, page 978478, 2020.

[13] Thanatip Chankong, Nipon Theera-Umpon, and Sansanee Auephanwiriyakul. Automatic cervical cell segmentation and classification in Pap smears. *Computer Methods and Programs in Biomedicine*, 113:539–556, 2014.

[14] Rafael Lozano, Mohsen Naghavi, Kyle Foreman, Stephen Lim, Kenji Shibuya, Victor Aboyans, Jerry Abraham, Timothy Adair, Rakesh Aggarwal, Stephanie Y. Ahn, and et al. Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the global burden of disease study 2010. *Lancet (London, England)*, 380(9859):2095–2128, Dec 2012. PMID: 23245604.

[15] NHS, UK. https://www.nhs.uk/conditions/cervical-cancer, 1948. Last accessed 14 May 2019.

[16] J. Paul. The cancer cell in vitro: a review. *Cancer Research*, 22:431–440, May 1962.

[17] J. Leighton. Contributions of tissue culture studies to an understanding of the biology of cancer: a review. *Cancer Research*, 17(10):929–941, Nov 1957.

[18] Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A. Margolin, Sungjoon Kim, Christopher J. Wilson, Joseph Lehár, Gregory V. Kryukov, Dmitriy Sonkin, and et al. The cancer cell line encyclopedia enables predictive modeling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, Mar 2012.

[19] Constantino Carlos Reyes-Aldasoro. The proportion of cancer-related entries in PubMed has increased considerably; is cancer truly "the emperor of all maladies"? *PLOS ONE*, 12(3):e0173671, Mar 2017.

[20] Rajeshwar P. Verma and Corwin Hansch. Chemical Toxicity on HeLa Cells. *Current Medicinal Chemistry*, 13(4):423–448, 2006.

[21] John R. Masters. HeLa cells 50 years on: the good, the bad and the ugly. *Nature Reviews Cancer*, 2:315–319, 2002.

[22] Brendan P. Lucey, Walter A. Nelson-Rees, and Grover M. Hutchins. Henrietta Lacks, HeLa Cells and Cell Culture Contamination. *Archives of Pathology & Laboratory Medicine*, 133:1463–1467, 2009.

[23] Marc Lacroix. Persistent use of "false" cell lines. *International Journal Cancer*, 122:1–4, 2008.

[24] Isabella Ellinger and Adolf Ellinger. Smallest Unit of Life: Cell Biology 2. *Comparative Medicine: Anatomy and Physiology*, 9(24):19–33, 2014.

[25] Bruce Alberts, Dennis Bray, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell.* Garland Publishing Inc., 1st edition, 1998.

[26] Xiaobo Zhou, Fuhai Li, Jun Yan, and Stephen T. C. Wong. A Novel Cell Segmentation Method and Cell Phase Identification Using Markov Model. *IEEE Transactions on Information Technology in Biomedicine*, 13(2):152–157, 2009.

[27] Chanho Jung, Changick Kim, Seoung Wan Chae, and Sukjoong Oh. Unsupervised Segmentation of Overlapped Nuclei Using Bayesian Classification. *IEEE Transactions on Biomedical Engineering*, 57(12):2825–2832, 2010.

[28] A.J. Perez, M. Seyedhosseini, T.J. Deerinck, E.A. Bushong, S. Panda, T. Tasdizen, and M.H. Ellisman. A workflow for the automatic segmentation of organelles in electron microscopy image stacks. *Frontiers in Neuroanatomy*, 8(126):1–13, Nov 2014.

[29] S.A. Wilke, J.K. Antonios, E.A. Bushong, A. Badkoobehi, E. Malek, and M. Hwang. Deconstructing complexity: serial block-face electron microscopic analysis of the hippocampal mossy fiber synapse. *Journal of Neuroscience*, 33(2):507–522, Jan 2013.

[30] D.V. Bohorquez, L.A. Samsa, A. Roholt, S. Medicetty, R. Chandra, and R.A. Liddle. An enteroendocrine cell-enteric glia connection revealed by 3D electron microscopy. *PLoS ONE 9:e89881*, 9(2):1–13, Feb 2014.

[31] Luis Pedro Coelho, Aabid Shariff, and Robert F. Murphy. Nuclear Segmentation in Microscope Cell Images: A Hand-Segmented Dataset and Comparison of Algorithms. *In Proceeding IEEE International Symposium Biomedical Imaging*, pages 518–521, 2009.

[32] The Cell Image Library, 2017. Last accessed 31 Jan 2020.

[33] Domenico Ribatti. An historical note on the cell theory. *Experimental Cell Research*, 364(1):1–4, Mar 2018.

[34] Julián Candia, Ryan Maunu, Meghan Driscoll, Angélique Biancotto, Pradeep Dagur, J. Philip McCoy, H. Nida Sen, Lai Wei, Amos Maritan, Kan Cao, and et al. From cellular characteristics to disease diagnosis: Uncovering phenotypes with supercells. *PLoS Computational Biology*, 9(9):e1003215, Sep 2013.

[35] Julián Candia, Jayanth R. Banavar, and Wolfgang Losert. Understanding health and disease with multidimensional single-cell methods. *Journal of Physics. Condensed Matter: An Institute of Physics Journal*, 26(7):073102, Feb 2014.

[36] J. Zhao. Cell individuality: a basic multicellular phenomenon and its role in the pathogenesis of disease. *Medical Hypotheses*, 44(5):400–402, May 1995.

[37] J. Zhao. A liability theory of disease: the foundation of cell population pathology. *Medical Hypotheses*, 48(4):341–346, Apr 1997.

[38] S. Orrenius. Apoptosis: molecular mechanisms and implications for human disease. *Journal of Internal Medicine*, 237(6):529–536, Jun 1995.

[39] Kathryn Prame Kumar, Alyce J. Nicholls, and Connie H. Y. Wong. Partners in crime: neutrophils and monocytes/macrophages in inflammation and disease. *Cell and Tissue Research*, 371(3):551–565, 2018.

[40] Christopher J. Peddie and Lucy M. Collinson. Exploring the third dimension: Volume electron microscopy comes of age. *Micron*, 61:9–19, Jun 2014.

[41] H. Denk, W.and Horstmann. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biol.*, 2(11), e329 2004.

[42] Paola De Magistris and Wolfram Antonin. The dynamic nature of the nuclear envelope. *Current Biology*, 28(8):R487–R497, 2018.

[43] Katherine L Wilson and Jason M Berk. The nuclear envelope at a glance. *J Cell Sci*, 123(12):1973–1978, 2010.

[44] Martin W Hetzer. The nuclear envelope. *Cold Spring Harbor perspectives in biology*, 2(3):a000539, 2010.

[45] Spyros D Georgatos. The inner nuclear membrane: simple, or very complex? *The EMBO journal*, 20(12):2989–2994, 2001.

[46] Thomas Hennig and Peter O'Hare. Viruses and the nuclear envelope. *Current opinion in cell biology*, 34:113–121, 2015.

[47] Anvita Bhargava, Xavier Lahaye, and Nicolas Manel. Let me in: Control of hiv nuclear entry at the nuclear envelope. *Cytokine & growth factor reviews*, 40:59–67, 2018.

[48] Kenza Snoussi and Michael Kann. Interaction of parvoviruses with the nuclear envelope. *Advances in biological regulation*, 54:39–49, 2014.

[49] Thomas C Mettenleiter. Breaching the barrier—the nuclear envelope in virus infection. *Journal of molecular biology*, 428(10):1949–1961, 2016.

[50] Sarah Cohen, Igor Etingov, and Nelly Panté. Effect of viral infection on the nuclear envelope and nuclear pore complex. In *International review of cell and molecular biology*, volume 299, pages 117–159. Elsevier, 2012.

[51] Kyle J Roux and Brian Burke. Nuclear envelope defects in muscular dystrophy. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1772(2):118–127, 2007.

[52] Kin-Hoe Chow, Rachel E Factor, and Katharine S Ullman. The nuclear envelope environment and its cancer connections. *Nature Reviews Cancer*, 12(3):196, 2012.

[53] Ashraf N Malhas and David J Vaux. Nuclear envelope invaginations and cancer. In *Cancer Biology and the Nuclear Envelope*, pages 523–535. Springer, 2014.

[54] Sanghee Lim, Ryan J Quinton, and Neil J Ganem. Nuclear envelope rupture drives genome instability in cancer. *Molecular biology of the cell*, 27(21):3210–3213, 2016.

[55] I Jose, Dzmitry G Batrakou, and Eric C Schirmer. Cancer biology and the nuclear envelope: a convoluted relationship. In *Seminars in cancer biology*, volume 23 (2), pages 125–137. Elsevier, 2013.

[56] Maria Alvarado-Kristensson and Catalina Ana Rosselló. The biology of the nuclear envelope and its implications in cancer biology. *International journal of molecular sciences*, 20(10):2586, 2019.

[57] Christopher Vidal, Sandra Bermeo, Diane Fatkin, and Gustavo Duque. Role of the nuclear envelope in the pathogenesis of age-related bone loss and osteoporosis. *BoneKEy reports*, 1, 2012.

[58] Brian Burke, Leslie C Mounkes, and Colin L Stewart. The nuclear envelope in muscular dystrophy and cardiovascular diseases. *Traffic*, 2(10):675–683, 2001.

[59] Can Zhou, Chen Li, Bin Zhou, Huaqin Sun, Victoria Koullourou, Ian Holt, Megan J Puckelwartz, Derek T Warren, Robert Hayward, Ziyuan Lin, et al. Novel nesprin-1 mutations associated with dilated cardiomyopathy cause nuclear envelope disruption and defects in myogenesis. *Human molecular genetics*, 26(12):2258–2276, 2017.

[60] Robert G Weintraub, Christopher Semsarian, and Peter Macdonald. Dilated cardiomyopathy. *The Lancet*, 390(10092):400–414, 2017.

[61] Agnieszka Dobrzynska, Susana Gonzalo, Catherine Shanahan, and Peter Askjaer. The nuclear lamina in health and disease. *Nucleus*, 7(3):233–248, 2016.

[62] Gisele Bonne. Nuclear envelope proteins in health and diseases. In *Seminars in cell & developmental biology*, volume 29, page 93, 2014.

[63] Howard J Worman, Cecilia Östlund, and Yuexia Wang. Diseases of the nuclear envelope. *Cold Spring Harbor perspectives in biology*, 2(2):a000760, 2010.

[64] Joke Robijns, Gaëlle Houthaeve, Kevin Braeckmans, and Winnok H De Vos. Loss of nuclear envelope integrity in aging and disease. In *International review of cell and molecular biology*, volume 336, pages 205–222. Elsevier, 2018.

[65] Pierre Cau, Claire Navarro, Karim Harhouri, Patrice Roll, Sabine Sigaudy, Elise Kaspi, Sophie Perrin, Annachiara De Sandre-Giovannoli, and Nicolas Lévy. Nuclear matrix, nuclear envelope and premature aging syndromes in a translational research perspective. In *Seminars in cell & developmental biology*, volume 29, pages 125–147. Elsevier, 2014.

[66] Muhammad Saad Ahmed, Sana Ikram, Nousheen Bibi, and Asif Mir. Hutchinson–gilford progeria syndrome: a premature aging disease. *Molecular neurobiology*, 55(5):4417–4427, 2018.

[67] N. C. Putnam. Searching medline free on the internet using the national library of medicine's pubmed. *Clinical excellence for nurse practitioners: the international journal of NPACE*, 2(5):314–316, Sep 1998.

[68] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[69] Karolina Nurzynska. Deep learning as a tool for automatic segmentation of corneal endothelium images. *Symmetry*, 10(60):1–16, 2018.

[70] Chen Chen, Chen Qin, Huaqi Qiu, Giacomo Tarroni, Jinming Duan, Wenjia Bai, and Daniel Rueckert. Deep learning for cardiac image segmentation: A review. *arXiv e-prints*, page arXiv:1911.03723, Nov 2019.

[71] Y. LeCun, Y. Bengio, and G. Hinton. Deep Learning. *Nature*, 521(7553):436–444, May 2015.

[72] Yousef Al-Kofahi, Alla Zaltsman, Robert Graves, Will Marshall, and Mirabela Rusu. A deep learning-based algorithm for 2D cell segmentation in microscopy images. *BMC bioinformatics*, 19(1):1–11, 2018.

[73] Bradley J. Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L. Kline. Machine learning for medical imaging. *RadioGraphics*, 37:505–515, 2017.

[74] Huynh B.Q., Li H., and Giger M.L. Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *JAMA*, 3(3, (034501)), 2016.

[75] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105. Curran Associates Inc., 2012. Lake Tahoe, Nevada.

[76] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and TinneEditors Tuytelaars, editors, *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, page 818–833. Springer International Publishing, 2014.

[77] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556 [cs]*, Sep 2014. arXiv: 1409.1556.

[78] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[79] Hamid Laga, Yulan Guo, Hedi Tabia, Robert Fisher, and Mohammed Bennamoun. *3D Shape Analysis: Fundamentals, Theory, and Applications*. Wiley-Blackwell, United States, 2019.

[80] Christian Szegedy, A. Toshev, and D Erhan. Deep neural networks for object detection. *Advances in Neural Information*, 35:1915–1929, 2013.

[81] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014.

[82] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[83] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[84] Lawrance Carin and Michael J. Pencina. On deep learning for medical image analysis. *JAMA*, 320(11):1192–1193, 2018.

[85] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research Archive*, 15(1):1929–1958, January 2014.

[86] Maryellen . Giger. Machine learning in medical imaging. *Journal of American College of Radiology*, 15(3PB):512–520, 2018.

[87] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec 2017.

[88] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F.Editors Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9350 of *Lecture Notes in Computer Science*, page 234–241. Springer International Publishing, 2015.

[89] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[90] Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *International conference on medical image computing and computer-assisted intervention*, pages 411–418. Springer, 2013.

[91] Hidetoshi Urakubo, Torsten Bullmann, Yoshiyuki Kubota, Shigeyuki Oba, and Shin Ishii. UNI-EM: An environment for deep neural network-based automated segmentation of neuronal electron microscopic images. *bioRxiv*, page 607366, 2019.

[92] I. Oztel, G. Yolcu, I. Ersoy, T. White, and F. Bunyak. Mitochondria segmentation in electron microscopy volumes using deep convolutional neural network. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1195–1200, Nov 2017.

[93] Sven Dorkenwald, Philipp J Schubert, Marius F Killinger, Gregor Urban, Shawn Mikula, Fabian Svara, and Joergen Kornfeld. Automated synaptic connectivity inference for volume electron microscopy. *Nature methods*, 14(4):435, 2017.

[94] R. Li, D. Si, T. Zeng, S. Ji, and J. He. Deep convolutional neural networks for detecting secondary structures in protein density maps from cryo-electron microscopy. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 41–46, Dec 2016.

[95] A. Fakhry, T. Zeng, and S. Ji. Residual deconvolutional networks for brain electron microscopy image segmentation. *IEEE Transactions on Medical Imaging*, 36(2):447–456, Feb 2017.

[96] Kohki Konishi, Masafumi Mimura, Takao Nonaka, Ichiro Sase, Hideo Nishioka, and Mitsuo Suga. Practical method of cell segmentation in electron microscope image stack using deep convolutional neural network. *Microscopy*, 2019.

[97] Jing Liu, Weifu Li, Chi Xiao, Bei Hong, Qiwei Xie, and Hua Han. Automatic detection and segmentation of mitochondria from SEM images using deep neural network. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 628–631. IEEE, 2018.

[98] J. C. Caicedo, J. Roth, A. Goodman, T. Becker, K. W. Karhohs, C. McQuin, S. Singh, and A. E. Carpenter. Evaulation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images. *IEEE Reviews in Biomedical Engineering*, 2:147–171, 2018.

[99] Tran Minh Quan, David G. C. Hildebrand, and Won-Ki Jeong. Fusionnet: A deep fully residual convolutional neural network for image segmentation in connectomics. *CoRR*, abs/1612.05360(2146), 2016.

[100] Giger ML Antropova N, Huynh BQ. A deep feature fusion methodology for breast cancer diagnosis demonstrated on three imaging modality datasets. *Medical Physics*, 44(10):5162–5171, October 2017.

[101] Gulshan V., Peng L., and et al. Coram M. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2016.

[102] Jake Luo, Min Wu, Deepika Gopukumar, and Yiqing Zhao. Big data application in biomedical research and health care: A literature review. *Biomedical Informatics Insights*, 8:BII.S31559, Jan 2016.

[103] M. Viceconti, P. Hunter, and R. Hose. Big data, big knowledge: Big data for personalized healthcare. *IEEE Journal of Biomedical and Health Informatics*, 19(4):1209–1215, Jul 2015.

[104] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[105] E.A. Jones and J.A. Summerfield. The liver: Biology and pathobiology. *Raven Press, New York*, pages 683–704, 1988.

[106] Ella Marushchenko. Scientific design and animation, 2018. Last accessed 31 Dec 2019.

[107] Etch A Cell. https://www.zooniverse.org/projects/h-spiers/etch-a-cell, 2015. Last accessed: 14 May 2020.

[108] The National Institutes of Health (NIH), USA. https://osp.od.nih.gov/scientific-sharing/hela-cells-landing/, 1887. Last accessed 14 May 2019.

[109] Marcos Malumbres and Mariano Barbacid. Cell cycle, CDKs and cancer: a changing paradigm. *Nature*, 9(24):153–166, 2009.

[110] Douglas Hanahan and Robert A. Weinberg. The Hallmarks of Cancer. *Cell*, 100:57–70, 2000.

[111] S Kredel, F Oswald, K Nienhaus, K Deuschle, C Röcker, and et al. mRuby, a bright monomeric red fluorescent protein for labeling of subcellular structures. *PLoS ONE*, 4, e4391(2):1–18, June 2009.

[112] Douglas Hanahan and Robert A. Weinberg. Hallmarks of cancer: The next generation. *Cell*, 144(5):646–674, Mar 2011. PMID: 21376230.

[113] S Iyer, RM Gaikwad, V Subba-Rao, CD Woodworth, and Igor Sokolov. Atomic force microscopy detects differences in the surface brush of normal and cancerous cells. *Nature nanotechnology*, 4(6):389–393, 2009.

[114] Rebecca Skloot. *The immortal life of Henrietta Lacks*. Broadway Paperbacks, 2017.

[115] Padma Nambisan. *An Introduction to Ethical, Safety and Intellectual Property Rights Issues in Biotechnology*. Academic Press, 2017.

[116] Jonathan JM Landry, Paul Theodor Pyl, Tobias Rausch, Thomas Zichner, Manu M Tekkedil, Adrian M Stütz, Anna Jauch, Raeka S Aiyar, Gregoire Pau, Nicolas Delhomme, et al. The genomic and transcriptomic landscape of a hela cell line. *G3: Genes, Genomes, Genetics*, 3(8):1213–1224, 2013.

[117] Jørgen Fogh, William C. Wright, and James D. Loveless. Absence of hela cell contamination in 169 cell lines derived from human tumors. *JNCI: Journal of the National Cancer Institute*, 58(2):209–214, Feb 1977.

[118] D.A. Kniss and T.L. Summerfield. Discovery of hela cell contamination in HES cells: Call for cell line authentication in reproductive biology research. *Reproductive Sciences*, 21(8):1015–1019, 2014.

[119] Lei Lu, Mark S. Ladinsky, and Tomas Kirchhausen. Formation of the postmitotic nuclear envelope from extended ER cisternae precedes nuclear pore assembly. *The Journal of Cell Biology*, 194(3):425–440, Aug 2011. PMID: 21825076 PMCID: PMC3153650.

[120] Joachim Walter, Lothar Schermelleh, Marion Cremer, Satoshi Tashiro, and Thomas Cremer. Chromosome order in HeLa cells changes during mitosis and early G1, but is stably maintained during subsequent interphase stages. *The Journal of Cell Biology*, 160(5):685–697, 2003.

[121] E. Hatch and M. Hetzer. Breaching the nuclear envelope in development and disease. *J Cell Biol*, 205(2):133–141, Apr 2014.

[122] Carol V. Robinson, Andrej Sali, and Wolfgang Baumeister. The molecular sociology of the cell. *Nature*, 450:973–982, 2007.

[123] Julia Mahamid, Stefan Pfeffer, Miroslava Schaffer, Elizabeth Villa, Radostin Danev, Luis Kuhn Cuellar, Friedrich Förster, Anthony A. Hyman, Jürgen M. Plitzko, and Wolfgang Baumeister. Visualizing the molecular sociology at the hela cell nuclear periphery. *Science (New York, N.Y.)*, 351(6276):969–972, Feb 2016. PMID: 26917770.

[124] John C. Russ. *The Image Processing Handbook*. CLC Press LLC and Springer-Verlag GmbH & Co.KG, 3rd edition, 1999.

[125] F Boray Tek, Andrew G Dempster, and Izzet Kale. Computer vision for microscopy diagnosis of malaria. *Malaria journal*, 8(1):153, 2009.

[126] Constantino Carlos Reyes-Aldasoro. *Biomedical Image Analysis Recipes in MATLAB For Life Scientist and Engineers*. John Wiley & Sons, Ltd, 1st edition, 2015.

[127] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson, 4th edition, 2018.

[128] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision*. Brooks/Cole Publishing Company, 2nd edition, 1999.

[129] John L. Semmlow and Benjamin Griffel. *Biosignal and Medical Image Processing*. CRC Press, 3rd edition, 2014.

[130] Ilya Belevich, Merja Joensuu, Darshan Kumar, Helena Vihinen, and Eija Jokitalo. Microscopy image browser: A platform for segmentation and analysis of multidimensional datasets. *PLOS Biology*, 14(1):e1002340, Jan 2016.

[131] L. Ostroff and H. Zeng. Electron Microscopy at Scale. *Cell*, 162:474–475, July 2015.

[132] A. H. K. Roeder, Alexandre Cunha, M. C. Burl, and E. M. Meyerowitz. A computational image analysis glossary for biologists. *Nature Reviews Cancer*, 139:3071–3080, 2012.

[133] J. Cheng and J. C. Rajapakse. Segmentation of clustered nuclei with shape markers and marking function. *IEEE Transactions on Biomedical Engineering*, 56(3):741–748, 2009.

[134] Xiaodong Yang, Houqiang Li, and Xiaobo Zhou. Nuclei Segmentation Using Marker-Controlled Watershed, Tracking Using Mean-Shift, and Kalman Filter in Time-Lapse Microscopy. *IEEE Transactions on Circuits and Systems — I: Regular Papers*, 53(11):2405–2414, 2006.

[135] Behiye Kaya, Evgin Goceri, Aline Beker, Brad Elder, Vinay Puduvalli, Jessica Winter, Metin Gurcan, and Jose Javier Otero. Automated fluorescent miscroscopic image analysis of PTBP1 expression in glioma. *PLoS One*, 12(3):e0170991, 2017.

[136] Evgin Goceri, Behiye Goksel, James B. Elder, Vinay K. Puduvalli, and Jose J. Otero. Quantitative validation of anti-PTBP1 antibody for diagnostic neuropathology use: Image analysis approach. *Journal of Numerical Methods in Biomedical Engineering*, 33:e2862, 2017.

[137] Christopher Beckham, Mark Hall, and Eibe Frank. WekaPyScript: Classification, regression, and filter schemes for WEKA implemented in Python. *Journal of Open Research Software*, 4(1):e33, 2016.

[138] I. Luengo, M.C. Darrow, M.C. Spink, Y. Sun, W. Dai, C.Y. He, W. Chiu, T. Pridmore, A.W. Ashton, E.M.H. Duke, M. Basham, and AP French. Survos: Super-region volume segmentation workbench. *J Struct Biol*, 198(1):43–53, 2017.

[139] N.J. Kramer, M.S. Haney, D.W. Morgens, A. Jovičić, A. Couthouis, J.and Li, J. Ousey, R. Ma, G. Bieri, C.K. Tsui, Y. Shi, N.T. Hertz, M. Tessier-Lavigne, J.K. Ichida, M.C. Bassik, and A.D. Gitler. Crispr-cas9 screens in human cells and primary neurons identify modifiers of c9orf72 dipeptide repeat protein toxicity. *Nature Genetics*, 50(4):603–612, April 2018.

[140] G. Sepulveda, M. Antkowiak, I. Brust-Mascher, K. Mahe, T. Ou, N.M. Castro, L.N. Christensen, L. Cheung, X. Jiang, D. Yoon, B. Huang, and L.E. Jao. Co-translational protein targeting facilitates centrosomal recruitment of PCNT during centrosome maturation in vertebrates. *eLife*, 7:1–28, Nov 2018.

[141] M. Luckner and G. Wanner. From light microscopy to analytical scanning electron microscopy (SEM) and focused ion beam (FIB)/SEM in biology: Fixed coordinates, flat embedding, absolute references. *Microscopy and Microanalysis*, 24(5):526–544, Jul 2018.

[142] Jerald L. Schnoor. Citizen science. *Environmental Science & Technology*, 41(17):5923–5923, Sep 2007.

[143] Katherine M Henry, Luke Pase, Carlos Fernando Ramos-Lopez, Graham J Lieschke, Stephen A Renshaw, and Constantino Carlos Reyes-Aldasoro. Phagosight: an open-source matlab® package for the analysis of fluorescent neutrophil and macrophage migration in a zebrafish model. *PloS one*, 8(8):e72636, 2013.

[144] Ignacio Arganda-Carreras and Philippe Andrey. Designing image analysis pipelines in light microscopy: A rational approach. In Yolanda Markaki and HartmannEditors Harz, editors, *Light Microscopy: Methods and Protocols*, Methods in Molecular Biology, page 185–207. Springer New York, 2017.

[145] G. Mazzamuto, L. Silvestri, L. Silvestri, I. Costantini, F. Orsini, M. Roffilli, P. Frasconi, L. Sacconi, L. Sacconi, F. S. Pavone, and et al. Software tools for efficient processing of high-resolution 3D images of macroscopic brain

samples. In *Biophotonics Congress: Biomedical Optics Congress 2018 (Microscopy/Translational/Brain/OTS) (2018)*, page JTh3A.64. Optical Society of America, Apr 2018.

[146] Kota Miura and Sébastien Tosi. *Standard and Super-Resolution Bioimaging Data Analysis. Epilogue, A Framework for Bioimage Analysis*, page 269–284. Wiley-Blackwell, 2017.

[147] Veit Wiesman, Christian Held, Ralf Palmisano, and Thomas Wittenberg. Segmentation of HeLa cells in phase-contrast images with and without DAPI stained cell nuclei. *Biomedizinische Technik/Biomedical Engineering 57(SI-1 Track-B)*, 57:519–522, 2012.

[148] F. Haguenau, P. W. Hawkes, J. L. Hutchison, B. Satiat-Jeunemaitre, and G. T. Simon. Key Events in the History of Electron Microscopy. *Microscopy and Microanalysis*, 9(4):96–138, 2003.

[149] Barry R. Masters. History of the Electron Microscope in Cell Biology. *Encyclopedia of Life Sciences, John Wiley & Sons, Ltd*, 2009.

[150] https://www.sciencelearn.org.nz/images/526-resolving-power-of-microscopes, 2018. Last accessed: 11 May 2020.

[151] Ulrich Kubitscheck and Reiner Peters. *Fluorescence microscopy.* Wiley Online Library, 2013.

[152] Michael J Sanderson, Ian Smith, Ian Parker, and Martin D Bootman. Fluorescence microscopy. *Cold Spring Harbor Protocols*, 2014(10):pdb–top071795, 2014.

[153] David Bernard Williams and C Barry Carter. *Transmission Electron Microscopy: Spectrometry. IV.* Plenum Press, 1996.

[154] Peter Bernhard Hirsch. *Electron Microscopy of Thin Crystals: By PB Hirsch et al.* Butterworth, 1965.

[155] K. J. Hayworth, J. L. Morgan, R. Schalek, D. R. Berger, D. G. C. Hildebrand, and Lichtman J. W. Imaging ATUM ultrathin section libraries with WaferMapper: a multi-scale approach to EM reconstruction of neural circuits. *Frontiers in Neural Circuits*, 8, Article 68:1–18, June 2014.

[156] E. M. Slayter and H. S. Slayter. *Light and Electron Microscopy.* Cambridge University Press, 1st edition, 1992.

[157] A. Weston, H. Armer, and L. Collinson. Towards native-state imaging in biological context in the electron microscope. *J Chem Biol*, 2009.

[158] Matthew R. G. Russell, Thomas R. Lerner, Jemima J. Burden, David O. Nkwe, Annegret Pelchen-Matthews, Marie-Charlotte Domart, Joanne Durgan, Anne Weston, Martin L. Jones, Christopher J. Peddie, Raffaella Carzaniga, Oliver Florey, Mark Marsh, Maximiliano G. Gutierrez, , and Lucy M. Collinson. 3D correlative light and electron microscopy of cultured cells using serial blockface scanning electron microscopy. *Journal of Cell Science*, 130(1):278–291, 2017.

[159] Kevin L Briggman and Davi D Bock. Volume electron microscopy for neuronal circuit reconstruction. *Current Opinion in Neurobiology*, 22(1):154–161, Feb 2012.

[160] Kevin L Briggman and Winfried Denk. Towards neural circuit reconstruction with volume electron microscopy techniques. *Current Opinion in Neurobiology*, 16(5):562–570, Oct 2006.

[161] Joseph I Goldstein, Dale E Newbury, Joseph R Michael, Nicholas WM Ritchie, John Henry J Scott, and David C Joy. *Scanning electron microscopy and X-ray microanalysis.* Springer, 2017.

[162] M. Adrian, J. Dubochet, J. Lepault, and A.W. McDowall. Cryo-electron microscopy of viruses. *Nature*, 308:32–36, 1984.

[163] Eva Nogales. Profile of Joachim Frank, Richard Henderson, and Jacques Dubochet, 2017 Nobel Laureates in Chemistry. *PNAS*, 115(3):441–444, January 2018.

[164] Joseph P Hornak. *Encyclopedia of imaging science and technology.* Wiley New York, 2002.

[165] T. J. Deerinck, E. Bushong, A. Thor, and M. H. Ellisman. NCMIR - National Center for Microscopy and Imaging Research. NCMIR methods for 3D EM: A new protocol for preparation of biological specimens for serial block-face SEM microscopy, 2010. Accessed: 2018-02-09.

[166] Jennifer K Lee, Germán A Enciso, Daniela Boassa, Christopher N Chander, Tracy H Lou, Sean S Pairawan, Melody C Guo, Frederic YM Wan, Mark H Ellisman, Christine Sütterlin, et al. Replication-dependent size reduction precedes differentiation in chlamydia trachomatis. *Nature communications*, 9(1):1–9, 2018.

[167] B. P. Lathi. *Linear Systems and Signals.* Oxford University Press Inc., 2nd edition, 2010.

[168] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[169] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[170] Trygve Randen and John Håkon Husøy. Filtering for Texture Classification: A Comparative Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999.

[171] I.S. Kim and W. McLean. Computing the Hausdorff distance between two sets of parametric curves. *Commun. Korean Math. Soc.*, 28(4):833–850, 2013.

[172] A. A Dubuisson, M.P.; Jain. A modified hausdorff distance for object matching. *In Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 1(60):566–568, 1994.

[173] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5 (1-2)(1–2):75–91, 1995.

[174] A. Maheshwari and J. Yi. On Computing Fréchet Distance of Two Paths on a Convex Polyhedron. *In Proceedings 21st European Workshop on Computational Geometry*, pages 41–44, 2005.

[175] E. W. Chambers, de Verdière É. C., J. Erickson, S. Lazard, F. Lazarus, and Thite S. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry: Theory and Applications*, 43:295–311, 2010.

[176] K. Cetina, J.M. Buenaposada, and L. Baumela. Multi-class segmentation of neuronal structures in electron microscopy images. *BMC Bioinformatics*, 19(298):1–13, 2018.

[177] Rocchini. A schematic of HD. Components of the calculation of the Hausdorff distance between the green line X and the blue line Y., 2007. Last accessed 14 March 2019.

[178] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[179] Cefa Karabağ, Martin L. Jones, Christopher J. Peddie, Anne E. Weston, Lucy M. Collinson, and Constantino Carlos Reyes-Aldasoro. Automated Segmentation of HeLa Nuclear Envelope from Electron Microscopy Images. In Mark Nixon, Sasan Mahmoodi, and ReyerEditors Zwiggelaar, editors, *Medical Image Understanding and Analysis*, Communications in Computer and Information Science, page 241–250. Springer International Publishing, 2018.

[180] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[181] Aram Ter-Sarkisov, Robert Ross, John Kelleher, Bernadette Earley, and Michael Keane. Beef cattle instance segmentation using fully convolutional neural network. *arXiv preprint arXiv:1807.01972*, 2018.

[182] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[183] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.

[184] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[185] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.

[186] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Unsupervised feature learning by augmenting single images. *arXiv preprint arXiv:1312.5242*, 2013.

[187] http://cs231n.stanford.edu/, 2020. Last accessed: 11 May 2020.

[188] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *arXiv preprint arXiv:2001.05566*, 2020.

[189] C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, A Rabinovich, et al. Going deeper with convolutions. arxiv 2014. *arXiv preprint arXiv:1409.4842*, 1409, 2014.

[190] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159, 2011.

[191] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[192] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[193] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[194] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, Apr 2017.

[195] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, and et al. U-net: Deep learning for cell counting, detection, and morphometry. *Nature Methods*, 16(1):67, Jan 2019.

[196] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In María Valdés Hernández and Víctor González-Castro, editors, *Medical Image Understanding and Analysis*, volume 723 of *Communications in Computer and Information Science*, page 506–517. Springer International Publishing, 2017.

[197] Cefa Karabağ, Jo Verhoeven, Naomi Rachel Miller, and Constantino Carlos Reyes-Aldasoro. Texture segmentation: An objective comparison between five traditional algorithms and a deep-learning U-Net architecture. *Applied Sciences*, 9(18):3900, 2019.

[198] Stanford University Li Fei-Fei et al. Imagenet, 2015. Last accessed 13 Nov 2019.

[199] Aram Ter-Sarkisov. Network of steel: Neural font style transfer from heavy metal to corporate logos. *arXiv preprint arXiv:2001.03659*, 2020.

[200] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[201] Long D Nguyen, Dongyun Lin, Zhiping Lin, and Jiuwen Cao. Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.

[202] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[203] Yunjun Wang, Qing Guan, Iweng Lao, Li Wang, Yi Wu, Duanshu Li, Qinghai Ji, Yu Wang, Yongxue Zhu, Hongtao Lu, and Jun Xiang. Using deep convolutional neural networks for multi-classification of thyroid tumor by histopathology: a large-scale pilot study. *Annals of Translational Medicine*, 7(18), 2019.

[204] T. Agarwal and H. Mittal. Performance comparison of deep neural networks on image datasets. In *2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India*, pages 1–6, Aug 2019.

[205] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018.

[206] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

[207] https://www.mathworks.com/help/deeplearning/ug/ pretrained-convolutional-neural-networks.html, 2020. Last accessed: 11 May 2020.

[208] Cefa Karabağ, Martin L. Jones, Christopher J. Peddie, Anne E. Weston, Lucy M. Collinson, and Constantino Carlos Reyes-Aldasoro. Segmentation and Modelling of HeLa Nuclear Envelope. In Maria J. Ledesma Carbayo and Miguel A. Gonzalez Ballester, editors, *IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE and IEEE Xplore, 2019.

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

[210] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, January 2001.

[211] David Nam, Judith Mantell, David Bull, Paul Verkade, and Alin Achim. A novel framework for segmentation of secretory granules in electron micrographs. *Medical Image Analysis*, 18(1):411–424, 2014.

[212] Zenodo. https://zenodo.org/record/3834609, 2013. Last accessed 23 June 2020.

[213] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[214] Timo Ojala, Matti Pietikäinen, and David Harwood. A Comparative Study of Texture Measures with Classification based on Feature Distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[215] T Ojala, K Valkealahti, E Oja, and M Pietikäinen. Texture Discrimination with Multidimensional Distributions of Signed Gray Level Differences. *Pattern Recognition*, 34(3):727–739, 2001.

[216] Norberto Malpica, Juan E. Ortuño, and Andrés Santos. A Multichannel Watershed-based Algorithm for Supervised Texture Segmentation. *Pattern Recognition Letters*, 24(9):1545–1554, 2003.

[217] C. C. Reyes-Aldasoro and A. Bhalerao. The bhattacharyya space for feature selection and its application to texture segmentation. *Pattern Recogn.*, 39(5):812–826, May 2006.

[218] Josef Bigun. Multidimensional Orientation Estimation with Applications to Texture Analysis and Optical Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.

[219] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel Texture Analysis Using Localized Spatial Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55 – 73, 1990.

[220] George R. Cross and Anil K. Jain. Markov Random Field Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39, 1983.

[221] Robert M. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

[222] Hideyuki Tamura, Shunki Mori, and Takahashi Yamawaki. Texture Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–473, 1978.

[223] Mihran Tuceryan and Anil K. Jain. Texture Analysis. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 207–248. World Scientific Publishing, 1998.

[224] Charles Bouman and Bede Liu. Multiple Resolution Segmentation of Textured Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):99–113, 1991.

[225] A. K. Jain and F. Farrokhnia. Unsupervised Texture Segmentation using Gabor Filters. *Pattern Recognition*, 24(12):1167– 1186, 1991.

[226] Alexander Kadyrov, A Talepbour, and Maria Petrou. Texture Classification with Thousand of Features. In *British Machine Vision Conference (BMVC)*, pages 656–665, Cardiff, UK, 2-5 September 2002.

[227] Charles Kervrann and Fabrice Heitz. A Markov Random Field Model-based Approach to Unsupervised Texture Segmentation using Local and Global Spatial Statistics. *IEEE Transactions on Image Processing*, 4(6):856–862, 1995.

[228] Michael Unser. Texture Classification and Segmentation Using Wavelet Frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, 1995.

[229] J.S. Weszka, C.R. Dyer, and A. Rosenfeld. A Comparative Study of Texture Measures for Terrain Classification. *IEEE Transactions Systems, Man and Cybernetics*, 6(4):269–285, April 1976.

[230] C.W. Tai and K.Z. Baba-Kishi. Microtexture Studies of PST and PZT Ceramics and PZT Thin Film by Electron Backscatter Diffraction Patterns. *Textures and Microstructures*, 35(2):71–86, 2002.

[231] Alexis Carrillat, T. Randen, L. Sønneland, and G. Elvebakk. Seismic Stratigraphic Mapping of Carbonate Mounds using 3D Texture Attributes. In *Extended Abstracts, Annual Meeting, European Association of Geoscientists and Engineers*, Florence, Italy, May 27-30 2002.

[232] Trygve Randen, Erik Monsen, Arve Abrahamsen, Jan Ove Hansen, Jürgen Schlaf, and Lars Sønneland. Three-dimensional Texture Attributes for Seismic Data Analysis. In *Ann. Int. Mtg., Soc. Expl. Geophys., Exp. Abstr.*, Calgary, Canada, August 2000.

[233] Anette K Thybo and Magni Martens. Analysis of sensory assessors in texture profiling of potatoes by multivariate modelling. *Food Quality and Preference*, 11(4):283–288, Jul 2000.

[234] J Létal, D Jirák, L Šuderlová, and M Hájek. MRI 'texture' analysis of MR images of apples during ripening and storage. *LWT - Food Science and Technology*, 36(7):719–727, Nov 2003.

[235] Rocio A Lizarraga-Morales, Raul E Sanchez-Yanez, and Roberto Baeza-Serrato. Defect detection on patterned fabrics using texture periodicity and the coordinated clusters representation. *Textile Research Journal*, 87(15):1869–1882, 2017.

[236] Francesco Bianconi, Elena González, Antonio Fernández, and Stefano A. Saetta. Automatic classification of granite tiles through colour and texture features. *Expert Systems with Applications*, 39(12):11212–11218, Sep 2012.

[237] V. A. Kovalev, M. Petrou, and Y. S. Bondar. Texture Anisotropy in 3D Images. *IEEE Transactions on Image Processing*, 8(3):346–360, 1999.

[238] C. C. Reyes-Aldasoro and Abhir Bhalerao. Volumetric Texture Description and Discriminant Feature Selection for MRI. In C. Taylor and A. Noble, editors, *Proceedings of Information Processing in Medical Imaging*, pages 282–293, Ambleside, UK, July 2003.

[239] R.A. Lerski, K Straughan, L. R. Schad, D. Boyce, S. Bluml, and I. Zuna. MR Image Texture Analysis - An Approach to tissue Characterization. *Magnetic Resonance Imaging*, 11(6):873–887, 1993.

[240] L R Schad, S Bluml, and I Zuna. MR Tissue Characterization of Intracranial Tumors by means of Texture Analysis. *Magnetic Resonance Imaging*, 11(6):889–896, 1993.

[241] C. C. Reyes Aldasoro and A. Bhalerao. Volumetric texture segmentation by discriminant feature selection and multiresolution classification. *IEEE Transactions on Medical Imaging*, 26(1):1–14, Jan 2007.

[242] Yiqiang Zhan and Dinggang Shen. Automated Segmentation of 3D US Prostate Images Using Statistical Texture-Based Matching Method. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 688–696, Canada, November 16-18 2003.

[243] Jun Xie, Yifeng Jiang, and Hung tat Tsui. Segmentation of kidney from ultrasound images based on texture and shape priors. *IEEE Transactions on Medical Imaging*, 24(1):45–57, Jan 2005.

[244] Eric A. Hoffman, Joseph M. Reinhardt, Milan Sonka, Brett A. Simon, Junfeng Guo, Osama Saba, Deokiee Chon, Shaher Samrah, Hidenori Shikata, Juerg Tschirren, Kalman Palagyi, Kenneth C. Beck, and Geoffrey McLennan. Characterization of the Interstitial Lung Diseases via Density-Based and Texture-Based Analysis of Computed Tomography Images of Lung Structure and Function. *Academic Radiology*, 10(10):1104–1118, 2003.

[245] Manuel Segovia-Martínez, Maria Petrou, Vassili A. Kovalev, and Petra Perner. Quantifying Level of Brain Atrophy Using Texture Anisotropy in CT Data. In *Medical Imaging Understanding and Analysis*, pages 173–176, Oxford, UK, July 1999.

[246] Balaji Ganeshan, Vicky Goh, Henry C. Mandeville, Quan Sing Ng, Peter J. Hoskin, and Kenneth A. Miles. Non–small cell lung cancer: Histopathologic correlates for texture parameters at ct. *Radiology*, 266(1):326–336, Jan 2013.

[247] Daniela Mayumi Ushizima Sabino, Luciano da Fontoura Costa, Edgar Gil Rizzatti, and Marco Antonio Zago. A texture approach to leukocyte recognition. *Real-Time Imaging*, 10(4):205–216, Aug 2004.

[248] X. Wang, W. He, D. Metaxas, R. Mathew, and E. White. Cell segmentation and tracking using texture-adaptive snakes. In *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, page 101–104, Apr 2007.

[249] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M. Melchers, Lothar R. Schad, Timo Gaiser, Alexander Marx, and F.G. Zollner. Multi-class texture analysis in colorectal cancer histology. *Scientific Reports*, 6:27988, Jun 2016.

[250] D. Dunn, W.E. Higgins, and J. Wakeley. Texture Segmentation using 2-D Gabor Elementary Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149, February 1994.

[251] J. Bigun and J. M. H. du Buf. N-Folded Symmetries by Complex Moments in Gabor Space and Their Application to Unsupervised Texture Segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 16(1):80–87, 1994.

[252] Francesco Bianconi and Antonio Fernández. Evaluation of the effects of gabor filter parameters on texture classification. *Pattern Recognition*, 40(12):3325 – 3335, 2007.

[253] N. M. Rajpoot. Texture Classification Using Discriminant Wavelet Packet Subbands. In *Proceedings 45th IEEE Midwest Symposium on Circuits and Systems (MWSCAS 2002)*, Tulsa, OK, USA, August 2002.

[254] Tianhorng Chang and C. C. Jay Kuo. Texture Analysis and Classification with Tree-Structured Wavelet Transform. *IEEE Transactions on Image Processing*, 2(4):429–441, 1993.

[255] R. Chellapa and A. Jain. *Markov Random Fields*. Academic Press, Boston, 1993.

[256] Vincent Andrearczyk and Paul F Whelan. Deep learning in texture analysis and its application to tissue image classification. In *Biomedical Texture Analysis*, pages 95–129. Elsevier, 2017.

[257] Z. Zhang, Q. Liu, and Y. Wang. Road extraction by deep residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, May 2018.

[258] Andreas Jansson, Eric J. Humphrey, Nicola Montecchio, Rachel M. Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. In *18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017.

[259] Trygve Randen and John Håkon Husøy. Texture Segmentation using Filters with Optimized Energy Separation. *IEEE Transactions Image Processing*, 8:571–582, 1999.

[260] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, U.S.A, 1996.

[261] R. Yamada, H. Ide, N. Yudistira, and T. Kurita. Texture segmentation using siamese network and hierarchical region merging. In *2018 24th International Conference on Pattern Recognition (ICPR)*, page 2735–2740, Aug 2018.

[262] M. Petrou and P. Garcia-Sevilla. *Image Processing: Dealing with Texture*. John Wiley & Sons, 2006.

[263] C. C. Reyes-Aldasoro and A. H. Bhalerao. Volumetric texture analysis in biomedical imaging. In M. Pereira and M.Editors Freire, editors, *Biomedical Diagnostics and Clinical Technologies: Applying High-Performance Cluster and Grid Computing*, page 200–248. IGI Global, 2011.

[264] Majid Mirmehdi, Xianghua Xie, and Jasjit Suri. *Handbook of Texture Analysis*. Imperial College Press, 2009.

[265] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, Jun 1991.

[266] Dennis Gabor. Theory of Communication. *Journal of the IEE*, 93(26):429–457, 1946.

[267] H. Knutsson and G. H. Granlund. Texture Analysis Using Two-Dimensional Quadrature Filters. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management - CAPAIDM*, pages 206–213, Pasadena, October 1983.

[268] Trygve Randen and John Håkon Husøy. Multichannel filtering for image texture segmentation. *Optical Engineering*, 33(8):2617–2625, 1994.

[269] Jo Verhoeven, Naomi Rachel Miller, Luc Daems, and Constantino Carlos Reyes-Aldasoro. Visualisation and analysis of speech production with electropalatography. *Journal of Imaging*, 5(3):40, Mar 2019.

[270] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[271] J. M. S. Prewitt. Object enhancement and extraction. *Picture Processing and Psychopictorics*, pages 75–149, 1970.

[272] I.E. Sobel. *Camera Models and Machine Perception*. PhD thesis, Stanford University, Palo Alyto CA, 1970.

[273] O. Kafri and Keren E. Encryption of pictures and shapes by random grids. *OPTICS LETTERS*, 12(6):377–379, 1987.

[274] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels. Technical Report. *EPFL*, 149300, June 2010.

[275] R. Achanta and et al. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Trans Pattern Anal. Mach. Intell.*, 34(11):2274–2281, June 2012.

[276] Farah Sinada, Saumya Reni, Izzet Kale, Peter Chiodini, and Jaya Shrivastava. A reflective analysis of image processing operations on kato-katz images for the pathological diagnosis of neglected tropical diseases. In *2019 E-Health and Bioengineering Conference (EHB)*, pages 1–4. IEEE, 2019.

[277] A. Kass, M.and Witkin and D. Terzopoulos. Snakes: active contour models. *Int.J. Comput Vision 1*, 4:321–331, 1988.

[278] Mark Nixon and Alberto Aguado. *Feature extraction and image processing for computer vision.* Academic press, 2019.

[279] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[280] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003.

[281] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[282] Raja Ebsim, Jawad Naqvi, and Timothy F. Cootes. Automatic detection of wrist fractures from posteroanterior and lateral radiographs: A deep learning-based approach. In Guoyan Zheng, Tomaž Vrtovec, Jianhua Yao, and Jose M. Pozo, editors, *Computational Methods and Clinical Applications in Musculoskeletal Imaging - 6th International Workshop, MSKI 2018, Held in Conjunction with MICCAI 2018, Revised Selected Papers*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 114–125, United States, 2019. Springer Nature.

[283] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Appendix A

Texture segmentation

As future work could include textures in semantic segmentation of HeLa cells and as U-Net has been used as one of the alternatives to the image processing algorithm, in this appendix, a different application of U-net is presented. This work has been published as shown in the List of publications related to this work - Journal publications 3.

A comparison of a series of traditional and deep learning methodologies for the segmentation of textures is presented. Six well-known texture composites first published by Randen and Husøy [170] were used to compare traditional segmentation techniques (co-occurrence [213], filtering [170], local binary patterns [214, 215], watershed [216], multiresolution sub-band filtering [217]) against a deep-learning approach based on the U-Net architecture [88]. For the latter, the effects of depth of the network, number of epochs and different optimisation algorithms were investigated.

A.1   Introduction

Texture, and more specifically textural characteristics in images, has been widely studied in the past decades as texture is one of the most important features present in images and can be used for feature extraction [218–221, 213, 222, 223, 217] and classification and segmentation [224–229]. The areas of study where texture is present range from crystallographic texture [230], stratigraphy [231, 232], food science of

potatoes [233] or apples [234], patterned fabrics [235] to natural stone industry [236]. In medical imaging, there is a large volume of research which exploits the use of texture for different purposes like segmentation of classification in most acquisition modalities like magnetic resonance imaging (MRI) [237–241], ultrasound [242, 243], computed tomography (CT) [244–246], microscopy [247, 248] and histology [249]. There are numerous approaches to texture: Haralick's co-occurrence matrix [213, 221] on the spatial domain, Gabor filters [250–252] and ordered pyramids [217] on the spectral domain, wavelets [253, 254] or Markov random fields [220, 255].

In recent years, advances in artificial intelligence have been revolutionised image processing tasks. Several deep learning approaches [75–77] have achieved outstanding results in difficult tasks such as those of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [78]. Convolutional Neural Networks (CNNs) are well suited to analyse textures as their repetitive patterns can be learned and identified by filter banks [256]. The U-Net architecture proposed by Ronneberger [88] has become a very widely used tool for segmentation and analysis reaching thousands of citations in few years since it was published. U-Nets have been used widely, for instance, road extraction [257], singing voice separation [258], automatic brain tumour detection and segmentation [196] and cell counting, detection, and morphometry [195]. The success of these deep learning approaches in very different areas invite for its application on texture analysis.

In this work, a U-Net architecture for the segmentation of textures is implemented and objectively compared against several popular traditional segmentation strategies. The traditional algorithms (co-occurrence matrices) [213], watershed [216], local binary patterns (LBP) [214, 215], filtering [170] and multiresolution sub-band filtering (MSBF) [217] were selected as these have been previously published using the texture composites proposed by Randen [259] and thus an objective numerical comparison is possible.

To perform an objective comparison, six well-known texture composites from the Brodatz [260] album, first published by Randen and Husøy [170], are segmented with U-Nets of different configurations and parameters and the results compared against

previously published results. The effects of the configuration of the networks, namely, number of epochs, depth of the network in the number of layers, and type of optimisation algorithm are assessed.

A.2 Materials and Methods

A.2.1 Texture composite images

Six composite texture images were segmented in this work (Fig. A.1). The first five composites are images of 256×256 pixels and consist of five different textures whilst the last one is 512×512 pixels and is formed with 16 different textures. The masks with which these were formed are shown in Fig. A.2. It should be highlighted that these textures have been histogram equalised prior to the arrangement and thus they cannot be distinguished by the general intensity of each region. It is frequent that comparisons are made over textures that are not equalised (e.g. [261] Figure 3, [256] (Figure 2) and thus the segmentation is not only based on the texture but the average intensity of the regions. Furthermore, whilst some textures are easy to distinguish, there are some that are quite challenging, for instance, the difference between the central and bottom regions in Fig. A.1(c) or the top left corners of Fig. A.1(d,e).

A.2.2 Training data

The training data in [170] is provided separately and is shown in Fig. A.3 for the first five composites and in Fig. A.4 for the last case. For the purpose of training the U-Nets, the training images were tessellated into sub-regions of 32×32 pixels each.

Pairs of textures and labels were constructed simultaneously in the following way: two training images were selected. Sub-regions of each image were selected and for every pair of the sub-regions, half of each was selected and placed together so that a new 32×32 patch with both textures was created with a corresponding 32×32 patch with the classes. The patches were created with diagonal, vertical and horizontal pairs.
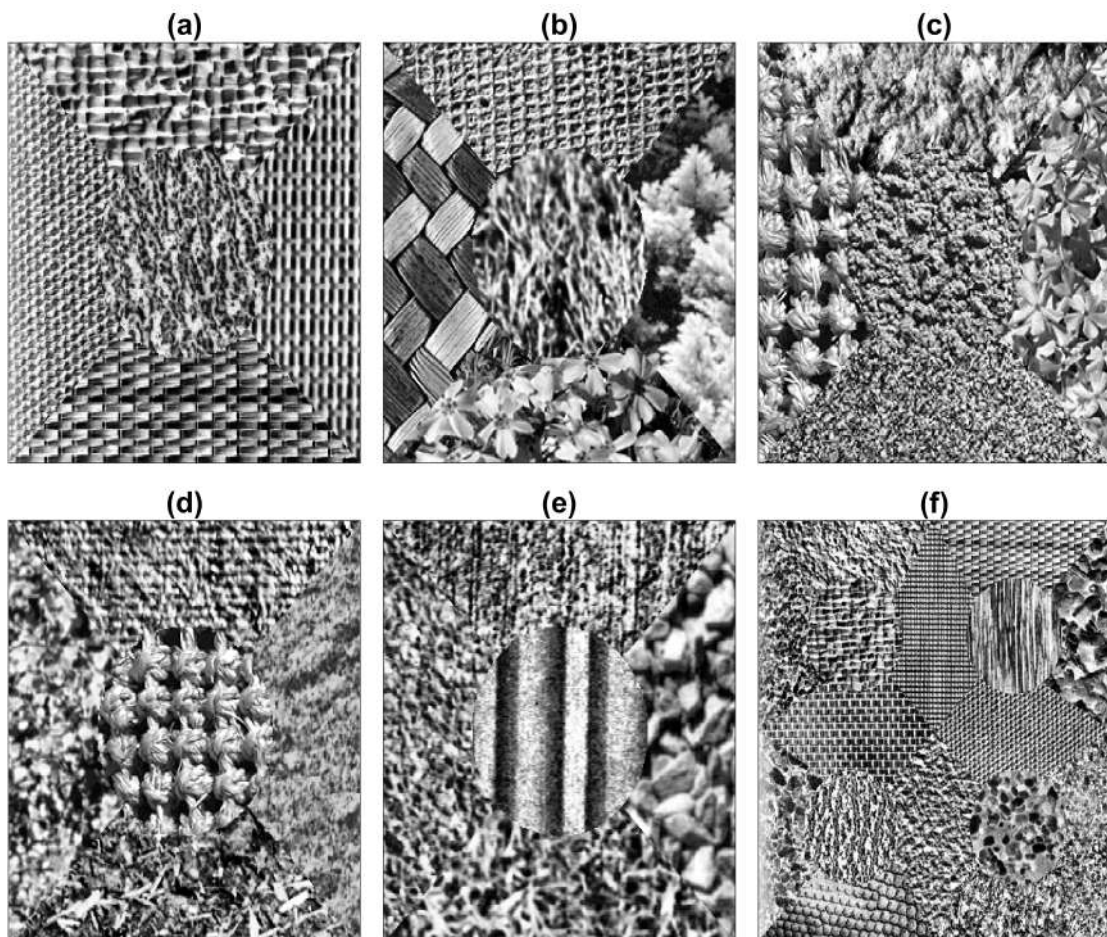
Fig. A.1 Six composite texture images. (a-e) Texture arrangements with five textures. (f) Texture arrangement with sixteen textures. Notice first, that individual textures have been histogram equalised and thus each region cannot be distinguished by the intensity, and second, some textures area easier to distinguish (e.g. (a)) than others (e.g. (d)).
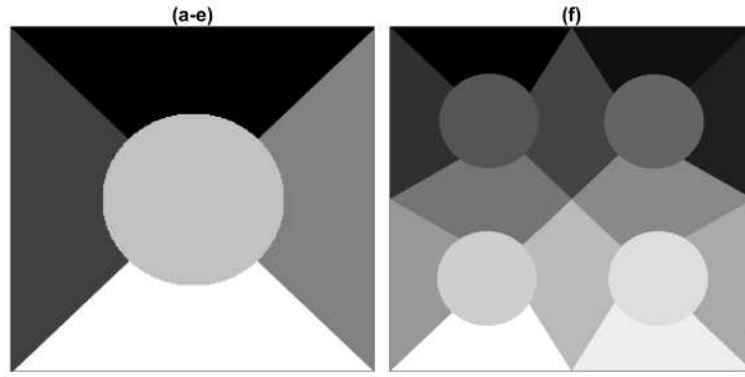
Fig. A.2 (a) Mask corresponding to texture arrangements of Figs. A.1(a-e). (b) Mask corresponding to texture arrangements of Fig. A.1(f).

The training images were traversed horizontally and vertically without overlap creating numerous training pairs. A montage of the texture pairs and labels corresponding to Fig. A.1(a) is illustrated in Fig. A.5. All pairs between classes were considered i.e. $1-2, 1-3, 1-4, 1-5, 2-1, 2-3, \ldots, 5-3, 5-4$. In total, 2940 patches were created for the five composites with five textures and 35280 were created for the composite with sixteen textures.

A.2.3 Traditional texture segmentation algorithms

We compared the results of the following texture segmentation algorithms: co-occurrence matrices [213], watershed [216], local binary patterns (LBP) [214, 215], filtering [170] and multiresolution sub-band filtering (MSBF) [217] against a U-Net architecture [88].

The traditional algorithms have been thoroughly described in the literature; however, for completeness, a short explanation of how features are extracted with each algorithm will follow. For a review of traditional texture techniques, the reader is referred to any of the following reviews [262–264].

Co-occurrence matrices are constructed from a quantised version of a grey level image so that if an image is quantised to 8 levels, the co-occurrence matrix will have 8 rows and columns. The values of each location of the matrix will depend on the number of times that a pair of grey levels jointly occur at a neighbouring distance (e.g. 1 pixel
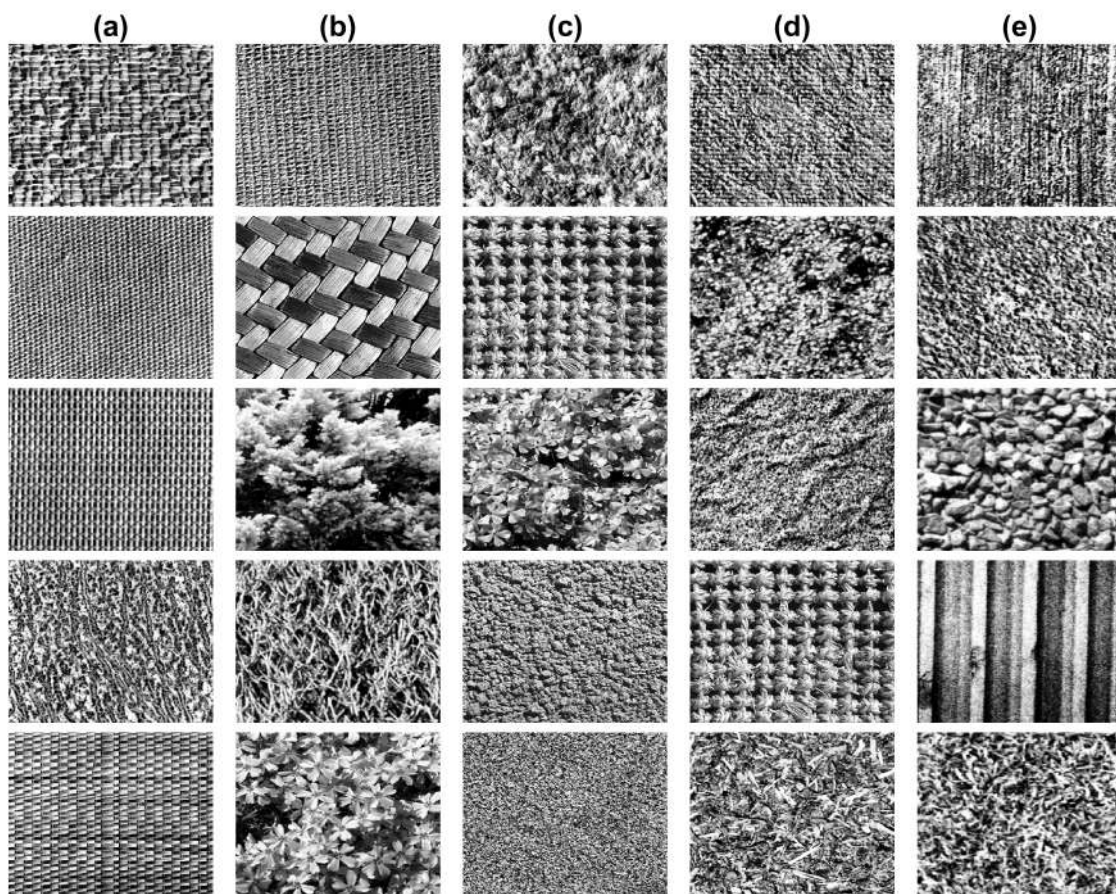
Fig. A.3 Training images corresponding to the texture arrangements of Figs. A.1(a-e).
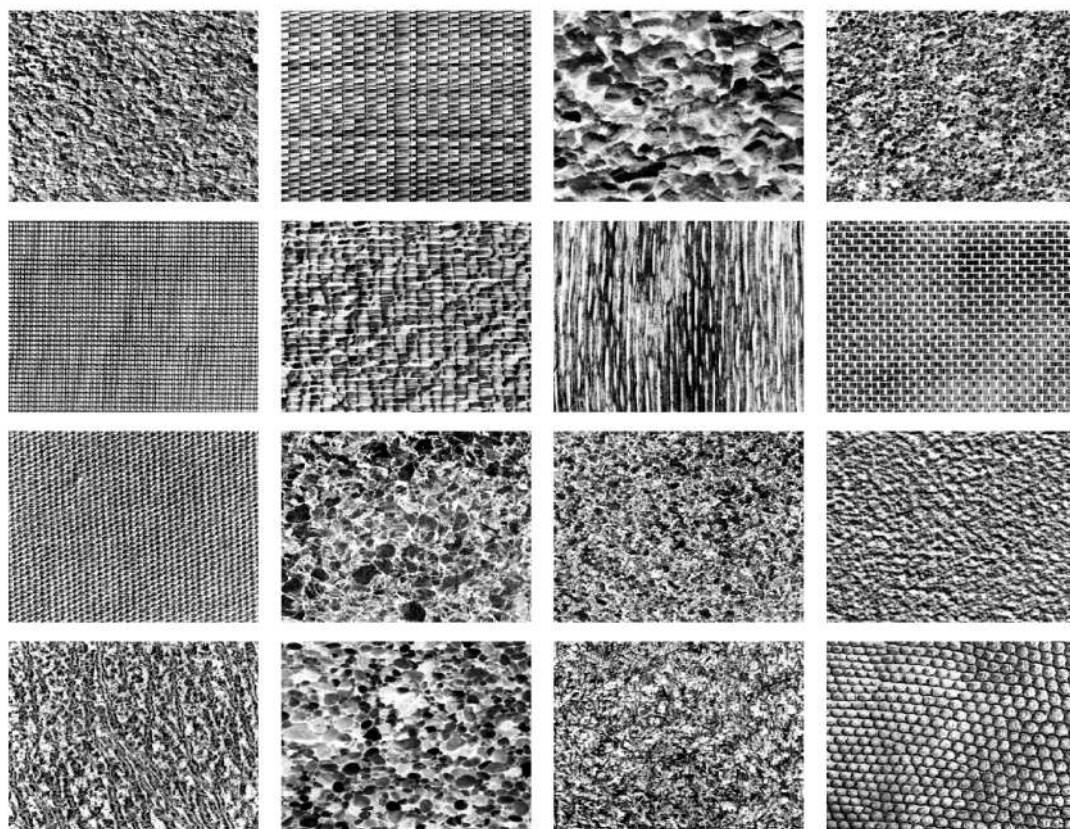
Fig. A.4 Training images corresponding to the texture arrangements of Fig. A.1(f).
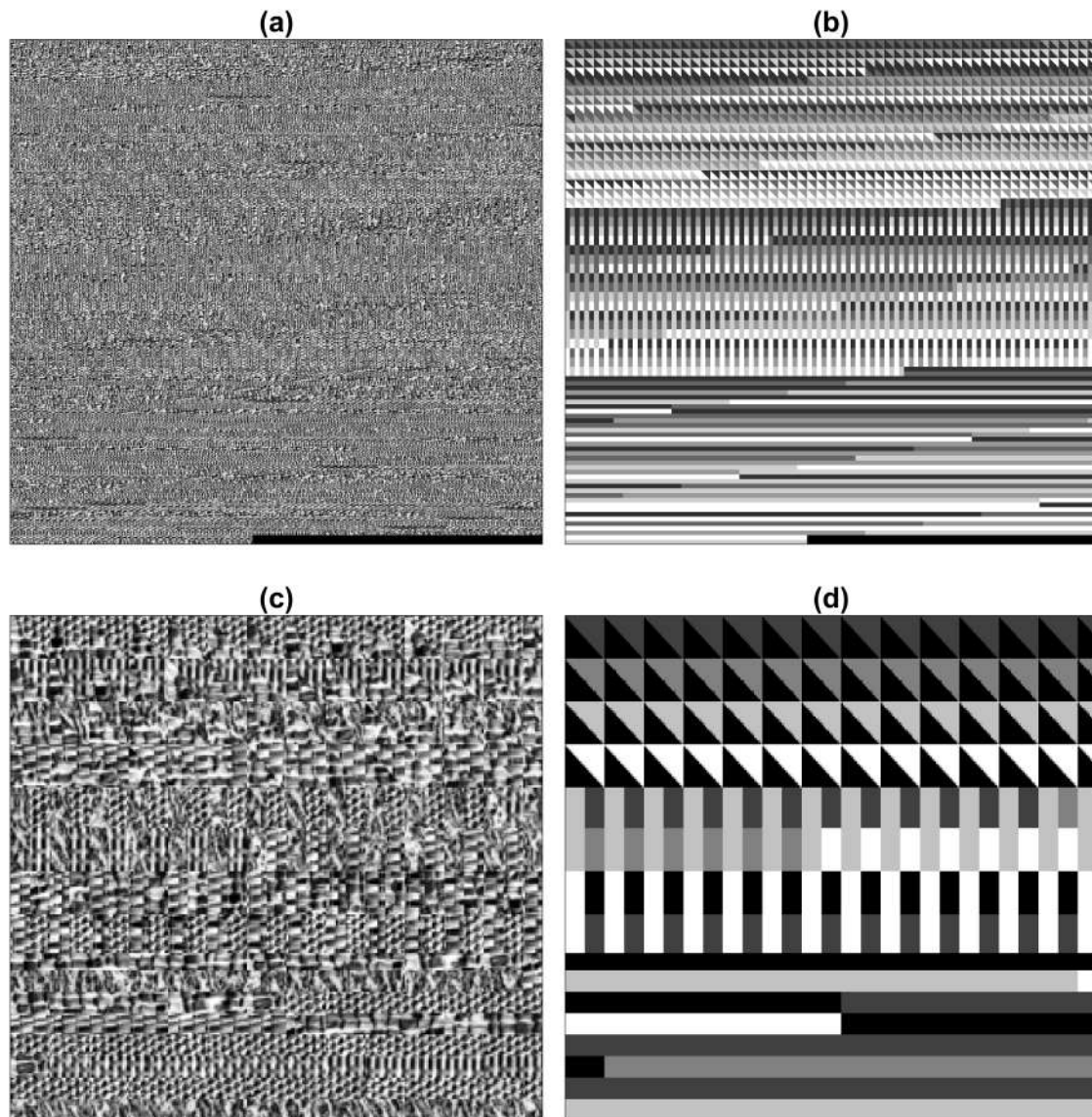
Fig. A.5 Montages of the texture pairs created to train the deep learning networks. Training images shown in Figs. A.3,A.4 were tessellated and arranged in diagonal, vertical and horizontal pairs. (a) Texture pairs. (b) Labels. (c) Detail of the texture pairs. (d) Detail of the labels.

away) with a certain orientation (e.g. horizontally). In this way, a co-occurrence matrix is able to measure local grey level dependence: textural coarseness and directionality. For example, in coarse images, the grey level of the pixels change slightly with distance, while for fine textures the levels change rapidly. From this matrix, different features like entropy, uniformity, maximum probability, contrast, correlation, difference moment, inverse difference moment, correlation can be calculated [213]. Once the features have been calculated, classifiers can be applied directly, or further processing like the watershed transforms can be applied.

Watershed transforms are based on a topographical analogy of a landscape. Should water fall in this landscape, it would find the path through which it could reach a region of minimum altitude, i.e. a basin, sometimes called lake or sea. For each point in the landscape (or pixel of the image) there is a path towards one and only one basin. Thus, the landscape can be partitioned into catchment basins or regions of influence of the regional minima and the boundaries between the basis (e.g. points of inflection) are called the watershed lines. [265]. The watershed transform can be applied to features extracted from the co-occurrence matrix[216]. The basins produced can further be iteratively merged to segment textured regions.

Local binary patterns (LBP) [214], explore the relations between neighbouring pixels. These methods concentrate in the relative intensity relations between the pixels in a small neighbourhood and not in their absolute intensity values or the spatial relationship of the whole data. The underneath assumption is that texture is not properly described by the Fourier spectrum and traditional frequency filters. The texture analysis is based on the relationship of the pixels of a 3×3 neighbourhood. A Texture Unit is first calculated by differentiating the grey level of a central pixel with the grey level of its neighbours. The difference is measured if the neighbour is greater or lower than the central pixel. Two advantages of LBP is that there is no need of quantising images and there is a certain immunity to low frequency artefacts. In a more recent paper, Ojala [215] presented another variation to the LBP by considering the sign of the difference of the grey-level differences histograms. Under the new

consideration, LBP is a particular case of the new operator called p_8. This operator is considered as a probability distribution of grey levels, when $p(g_0, g_1)$ denotes the co-occurrence probabilities, they use $p(g_0, g_1 - g_0)$ as a joint distribution.

Filtering, in the context of image processing, consists of a process that will modify the pixel values. There are spatial filters, which are applied directly to the values of the images (e.g. average neighbouring pixels to blur an image) and filters which are applied after a transformation of the data has been performed. Thus a filter in the frequency or Fourier domain will be applied after the image has been converted through the Fourier transform. The filters in the Fourier domain are sometimes named after the frequencies that are to be allowed to pass through them: low pass, band pass and high pass filters. Since textures can vary in their spectral distribution in the frequency domain, a set of sub-band filters can help in their discrimination. One common frequency filtering approach is that of Gabor multichannel filter banks [266, 219, 225, 267, 268].

The partitioning of the Fourier space can be achieved in different ways, Gabor being only one. A multiresolution approach, based on finite prolate spheroidal sequences is described in [217]. The Fourier space is divided into frequencies and orientations, which are further subdivided in a multiresolution approach. Each filter then produces a feature; different textures are captured by different filters. In addition, a feature selection strategy can improve the texture segmentation.

A.2.4 U-Net configuration

The basic U-Net architecture was formed with the following layers: *Input, Convolutional, ReLU, Max Pooling, Transposed Convolutional, Convolutional, Soft-max* and *Pixel Classification.* Two levels of depth were investigated by repeating the down-sampling and up-sampling blocks in the following configurations:

15 layers:

> Input,
>
> Convolutional, ReLU, Max Pooling,
>
> Convolutional, ReLU, Max Pooling,

Convolutional, ReLU,

Transposed Convolutional, Convolutional,

Transposed Convolutional, Convolutional,

Soft-max,

Pixel Classification

20 layers:

Input,

Convolutional, ReLU, Max Pooling,

Convolutional, ReLU, Max Pooling,

Convolutional, ReLU, Max Pooling,

Convolutional, ReLU,

Transposed Convolutional, Convolutional,

Transposed Convolutional, Convolutional,

Transposed Convolutional, Convolutional,

Soft-max,

Pixel Classification.

The image input layer was configured for the 32×32 patches. The convolutional layers consisted of 64 filters of size 3 and padding of 1. The pooling size was 2 with stride of 2. The transposed convolutional had a filter size of 4, stride of 2 and cropping of 1. The number of epochs evaluated were $10, 20, 50, 100$. The following optimisation algorithms were analysed: stochastic gradient descent (sgdm), Adam (Adam) [191] and Root Mean Square Propagation (RMSprop). One last investigation was performed by training the 20 layer network two separate times to investigate the variability of the process.

A.2.5   Misclassification

For the purposes of assessing the algorithms, a pixel-based assessment will be considered. Each pixel whose class is correctly determined by the segmentation algorithm will be counted as **Correct**, every pixel which the algorithm assigns a different class will be considered as **Incorrect**. Notice that since there is no foreground/background distinction but rather correct or incorrect, both **True Positive** (TP) and **True Negative** (TN) are included as correct, and **False Positive** (FP) and **False Negative** (FN) are included in the incorrect. Thus, the **misclassification** in percentage, or classification error, will be calculated as number of incorrect pixels divided by the total number of pixels of the image $m = 100 * (FP + FN)/(TP + TN + FP + FN)$. The accuracy can be calculated as the complement $a = 100 * (TP + TN)/(TP + TN + FP + FN)$.

A.3   Results

For each image, the networks were trained with the 3 different optimisation algorithms, 3 layer configurations and 4 epoch numbers, for a total of 36 different combinations. Thus for the 6 composites images there were 216 results. The misclassification of each segmentation was measured against the ground truth as the percentage of pixels classified incorrectly. These results are summarised in table A.1.

The best results for each image were selected and compared against traditional methodologies and are shown in table A.2. The results are illustrated graphically in two ways. Fig. A.6 shows segmented the classes overlaid as different colours over the original textured images. Fig. A.7 shows correctly segmented pixels in white and the misclassified pixels in black.

Table A.1 Comparative misclassification (%) results of the different U-Net configurations. (Bold and underline denotes the best result for each image).

Method			Figures					
Layers	Optimisation Algorithm	Epochs	a	b	c	d	e	f
15	sgdm	10	6.8	21.5	40.8	31.2	27.2	20.9
20	sgdm	10	33.0	59.0	74.3	79.1	77.3	41.9
20	sgdm	10	71.9	62.9	74.3	78.8	72.1	39.0
15	Adam	10	3.2	10.4	7.9	**7.1**	17.8	19.3
20	Adam	10	7.4	15.5	46.5	25.0	45.1	94.2
20	Adam	10	6.4	15.5	36.0	21.1	26.7	32.9
15	RMSprop	10	5.1	**8.9**	14.0	18.3	12.1	17.6
20	RMSprop	10	5.3	42.4	45.3	59.9	56.2	27.7
20	RMSprop	10	20.2	37.4	47.0	43.7	44.2	26.1
15	sgdm	20	3.8	23.1	17.5	15.9	14.1	19.8
20	sgdm	20	27.3	60.5	74.8	69.3	73.9	27.4
20	sgdm	20	23.8	51.0	63.6	66.8	56.5	26.7
15	Adam	20	3.7	11.6	7.5	7.4	9.5	71.7
20	Adam	20	6.1	13.3	28.7	18.5	40.8	32.2
20	Adam	20	5.6	17.9	27.4	22.5	39.3	94.0
15	RMSprop	20	3.8	11.7	14.5	19.2	11.7	17.9
20	RMSprop	20	6.1	42.2	54.7	47.5	42.6	22.3
20	RMSprop	20	19.1	30.3	44.7	51.7	37.1	26.9
15	sgdm	50	3.2	15.3	9.2	7.7	13.8	19.6
20	sgdm	50	18.2	32.2	60.3	42.8	30.2	28.9
20	sgdm	50	9.4	55.2	56.0	16.0	32.4	32.4
15	Adam	50	3.4	10.4	9.8	9.9	39.1	22.6
20	Adam	50	8.3	80.3	19.8	82.3	79.6	34.8
20	Adam	50	7.2	9.6	41.4	10.0	27.6	23.6
15	RMSprop	50	3.4	18.7	10.0	8.3	11.2	**17.5**
20	RMSprop	50	5.6	33.2	25.7	34.8	34.4	22.4
20	RMSprop	50	5.4	22.8	45.3	20.0	34.7	29.2
15	sgdm	100	3.9	10.6	7.9	7.7	**7.7**	21.4
20	sgdm	100	9.6	22.1	39.4	39.7	30.3	23.8
20	sgdm	100	13.7	17.1	52.8	26.3	37.1	30.5
15	Adam	100	2.7	16.6	80.3	7.2	18.2	21.9
20	Adam	100	**2.6**	38.9	79.9	80.1	31.1	25.7
20	Adam	100	3.4	80.0	79.7	80.9	80.3	28.6
15	RMSprop	100	4.8	11.2	**7.2**	8.1	9.5	18.1
20	RMSprop	100	7.1	66.0	46.0	28.6	30.9	24.0
20	RMSprop	100	5.6	29.5	26.9	18.5	29.3	22.9
Max			71.9	80.3	80.3	82.3	80.3	94.1
Mean			10.4	30.7	39.4	33.7	35.6	30.7
Min			2.6	8.9	7.2	7.1	7.7	17.5

Table A.2 Comparative misclassification (%) results with co-occurrence [213], best filtering result from Randen [170], p_8 and LBP [215], Watershed [216], Multiresolution sub-band filtering (MSBF) [217] and U-Net [88]. (Bold is the best for each image).

Method	Figures						
	a	b	c	d	e	f	Average
Co-occurrence [213]	9.9	27.0	26.1	51.1	35.7	49.6	33.23
Best in Randen [259]	7.2	18.9	20.6	16.8	17.2	34.7	19.23
p_8 [214]	7.4	12.8	15.9	18.4	16.6	27.7	16.46
LBP [214]	6.0	18.0	12.1	9.7	11.4	**17.0**	12.36
Watershed [216]	7.1	10.7	12.4	11.6	14.9	20.0	12.78
MSBF [217]	2.8	14.8	8.4	7.3	**4.3**	17.9	9.25
U-Net [88]	**2.6**	**8.9**	**7.2**	**7.1**	7.7	17.5	**8.50**
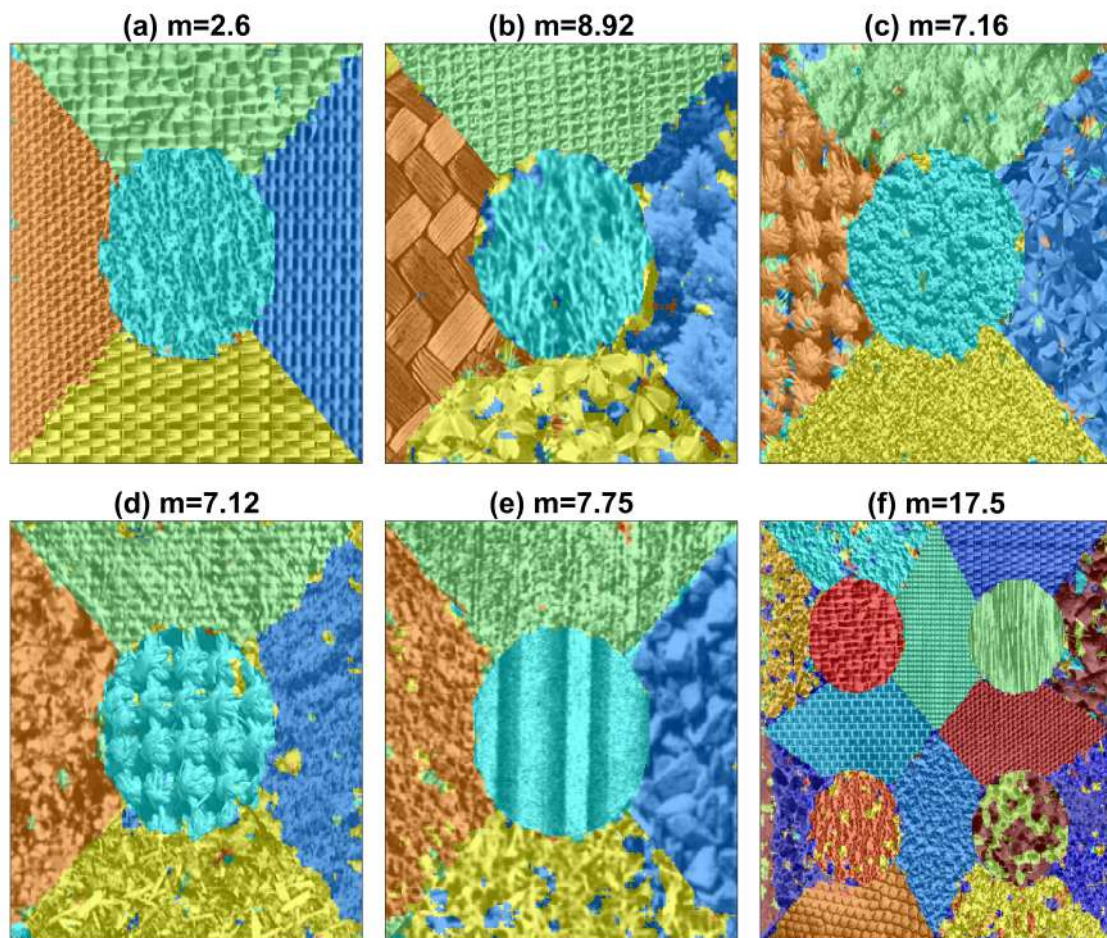
Fig. A.6 (a-f) Results of the segmentation with U-Nets for the six texture arrangements. The misclassification (%) is shown in each case. The classes are shown as overlaid colours.
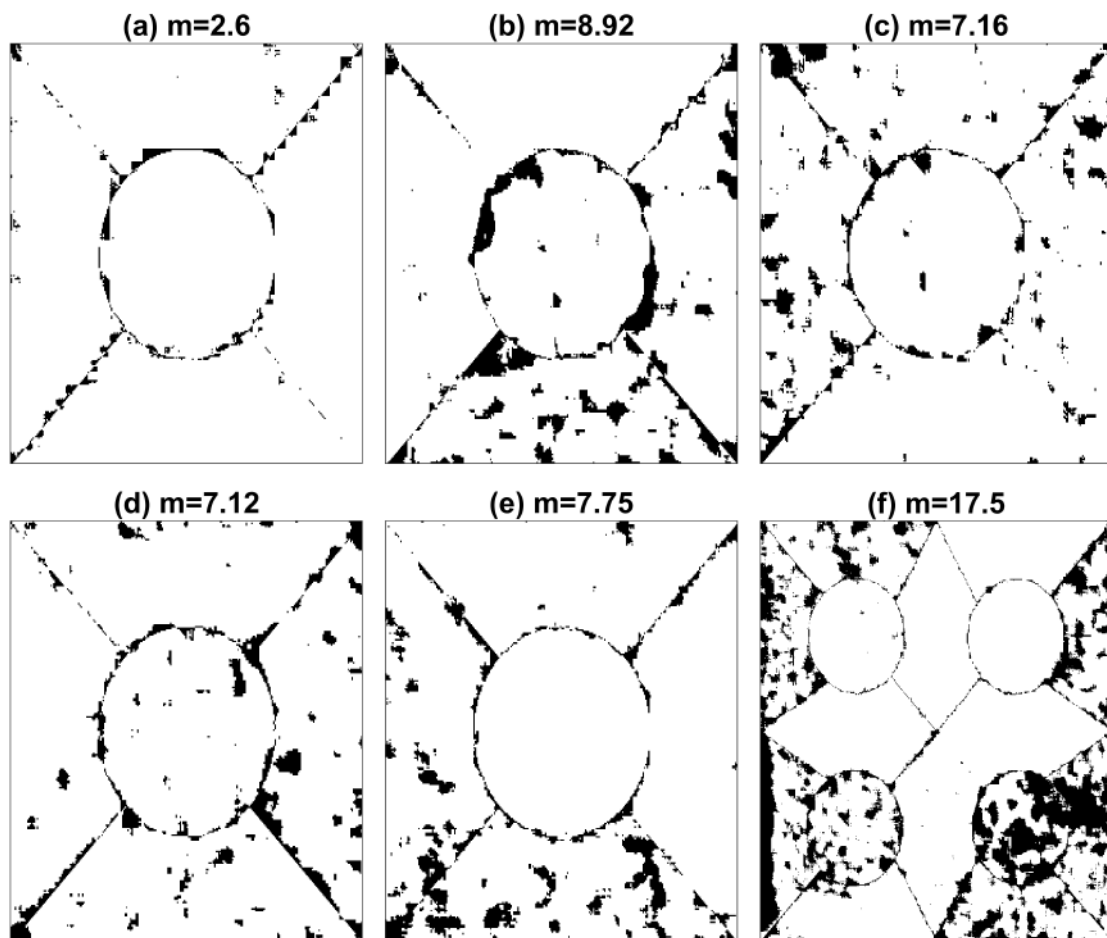
Fig. A.7 (a-f) Results of the segmentation with U-Nets for the six texture arrangements. The misclassification (%) is shown in each case. Pixels that are correctly classified appear in white.

A.4 Discussion

The results provided by the U-Net algorithm provided interesting results in terms of the actual misclassification results against traditional algorithms, and the variability of the U-Net cases. The segmentation results provided by the U-Nets were better in four of the six images. In some cases, the results were very close to the second best option (a: 2.8/2.6, d: 7.3/7.1) and in two cases (e,f) traditional algorithms provided better results (e: 4.3/7.7, f: 17.0/17.5). The average for all the six composites was best for U-Nets, however, given the fact that the difference with the second best is relatively small (0.75), and that traditional algorithms provided better results in 1/3 cases shows that care should be taken when selecting algorithms. This is similar to the conclusion of Randen who stated that "No single approach did perform best or very close to the best for all images" [259].

In terms of the U-Net configuration there are several interesting observations. First, there was a great variability in the results produced by the different U-Net configurations. It was surprising that the maximum value of the misclassification in some cases was extremely high, 80% in the cases of 5 textures and 94% in the case of 16 textures, those cases are equivalent of selecting a single class for all textures. Second, three of the best results were obtained with 100 epochs, 2 with 10 epochs, and 1 with 50, which is counter-intuitive as it would be expected that longer training times would provide better results. Third, three of the best results were provided by RMSprop optimisation, two by Adam and one by sgdm. Fourth, and perhaps the most surprising result was that the results provided by the two 20 layer configurations were very different. In a few cases the result were equal (e.g. image c, sgdm, 10 epochs; image b, Adam, 10 epochs) but in others the variation was huge (e.g. image b, Adam, 50 epochs).

In terms of texture, it can be highlighted that not all textures are the same, the five textures of image (a) are far easier to distinguish and correctly segment than those of image (b) and image (f). The U-Net was capable of segmenting these textures with accuracy comparable or better than traditional techniques. As mentioned previously,

the fact that the textures have been histogram equalised removes the discrimination of the regions by their average intensities. More complex architectures, e.g. Siamese Networks [261] could provide better results, but it is important to use a standard benchmark such as that provided by Randen [259].

There are many other configuration parameters that could be varied; *learning rate, batch size, variations of the training data, different number of layers*, but for the purpose of this work, the results show first, the capability of deep learning architectures for segmentation of textured images and second, in some cases better results that traditional methodologies. However, the configuration of the network is not trivial and variations of some parameters can provide sub-optimal results. The experiments conducted in this work did not provide conclusive evidence for the selection of any of the parameters evaluated. Furthermore, training of the networks requires considerable resources. The training times for the images with 5 textures took around 5 hours and for the image with 16 textures around 96 hours on a Mac Pro (Late 2013) with a 3.7GHz Quad-Core and 32 GB Memory with Dual AMD FirePro D300 graphics processors.

Therefore, it can be concluded that U-Net convolutional neural networks can be used for texture segmentation and provide results that are comparable or better than traditional texture algorithms. Furthermore, these results encourage the application of deep learning to other areas. If we assume that different textures are characterised by patterns, i.e. repetitions of certain sequences or particular variation of intensities, then any data which is characterised by patterns could be analysed. For instance, phonemes in human speech have different patterns, which when combined form words. Thus one line of an image with different textures would have similar characteristics as a the intensity variation of a phrase with different phonemes. Moreover, voice signals, which are one-dimensional can be converted into two-dimensional spectrograms [269] with time on one axis and frequency in another axis. In these cases, the spectrograms can be analysed for texture directly.

Appendix B

Preliminaries of image analysis

In this appendix, an overview on the broad field of image analysis is presented. Classical approaches are overviewed and in some cases tested on generic images. The techniques discussed here present a thorough explanation of the methods used as exploration of the data in chapter 4.

B.1   Pre-processing

Pre-processing of an image in this report refers to the operations performed on an image that allows for a more precise segmentation or analysis of the data. These operations can highlight some of the images' features, remove noise, modify dimensions and should improve the results of posterior operations. One classical classification of pre-processing techniques is from the work by Sonka et al. [128], which classifies such techniques into pixel brightness transformations, geometric transformations and transformations in local neighbourhoods. Pixel brightness transformations refer to modifications in the image regarding the pixels themselves, which normally address brightness or grey-scale corrections. Geometric transformations are made to address different types of distortion an image may present. Finally, local-preprocessing involves the operation in the neighbourhood of a given pixel in order to produce a new value for that location.

Furthermore, another approach in the work by Gonzalez and Woods [127] classifies the techniques into transformations of intensities and spatial filtering. Transformations of intensities, refer to transformations to the intensity levels $I = 1, 2 \cdot \cdot \cdot , L$ in each pixel, which produce modifications of the statistical distribution of the intensities. Spatial filtering refers to the local neighbourhood operations. Techniques which involve the convolution of the image with a filter kernel, to produce a new image in which each pixel is the result of a local operation.

B.1.1 Noise

Real images are often degraded by some random errors - this degradation is usually called noise [128]. Noise can occur during image capture, transmission, or processing, and may be dependent on, or independent or, image content. Left side of Figure B.1a,b show the original image with grainy texture used in this work. When looking at individual pixel intensities and their neighbourhoods, noise can be seen as intensity values that are quite different to those within the pixel's neighbourhood. Thus techniques for dealing with noise are commonly associated with smoothing techniques.

Filtering

This section provides an overview of the techniques in which each pixel in the image is assigned a specific neighbourhood in which a predefined operation is performed.

Filtering can be viewed differently depending on the field of study. Apart from reducing noise in the images, other filters can be used to enhance some properties or features of the image, extracting relevant information from it. The term filtering is taken from the field of digital signal analysis and processing in which properties of a signal are selected, enhanced or suppressed through convolution with a finite function with special characteristics.

Filter kernels consist of matrices commonly of small sizes that can have different parameters that determine the values within them. Throughout this work, it can be assumed that all kernels, represented with letter $\mathbb{K}$, are square matrices of size n, with n $= 2a + 1$, $a \in N$, unless explicitly determined.

Depending on the type of filter applied, the values will change to highlight certain features in the image, like edges or corners. The following sections discuss two of the principal filtering approaches: smoothing and sharpening of image features. The difference lies in the properties of the kernels chosen to perform the convolution. To highlight the processes performed in each image, two levels of detail will be provided.

In this section, examples of smoothing filters are presented, as well as its application on relevant images for this work. The filters are presented in terms of the kernels that contain them.

Smoothing (also called averaging) spatial filters are used to reduce sharp transitions in intensity [127]. Because random noise typically consists of sharp transitions in intensity, an obvious application of smoothing is noise reduction. Smoothing is used to reduce irrelevant detail in an image, where "irrelevant" refers to pixel regions that are small with respect to the size of the filter kernel. The kernel is an array whose size defines the neighboured of operation, and whose coefficients determine the nature of the filter. Smoothing filters are used in combination with other techniques for image enhancement. Spatial filtering consists of convolving an image with a filter kernel. Convolving a smoothing kernel with an image blurs the image, with the degree of blurring being determined by the size of the kernel and the values of its coefficients.

In general, smoothing filters are useful if the data presents high frequency noise, and thus serve as low pass filters, in the context of signal analysis. Several filters were applied to the data exploring their individual parameters.

Averaging kernel

In this type of kernels, the values of the elements in $\mathbb{K}$ represents a weight or proportion, that is, the values are between $(0, 1)$ and the sum of all the values in the kernel add up to 1.

In the convolution $\mathbb{I} \circledast \mathbb{K} = \text{imfilt}(\mathbb{I}, \mathbb{K})$, every pixel value is replaced by the weighted average of the pixels in the n-neighbourhood [2].

The most straightforward implementation of this filter is what is known as a box filter, which has uniform weights assigned to all the pixels $\mathbb{K}(i, j) = 1/n^2$. The only

parameter is the size of the filter, which at each point in the resulting matrix produces the mean of the intensity values contained within the scope of the filter. Another example of the weight distribution in the kernel is the disk filter, an averaging filter acting upon a circular area. The circle is or radius r and the size of the kernel is $2r + 1$. The filter highlights circular structures which match the size of the radius of the circle. The principle is the same as the one in the box filter, given the size of the filter, taking the central element $K(r + 1, r + 1)$, all elements within the radius will contain a value higher than zero. As it is a discrete filter, the edges of the circle will not coincide with the finite grid of the kernel.

B.1.2   Gaussian filter

Gaussian filters are another type of smoothing filter in which the values follow a discrete 2D Gaussian of the size of the kernel. The parameters in the filter are zero mean (0) and variance (σ^2), as well as the size of the filter (n). The 2D shape of the Gaussian resembles a bell, where the width of it is determined by the variance. It is important to note that the Gaussian kernel must consider a value for n large enough for an adequate discretisation of $\mathbb{K}$.

Image smoothing is the set of local pre-processing methods whose predominant use is the suppression of image noise - it uses redundancy in the image data [128]. Noise is often described by its probabilistic characteristics, and as such can be associated with the Gaussian distribution as a very good approximation to noise that occurs in many practical cases [128]. The Gaussian distribution is characterised in one dimension by a bell-shaped curve, in two by a bell shape and has one parameter, σ, the standard deviation of the distribution. The 2D Gaussian smoothing operator G(x,y), also called a Gaussian filter, or simply a Gaussian is given by Equation B.1

$$G(x, y) = e^{-(x^2 + y^2)/2\sigma^2} \tag{B.1}$$

<center>(a)   (b)</center>
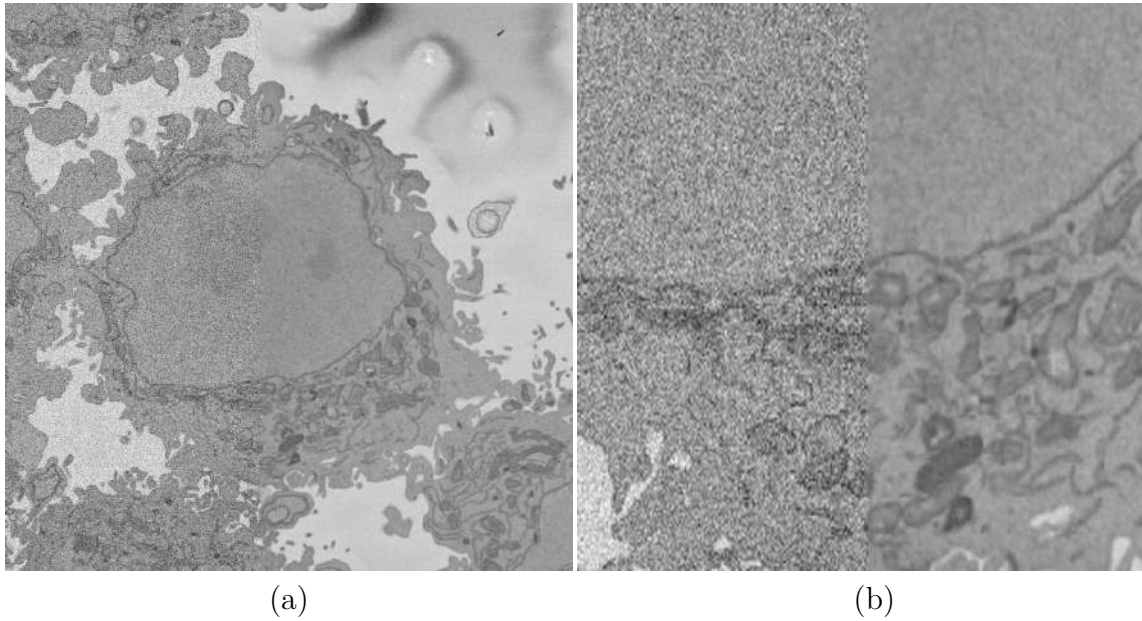
Fig. B.1 (a) Illustration of the effects of low-pass filtering on one representative image of a HeLa cell. Left half: Original image with grainy texture. Right half: image filtered with a low-pass filter with a Gaussian kernel with size $h = 7$ and standard deviation $\sigma = 2$ to remove high frequencies. (b) Region of interest of (a).

where x,y are the image co-ordinates and σ is a standard deviation of the associated probability distribution. Sometimes this is presented with a normalising factor as in Equation B.2 :

$$G(x,y) = \frac{1}{2\pi\sigma}e^{-(x^2+y^2)/2\sigma^2} \tag{B.2}$$

The standard deviation σ is the only parameter of the Gaussian filter - it is proportional to the size of the neighbourhood on which the filter operates. Pixels more distant from the centre of the operator have smaller influence, and pixels further than 3σ from the centre have negligible influence. For this distribution to be used as a smoothing operator we modify the parameter σ to give the approximate neighbourhood size that we wish to consider. Larger values of σ correspond to wider bell shapes and thus larger neighbourhoods over which smoothing takes place. Since the image is discrete but the Gaussian distribution is continuous, the kernel we define is a discrete approximation to the continuous Gaussian for a particular value of σ.

Figure B.1 show the effect of low pass filtering on the original image of a HeLa cell. Image was filtered with a low-pass filter with a Gaussian kernel with size $h = 7$ and standard deviation $\sigma = 2$. Filtered image improves segmentation as the Gaussian blurs the original image and to removes high frequencies.

Order statistic filtering.

Most filtering is performed through convolution. As shown in Figure C.1, the convolution can be interpreted as a moving window of the size of the kernel, which at every step, selects the pixels in the input image that will contribute to the new intensities certain position in the output; normally through the sum of the array multiplication of the elements in the kernel by the elements in the image. Order statistic filtering, the same idea of a moving window is taken, but at every stage, the calculation of a percentile is performed in the pixels selected at a certain moment. The most common calculations are the minimum, maximum and median filters.

Sharpening filters

To analyse a digital image, it is sometimes useful to locate certain geometric objects, such as lines which could correspond to edges and ridges in the image where certain key objects are located. Thus, highlighting transitions in intensity, with a sense of the magnitude of such transitions becomes key to image processing [127]. In calculus, the operator that describes the rate of change at a certain point is the derivative, or in multivariate calculus, the gradient. As images were constructed as discretisations of continuous functions of two variables, applying the numerical concepts of differentiation to the images would provide an insight to the characterisation of intensity transitions. On the other hand, as an image can be observed as a signal in the spatial domain, abrupt transitions could be linked to locations of a high frequency.

B.2 Image segmentation by intensity thresholding

Segmentation is the process of classifying pixels into one of two categories: foreground, or objects of interest, and background. In a way, segmentation provides context to an

image selecting the regions which - given an application- are important. In particular cell segmentation is a widely studied problem which has produced considerable amount of research output.

Intensity levels on an image are referred to be a finite set of L levels, $\{1, 2, 3, \ldots, L\}$. Thus, performing a segmentation by intensity thresholding implies finding a value k which separates the intensity levels into groups: $\{1, 2, 3, \ldots, k-1\}$ and $\{k, k+1, k+2, \ldots, L\}$. Let $\mathcal{C}_R$ be the set of intensities in which the intensities of an image are categorised, where R refers to the region in which the image will be segmented, and can be either a subscript or a number depending on the context. It is important to remember that the segmentation output of a method will be a binary image which takes the value 1 at the detections of foreground and 0 at the background. The concept of hysteresis thresholding is a technique for image segmentation that uses two levels to segment an image, a higher one that determines what will be part of the foreground and a lower one that will distinguish what will be considered as background.

In the following sections, the description of different techniques to automatically select the threshold from the image intensities in the image will be given. Two main approaches are compared: the global threshold, which uses a single level throughout the image and the adaptive threshold which selects a threshold locally.

B.2.1   Otsu's threshold

Otsu [270] developed an algorithm for differentiating between classes, i.e. finding a threshold, by maximising the quotient of between and within variances of the classes. In simple terms, the algorithm would select a threshold that would distinguish the classes as much as possible, while at the same time would make the intensity levels within the class as alike as possible, one advantage is that this procedure can simply be extended to more than two classes. The derivation of the algorithm is presented for a single threshold as the extension to multiple thresholds is straightforward.

Given the levels of intensity found in an image $I = \{1, 2, 3, \cdots, L\}$, the Otsu method finds a level of intensity $k, 1 \leq k \leq L$, that maximises the quotient σ_B^2/σ_W^2,
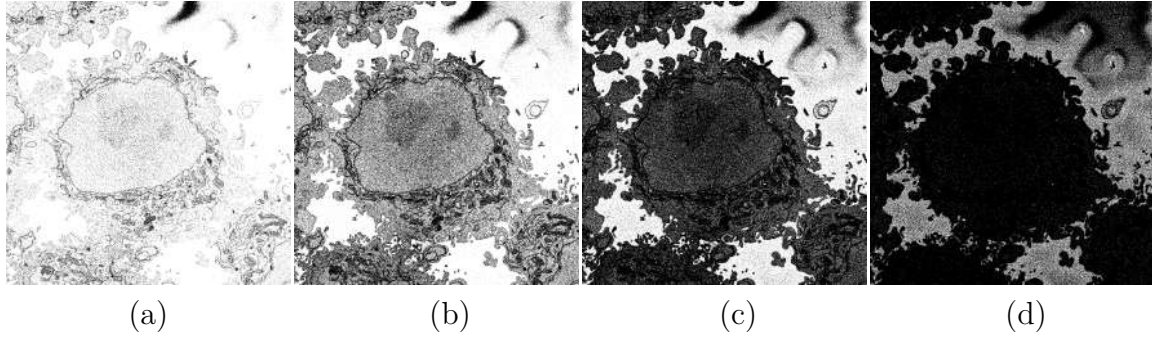
Fig. B.2 Illustration of a HeLa cell image after applying different Otsu thresholds. Different number of thresholds were applied producing different number of classes $\mathcal{C}_i$. (a)-(d) show the results of applying 4 thresholds selected by the Otsu method. The following threshold values were used - 115, 145, 171, and 201 - range of values is 86. There may seem to be grey levels besides white and black, but this is due to the grainy textures.

where σ_B^2 is the variance between the two classes (foreground and background) and σ_W^2 is the variance within the classes. Otsu's method, while very fast, is very sensitive to outliers on the data. Figure B.2 illustrates the results of applying 4 thresholds selected by the Otsu method on a HeLa cell image.

B.2.2   Adaptive thresholding

Variations of the thresholding technique are presented in the works by Pappas and Bradley and Roth [2]. In the work by Pappas the segmentation achieved was different from the global ones, as the threshold level chosen is dependent on the local pixels at given positions. Instead of selecting thresholds that take into consideration the statistical properties of the entire image, the method starts with global estimates of the levels and adapts them to the local characteristics of each region.

The concept of an adaptive threshold becomes relevant when the objects of interest in images varies considerably, and a single threshold might not be enough, even a robust one like a hysteresis threshold. In addition, it is possible that the images suffer from shading due to a variety of reasons. The work by Pappas can be seen as a generalisation of the k-means algorithm that include spatial constraints and account for local variations of intensity. The algorithm works through the estimation of the regions

of an image (segmentation) and the parameters in an a posteriori density function that computes the probability of the distribution of regions (x) given the observed image (y).

On the other hand, the work by Bradley and Roth presents an adaptive threshold methodology that takes into account spatial variations in illumination. It achieves its purpose by selecting a threshold value based on local mean intensity in the neighbourhood of each pixel. A neighbourhood of 1/8th of the size of the image is used around each pixel. Each pixel gets assigned a different threshold value in which each pixel is compared against. A parameter called sensitivity is included, which allows for some control from the user into the threshold taken by the algorithm.

B.3   Methods based on derivatives

In this section, methods based on derivatives will be described. The derivatives will be outlined both in terms of the gradient and the filter which can be used to perform the analysis. The image gradient can be obtained from an estimation of the derivative through central differences. In this work, the symbol used for partial derivatives applied to an image is $\partial_x \mathbb{I}(x,y), \partial_y \mathbb{I}(x,y)$, will refer to the point-wise estimation of the derivative. While the symbol for gradient $\nabla_x \mathbb{I}$ will refer to the image which contains all the partial derivative calculations per position, i.e. $\nabla_x \mathbb{I} = (\partial_x \mathbb{I}(i,j))$. Consider a kernels given by the matrices $\mathbb{K}_x = (1,0,1)^T$ and $\mathbb{K}_y = (1,0,1)$. Centred around position (x,y), a convolution of certain filters in the image would produce at each pixel the approximation of the derivative per point.

The image gradient will consist of both images containing all the partial derivatives per point, and per direction: $\mathbb{G}_x = \nabla_x \mathbb{I}, \mathbb{G}_y = \nabla_y \mathbb{I}$, as well as the gradient magnitude $\mathbb{G} = |\nabla \mathbb{I}| = \sqrt{\nabla_x \mathbb{I}^2 + \nabla_y \mathbb{I}^2}$. Different calculations of the gradients can be applied, commonly, **Sobel**, **Prewitt** and **Roberts**. The difference lies in the kernels used.

Methods based on derivatives			Laplacian
Sobel	Prewitt	Roberts	

$$\mathbb{K}_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad \mathbb{K}_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad \mathbb{K}_{d1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \mathbb{K}_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

$$\mathbb{K}_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \mathbb{K}_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad \mathbb{K}_{d2} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \mathbb{K}_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Fig. B.3 List of kernels of common sharpening filters in different directions.

Table 3.2 shows the different kernels to calculate the gradients $\mathbb{G}_x, \mathbb{G}_y$ for an image. The Laplacian filter is also presented, which provides the estimation of the Laplacian operator $\nabla^2 f = \partial^2_{xx} f + \partial^2_{yy} f,$.

Figure B.3 shows the table of kernels of common sharpening filters in different directions.

To showcase the different implementations of the derivatives, Figure B.4 includes the detail of a Wild-type of HeLa cell data set in which the image has been processed with each of the filters.

B.4   Edge detection

An important problem in image analysis is the detection of contours of objects or edges. Note that filtering highlights edges and points of abrupt changes, however edge detection refers to the creation of a binary image where positive values are located where changes are more abrupt. In particular, the Canny [178] edge detection algorithm has been a widely referenced image analysis technique with thousands of citations since

(a)   (b)

(c)   (d)   (e)
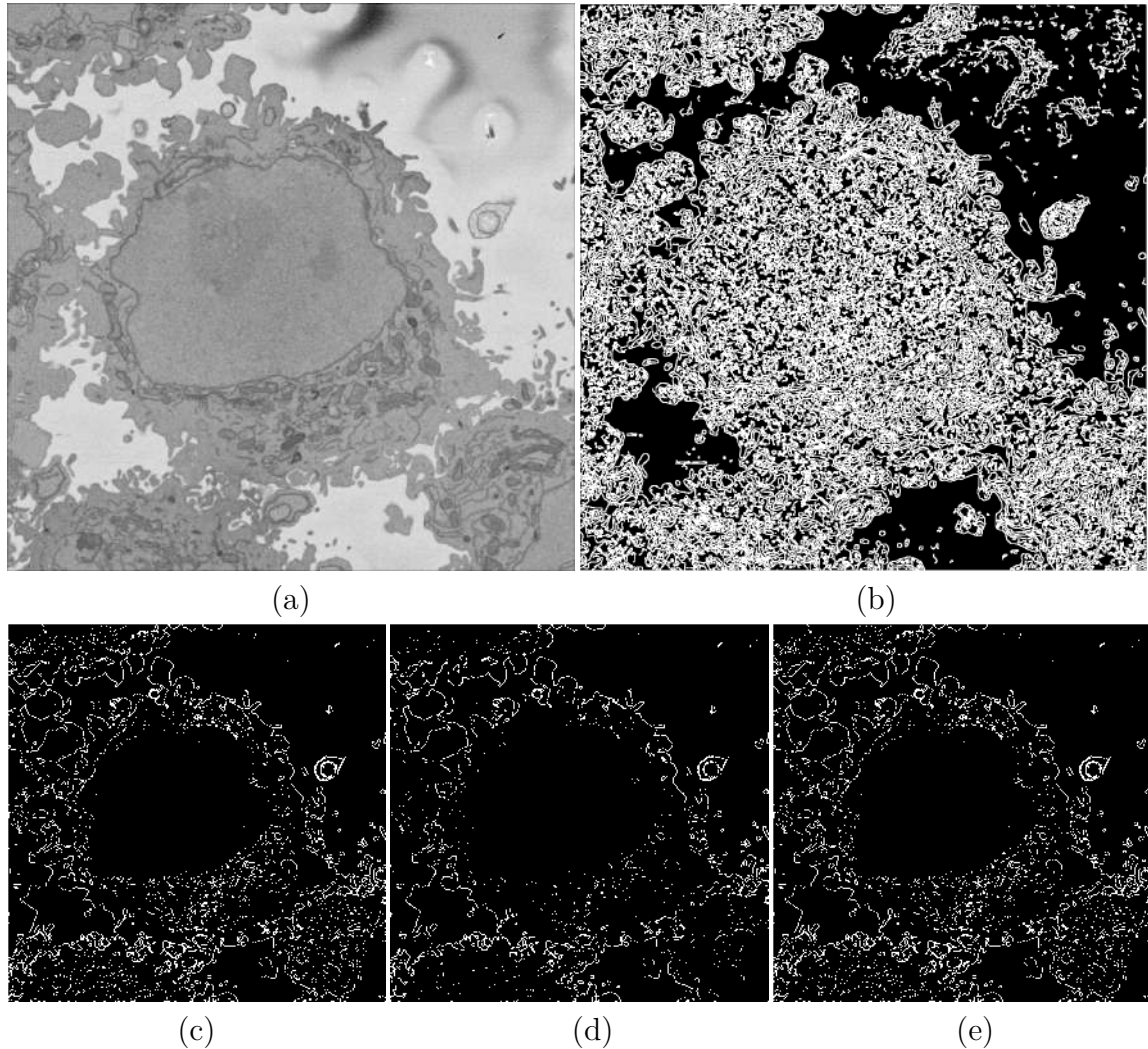
Fig. B.4 Representation of different edge detection methods applied on HeLa cell image. Edge detection is an image processing technique for finding the boundaries of objects within images. Detail of a slice is presented after applying the different edge detection methods analysed in this section, each one presenting the gradient's magnitude. (a) HeLa raw image with high frequency noise. (b) Raw image was low-pass filtered with a Gaussian kernel with size $h = 7$ and standard deviation $\sigma = 2$ to remove high frequency noise. (c) Canny, (d) Sobel, (e) Roberts. (f) Prewitt. The edges were dilated for visualisation purposes.

its publication in 1986. As it is relevant to this work, an overview of the algorithm is provided and some examples of its implementations, while varying its key parameters.

The two main advantages of the algorithm are (i) the ability to modify the Gaussian kernels, allowing different edges to be visible and (ii) the non-maximum suppression which allows to select the more relevant ones. The technique would be better classified as a segmentation of the edges in the image, as the output is a binary image with two levels, $I = 0, 1$. It is included in this section to provide context of the application of sharpening filters and because the term segmentation, in this work, refers to the distinction of cells from other cells and from the background.

A significant class of algorithms utilise the edges found from an image that are defined by changes in colour, intensity or texture as a gradient within the image. Among the most popular methods are edge detecting operators, such as the Laplacian, Prewitt and Sobel convolution filters. All these operators convolve a small square kernel, often of dimension 3×3, with the image to compute the gradient or second derivative at each pixel in the image. The resulting gradient map can then be thresholded to decide which edges are significant. The results of such operators most often cannot be used directly for segmentation, instead additional processing is required to form a continuous edge map from the convolved image. Noise in the original image is also a source of errors in segmentation; edges can appear where there is no border in the original image, and vice versa. A popular technique is to smooth the image before edge detection in an attempt to reduce noise; this is most commonly achieved using a Gaussian filter.

B.4.1 Canny edge detection

Edge detectors are a collection of very important local image pre-processing methods used to locate changes in the intensity function; edges are pixels where this function changes abruptly [128]. A significant class of algorithms utilise the edges found from an image that are defined by changes in colour, intensity or texture as a gradient within the image. Among the most popular methods are edge detecting operators, such as the Laplacian, Prewitt [271] and Sobel [272] convolution filters. All these operators

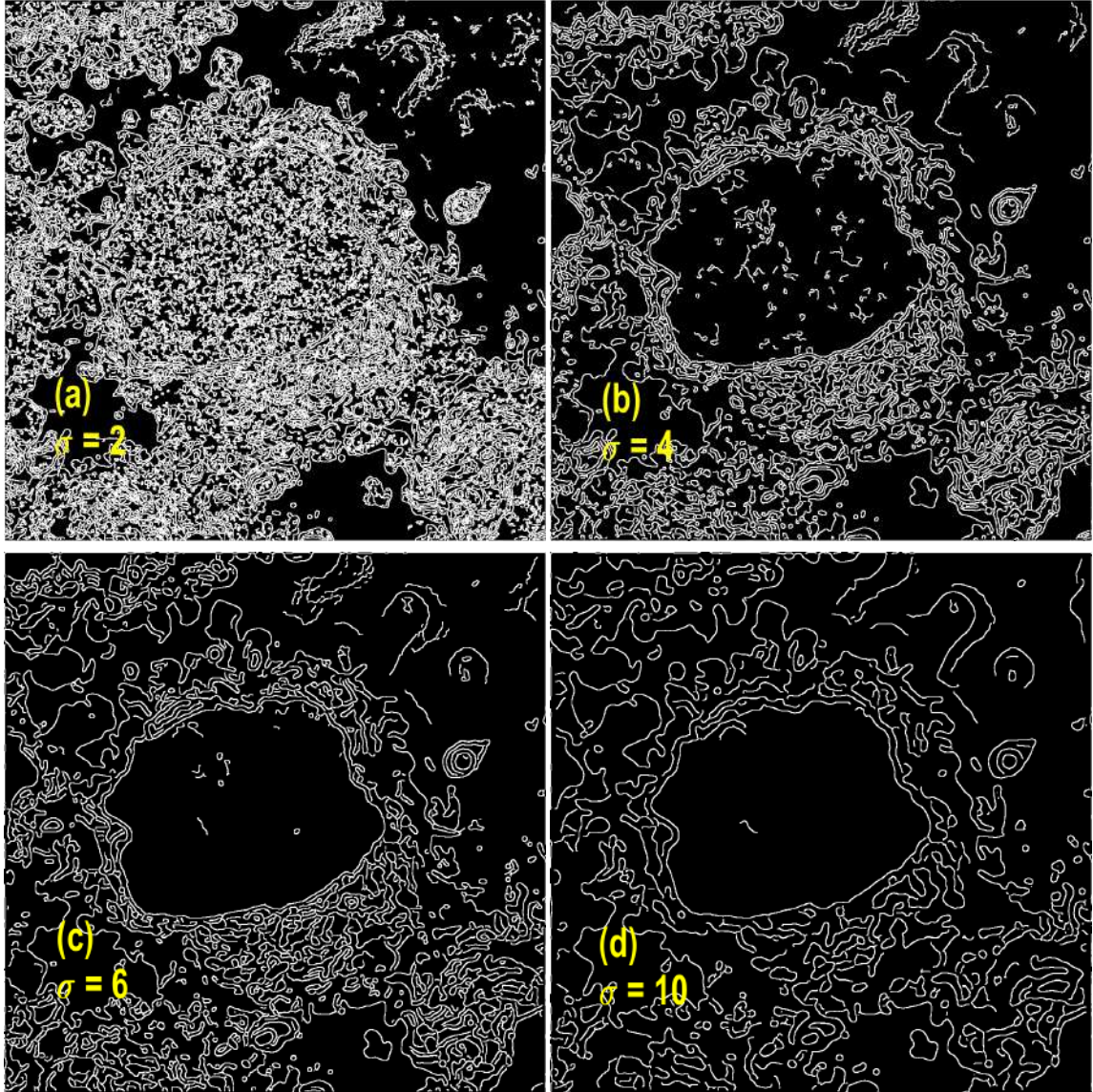Fig. B.5 Detail of output of Canny algorithm on HeLa cell image. Different values of σ^2 are presented to showcase the output of the algorithm. Notice that the size of the variance σ is inversely proportional to the level of detail being analysed by the algorithm. (a) Canny standard deviation $\sigma = 2$, (b) $\sigma = 4$, (c) $\sigma = 6$ and (d) $\sigma = 10$. Edges were dilated for visualisation purposes.

convolve a small square kernel, often of dimension 3×3, with the image to compute the gradient or second derivative at each pixel in the image. The resulting gradient map can then be thresholded to decide which edges are significant.

Introduced by John Canny in 1986 [178], a commonly used algorithm is that of Canny edge detection [127, 128] and its performance is superior in general to the other edge detectors mentioned earlier. It is based upon three criteria that Canny aimed to satisfy to find an optimal edge detector:

(a) Low error rate. Important edges should not be missed - all edges should be found, and there should be no spurious responses. (b) Edge points should be well localized. Distance between the actual and located position of the edge should be minimal. (c) Single edge point response. Any given edge should only be marked once and noise should not create false edges. If there are two responses to any given edge, one should be marked false.

The first stage of his algorithm is to perform noise reduction by a convolution with a Gaussian filter. From here, estimates of the edge normal directions are calculated for each pixel using an edge detection operator, which are then rounded to one of the four directions that represent vertical, horizontal and diagonal gradients. This is then used to perform non-maximum suppression of edges; a pixel is determined to be an edge if its gradient magnitude is greater than those of both its immediate neighbours in the same direction as the gradient magnitude.

At this stage we have a binary image consisting of edge points. An important step is to now go back over the image and compute the magnitude of each edge point such that thresholding can be applied; edge points with higher gradient magnitude are more likely to be edges than those with a smaller magnitude. Thresholding with hysteresis, requiring the definition of two thresholds, is then used to identify the edges. A high threshold determines those pixels that are definitely an edge whereas a low threshold determines pixels that are also an edge if they are connected to any pixels above the high threshold. This is useful as low gradients often correspond to noise in images, however being connected to a high gradient pixel should increase the likelihood that it is
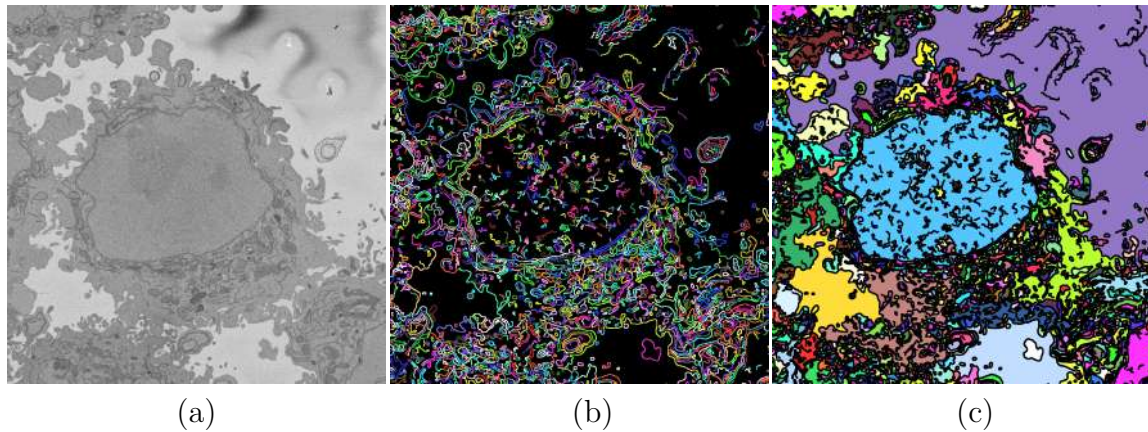
(a) (b) (c)

Fig. B.6 Edges obtained by Canny algorithm and superpixels created on HeLa cell. (a) One representative slice of a 3D stack imaged with Serial Blockface Electron Microscope (SBF SEM) with region of interest (ROI) of one HeLa cell centred. At this resolution it is easy to distinguish the nuclear envelope (NE), borders of the cell and other structures. The image was filtered with 5×5 Gaussian low pass filter with the standard deviation $\sigma = 2$. (b) Edges detected by Canny algorithm in order to detect abrupt changes of intensity. The edges were further dilated to connect those edges that may belong to the NE but were disjoint due to the variations of the intensity of the envelope itself. (c)The edges previously detected were dilated, removed from the image and the remaining regions were labelled to become superpixels.

part of an edge. An optional final step in Canny's algorithm is to perform the previous steps at a number of different spatial resolutions by changing the standard deviation of the Gaussian used to smooth the image, and to then compile the information from each resolution into the result. Whilst this can produce an improved edge detection, it is common for an implementation to choose just one value for the standard deviation based upon the objects in the image and omit this final step.

Figure B.6b shows edges of the same slice of HeLa cell image shown in Figure B.1 detected by Canny algorithm. The edges were further dilated to connect those edges that may belong to the NE but were disjoint due to the variations of the intensity of the envelope itself.

Post-processing

Post-processing is the set of operations performed on binary images which aid the segmentation to reduce noise or over-detection. In the context of segmentation of cells, some of the operations can aid in reducing noise, smoothing the edges of the shapes
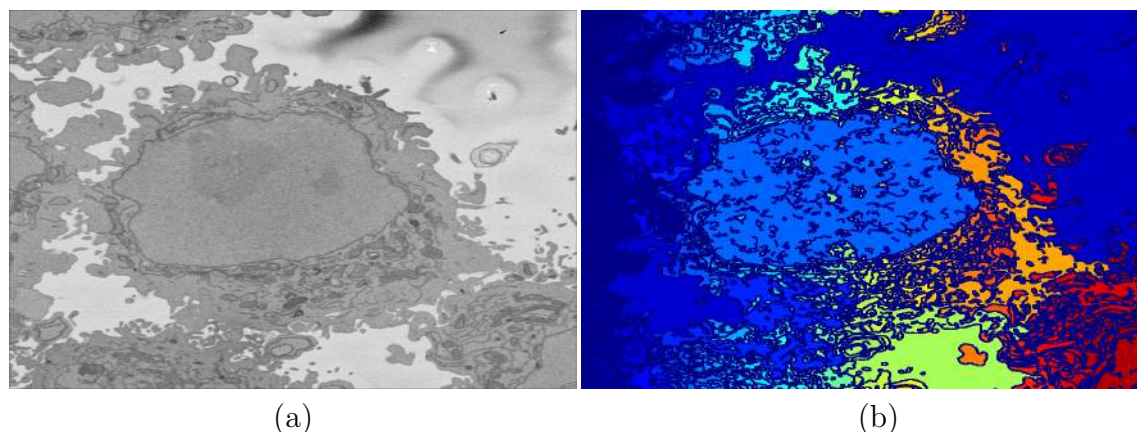
Fig. B.7 Illustration of superpixels of a HeLa cell image. (a) The original filtered HeLa cell image with the size of 2000×2000. There are 4000000 pixels in total in this image. (b) Superpixels obtained from the original image in (a). In order to obtain superpixels, the standard pixel grid was replaced by grouping pixels into primitive regions that are more perceptually meaningful than individual pixels.

and help give a more robust segmentation. Measurements can be performed on the binary images, like area and orientation.

In this section, some techniques of morphology are over-viewed. Morphology in this work refers to mathematical morphology, the branch of mathematics that deals with the representation and description of region shapes in a space [127].

As morphological operations are based on sets, in the next section, a description and basic ideas of set theory are presented. All notations and the operations shown in this section are used throughout the dissertation.

B.5   Superpixels

The idea behind superpixels is to replace the standard pixel grid by grouping pixels into primitive regions that are more perceptually meaningful than individual pixels [273] (Figure B.7. The objectives are to lessen computational load, and to improve the performance of segmentation algorithms by reducing irrelevant detail. Superpixels are a computationally efficient representation of the pixels in an image which take into account and have a better fit to the objects and structures within the image [274]. Pixels

are grouped into larger regions based on similarity of colour or texture, and savings come from calculating features for just superpixels and not all the pixels it contains. Little global information and fine detail is lost by moving to this representation as boundaries of structures of interest most often correspond to large changes in the colour or texture feature with which the superpixels were formed. The superpixel representation can come close to an accurate segmentation, but is most frequently used as a pre-processing step for a different segmentation algorithm. One technique for finding a fast superpixel segmentation is simple linear iterative clustering, or SLIC [275]. Its performance has been shown to be better than many other techniques for generating superpixels, and has an advantage of only requiring one parameter, the target number of superpixels in the output.

Figure B.6c shows superpixels created in the following way. The previously detected edges (Figure B.6b) were dilated and removed from the image, the remaining unconnected regions were labelled and became superpixels. There was no restriction in size of these superpixels as it was important to allow for large superpixels that corresponded to the nucleus and the background.

B.6   Set theory

In this section, the concept of a digital image remains, in which the number of levels is binary, $L = 2$, and normally the levels are integers [2]. Each pixel will take a value, 0 or 1 where contiguous areas of pixels with the same value will form areas and objects. In this work, the contiguous areas of pixels with a value of 1 will be considered to be objects or sets. This section is referred to when exploring the fundamentals of image segmentation and when discussing post-processing.

In the case of images, the entirety of the ordered pixels within it can be referred to as the universe as it contains all possible sets of pixels in it. Thus, each pixel $p = \mathbb{I}(x, y)$ will be an element of the universe. A collection of pixels, will be called a set or a region

Table B.1 Basic set operations. Let $\mathbb{A}, \mathbb{B}$ be sets within the universe $\mathbb{U}$, this table summarises the basic set theory operations and their definitions.

Name	Symbol	Definition
Union	$\mathbb{A} \cup \mathbb{B}$	$p \in \mathbb{A}$ OR $p \in \mathbb{B}$
Intersection	$\mathbb{A} \cap \mathbb{B}$	$p \in \mathbb{A}$ AND $p \in \mathbb{B}$
Complement	$\mathbb{A}^c$	$p \notin \mathbb{A}$
Difference	$\mathbb{A} - \mathbb{B}$	$p \in \mathbb{A} \cap \mathbb{B}^c$

within the image and represented by $\mathbb{A} \subset \mathbb{U}$. Table B.1 shows a brief description of the basic set operations.

In the context of images, pixel positions with a particular intensity - commonly 1 - will correspond to elements in the sets. Therefore, the regions in an image that constitute a set will be described by positions. In turn, this representation of sets allows for size to be measured in set as the number of elements in the image region.

Connected regions in binary images with a value of 1 can be modelled as sets in which the elements correspond to the pixel locations and the universe set corresponds to the entire collection, of pixels, i.e. the image $\mathbb{I}$.

Let the set $\mathbb{A}$ be a connected region in the image $\mathbb{I}$, then the set will be defined as the collection of points (x, y) in where $\mathbb{A} = \{(x, y) | \mathbb{I}(x, y) = 1\}$. Therefore, cardinality of a set $|\mathbb{A}|$, also referred to as size of the set, is defined by the number of connected pixels in a given region.

In previous section, the construction of filter kernels and the notion of convolution was explored in order to modify the intensities by assigning an operation performed on a neighbourhood of a pixel. To extend the notion into binary images and sets, some operations must be defined in the context of sets, and then translated into binary arrays to define each of the operations.

Structuring elements and notation. Sets are represented by connected regions in an image with the same value. As mentioned in the classic work by Gonzalez and Woods [[127], Chapter 10], a structuring element is a small set of sub-image used to probe an image under study for properties of interest. Such objects can be thought of
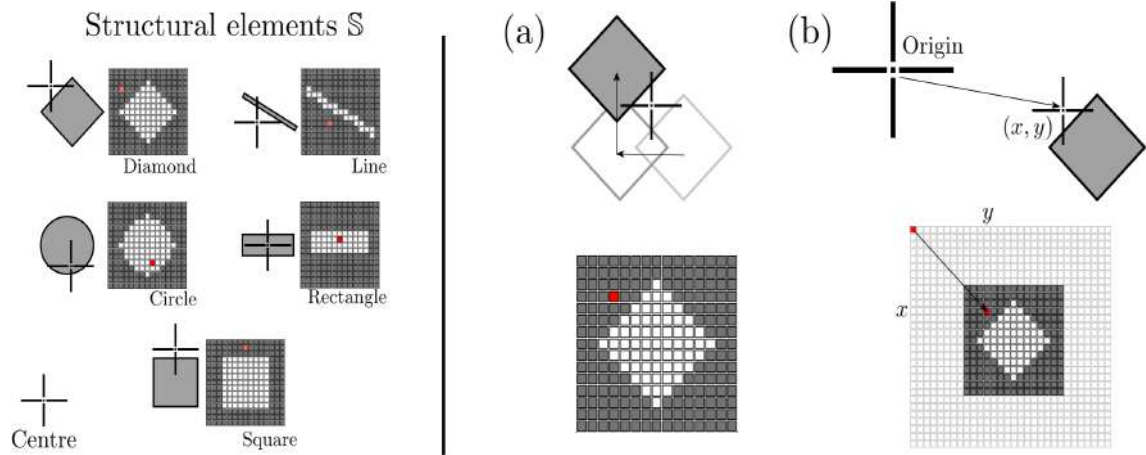
Fig. B.8 Representation of different structural elements in abstract terms and as a binary kernel $\mathbb{S}$. The diagram shows the abstract representation of a structuring element as a set with a particular shape. The notion of a centre refers to a point of reference in which the element can pivot and from which it can move in a plane. (a) Shows the operation of reflection while (b) shows translation. Notice that if the structural element is symmetric, and the centre is in the midpoint of the element, then $\mathbb{S} = \mathbb{S}$. Image credit to [2].

in abstract as sets with a centre, or reference point, or as a binary kernel, represented by the symbol $\mathbb{S}$.

B.7   Morphological operations

Mathematical Morphology is a theory which provides a number of useful tools for image analysis. Morphology is an approach to image analysis which is based on the assumption that an image consists of structures which may be handled by set theory [127].

As morphological operations are based on sets and so the fundamental objects are sets, set notation will be used in this chapter [128]. Sets are simply groups of pixels, and the terminology is just a convenient way of describing what pixels lie in particular groups. For morphology of binary images, the sets consist of pixels in an image.

In a binary image the pixels are either labelled 0 (displayed as black) or those labelled 1 (displayed as white) may comprise the sets of interest. Unless stated otherwise, we
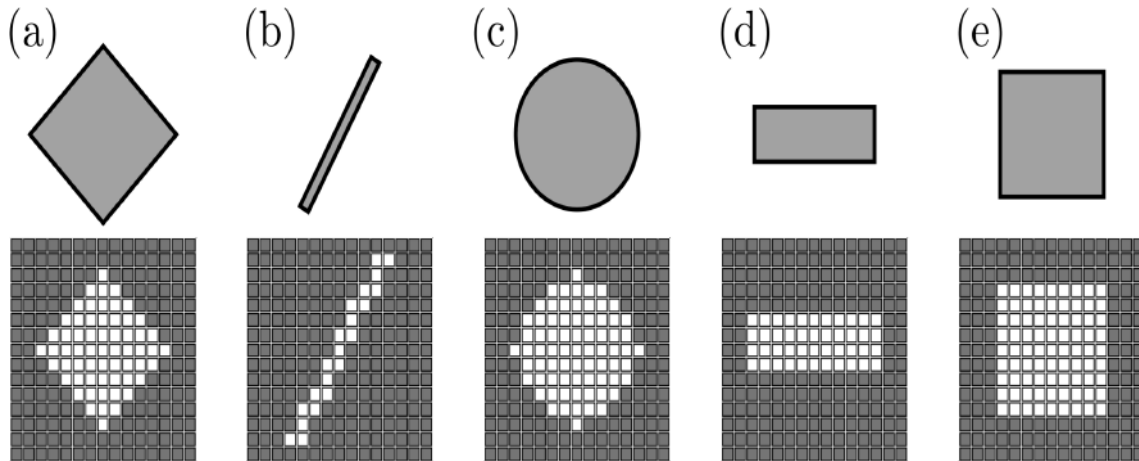
Fig. B.9 Examples of structural elements. Representations of structural elements in abstract form (top) and represented as discretisation in matrices (bottom). (a) Diamond. (b) Line. (c) Circle. (d) Rectangle. (e) Square. Image credit to [2].

will take the black pixels to be the sets of interest. When we refer to operations on the image, we shall be referring to operations on the set of all black pixels. Usually this will be the union of several separate sets of black pixels — what we think of as individual objects. The white pixels are the complement (Figure B.10e) of the black pixels, the complement of a set being the set of elements it does not contain. Any operation which affects the set of black pixels will also affect the set of white pixels. For example, removing a pixel from the set of black pixels naturally creates a new white pixel.

Morphological operations primarily operate on point sets, connectivity and shape rather than pixel intensity values and distributions [128]. They are typically used to simplify shapes, filter noise or modify the object structure in a meaningful and well defined manner. These operations also require a relation that identifies which pixels in the image are considered as objects and which as background. Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape [127]. Morphological operations are defined in terms of sets, set operations and relationships (Figure B.10). In image processing morphology is used with two types of sets of pixels: objects and structuring elements

Table B.2 Description of main morphological operations and equations in the context mathematical morphology. Let $\mathbb{I}$ be a binary image and $\mathbb{S}$ a structuring element of size $2n + 1$.

Name	Mathematical expression	Description
Erosion	$\mathbb{I} \ominus \mathbb{S} = \{(x,y)\|\mathbb{S}_{x,y} \subseteq \mathbb{I}\}$	Reduce the area of present regions in $\mathbb{I}$
Dilation	$\mathbb{I} \oplus \mathbb{S} = \left\{(x,y)\|\hat{\mathbb{S}}_{x,y} \cap \mathbb{I} \neq \emptyset\right\}$	Expands the area of present regions in $\mathbb{I}$.
Opening	$\mathbb{I} \bigcirc \mathbb{S} = (\mathbb{I} \ominus \mathbb{S}) \oplus \mathbb{S}$	Erosion followed by a dilation.
Closing	$\mathbb{I} \odot \mathbb{S} = (\mathbb{I} \ominus \mathbb{S}) \oplus \mathbb{S}$	Dilation followed by erosion.

(SE's). Typically, objects are defined as sets of foreground pixels. Structuring elements can be specified in terms of both foreground and background pixels.

In most cases, morphological operations are performed similarly to convolution, as a structuring element will be translated along a binary image, performing set operations as it moves [276]. In this section, the following operations will be defined: (i) erosion, (ii) dilation, (iii) opening and closing; as well as some algorithms like (iv) boundary extraction and (v) hole filling. For the coming definitions, let $\mathbb{S}$ be an image and $\mathbb{S}$ a structuring element of size $2n+1$, with a centre at position $(n+1, n+1)$. For simplicity, assume that $\mathbb{I}$ contains only one set — or detected object — in it. The operations will result in new sets, described by the elements in it. In terms of images, the positions that belong to a particular set will have a value of 1 in the resulting image.

Figure B.11 represents the application of the different morphological operations of a structural element $\mathbb{S}$ into a set with disjoint regions $\mathbb{A}$. The structural element present in the picture will be translated throughout the plane containing the set $\mathbb{A}$, and the output produced will follow the equations in Table B.2.

Morphological operations transform the image. The morphological transformation of an image requires the definition of an additional smaller SE expressed with respect to a local origin. The SE is moved systematically across the entire image with each pixel placed at the local origin of the element, and the output of the operation is stored in a separate output image. Typical SE include squares, circles and lines, but can also be arbitrarily shaped according to the application.
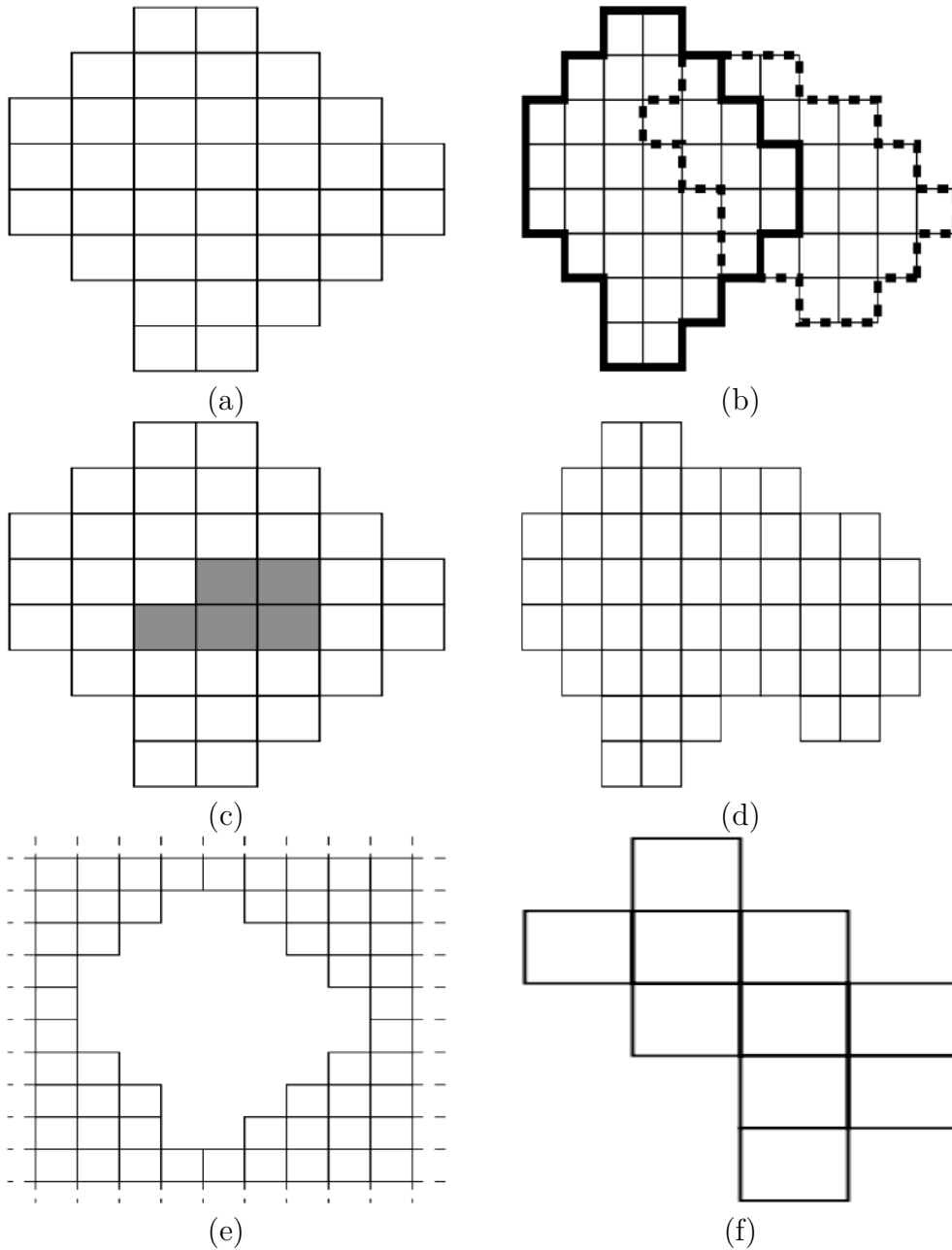
Fig. B.10 Basic ideas of set theory. (a) The set A. The pixel x is an element of A. (written x ∈ A); (b) Two overlapping sets, A and B; (c) Subsets: The shaded pixels (C) are a subset of A. (written C ⊂ A); (d) The union of A and B (written A ∪ B; (e) The complement of A (written A^c); (f) The intersection of A and B (written A ∩ B). Top row from left to right (a), (b), (c) and (d). Second row from left to right (e), (f), (g) and (h). (a) Venn diagrams corresponding to some of the set operations.(a) The sample space also called the set universe. In image processing Ω is defined to be the rectangle containing all the pixels in an image. (b) Complement of set A. (c) Intersection of set A and set B. (d) Union of set A and set B. (e) Difference between set A and set B. (f) Complement of set B. (g) Intersection of set A and complement of set B. (h) Intersection of set A and union of set B and set C. The results of the operations, such as A^c, are shown shaded. Figures (e) and (g) are the same, proving via Venn diagrams that A-B = A ∩ B^c. Images adapted from [127].

There are four primary morphological operations that can be defined based on this notion [128, 127]. The most basic morphological operation is that of erosion. The dual operator of a dilation is erosion, where an object pixel is only carried over to the output image if all object pixels in the SE are present as object pixels in the input image. An erosion can thus be seen as shrinking or reducing the input image according to the SE while dilation expands the components of a set. A dilation places the SE over each object pixel in the original image. Each pixel which is then defined as an object pixel by the SE being placed there is defined as an object pixel in the output image. Hence this can be seen as an increasing or growing transformation. Schematics of the erosion and dilation operations are shown in Figure B.11c, and Figure B.11d.

Two further morphological operators are opening and closing; an opening is an erosion followed by a dilation and a closing is a dilation followed by an erosion [127]. These operations in general are used to eliminate image details that are smaller than the SE without distorting the overall shape of objects in the input image [128]. The closing operator is useful for filling small holes, connecting close objects and smoothing the outline of objects, with the opening operator effectively performing the inverse by opening holes. Schematics of the opening and closing operations are shown in Figure B.11e and Figure B.11f.

Morphological operators are useful for refining regions that may have small holes due to noise, or for general manipulation of object and region shapes. Their extension from binary to grey scale images is simple using minimum and maximum operations. Erosion assigns the minimum intensity value found in the neighbourhood of a pixel in the input image, and dilation the maximum value. The SE in this case simply defines the neighbourhood over which intensity values are considered.

Combinations of techniques like in morphological opening, could be useful, as the erosion could reduce granular noise and the subsequent dilation would grow the area of the detected object and close certain gaps between disjoint sections. An illustration of opening and closing operations outputs when applying on a HeLa cell superpixels is shown in Figure B.12b,c.
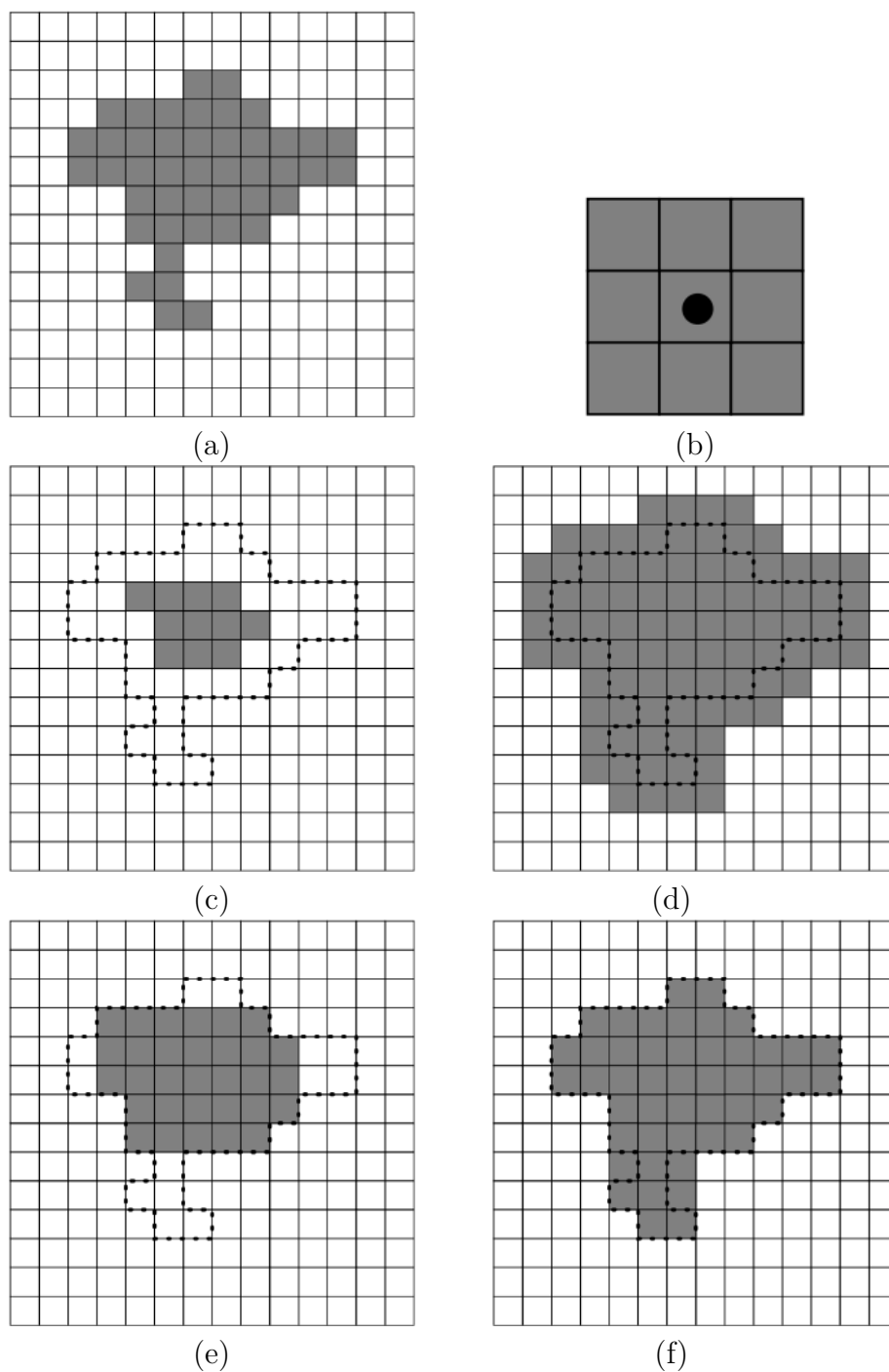
Fig. B.11 Illustration of basic morphological operations: (a) Original set, extracted as a subset from the turbinate image, (b) structuring element: a square of side 3. The reference pixel is at the centre. (c) erosion, (d) dilation, (e) opening and (f) closing. Images adapted from [127].
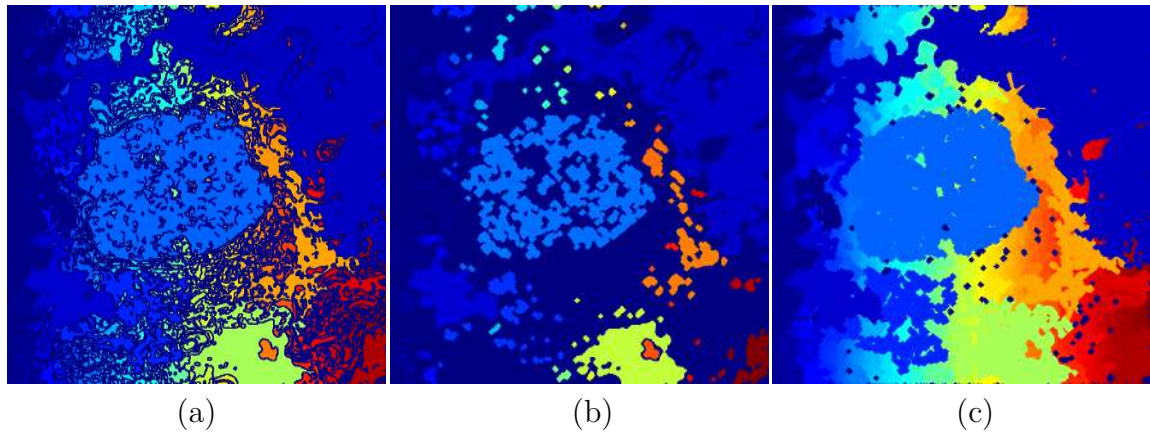
(a)   (b)   (c)

Fig. B.12 Illustration of opening and closing operations outputs when applying on a HeLa cell superpixels. (a) Original set of superpixels extracted from HeLa image through image processing algorithm. A diamond-shaped structuring element kernel $\mathbb{S}$ with r = 15 (where r specifies the distance from the structuring element origin to the points of the diamond) was used to open and close the superpixels of a HeLa cell image. (b) opening and (c) closing.

B.8   Active contours (Snakes)

Active contours or snakes [277] area completely different approach to feature extraction. An active contour is a set of points which aims to enclose a target feature [278]. A good analogy also taken from is that of a balloon placed around a shape so as to enclose it fully. If enough air is taken out of the balloon it will surround the object that it encloses; active contours aim to describe a shape by enclosing it in this manner. A snake is defined as an energy-minimising spline, and its energy is defined by its shape, location in the image and certain other image properties. The minimisation task is to find local minima of energy which then correspond to desired image features.

The snake is initialised at some position in the image and is subsequently deformed to match the nearest contour. The definition of energy depending on the shape of the snake limits its applicability to recognising shapes that are intrinsically smooth in their shape. Shapes that have sharp changes of direction in their border may struggle to be matched exactly by active contours. Further limitations upon general purpose use arise from requiring user input to position the initial snakes; active contours are a deformable model of the shape, so some segmentation is already done by a user in

order for it to be refined using this method. The outcome of the model will therefore also be sensitive to how it was initialised by the user.

Snakes are actually quite recent compared with many computer vision techniques and their original formulation was as an interactive extraction process, though they are now usually deployed for automatic feature extraction.

Active contours are actually expressed as an energy minimisation process. The target feature is a minimum of a suitably formulated energy functional. This energy functional includes more than just edge information: it includes properties that control the way the contour can stretch and curve. In this way, a snake represents a compromise between its own properties (like its ability to bend and stretch) and image properties (like the edge magnitude). Accordingly, the energy functional is the addition of a function of the contour's internal energy, its constraint energy, and the image energy: these are denoted $E_{\text{int}}, E_{\text{con}}$ and E_{image}, respectively. These are functions of the set of points which make up a snake, $\mathbf{v}(s)$, which is the set of x and y coordinates of the points in the snake. The energy functional is the integral of these functions of the snake, given $s \in [0, 1)$ is the normalised length around the snake. The energy functional E_{snake} is then

$$E_{\text{snake}} = \int_{s=0}^{1} E_{\text{int}}\left(\mathbf{v}(s)\right) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s))ds$$

In this equation: the internal energy, E_{int}, controls the natural behaviour of the snake and hence the arrangement of the snake points; the image energy, E_{image}, attracts the snake to chosen low-level features (such as edge points) and the constraint energy, E_{con} allows higher level information to control the snake's evolution. The aim of the snake is to evolve by minimising Eq. (6.8). New snake contours are those with lower energy and are a better match to the target feature (according to the values of $E_{\text{int}}, E_{\text{image}}$ and E_{con}) than the original set of points from which the active contour has evolved.

In this manner, we seek to choose a set of points $\mathbf{v}(s)$ such that

$$\frac{dE_{snake}}{d\mathbf{v}(s)} = 0$$

This can of course select a maximum rather than a minimum, and a second order derivative can be used to discriminate between a maximum and a minimum. However, this is not usually necessary as a minimum is usually the only stable solution (on reaching a maximum, it would then be likely to pass over the top to then minimise the energy). Prior to investigating how we can minimise Eq. (6.8), let us first consider the parameters which can control a snake's behaviour.

The energy functionals are expressed in terms of functions of the snake, and of the image. These functions contribute to the snake energy according to values chosen for respective weighting coefficients. In this manner, the internal image energy is defined to be a weighted summation of first- and second-order derivatives around the contour.

$$E_{int} = \alpha(s) \left| \frac{d\mathbf{v}(s)}{ds} \right|^2 + \beta(s) \left| \frac{d^2\mathbf{v}(s)}{ds^2} \right|^2$$

The first-order differential, $d\mathbf{v}(s)/ds$, measures the energy due to stretching which is the elastic energy since high values of this differential imply a high rate of change in that region of the contour. The second-order differential, $d^2\mathbf{v}(s)/ds^2$, measures the energy due to bending, the curvature energy. The first-order differential is weighted by $\alpha(s)$ which controls the contribution of the elastic energy due to point spacing; the second-order differential is weighted by $\beta(s)$ which controls the contribution of the curvature energy due to point variation. Choice of the values of α and β controls the shape the snake aims to attain. Low values for α imply the points can change in spacing greatly, whereas higher values imply that the snake aims to attain evenly spaced contour points. Low values for β imply that curvature is not minimised and the contour can form corners in its perimeter, whereas high values predispose the snake to smooth contours. These are the properties of the contour itself, which is just part of a snake's compromise between its own properties and measured features in an image.

The image energy attracts the snake to low-level features, such as brightness or edge data, aiming to select those with least contribution. The original formulation suggested that lines, edges and terminations could contribute to the energy function. Their energy is denoted $E_{\text{line}}, E_{\text{edge}}$ and E_{term}, respectively, and are controlled by weighting coefficients $w_{\text{line}}, w_{\text{edge}}$ and w_{term}, respectively. The image energy is then

$$E_{\text{image}} = w_{\text{line}}E_{\text{line}} + w_{\text{edge}}E_{\text{edge}} + w_{\text{term}}E_{\text{term}}$$

The line energy can be set to the image intensity at a particular point. If black has a lower value than white, then the snake will be extracted to dark features. Altering the sign of w_{line} will attract the snake to brighter features. The edge energy can be that computed by application of an edge detection operator, the magnitude, say, of the output of the Sobel edge detection operator. The termination energy, E_{term} as measured by Eq. (4.52) can include the curvature of level image contours (as opposed to the curvature of the snake, controlled by $\beta(s)$), but this is rarely used. It is most common to use the edge energy, though the line energy can find application.

The technique which many people compare the result of their own new approach with is a GAC called the active contour without edges, introduced by Chan and Vese [210], which is based on the Mumford Shah functional. Their model uses regional statistics for segmentation, and as such is a region-based level set model. The overall premise is to avoid using gradient (edge) information since this can lead to boundary leakage and cause the contour to collapse. A further advantage is that it can find objects when boundary data is weak or diffuse. The main strategy is to minimise energy, as in an active contour. The active contour without edges model can address problems with initialisation, noise, and boundary leakage (since it uses regions, not gradients) but still suffers from computational inefficiency and difficulty in implementation because of the level set method.

The process is shown in Figure B.13 where the target feature is the perimeter of the NE. First, an initial contour is placed inside the NE (Figure B.13a). The contour
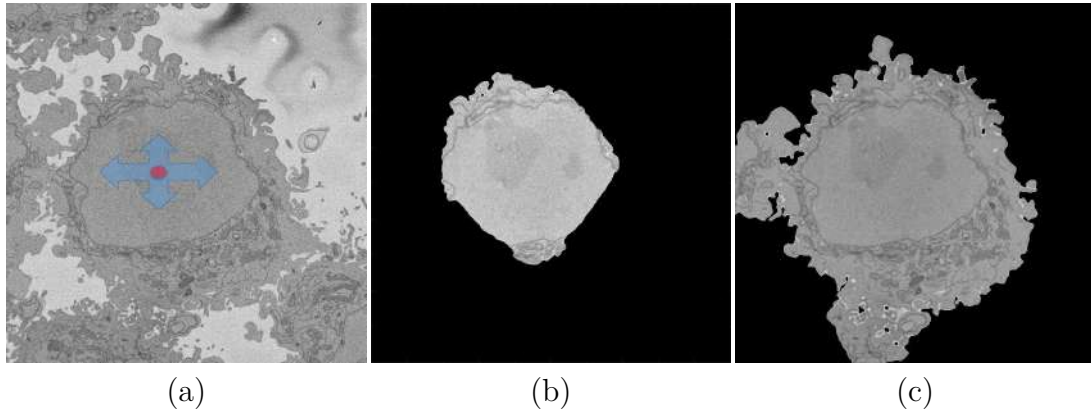
(a) (b) (c)

Fig. B.13 Effect of active contours (snakes) to find the nuclear envelope of HeLa cell. Using a circular snake to find the nuclear envelope (NE) of a HeLa cell in one of the 300 slices. An initial contour (a small circle) is placed inside the target feature which is the perimeter of the NE and the contour is then evolved so as to enclose the NE. The contour is then maximized to find a new contour which grows so as to be closer to the NE. (a) One representative slice from one HeLa cell. (b) Active contours segmentation with 2200 iterations. The contour points start to match the NE perimeter. (d) Active contours segmentation with 5000 iterations included large sections of the cell showing the influence of the iterations on the result.

is then maximised to find a new contour which grows so as to be closer to the NE. (Figure B.13b) shows active contours segmentation with 2200 iterations. The contour points start to match the NE perimeter. After five thousand iterations, the contour points can be seen to match the NE perimeter well (Figure B.13c).

Appendix C

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs), as a part of the deep learning techniques has become really popular amongst researchers because of the ease of use and relatively easy implementation therefore in this appendix basics of CNNs will be presented.

C.1 Convolutional neural networks

In this part, we will introduce convolutional neural network (CNN), which is the most common type of deep neural networks for image analysis. CNN have been successfully applied to advance the state of-the-art on many image classification, object detection and segmentation tasks.

We first describe what convolution is. Next, we explain the motivation behind using convolution in a neural network. We then describe some operations called pooling, ReLU, and Fully Connected Layers which almost all convolutional networks employ.

The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution [68]. Convolution is a specialised kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

The convolution operation

A convolution is defined as an operation on two functions. In image analysis, one function consists of input values (e.g. pixel values) at a position in the image, and the second function is a filter (or kernel); each can be represented as array of numbers. Computing the dot product between the two functions gives an output. The filter is then shifted to the next position in the image as defined by the stride length. The computation is repeated until the entire image is covered, producing a feature (or activation) map. Figure C.1.

The convolution operation is defined by the $\otimes$ symbol. An output (or feature map) s(t) is defined below when input I(t) is convolved with a filter or kernel K(a).

Let $f(x, y)$ be an image of size $N_r \times N_c$, and let $\mathbb{K}$ be a filter kernel, i.e. a small square matrix of size $n^2, n << \min(N_r, N_c)$ with n odd $(n = 2a + 1)$. Then the convolution $g(x, y) = f \otimes \mathbb{K}$ is given by the following equation,

$$g(x, y) = \sum_{i=-a}^{a} \sum_{j=-a}^{a} \mathbb{K}(i, j) f(x + i, y + j)$$

Notice that convolution can be represented mathematically as in the equation, with the operator $\otimes$, or in MATLAB® notation, imfilt($\mathbb{I}, \mathbb{K}$). Neural networks implement the cross-correlation function, which is the same as convolution but without flipping the kernel.

In convolutional network terminology, the first argument to the convolution is often referred to as the input, and the second argument as the kernel [68]. The output is sometimes referred to as the feature map. This is a map of where the filter is strongly activated and 'sees' a feature such as a straight line, a dot, or a curved edge. If a photograph of a face was fed into a CNN, initially low-level features such as lines and edges are discovered by the filters. These build up to progressively higher features in subsequent layers, such as a nose, eye or ear, as the feature maps become inputs for the next layer in the CNN architecture.
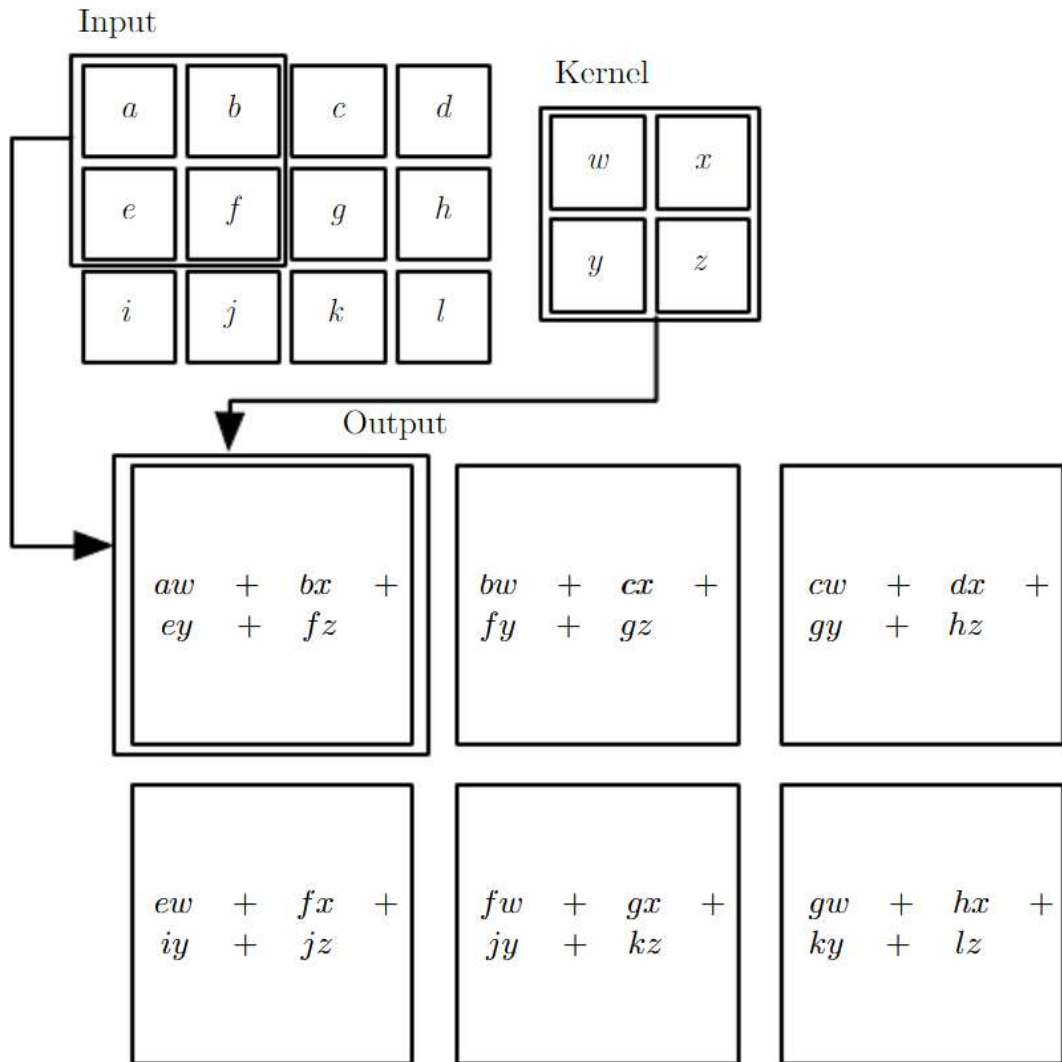
Fig. C.1 An example of 2D convolution without kernel flipping. We restrict the output to only positions where the kernel lies entirely within the image, called "valid" convolution in some contexts. We draw boxes with arrows to indicate how the upper-left element of the output tensor is formed by applying the kernel to the corresponding upper-left region of the input tensor. This is a simplified diagram of a 2D convolution of an image and a 2×2 kernel. In this case, the kernel, $\mathbb{K}$, is represented by a 2×2 matrix as it moves onto different parts of the image. The position of the origin in $\mathbb{K}$ will be the position in the resulting image. Images adapted from [68]

The input is usually a multidimensional array of data, and the kernel is usually a multidimensional array of parameters that are adapted by the learning algorithm. Finally, we often use convolutions over more than one axis at a time. For example, if we use a two-dimensional image as our input, we probably also want to use a two-dimensional kernel.

In the context of machine learning, the learning algorithm will learn the appropriate values of the kernel in the appropriate place, so an algorithm based on convolution with kernel flipping will learn a kernel that is flipped relative to the kernel learned by an algorithm without the flipping. It is also rare for convolution to be used alone in machine learning; instead convolution is used simultaneously with other functions, and the combination of these functions does not commute regardless of whether the convolution operation flips its kernel or not.

Figure C.1 shows an example of convolution (without kernel flipping) applied to a 2-D tensor. Discrete convolution can be viewed as multiplication by a matrix, but the matrix has several entries constrained to be equal to other entries. For example, for uni-variate discrete convolution, each row of the matrix is constrained to be equal to the row above shifted by one element. This is known as a Toeplitz matrix. In two dimensions, a doubly block circulant matrix corresponds to convolution. In addition to these constraints that several elements be equal to each other, convolution usually corresponds to a very sparse matrix (a matrix whose entries are mostly equal to zero). This is because the kernel is usually much smaller than the input image. Any neural network algorithm that works with matrix multiplication and does not depend on specific properties of the matrix structure should work with convolution, without requiring any further changes to the neural network. Typical convolutional neural networks do make use of further specialisations in order to deal with large inputs efficiently, but these are not strictly necessary from a theoretical perspective.

Convolution exploits three ideas intrinsic to perform computationally efficient machine learning: sparse connections, parameter sharing (or weights sharing) and equi-variant (or invariant) representation. Unlike some neural networks where every

input neuron is connected to every output neuron in the subsequent layer, CNN neurons have sparse connections, meaning that only some inputs are connected to the next layer. By having a small, local receptive field (i.e., the area covered by the filter per stride), meaningful features can be gradually learnt, and the number of weights to be calculated can be drastically reduced, increasing the algorithm's efficiency. In using each filter with its fixed weights across different positions of the entire image, CNNs reduce memory storage requirements. This is known as parameter sharing.

Parameter sharing results in the quality of equi-variant representation to arise. This means that input translations result in a corresponding feature map translation.

By far the most popular and extensive use of parameter sharing occurs in convolutional neural networks (CNNs) applied to computer vision. Natural images have many statistical properties that are invariant to translation. For example, a photo of a cat remains a photo of a cat if it is translated one pixel to the right. CNNs take this property into account by sharing parameters across multiple image locations. The same feature (a hidden unit with the same weights) is computed over different locations in the input. This means that we can find a cat with the same cat detector whether the cat appears at column i or column $i + 1$ in the image. Parameter sharing has enabled CNNs to dramatically lower the number of unique model parameters and to significantly increase network sizes without requiring a corresponding increase in training data. It remains one of the best examples of how to effectively incorporate domain knowledge into the network architecture.

Convolutional Neural Networks (CNNs), as a part of the deep learning techniques has become really popular amongst researchers because of the ease of use and relatively easy implementation [68]. In general, CNNs require a large amount of data, and because of the numbers of parameters in all the layers, the processing of each one has increased significantly. CNNs are a specialised kind of neural network for processing data that has a known grid-like topology. Examples include time-series data, which can be thought of as a 1D grid taking samples at regular time intervals, and image data, which can be thought of as a 2D grid of pixels. Convolutional networks have been

tremendously successful in practical applications have played an important role in the history of deep learning [70]. They were some of the first deep models to perform well, long before arbitrary deep models were considered viable. Convolutional networks were also some of the first neural networks to solve important commercial applications and remain at the fore front of commercial applications of deep learning today. For example, in the 1990s, the neural network research group at AT&T developed a convolutional network for reading checks [279]. By the end of the 1990s, this system deployed by NCR was reading over 10% of all the checks in the United States. Later, several Optical Character Recognition (OCR) and handwriting recognition systems based on convolutional nets were deployed by Microsoft [280].

Convolutional networks were also used to win many contests. The current intensity of commercial interest in deep learning began when Krizhevsky et al. won the ImageNet object recognition challenge, but convolutional networks had been used to win other machine learning and computer vision contests with less impact for years earlier.

Convolutional networks were some of the first working deep networks trained with back-propagation. It is not entirely clear why convolutional networks succeeded when general back-propagation networks were considered to have failed. It may simply be that convolutional networks were more computationally efficient than fully connected networks, so it was easier to run multiple experiments with them and tune their implementation and hyper-parameters. Larger networks also seem to be easier to train. With modern hardware, large fully connected networks appear to perform reasonably on many tasks, even when using data sets that were available and activation functions that were popular during the times when fully connected networks were believed not to work well. It may be that the primary barriers to the success of neural networks were psychological (practitioners did not expect neural networks to work, so they did not make a serious effort to use neural networks). Whatever the case, it is fortunate that convolutional networks performed well decades ago. In many ways, they carried the torch for the rest of deep learning and paved the way to the acceptance of neural networks in general.

Currently, CNNs are the most researched machine learning algorithms in medical image analysis [281]. The reason for this is that CNNs preserve spatial relationships when filtering input images.

An architecture of Convolutional Neural Networks (CNNs) is shown in Figure C.2. Convolutional Neural Networks (CNNs) preserve spatial structure. The image dimension is $32 \times 32 \times 3$ and a filter with the same depth of the image is used ($5 \times 5 \times 3$). We convolve the filter with the image - "slide over the image spatially, computing dot products." Filters always extend the full depth of the input volume. The filter is placed at the upper left hand corner and centred on top of every pixel in the input volume. At every position a dot product is obtained and this produces one value in the activation map. Then the filter is slid around and the next dot product produces another value in the activation map. At every pixel this operation is done and the corresponding point in the activation is filled in. Depending on how we choose to slide the filter we get different size outputs.

A CNN takes an input image of raw pixels, and transforms it via Convolutional Layers, Rectified Linear Unit (RELU) Layers and Pooling Layers. This feeds into a final fully Connected Layer which assigns class scores or probabilities, thus classifying the input into the class with the highest probability. In the next subsections these layers will be explained in detail.

CNNs are a class of deep feed-forward artificial neural networks for processing data that has a known grid-like topology [282]. They emerged from the study of the brain's visual cortex and benefited from the recent increase in the computational power and the amount of available training data. A CNN is generally composed of five types of neuron layers [79]:

The input layer. The neurons in this layer hold the raw pixel values of the input image. In the case of an RGB image of size $w \times h$, this layer is a 3D volume of size $w \times h \times 3$.

The convolution layer. Each neuron in this layer computes the dot product between their weights and a small region in the input volume to which they are

32x32x3 image

5x5x3 filter w

32

32

3

1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

(a)

activation map

32x32x3 image

5x5x3 filter

32

32

3

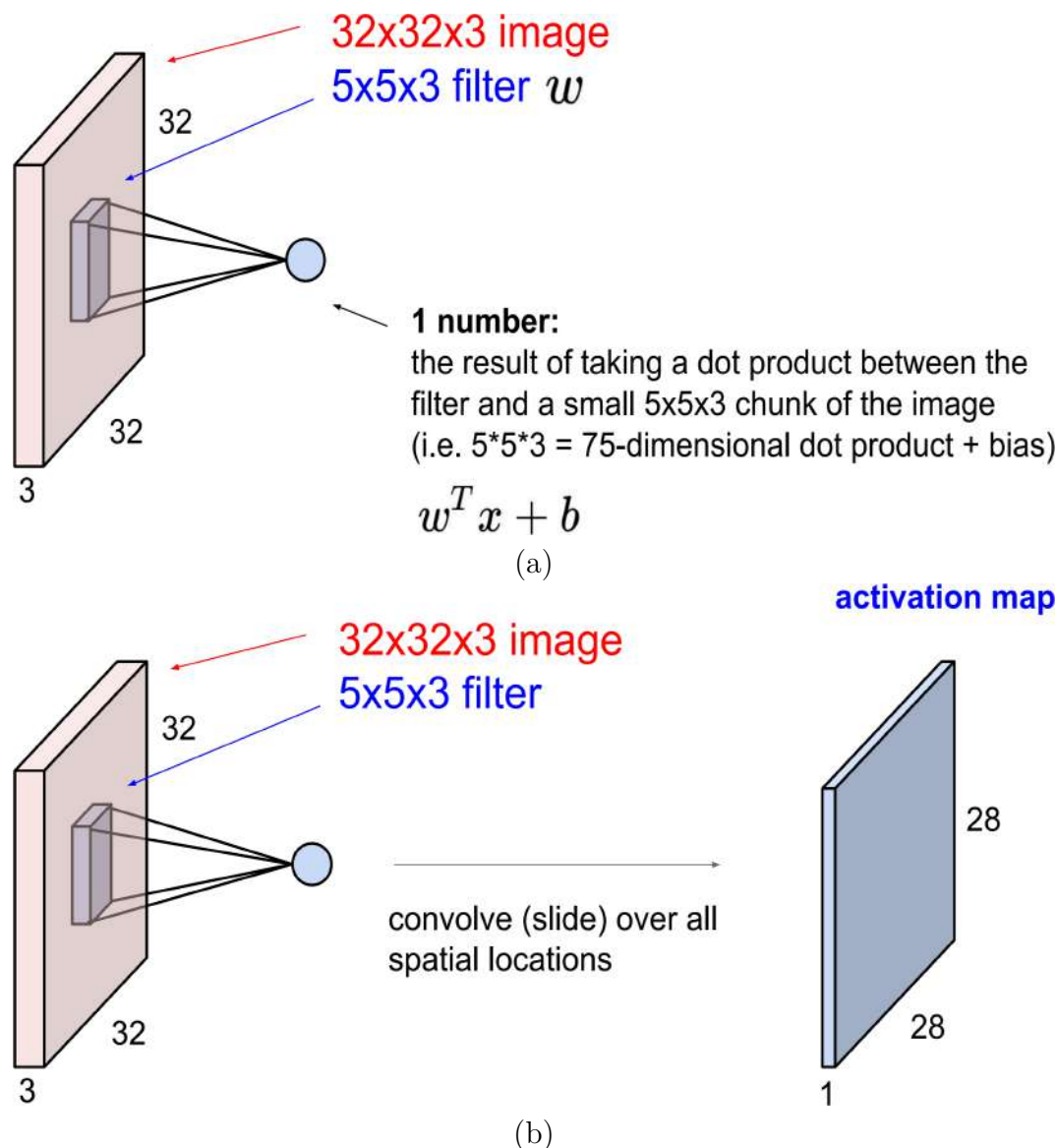convolve (slide) over all spatial locations

28

28

1

(b)

Fig. C.2 Architecture of Convolutional Neural Networks (CNNs). Convolutional Neural Networks (CNNs) preserve spatial structure. a) The image dimension is $32 \times 32 \times 3$ and a filter with the same depth of the image is used $(5 \times 5 \times 3)$. We convolve the filter with the image, i.e. "slide over the image spatially, computing dot products." Filters always extend the full depth of the input volume. b) The filter is placed at the upper left hand corner and centred on top of every pixel in the input volume. At every position a dot product is obtained and this produces one value in the activation map. Then the filter is slid around and the next dot product produces another value in the activation map. At every pixel this operation is done and the corresponding point in the activation is filled in. Depending on how we choose to slide the filter we get different size outputs. Image credit to [187]

.

connected. This is equivalent to performing a convolution of the input with the filter whose elements are the weights of the neuron.

The ReLU layer. This applies an element-wise activation function, e.g. the max(0, x), to its input.

The Pool layer. This layer down-samples the input along the spatial dimensions.

The Fully-Connected (FC) layer. This is a classification layer. It computes the class scores, i.e. the likelihood that the input to the network belongs to one of the classes.

C.1.1 Activation Functions in CNN

There are different kind of activation functions used in CNNs. Figure C.3 illustrate some of these activation functions.

Sigmoid : Squashes numbers to range [0,1], historically popular since they have nice interpretation as a saturating "firing rate" of a neuron with 3 problems: 1. Saturated neurons "kill" the gradients, 2. Sigmoid outputs are not zero-centered, 3. exp() is a bit compute expensive.

tanh: Squashes numbers to range [-1,1] and zero centred but still kills gradients when saturated.

ReLU: Does not saturate (in +region), very computationally efficient, converges much faster than sigmoid/tanh in practice. Actually more biologically plausible than sigmoid but not zero-centred output.

The ReLU layer is an activation function that sets negative input values to zero. This simplifies and accelerates calculations and training, and helps to avoid the vanishing gradient problem. Mathematically it is defined as:

Computes $f(x) = max(0, x)$

where x is the input to the neuron. Other activation functions include the sigmoid, tanh, leaky RELUs, Randomized RELUs and parametric RELUs.

Leaky ReLU: Does not saturate, computationally efficient, converges much faster than sigmoid/tanh in practice, and will not "die".

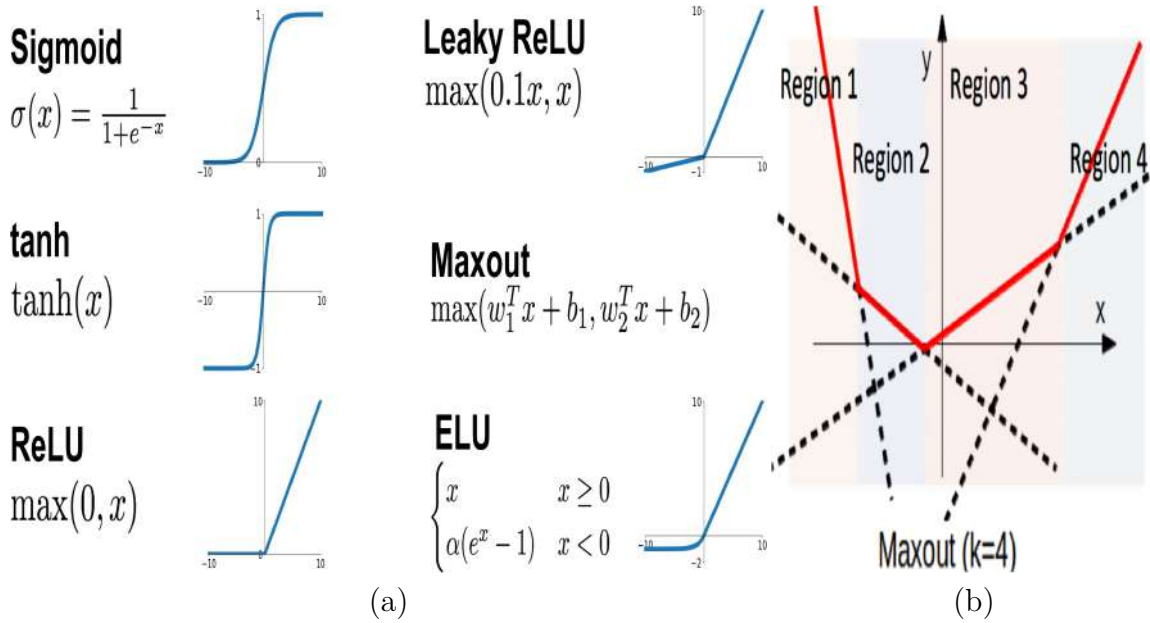(a)   (b)

Fig. C.3 (a) Different types of activations functions used in Convolutional Neural Networks (CNNs). (b) The curve of maxout. The maxout model is simply a feedforward architecture, such as a multilayer perceptron or deep convolutional neural network, that uses a new type of activation function: the maxout unit. Maxout activation functions are very robust and easy to train with dropout, and achieve excellent performance. Images credit to [187] and [283].

ELU: It has all benefits of ReLU, closer to zero mean outputs, Negative saturation regime compared with Leaky ReLU, adds some robustness to noise but computation requires exp().

C.1.2 Pooling layer

A typical layer of a convolutional network consists of three stages [68]. In the first stage, the layer performs several convolutions in parallel to produce a set of linear activations. In the second stage, each linear activation is run through a nonlinear activation function, such as the rectified linear activation function (ReLU). This stage is sometimes called the detector stage. In the third stage, we use a pooling function to modify the output of the layer further (Figure C.4a). A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs. For example, the max pooling (Figure C.4b) operation reports the maximum output

within a rectangular neighbourhood. Other popular pooling functions include the average of a rectangular neighbourhood, the L2 norm of a rectangular neighbourhood, or a weighted average based on the distance from the central pixel. In all cases, pooling helps to make the representation approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change. Invariance to local translation can be a useful property if we care more about whether some feature is present than exactly where it is. For example, when determining whether an image contains a face, we need not know the location of the eyes with pixel-perfect accuracy, we just need to know that there is an eye on the left side of the face and an eye on the right side of the face.

In other contexts, it is more important to preserve the location of a feature. For example, if we want to find a corner defined by two edges meeting at a specific orientation, we need to preserve the location of the edges well enough to test whether they meet. The use of pooling can be viewed as adding an infinitely strong prior that the function the layer learns must be invariant to small translations. When this assumption is correct, it can greatly improve the statistical efficiency of the network. Pooling over spatial regions produces invariance to translation, but if we pool over the outputs of separately parametrised convolutions, the features can learn which transformations to become invariant to. Because pooling summarises the responses over a whole neighbourhood, it is possible to use fewer pooling units than detector units, by reporting summary statistics for pooling regions spaced k pixels apart rather than 1 pixel apart. This improves the computational efficiency of the network because the next layer has roughly k times fewer inputs to process. When the number of parameters in the next layer is a function of its input size (such as when the next layer is fully connected and based on matrix multiplication), this reduction in the input size can also result in improved statistical efficiency and reduced memory requirements for storing the parameters.

For many tasks, pooling is essential for handling inputs of varying size. For example, if we want to classify images of variable size, the input to the classification layer must have a fixed size. This is usually accomplished by varying the size of an offset between pooling regions so that the classification layer always receives the same number of summary statistics regardless of the input size. For example, the final pooling layer of the network may be defined to output four sets of summary statistics, one for each quadrant of an image, regardless of the image size.

The Pooling layer is inserted between the Convolution and ReLU layers to reduce the number of parameters to be calculated, as well as the size of the image (width and height, but not depth). Max-pooling is most commonly used; other pooling layers include "average pooling" and "L2-normalisation" pooling. Max-pooling simply takes the largest input value within a filter and discards the other values; effectively it summarises the strongest activation over a neighbourhood. The rationale is that the relative location of a strongly activated feature to another is more important than its exact location.

C.1.3 Fully connected (FC) layer

As shown in Figure C.6, a standard CNN consists of an input layer, an output layer and a stack of functional layers in between that transform an input into an output in a specific form (e.g. vectors) [70]. These functional layers often contains convolutional layers, pooling layers and/or fully-connected layers. In general, each convolution uses a n × n kernel (for 2D input) or n × n × n kernel (for 3D input) followed by batch normalisation [202] after which the output is passed through a nonlinear activation function (e.g. rectified linear unit (ReLU)), which is used to extract feature maps from an image. These feature maps are then down-sampled by pooling layers, typically by a factor of 2, which removes redundant features to improve the statistical efficiency and model generalisation. After that, fully connected layers are applied to reduce the dimension of features and find the most task-relevant features for inference. The output of the network is a fix-sized vector where each element can be a probabilistic score for
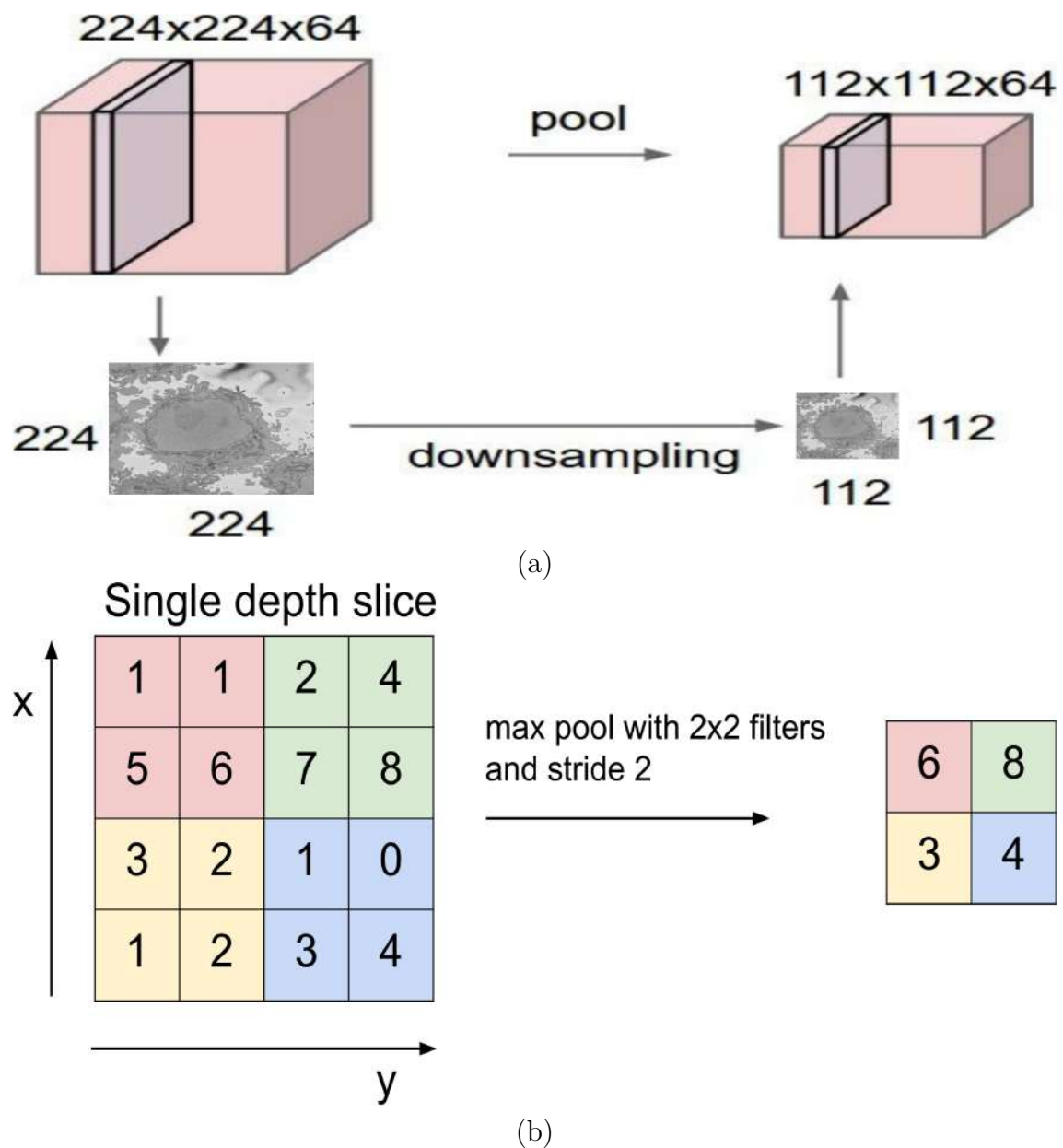
(a)

(b)

Fig. C.4 Pooling layer is used in Convolutional Neural Networks (CNNs) to reduce the number of parameters to be calculated, as well as the size of the image (width and height, but not depth). (a) Pooling makes the representations smaller and more manageable and it operates over each activation map independently. (b) Max-pooling simply takes the largest input value within a filter and discards the other values; effectively it summarises the strongest activation over a neighbourhood. Image credit to [187].
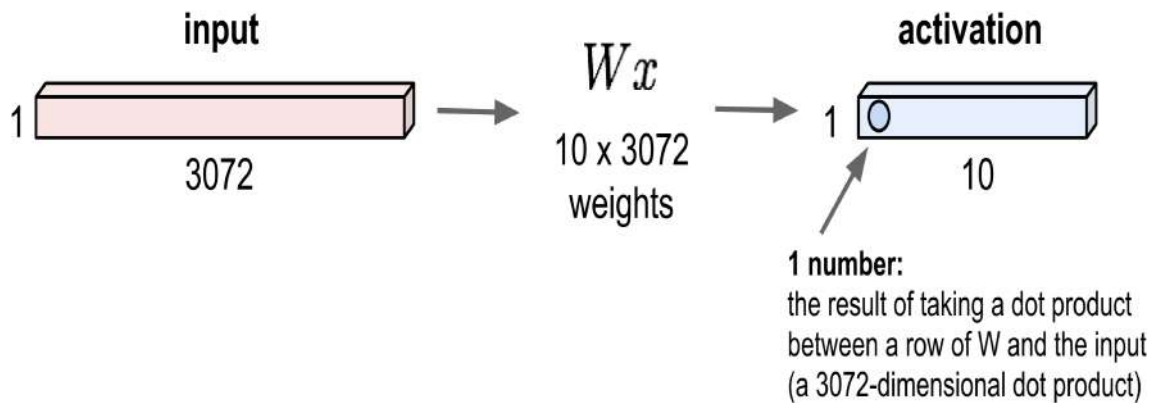
Fig. C.5 Architecture of Fully Connected Layer (FCN) in Convolutional Neural Networks (CNNs). FCN contains neurons that connect to the entire input volume, as in ordinary Neural Networks. This layer takes the output from the preceding layer (Convolutional, ReLU or Pooling) as its input, and computes a probability score for classification into the different available classes. In essence, this layer looks at the combination of the most strongly activated features that would indicate the image belongs to a particular class. Image credit to [187].

each category (for image classification), a real value for a regression task or a set of values (e.g. the coordinates of a bounding box for object detection and localisation).

The final layer in a CNN is the Fully Connected Layer (FCN), meaning that every neuron in the preceding layer is connected to every neuron in the fully Connected Layer [68]. Like the convolution, ReLU and pooling layers, there can be 1 or more fully connected layers depending on the level of feature abstraction desired. This layer takes the output from the preceding layer (Convolutional, ReLU or Pooling) as its input, and computes a probability score for classification into the different available classes. In essence, this layer looks at the combination of the most strongly activated features that would indicate the image belongs to a particular class. FCN contains neurons that connect to the entire input volume, as in ordinary Neural Networks. Figure C.5 shows an input image with dimensions $32 \times 32 \times 3$ is stretched to 3072×1.
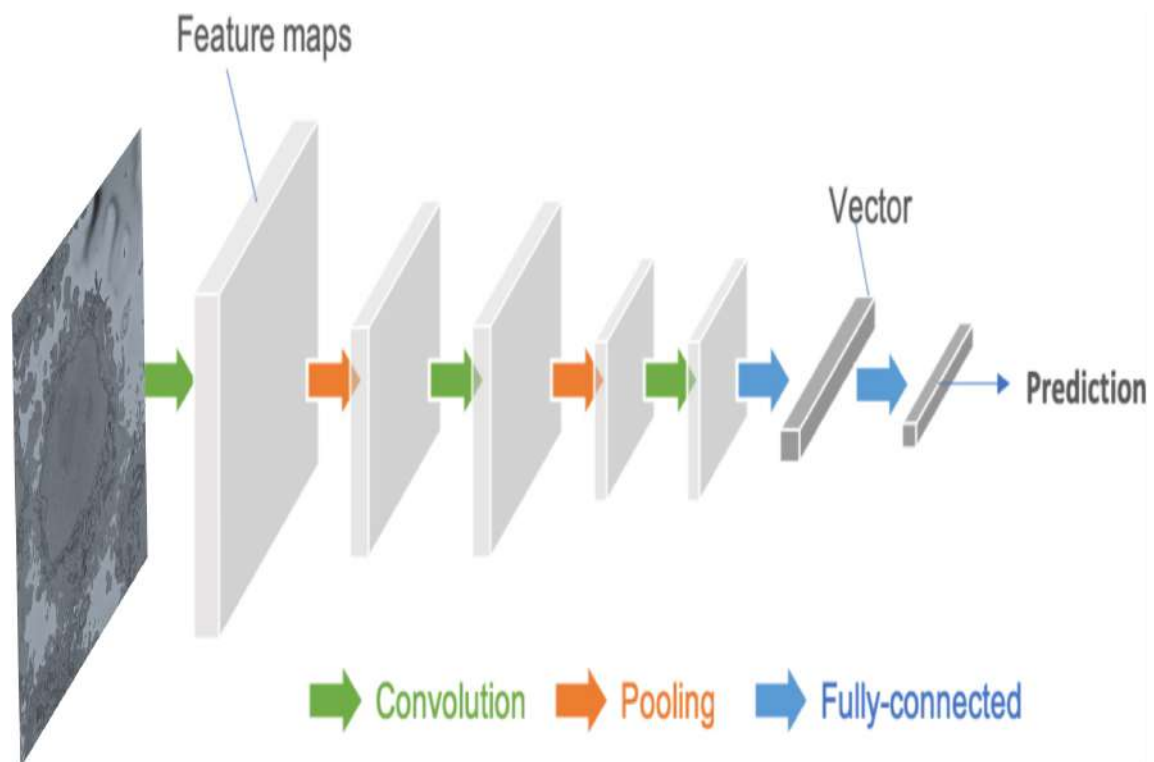
Fig. C.6 Architecture of Convolutional Neural Networks (CNNs). A CNN takes a HeLa image as input, learning hierarchical features through a stack of convolutions and pooling operations. These spatial feature maps are then flattened and reduced into a vector through fully connected layers (FCN). This vector can be in many forms, depending on the specific task. It can be probabilities for a set of classes (image classification) or coordinates of a bounding box (object localisation) or a predicted label for the centre pixel of the input (patch-based segmentation) or a real value for regression tasks. Image adapted from [70].

The size of convolution kernel n is chosen to be small in general, e.g. n = 3, in order to reduce computational costs. While the kernels are small, one can increase the receptive field (the area of the input image that potentially impacts the activation of a particular convolutional kernel/neuron) by increasing the number of convolutional layers.

In general, increasing the depth of convolution neural networks (the number of hidden layers) to enlarge the receptive field can lead to improved model performance, e.g. classification accuracy [77].

CNNs for image classification can also be employed for image segmentation applications without major adaptations to the network architecture [89]. However, this requires to divide each image into patches and then train a CNN to predict the class label of the centre pixel for every patch. One major disadvantage of this patch-based approach is that, at inference time, the network has to be deployed for every patch individually despite the fact that there is a lot of redundancy due to multiple overlapping patches in the image. As a result of this inefficiency, the main application of CNNs with fully connected layers is object localisation, which aims to estimate the bounding box of the object of interest in an image. This bounding box is then used to crop the image, forming an image pre-processing step to reduce the computational cost for segmentation. For efficient, end-to-end pixel-wise segmentation, a variant of CNNs called fully convolutional neural network (FCN) is more commonly used, which will be discussed in the next section.