TESE DE DOUTORADO

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM

CIÊNCIA DA COMPUTAÇÃO

"Automated analysis of leukocyte recruitment for in vivo studies using a spatiotemporal approach and multiple image features"

ALUNO: Bruno César Gregório da Silva
ORIENTADOR: Prof. Dr. Ricardo José Ferrari

São Carlos
Março / 2020

CAIXA POSTAL 676
FONE/FAX: (16) 3351-8233
13565-905 - SÃO CARLOS - SP
BRASIL

FEDERAL UNIVERSITY OF SÃO CARLOS

TECHNOLOGY AND EXACT SCIENCES CENTER

COMPUTER SCIENCE GRADUATION PROGRAM

AUTOMATED ANALYSIS OF LEUKOCYTE RECRUITMENT FOR IN VIVO STUDIES USING A SPATIOTEMPORAL APPROACH AND MULTIPLE IMAGE FEATURES

BRUNO CÉSAR GREGÓRIO DA SILVA

Thesis presented to the Computer Science Graduate Program of the Federal University of São Carlos as part of the requisites to obtain the title of Doctor in Computer Science, concentration area: Image and Signal Processing
Supervisor: Dr. Ricardo José Ferrari

São Carlos – SP

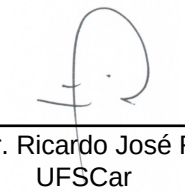
March/2020

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Bruno César Gregório da Silva, realizada em 30/03/2020:



Prof. Dr. Ricardo José Ferrari
UFSCar

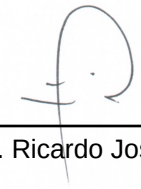
Prof. Dr. Ricardo Cerri
UFSCar

Prof. Dr. Cesar Henrique Comin
UFSCar

Prof. Dr. Paulo Mazzoncini de Azevedo Marques
USP

Prof. Dr. Marcelo Zanchetta do Nascimento
UFU

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Ricardo Cerri, Cesar Henrique Comin, Paulo Mazzoncini de Azevedo Marques, Marcelo Zanchetta do Nascimento e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.



Prof. Dr. Ricardo José Ferrari

ACKNOWLEDGMENTS

I would like to acknowledge and thank the following important people who have supported me, not only during the course of this research but also throughout my Master's degree.

Firstly, I would like to express my gratitude to my supervisors Prof. Dr. Ricardo José Ferrari and Prof. Dr. Roger Tam, for their unwavering support, guidance, and insight throughout this research work.

I would also like to thank Prof. Dr. Juliana Carvalho Tavares from UFMG, and my laboratory colleagues for their constructive discussions and observations.

I wish to express my sincere thanks to all my close friends and family. You have all encouraged and believed in me. In special, I thank my wife Tamara, who has helped me to focus on what has been a hugely rewarding and enriching process.

Last, but not least, I would like to thank the BIPG group[1], the GAPIS laboratory, the DC-UFSCar for the structure provided, and the CAPES for the financial support during my research (finance code 001).

[1] http://www.bipgroup.dc.ufscar.br

The greatest enemy of knowledge is not ignorance, it is the illusion of knowledge.

Stephen Hawking

RESUMO

Nos últimos anos, um grande número de pesquisadores tem direcionado seus esforços e interesses para estudos *in vivo* dos mecanismos celulares e moleculares na microcirculação de vários tecidos e em várias condições inflamatórias. O principal objetivo desses estudos é desenvolver estratégias terapêuticas mais eficazes para o tratamento de doenças inflamatórias e autoimunes. A análise do recrutamento leucocitário é um passo importante para entender as interações entre os leucócitos e as células endoteliais na microcirculação de animais vivos. Realizado preferencialmente através da técnica de microscopia intravital (MI), esse procedimento geralmente requer a análise visual de um especialista, que é propensa à intra- e inter-variabilidade do observador, além de ser uma atividade tediosa e demorada. Tal problema reivindica, portanto, um método automatizado para a detecção e rastreamento dessas células. Para tanto, este trabalho visa o estudo e o desenvolvimento de técnicas computacionais para a detecção e rastreamento de leucócitos em imagens de MI. Para isso, propusemos um arcabouço de desenvolvimento computacional automático que, após uma etapa de pré-processamento, combina os resultados da detecção quadro-a-quadro do vídeo (processamento espacial – 2D) com os resultados de uma análise tridimensional (processamento espaço-temporal – 3D=2D+t) feita em imagens volumétricas formadas pelo empilhamento de todos os quadros do vídeo. Neste caso, enquanto o processamento 2D visa a detecção dos leucócitos sem se preocupar com a tarefa de rastreamento, o processamento 2D+t tem o objetivo de auxiliar na análise da dinâmica celular (rastreamento). Nós testamos três abordagens diferentes para o processamento espacial, denominadas MTM-PCA, MTM-DCNN e DCNN. Nossos resultados foram obtidos por meio de avaliações qualitativas e quantitativas realizadas em seis diferentes vídeos de MI, em que as células detectadas foram comparadas com as marcações manuais de um especialista. Esses resultados mostraram que a combinação das duas etapas de processamento foi capaz de minimizar a maioria dos problemas envolvidos na detecção e rastreamento celular em imagens de MI, como a oclusão e a discriminação adequada das trajetórias das células.

Palavras-chave: Detecção de células, rastreamento de células, microscopia intravital, análise espaço-temporal, recrutamento leucocitário.

ABSTRACT

Over the last few years, many researchers have directed their efforts and interests toward in vivo studies of the cellular and molecular mechanisms in the microcirculation of many tissues under different inflammatory conditions. These studies' main goal is to develop more effective therapeutic strategies for the treatment of inflammatory and autoimmune diseases. Leukocyte recruitment analysis is a crucial step to understand the interactions between leukocytes and endothelial cells in the microcirculation of living animals. Performed preferably by the intravital video microscopy (IVM) technique, this procedure usually requires an expert to perform visual analysis, which is prone to the inter- and intra-observer variability, besides being a tedious and time-consuming task. This problem claims, therefore, an automated method to detect and track these cells. To this end, this work aims to study and develop computational techniques for the detection and tracking of leukocytes in IVM images. We proposed an automatic computational pipeline where, after a preprocessing stage, we combined the results of frame-basis detection (2D – spatial processing) with those from three-dimensional analysis (3D=2D+t – spatiotemporal processing) of volumetric images formed by stacking all the video frames. While the 2D processing focuses on leukocytes detection without worrying about their tracking, 2D+t processing was intended to assist in the dynamic analysis of cell movement (tracking). We tested three different detection approaches for the spatial processing, named as MTM-PCA, MTM-DCNN, and DCNN. Our results were obtained by qualitative and quantitative evaluations performed over six different IVM videos, where the detected cells were compared with the manual annotations of an expert. They showed the combination of these both processing stages minimized most of the problems involved in IVM cell detection and tracking, such as cell occlusion and the proper discrimination of cell trajectories.

Keywords: Cell detection, cell tracking, intravital video microscopy, spatiotemporal analysis, leukocyte recruitment.

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

ANN	Artificial Neural Networks
AP	Average Precision
ATM	Adaptive Template Matching
AUCPR	Area Under Curve Precision-Recall
CARS	Coherent Anti-stokes Raman Scattering
CLR	Cyclical Learning Rate
CM	Cumulative Matrix
CNN	Convolutional Neural Networks
CNS	Central Nervous System
CTC	Cell Tracking Challenge
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Networks
EAE	Experimental Autoimmune Encephalomyelitis
FCN	Fully Convolutional Neural Network
FLIM	Fluorescent Lifetime Imaging Microscopy
FN	False-Negative
FOV	Field Of View
FP	False-Positive
FPN	Feature Pyramid Network

fps	Frames Per Second
GBA	Grid-based BAyesian
GGVF	Generalized Gradient Vector Flow
GICOV	Gradient Inverse Coefficient of Variation
GLCM	Gray-Level Co-Ocorrence Matrix
GVF	Gradient Vector Flow
HOG	Histogram of Oriented Gradients
IoU	Intersection Over Union
IR	Infrared
IVM	Intravital Video Microscopy
KLT	Kanade-Lucas-Tomasi
LBP	Local Binary Pattern
LPS	Local Phase Symmetry
LR	Learning Rate
MC	Monte Carlo
MGVF	Motion Gradient Vector Flow
MI	Mutual Information
MOT	Multiple Object Tracking
MPM	MultiPhoton Microscopy
MS	Multiple Sclerosis
MTM	Multiple Template Matching
MTM-PCA	Multiple Template Matching - Principal Component Analysis
NCC	Normalized Cross Correlation
NMS	Non-Maximum Suppression

OFDI	Optical Frequency Domain Imaging
PC	Phase Congruency
PCA	Principal Component Analysis
PHD	Probability Hypothesis Density
POM	Probabilistic Occupancy Map
PSF	Point Spread Function
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
SHG	Second Harmonic Generation
SIFT	Scale-Invariant Feature Transform
SNR	Signal-to-Noise Ratio
SOT	Single Object Tracking
SSD	Single Shot MultiBox Detector
TBD	Tracking By Detection
TBM	Tracking By Matching
THG	Third Harmonic Generation
TNF	Tumor Necrosis Factor
TP	True-Positive
UV	Ultraviolet

CONTENTS

Chapter 1

INTRODUCTION

This chapter suits as an introduction to the following chapters, where we present the problem investigated and all the motivation behind the proposed work. We also illustrate the pipeline of a typical digital image processing framework for those who are not familiar with computer science techniques and the pipeline overview developed in this research. Finally, we highlight our primary goals and research contributions.

1.1 Context and motivation

In the middle of 1843, W. Addison (ADDISON, 1843) reported the discovery of the microcirculation involvement in the inflammatory insult by the leukocyte-endothelial interactions. An effective immune response to an inflamed tissue is critically dependent on the leukocytes ability to migrate from blood flow to sites of infection in the tissue. This process is mediated by a sequence of different interactions between the leukocytes and endothelial cells that lines blood vessels, and it is commonly called leukocyte recruitment.

Although there is a diverse number of in vitro experimental systems for the probing of morphology and molecular functions of immunological cells, the most effective scenario (if not the only one) for measuring these attributes is through in vivo techniques. Intravital video microscopy (IVM) (ELLINGER; HIRT, 1929, 1930) has become, therefore, an essential tool for studying in vivo systems mostly because it allows cellular traffic observation in lymphoid organs and peripheral tissues of the immune system (KILARSKI et al., 2013). Even in different inflammatory conditions, this imaging technique makes easier the comprehension of mechanisms related to immunologic diseases and, consequently, allows the design of new drugs and therapeutic strategies to fight inflammation (PINHO et al., 2011; ACTON; WETHMAR; LEY, 2002), which can be associated with several diseases, such as multiple sclerosis, atherosclero-

sis, rheumatoid arthritis, ischemia-reperfusion injury, and cancer (NOBIS et al., 2018; GAVINS, 2012). Also, the use of imaging methodologies has facilitated the research associated with the neurovascular regulation, which studies how cells of the immune system communicate with neurons and vice-versa. Understanding neurovascular coupling is fundamental for the pathogenesis elucidation of numerous neurological conditions (TAKANO et al., 2006).

To understand the underlying mechanisms of leukocyte recruitment, scientists usually evaluate and count the number of rolling and adhered leukocytes present in the microcirculation of living small animals (SANTOS et al., 2008). Semiautomatic techniques for tracking migrating cells (LACKIE; CHAABANE; CROCKET, 1987) in video frames are being used successfully for in vitro experiments. In these cases, the most common methods are the centroids' trackers (DIVIETRO et al., 2001; GHOSH; WEBB, 1994), which use the intensity of the center of mass of a cell to track its position in a sequence of images, and the correlation trackers (SCHüTZ; SCHINDLER; SCHMIDT, 1997; KUSUMI; SAKO; YAMAMOTO, 1993), which correlate one or more images of a particular cell with the next video frames to determine the cell location in the sequence of images.

However, unlike in vitro analyses, where the conditions of image acquisition can be controlled, in the in vivo analyses, the task of cell counting is still commonly performed by visual observation of the IVM. Besides being tedious and time-consuming, this manual task is error-prone and may introduce technician-related bias to the statistical results. Also, the region analyzed by the experts corresponds to a small section, in which only a few cells are considered – those crossing an imaginary line inside the microvessels.

Accordingly, the development of automatic techniques for in vivo experimental analysis is a critical task that arouses interest in clinical and research studies. However, although the advantages of IVM are of great importance to scientists and biologists, some inherent constraints must be considered: a) high variety of imaging protocols; b) images with low signal-to-noise ratio – SNR; c) photobleaching effect (ANDRESEN et al., 2012); d) cell occlusions and clutter; and e) image motion blur and motion artifacts, due primarily to respiratory and cardiac movement of the tested animal (ACTON; RAY, 2004; RAY; ACTON, 2004). Among the various problems cited, motion artifact is the most difficult to eliminate and may affect the success of automatic processing techniques in the in vivo studies. While this problem can be minimized by applying video stabilization techniques, the movement degrading the images can be quite complex, including horizontal and vertical components, depending on the anatomical region analyzed. For extreme cases, the employment of image restoration techniques is a required step. The most apparent problem resulting from animal motion is the momentary loss of leukocytes spatial position, which may cause failures in tracking these cells (ACTON; WETHMAR;

LEY, 2002) and, consequently, generate false statistics of their dynamic information.

Due to the constraints mentioned above, the algorithms used in the in vitro analyses, generally, are not robust enough for in vivo IVM applications. The gray-level intensity of centroids used in centroids' trackers, for example, is an attribute significantly affected by the presence of noise or the photobleaching effect. Besides suffering from the same problems, simple correlation trackers are also unable to track deformable targets and deal with cell occlusion successfully.

Therefore, although the frame-basis detection and tracking of leukocytes in IVM can be successfully performed in stabilized videos and good contrast images, the problem of animal movement and cell-changing appearance discussed previously can significantly degrade cell tracking. Thus, to prevent information losses, this work involves the use of several techniques as a preprocessing stage and follows an approach that aims to combine the results of frame-basis detection (2D processing) and three-dimensional segmentation (3D=2D+t) of the volume created by stacking all video frames. Thereby, while 2D processing can detect cells precisely, the 2D+t processing (or spatiotemporal analysis) can help both the study of cell dynamics and the elimination of ambiguities due to overlapping cells (occlusion problem).

Given this scenario, we hope that we contribute to a better analysis of leukocyte recruitment, providing more precise statistics about the cell analyses and, consequently, helping in the development of biological studies.

1.2 Objectives

The primary objective of this work was to research and develop an automatic computational pipeline to aid in the detection and tracking of leukocytes in IVM applied to in vivo experiments of different animal organs. The development of the system was based on image processing and computer vision techniques, and it combined spatial information from the frame-basis analysis (2D processing), and temporal information from the analysis of the 3D image created by stacking all video frames (2D+t processing).

To accomplish that, we created specific tasks that were defined as secondary objectives:

- Select a subset of consecutive frames from IVM experiments of different animal organs to be used in the development and test of the methods;
- Manually annotate leukocytes in the video frames selected;
- Perform the integration of preprocessing techniques;

- Test and evaluate the robustness of the preprocessing techniques;
- Develop and quantitatively evaluate techniques for leukocytes detection – 2D processing;
- Develop and quantitatively evaluate techniques for tubular-like structures detection in 2D+t processing;
- Analyze the combination of 2D and 2D+t processing outcomes;
- Test and evaluate the robustness of the proposed method in IVM study images;
- Automatically compute statistical measures for the leukocyte recruitment.

1.3 Pipeline overview

The main components of a machine vision system generally follow a standard sequential processing scheme. It starts by defining the problem domain and goal, as given in Section 1.1. The next step is responsible for acquiring the data that will be used in the processing scheme. All the steps adopted in our IVM data acquisition are detailed in Chapter 2. The following processes in the standard scheme involve the use of specialized computational techniques that can vary for each application but usually have the same goal in all problem domains. They are the preprocessing, processing, and post-processing stages.

The goal of the preprocessing stage is to improve the quality of the acquired data for the next stages. Conventional techniques employed during this stage include noise reduction, contrast improvement, and brightness correction. Next, the processing stage can include the most variable combination of methods according to the problem domain. Among the popular algorithms for this stage, we have those responsible for image segmentation, object detection, feature extraction, and classification. All these techniques have, however, one primary objective: to extract data information. Finally, some works may still have an additional step named post-processing. It aims to improve the information acquired in the previous stage with final adjustments. As a consequence of this processing scheme, we have the results for our problem domain.

Based on this standard scheme, in this work, we designed an automatic computational pipeline to solve the difficulties encountered in the biological analyses of in vivo studies. It combines both 2D and 2D+t information to detect and track leukocytes in IVM images. These images were acquired from different image acquisition protocols and animal organs. A pipeline overview of our approach is illustrated in Figure 1.1.

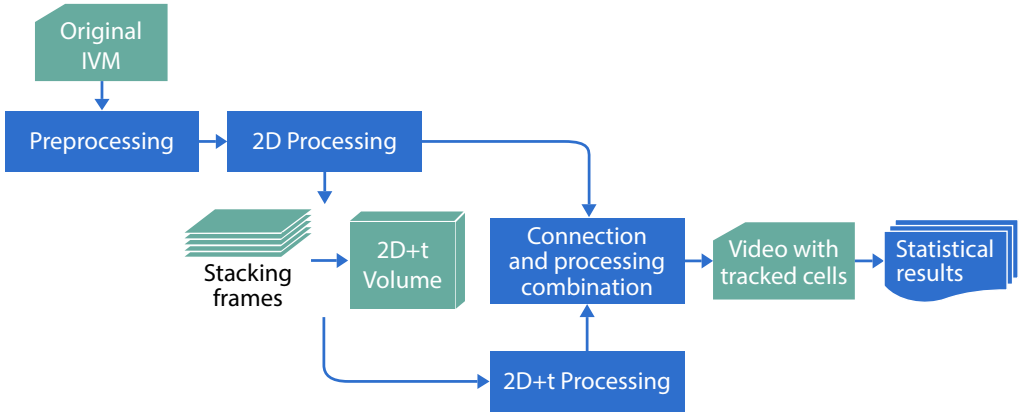


Figure 1.1: Pipeline overview of this thesis proposal.

Since the quality of input images inevitably influences the detection and tracking processes, we start our approach by applying some preprocessing techniques developed explicitly for the IVM application. This stage begins with the detection and removal of video frames strongly affected by motion blur. In the literature review, this issue is hardly reported since this problem is mostly observed in the images from the animal's central nervous system (CNS), where the mechanical stabilization is a delicate issue.

After frames removal, our framework reduces the potential noise present in the images and standardize the level of pixel intensities between all video frames to decrease the photobleaching effect over time. Only after that, we performed a registration process and got a final stabilized video. Once we have all video frames registered, a vessel segmentation procedure is employed to delimit the region in which the next techniques will operate.

Following the preprocessing stage, we split our processing step into spatial (2D) and spatiotemporal (2D+t) modules, i.e., into the detection and tracking approaches. In spatial processing, video frames were individually analyzed as two-dimensional images. On the other hand, in spatiotemporal processing, video frames were stacked to form a volumetric image (3D=2D+t) and facilitate the analysis of cell dynamics.

At this point, the detection algorithm used in spatial processing has its resulting points interpreted as candidates to cell centroids. When multiplied by a circular Gaussian kernel, all the candidate points look like blob structures in two-dimensional images and like tubular structures (representing the cell trajectories) in the image volume created. As a consequence, we have reduced our tracking stage to a detection of tubular-like structures problem in three-dimensional images, that is a well-studied problem. This approach ensures that the tracking algorithm can act in the same way for different detection techniques, which is a fundamental premise since detection methods in the literature are generally designed for particular cell types.

Local detections of 2D processing are combined with the trajectory information in a connection algorithm in the cases where trajectory gaps appear, or occlusions occur. This combination procedure results in a framework less susceptible to inherent problems from the image acquisition, such as the ambiguities arising from the overlapping of cells and the discontinuities of their movement caused by motion artifacts or by the removal of blurred frames. Additionally, it creates global improvement of cell tracking in dynamic scenes.

1.4 Contributions

The scientific contributions of this thesis consist of several open-source algorithms[1] and other punctual developed activities based on the secondary objectives of Section 1.2.

For instance, we created an IVM dataset composed of images from different animal organs and imaging protocols. All these images were frame-by-frame manually annotated for training and test purposes.

We created a preprocessing pipeline developed explicitly for the IVM application, in which we detected and removed extremely blurred frames from the videos, reduced the image noise, standardized the contrast of the images over the videos, performed video stabilization, and segmented structures of interest to further analysis.

We implemented and tested different algorithms for leukocyte detection, followed by specialized techniques of IVM data augmentation. As a part of our pipeline, we also developed a novel tracking method based on the spatiotemporal strategy. To combine these approaches, we created a new algorithm responsible for linking the cell trajectories in our tridimensional images.

Finally, we demonstrated the robustness of our methods by applying them in IVM study images in order to obtain statistical measures from the leukocyte recruitment.

The above contributions resulted directly in the following publications:

- Gregório da Silva, B. C.; Tam, R.; Ferrari, R. J. "Detecting cells in intravital video microscopy using a deep convolutional neural network." Submitted to: IEEE Transactions on Medical Imaging in April 2020.
- Gregório da Silva, B. C.; Ferrari, R. J. "Exploring deep convolutional neural networks as feature extractors for cell detection." Accepted in the 20th International Conference on Computational Science and its Applications (ICCSA 2020) in July 2020.

[1] `https://github.com/brunoggregorio`

- Gregório da Silva, B. C.; Carvalho-Tavares, J.; Ferrari, R. J. "Detecting and tracking leukocytes in intravital video microscopy using a Hessian-based spatiotemporal approach." Multidimensional Systems and Signal Processing 30(2), 815-839 (2019).
- Gregório da Silva, B. C., Carvalho-Tavares, J., and Ferrari, R. J. "Automated technique for in vivo analysis of leukocyte recruitment of mice brain microcirculation", in Proceedigns of XII Workshop de Visão Computacional (WVC), Campo Grande, MS, Brazil, 2016 (Best Paper Award - 1st Place).

While other indirectly publications were also published:

- Freire, P. G. L.; Gregório da Silva, B. C.; Pinto, C. H. V.; Moreira, C.; Ferrari, R. J. "Mid-sagittal plane detection in magnetic resonance images using phase congruency, Hessian matrix and symmetry information: a comparative study", in Proceedings of 18th International Conference on Computational Science and its Applications (ICCSA), Melbourne, VIC, Australia, 245-260, 2018.
- Elisa de Souza, K., Gregório da Silva, B. C., Carvalho-Tavares, J., and Ferrari, R. J., "Detection of leukocytes in intravital microscopy video images using the phase congruency technique". Revista de Informática Teórica e Aplicada 23(2), 33-55 (2016).

1.5 Thesis outline

This work is composed of eight chapters beyond this one. We decided to split our entire methodology into four different chapters to avoid a very long chapter. The content in each of them is as follows:

- Chapter 2: All the information needed to contextualize the reader in the IVM imaging technique, its possible applications, and a dataset description.
- Chapter 3: Theoretical background information and a literature review with the description of the principal published works related to cell tracking using IVM.
- Chapter 4: Description of the preprocessing techniques applied to the IVM images.
- Chapter 5: Description of all methods used for the 2D processing – detection.
- Chapter 6: Details of the techniques used in the 2D+t processing stage – tracking.
- Chapter 7: The results and discussions for each developed stage.

- Chapter 8: Conclusions drawn from the results and future investigations.
- Appendix A: Details of the Hessian-based local feature detector.

Chapter 2

TECHNICAL BACKGROUND

This chapter presents essential information regarding the IVM technique and its applications in the context of this research work. The following sections are split into a description of the fluorescence microscopes, the inherent challenges of IVM imaging, its applicability, and the details of our IVM dataset.

2.1 Initial considerations

As new imaging techniques for cultured cells have been emerging in the last two decades, cell biology studies have been growing fast. However, although cell culture is a very flexible handling system for genetic studies, in vitro models do not always accurately reconstitute the tissue environment of animals under several physiological conditions (WEIGERT et al., 2010). For instance, cell culture tests performed on glass or plastic surfaces (the most common ones) suffer from the loss of three-dimensional cellular organization, which is a crucial cellular process in many cases (CUKIERMAN et al., 2001). Moreover, even in three-dimensional in vitro models, which are also widely used today, there may be a lack of essential components in the analyzed environment, such as signaling molecules[1] and other types of cells (XU; BOUDREAU; BISSELL, 2009; GHAJAR; BISSELL, 2008; CUKIERMAN et al., 2001). Therefore, to reach the same analytical abilities as in vitro models, numerous efforts are being guided on new imaging technologies and, consequently, on the study of cellular events in organs of living animals. Among the imaging techniques for in vivo studies, IVM has stood out since it allows cellular traffic observation in lymphoid organs and peripheral tissues of the immune system (KILARSKI et al., 2013; WEIGERT; PORAT-SHLIOM; PARENTE AMORNPHIMOLTHAM, 2013; WEIGERT et al., 2010).

[1] Signaling molecules of cellular origin may belong to several families of biochemical substances and serve as messengers between two cells that are not distant from each other.

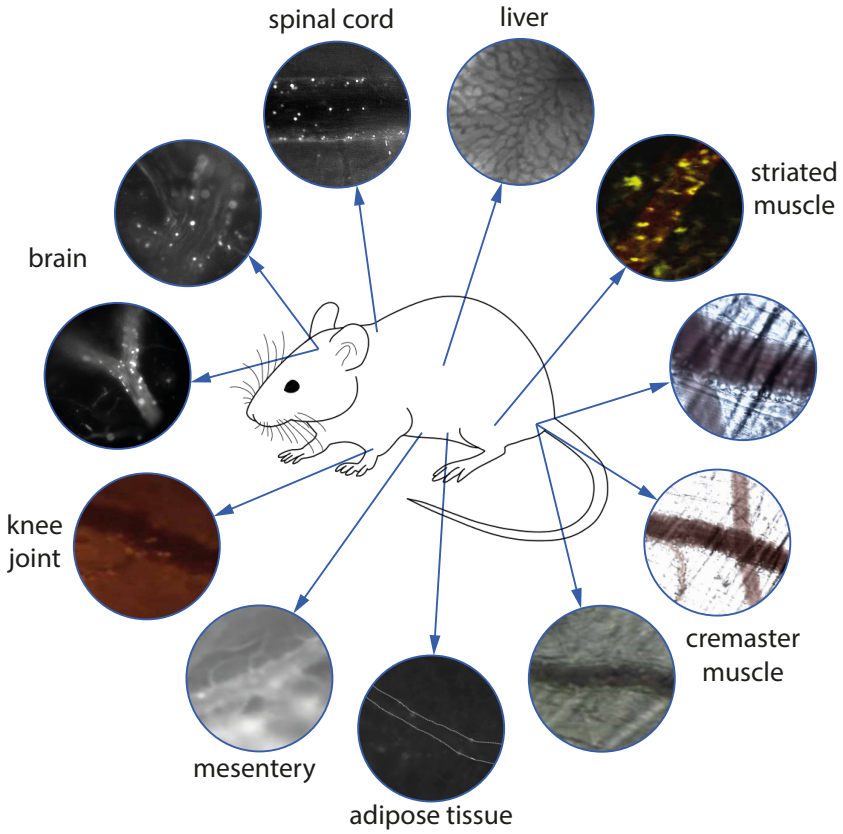


Figure 2.1: Examples of images resulting from studies in different organs of mice.

As visual examples, Figure 2.1 shows the most common animal organs observed through IVM. Even in different inflammatory conditions, this imaging technique helps in the comprehension of mechanisms related to immunologic diseases, and that is the reason why IVM images are being employed in this research work.

2.2 Intravital microscopy

The improvement of new technologies in fluorescent microscopes and proteins applied to the analyses of living cells resulted in a massive volume of information concerning almost every possible cellular process, which caused a significant advance in the knowledge encompassed by cellular biology (MASEDUNSKAS et al., 2012). Among the recently developed microscopy techniques, intravital microscopy emerges as a sophisticated research tool that allows the observation of in vivo microcirculation in organs of anesthetized or conscious animals, as well as the analysis of complex biological interactions and potential mechanisms of diseases (PITTET; WEISSLEDER, 2011).

Given the peculiarities of IVM in providing spatial and temporal responses under conditions close to those found in a natural environment, this imaging technique can reproduce cel-

lular interactions more realistically than in vitro systems. A reasonable justification for that lies in the fact that cells under analysis have their behavior influenced by several factors, including cytokine gradients, interactions with other cellular and extracellular components, anatomic compartmentalization, and blood flow forces (PITTET; WEISSLEDER, 2011).

The IVM technique is based on fluorescence microscopy, i.e., on the contrast generated by the excitation of the energy levels of a molecule through a component generally referred to as fluorophore or fluorochrome. The excitation is produced by illuminating the sample with a light source, such as a mercury lamp or a laser, which provides photons with wavelengths ranging from ultraviolet (UV) to infrared (IR).

In a fluorescence microscope, the wavelength of illumination is defined by the use of a filter (excitation filter – located after the light source) that limits the transmission of light to a narrow range of wavelengths. After passing through the excitation filter, the light reaches a dichroic mirror (beams separator) and is reflected down through the objective lens, and onto the sample. Molecules from inside the sample absorb the primary excitation and re-emit a longer wavelength light (lower energy). The objective lens collects the emitted fluorescent light and forwards it through the dichroic mirror. Any unwanted excitation light is blocked by a third filter (emission filter or barrier filter). Thus, only emitted lights from fluorescent molecules of the sample are observed and recorded. Figure 2.2 illustrates the fluorescence microscope schematic.

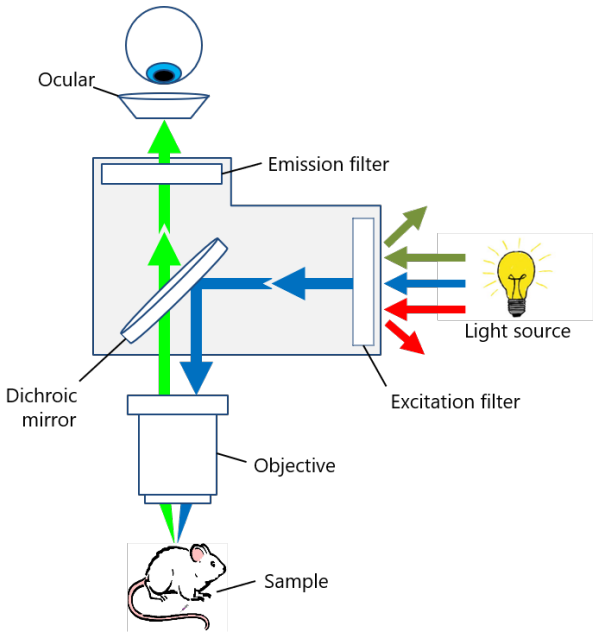


Figure 2.2: Basic scheme of a fluorescence microscope (BLACHNICKI, 2008).

There are two primary modalities of light excitation: linear and non-linear. The former is used in conventional instruments, such as fluorescence and confocal microscopes, and is based

Table 2.1: Brief comparison between different imaging techniques used for IVM (AMORNPHIMOLTHAM; MASEDUNSKAS; WEIGERT, 2011).

Imaging technique	Advantages	Limitations
Single photon microscopy	High temporal resolution High spatial resolution	Tissue penetration (50 - 60μm) Phototoxicity photobleaching Off-focus emission
Multiphoton microscopy (MPM)	Deep tissue penetration (hundreds of microns) No off-focus emission Reduced phototoxicity and photobleaching	Temporal resolution
Second and third harmonic generation (SHG and THG)	Imaging of endogenous molecules (collagen, myosin, lipids) No off-focus emission Reduced phototoxicity and photobleaching	Temporal resolution Tissue penetration
Coherent anti-Stokes Raman scattering (CARS)	Imaging of endogenous molecules (lipids, myelin) Reduced phototoxicity and photobleaching	Temporal resolution Tissue penetration
Fluorescent lifetime imaging microscopy (FLIM)	Deep tissue penetration (same as MPM) Metabolic information on tissue microenvironment	Temporal resolution
Optical frequency domain imaging (OFDI)	Deep tissue penetration (more than 1 mm) Fast acquisition of the data Reduced phototoxicity and photobleaching No need for exogenous labeling	Lower spatial resolution than MPM

on the fact that the emission intensity is linear regarding the intensity of the excitation light. The latter relies on more complex and non-linear interactions between the incidence light and the sample, in which they both absorb or scatter and recombine two or more photons (AMORNPHIMOLTHAM; MASEDUNSKAS; WEIGERT, 2011). The several and unique properties of non-linear processes led to the development of multiple imaging modalities that have been extensively explored to produce images in high resolution of living organisms. The most used modalities for IVM can be observed in the Table 2.1.

2.3 Inherent constraints of IVM

Unlike the in vitro analyses, in which the conditions of image acquisition can be better controlled, the development of automatic methods for in vivo studies using IVM presents many difficult challenges for researchers due to factors that directly affect the image quality. Some of these challenges are detailed as follows.

2.3.1 Different experimental procedures

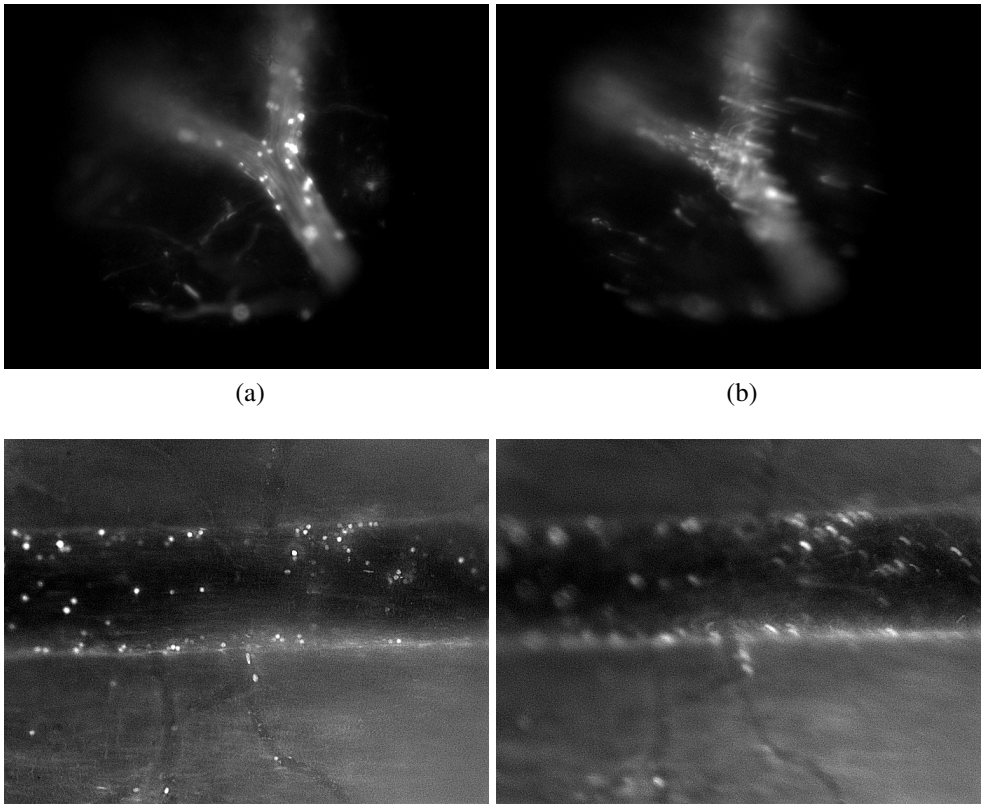
The feasibility of studying leukocyte recruitment in different animals and distinct organs results in images with diverse visual aspects that pose a challenging problem to computational approaches. Figure 2.1 illustrates examples of images obtained from various mice organs. When dealing with the same animal and the same organ, in vivo experiments can significantly differ by the way they are conducted. It means that the animal conditions before a surgery or the anesthetics used in that particular medical procedure can create an entirely different protocol of image acquisition. The same can be said about multiple experimental procedures performed by the experts, which involves the use of specific fluorescent dyes, as well as many mechanical instruments, microscopes from different brands, lenses, and other components that can modify the pictorial aspect of output images.

As an example of the complexity in designing an imaging acquisition protocol, we can cite experiments in the animal's CNS, which is one of the most challenging locations to observe leukocyte recruitment due to the presence of a continuous network of closed junctions and the lack of fenestra (SANTOS et al., 2008, 2005). This structure constitutes the blood-brain barrier and limits the exchange between soluble substances (hormones, cytokines, and immunoglobulins) and blood. Moreover, the cerebral microcirculation is partially different from the other vascular beds since vessels of the pia mater may have a diameter between 20-120 μm, there is an extensive capillary network involved, and the velocity of the blood flow is higher than in other tissues, like the liver. The blood flow also has a typical characteristic in the cerebral microcirculation: capillaries have an intermittent flow, and the arteries and veins of small and medium caliber have an oscillating stream, i.e., the blood can flow in one direction and then in another direction in the same blood vessel segment (ROSENBLUM; ZWEIFACH, 1963). Studies in the CNS have been made by inducing the Experimental Autoimmune Encephalomyelitis (EAE) model, which is one of the most commonly used models for the comprehension of inflammatory demyelinating diseases, such as multiple sclerosis (MS) (CONSTANTINESCU et al., 2011). However, CNS images are rarely reported in works for automated cell detection and tracking, which hinders our comparison in the next chapters with conventional methods in the literature.

2.3.2 Image motion blur

Movements caused by peristaltic motion or by the animal's breath and heartbeat can result in a combination of vertical and horizontal displacements of the organ under analysis. This combination of movements has as its main consequence the momentary loss of microscope

focus, which even using focal auto-adjustment is unable to correct for its focal plane in time, creating blurred and tremulous images. Figure 2.3 illustrates four examples of good and poor quality frames extracted from two IVM videos. The leftmost images (Figures 2.3(a) and 2.3(c)) represent frames without motion artifacts, i.e., those considered suitable frames, while the rightmost ones (Figures 2.3(b) and 2.3(d)) show examples of blurred and tremulous images, or poor quality frames, which make the analysis much more difficult.



(a) (b)

(c) (d)

Figure 2.3: Examples of good and poor (affected by motion blur) quality frames from two videos used in this work. (a)–(b) frames of good and poor quality from a mouse brain experiment, and (c)-(d) frames of good and poor quality from a mouse spinal cord experiment. Some images were contrast enhanced for better visualization. Leukocytes can be identified as bright circular objects.

In order to reduce the animal movement, the experts first need to perform particular surgical procedures and adequately place the exposed organs. However, the inclusion of mechanical components for sample stabilization can be a harmful task in procedures performed in the spinal cord or brain of animals, for example, since they can provoke undesirable cellular effects. As a consequence, we have a sequence of images containing motion artifacts that can interfere with structures of interest, complicating the characterization of structural changes, and causing the generation of false statistics in the quantification of image targets, like the loss of one or more cells in the detection and tracking processes.

Also, fluorescent microscopes have their resolution in the focal plane defined by the diffraction pattern produced by spherical waves exiting the rear circular aperture and converging in the focal point. Due to the wavelength of fluorescent bead illuminating the circular aperture, this pattern has a bright region in the center, commonly called Airy disk (BORN; WOLF, 1970), and a series of concentric bright rings around the center, forming the Airy pattern (DAVIDSON, 2019; TRANTHAM; REECE, 2015). For being an inherent point spread function (PSF) that also affects the z-axis, its appearance in microscope images is most noticed when the sample is out of focus. An example of the Airy pattern in our images can be seen in Figure 2.4.

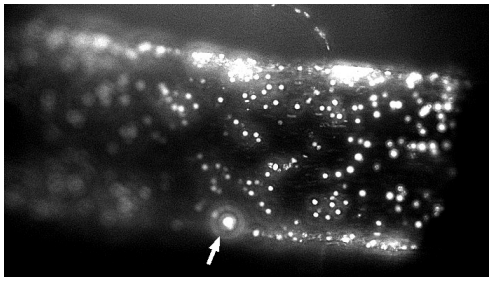


Figure 2.4: Example of Airy disk pattern in a real IVM image.

2.3.3 Photobleaching effect

The phenomenon of photobleaching occurs when the fluorophore permanently loses its ability to fluoresce due to chemical damage induced by photons and covalent modifications. The average number of excitation or emission cycles happening in a particular fluorophore previously to the photobleaching effect is dependent on the molecular structure and the local environment. Some fluorophores degrade rapidly after emitting only a few photons, while others (more robust) can suffer thousands or millions of cycles before the effect begins. As a consequence, there is a gradual loss of image contrast, as can be seen in Figure 2.5, for example.

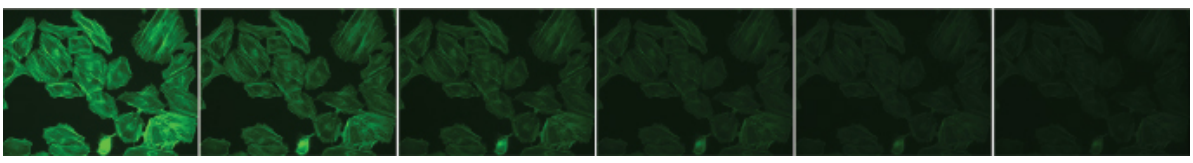
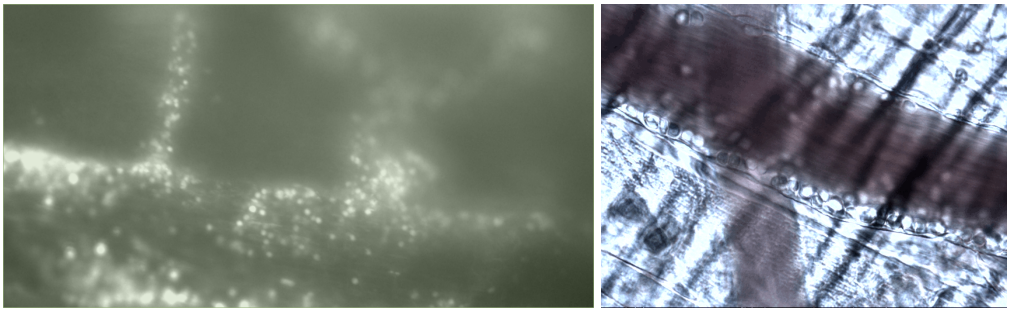


Figure 2.5: Photobleaching effect observed in a sequence of equally spaced images (12 seconds) (PROLONG, 2015).

Photobleaching effect can be reduced by limiting the fluorophores exposure time under the light source (animal exposure time) or by decreasing the applied excitation energy. However, these approaches may also restrict the sample observation period and reduce the measurable fluorescence signal.

2.3.4 Cell occlusion and clutter

Whereas the microcirculatory structures are three-dimensional, the IVM imaging also suffers from the spatial location of image targets. As the fluorescent-labeled cells move away from or approach the microscope focal plane, they experience a considerable loss of contrast. The same can be said of those cells in a complex environment consisting of many other cells or under biological structures, such as the animal muscles. A high density of cells can cause clutter or complete occlusion of targets in the analysis, impairing cell tracking, for example. Herein, gaps in cell trajectories over time (mostly caused by low frame rates or fast movements of cells) can also be treated as cell occlusion, since both are considered tracking losses by the algorithms. Figure 2.6(a) shows an image example from a mouse spinal cord in which the clutter problem is visible, while Figure 2.6(b) presents an image from the cremaster muscle, in which the diagonal structures (muscles) may cause a loss of contrast in the cells.



(a) (b)

Figure 2.6: Example of two frames where the problem of clutter and occlusion can be seen. (a) video frame from the spinal cord, and (b) video frame from the cremaster muscle of mice.

2.3.5 Cell traffic

Cell dynamics is also an obstacle to be considered when in vivo studies are performed. The cells entering and leaving the microscope field of view (FOV) happen continuously and can disrupt automatic tracking. A cell disappearance in a tracking process must be adequately investigated to determine if an occlusion occurred or if the cell left the FOV. The same happens with the rising of cells during video recording, which can characterize the arrival of a new cell in the microscope FOV or just the return of a particular cell already under analysis after an occlusion. Figure 2.7 shows an example of cells moving along the vessel in the blood flow direction.

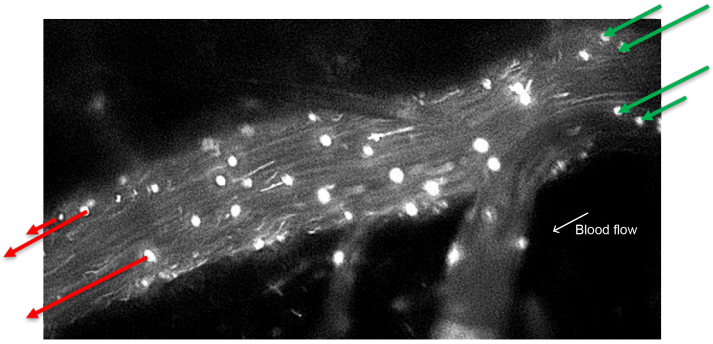


Figure 2.7: Example of cells entering and leaving the microscope field of view.

2.3.6 Ground truth

For the sake of training and quantitative validation of an automated method for the detection and tracking of cells in IVM images, it is essential to compare its results with those obtained from a visual analysis of one or more experts (researchers or laboratory technicians). The measures of inter- and intra-observers variability are also of great importance to define a confidence interval and to better interpret the developed method. However, as stated in Chapter 1, the task of manually annotate cells in hours of videos is time-consuming and prone to errors.

Although IVM imaging of different organs has distinct properties, the problems listed above must be considered in the development of algorithms for the detection and tracking of cells, regardless of the organ or tissue under analysis.

2.4 Leukocyte recruitment

In the mid-19th century, a rudimentary form of IVM, although being performed simplistically and mostly in transparent surface tissues, already revealed that blood flow occurred within microvessels (LEEUWENHOEK; HOOLE, 1800) and the movement of leukocytes could actively overflow into injured tissues (DUTROCHET, 1824; WAGNER, 1839). However, it was only in 1843 that the researcher W. Addison (ADDISON, 1843) finally reported the discovery of microcirculation involvement in the inflammatory insult through leukocyte-endothelium interactions.

Classically described as "swelling, redness, hot and painful", the inflammation is created in a particular portion of the body as a reaction to an injury or an infection (GAVINS; CHATTERJEE, 2004). The efficient formation of an immune response to the inflamed tissue is critically dependent on the ability of leukocytes to migrate from blood towards the inflammatory locus. This process is mediated by a sequence of interactions from different adhesion molecules over

leukocytes and endothelial cells of blood vessels. Among these cascade of events involved in leukocyte recruitment, we can include the initial contact of these cells with the vascular endothelium, the rolling process (weak and firm adhesion over endothelium), the chemotaxis, and the migration of cells from blood flow to the target tissue of inflammation (HYNES; LANDER, 1992).

The interactions originate with the moving leukocyte touching the vascular endothelium. Next, the leukocyte starts rolling along the vessel wall, which causes a substantial reduction of its velocity. This process is mediated by adhesion molecules from the selectins family and their respective binder carbohydrates or by α4-integrins (LUSTER; ALON, 2005). After the initial tethering of leukocytes by these molecules, the rolling process begins. The interaction of integrins with immunoglobulins causes a reduction of rolling velocity and stabilizes the leukocyte-endothelium interaction, resulting in the capture and firm adhesion of the cell to the vascular endothelium (STEEBER; CAMPBELL, 1998). Finally, this process leads to diapedesis (migration or transmigration of leukocytes to tissue). Figure 2.8 briefly illustrates the stages involving the described leukocyte recruitment.

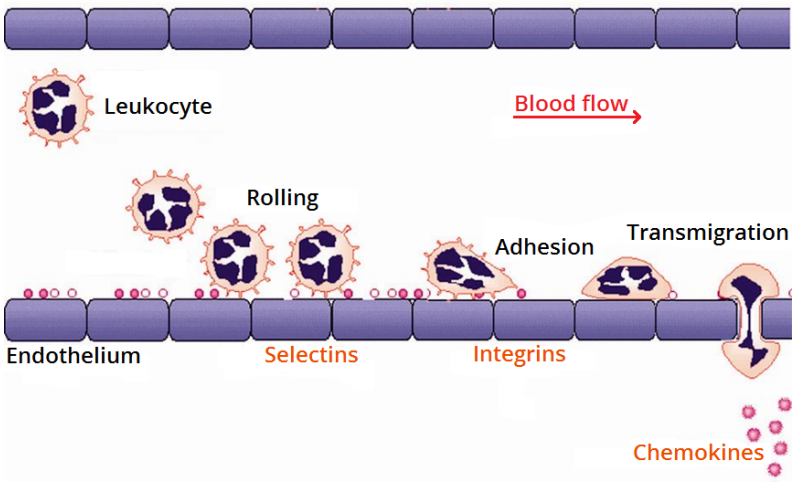


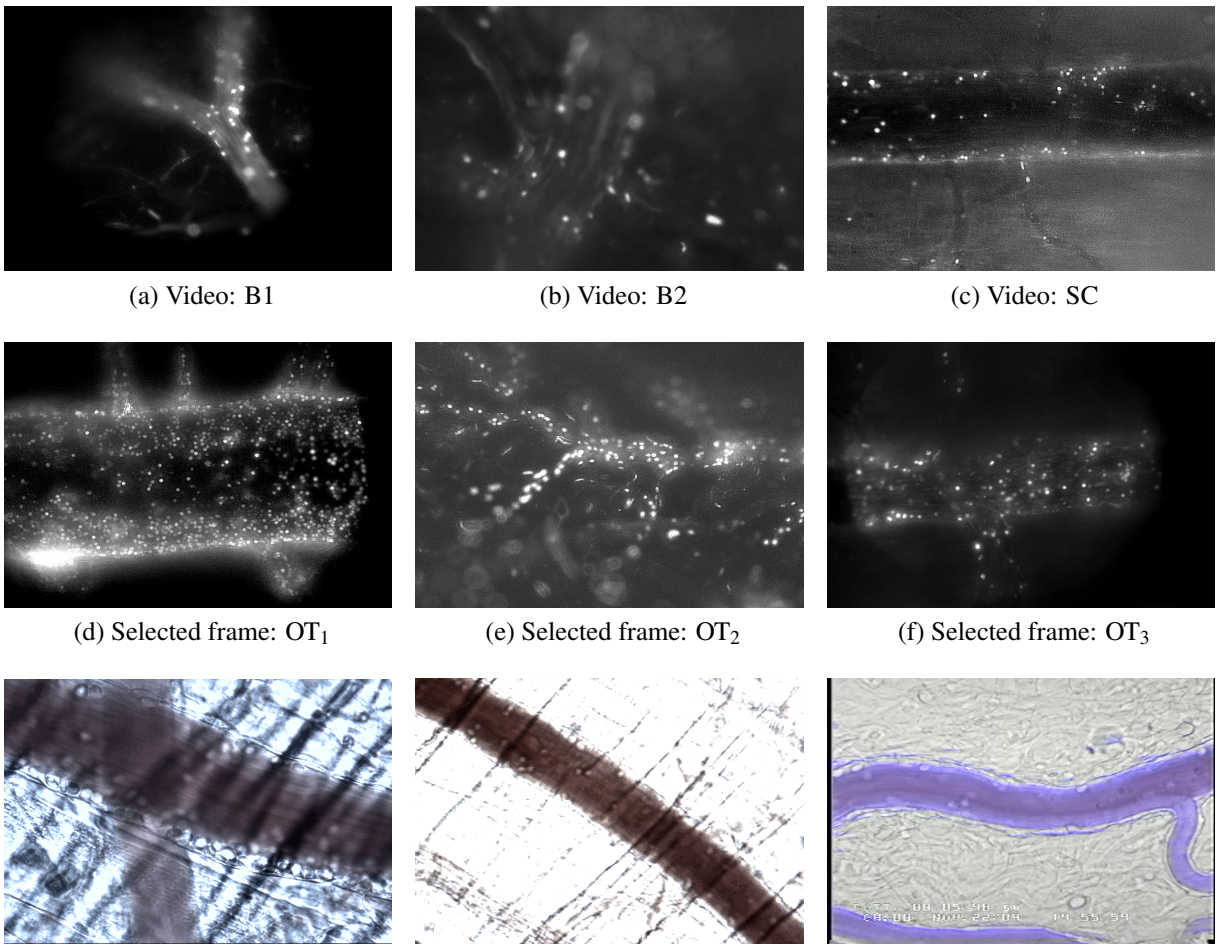
Figure 2.8: Leukocyte recruitment mechanism.

Through IVM, it is possible to study the movement of leukocytes in the microcirculation and, consequently, the effector functions that encompass the leukocyte's ability to engulf (phagocyte), eliminate, and digest several pathogens. We can say, therefore, that leukocyte recruitment plays a vital role in the immune response and, since it is visible through IVM analyses, this imaging technique has become an essential tool in multiple areas, such as neurobiology, immunology, tumor biology (MASEDUNSKAS et al., 2012), and for the pharmaceutical industry, with the study of new drugs, by identifying and validating their effectiveness (or failure).

2.5 IVM dataset

Intravital microscopy experiments require great effort regarding animal preparation and data acquisition. In general, they are conducted in a short period to avoid external interferences affecting the results, like the heat from the microscope light source.

To train and test our computational pipeline, we used six videos (IDs: B1, B2, SC, C1, C2, and ME) of IVM studies with 705 frames in total. They were obtained from distinct imaging acquisition protocols and applied in four different animal organs: brain, spinal cord, cremaster muscle, and mesentery of mice. Another 32 image frames (IDs: OT_i) were used in this work. These isolated images were included for a better evaluation of our deep learning model since they impose an additional challenge by presenting a considerable amount of cells per image compared to the others, as can be seen in Figures 2.9(d, e, and f). Examples of frames from each one of the videos can also be observed in Figure 2.9.



(a) Video: B1 (b) Video: B2 (c) Video: SC

(d) Selected frame: OT_1 (e) Selected frame: OT_2 (f) Selected frame: OT_3

(g) Video: C1 (h) Video: C2 (i) Video: ME

Figure 2.9: Examples of frames from videos of different mice organs.

As illustrated in Figure 2.9, the images present particular visual aspects considering the

Table 2.2: Description of the IVM dataset used in this work. B1 and B2: videos from the mice brain; SC: video from the mice spinal cord; OT: other selected frames from CNS; C1 and C2: videos from cremaster muscle; ME: video from the mesentery.

Category	ID	Matrix size (px)	Spatial resol. (pixels/μm)	Sample rate (fps)	Color	Avg cells size (px)	# frames analyzed	# annotated leukocytes
CNS	B1	692×520	0.66	16	No	7×7	220	5827
	B2	460×344	0.66	16	No	7×7	401	8048
	SC	692×520	0.98	16	No	5×5	21	1570
	OT	varying	varying	-	No	14×14	32	2725
						Subtotal:	*674*	*18170*
Cremaster muscle	C1	1392×1040	n/a	15	Yes	25×25	21	390
	C2	1392×1040	n/a	15	Yes	5×5	21	1603
						Subtotal:	*42*	*1993*
Mesentery	ME	720×480	n/a	30	Yes	11×11	21	291
						Subtotal:	*21*	*291*
						Total:	**737**	**20454**

organ in analysis. Experiments in the CNS (IDs: B1, B2, SC, and OT), for example, show leukocytes appearing as bright blob-like structures in a dark background, with cells varying their scales along the vessel in a cluttered environment. As another example, the colored images from the cremaster muscle present the striping diagonal shades in the microscopy images. Together with the mesentery images, they show leukocytes in different contrasts, with dark boundaries or even with a transparent appearance.

In all six videos, the leukocyte centroids were frame-by-frame manually annotated by an expert in the form (x,y,t), where (x,y) is the coordinate point in the spatial domain, and t is the corresponding frame number. The cells in the remaining 32 images (OT) were manually annotated using bounding boxes in the form (x_1,y_1,x_2,y_2), where (x_1,y_1) is the top-left point, and (x_2,y_2) is the down-right point of the box enclosing the cell. Depending on the algorithm output, we either extracted the central points of bounding boxes for the evaluation or created bounding boxes in the video annotations using the average cell radius to define the box sizes. All information necessary to describe our dataset are presented in Table 2.2.

Although some of these videos have a relatively small number of frames analyzed, the total number of manually annotated leukocytes is quite large (see values in Table 2.2), providing enough data for a proper quantitative evaluation of automated methods.

As the execution of animal procedures for image acquisition is outside the scope of this thesis, we refer the interested reader to the works of our collaborators Prof. Dr. Juliana Carvalho

Tavares[2] (SANTOS et al., 2008, 2005) and Prof. Dr. Mônica Lopes-Ferreira[3] (SANTOS et al., 2017) for further details.

2.6 Final considerations

Throughout this chapter, we presented the essential and necessary information to support the adoption of the IVM technique in the context of this research work, such as physical and chemical details of image formation, the inherent technical constraints, and what is reasonable to investigate in similar approaches. We also described the IVM dataset used over this thesis. In the next chapter, we will discuss in more detail the main techniques proposed in the literature related to the detection and tracking of cells in IVM images.

[2]Department of Physiology and Biophysics, Federal University of Minas Gerais, Belo Horizonte, MG, Brazil.

[3]Special Laboratory of Applied Toxinology (Center of Toxins Immune-Response and Cell Signaling), Butantan Institute, São Paulo, Brazil.

Chapter 3

THEORETICAL BACKGROUND

There are numerous techniques for object tracking nowadays. They include methods for single or multiple object tracking and a vast number of different kinds of input images. In this sense, this chapter starts describing and categorizing the main approaches used in various object tracking problems. Then, we briefly describe some cell tracking algorithms to show the diversity of methods in the literature. However, as these works do not necessarily use IVM images as input, they can move away from our work. Therefore, at the end of this chapter, we present a section detailing the studies mostly related to ours, i.e., those specifically developed to detect and track leukocytes in IVM studies.

3.1 Multiple object tracking – MOT

Multiple object tracking (MOT) has become an essential task in computer vision applications due to its academic and commercial potentials. MOT aims at locating multiple objects of interest, inferring their trajectories, and maintaining their identities given an input video or sequence of images. Among the objects of interest or targets, we can mention a variety of studies involving the tracking of vehicles (LOCHNER; TRICK, 2014; ZENG; MA, 2002; BETKE; HARITAOGLU; DAVIS, 2000), pedestrians (JIANG; HUYNH, 2018; XU; LIU, 2015; JIANG et al., 2010), sports players (LIU et al., 2013; XING et al., 2011), animals (RODRIGUEZ et al., 2017; ITSKOVITS et al., 2017; FUKUNAGA et al., 2015), cells (TüRETKEN et al., 2017; MASKA et al., 2014; LI et al., 2008), etc. These studies are essential for numerous applications, such as:

- Medical image processing: as mentioned in the previous chapters, labeling multiple cells in videos require laborious manual work. In that case, MOT helps to get more precise results and save a considerable amount of time and labeling costs.

- Visual surveillance: to identify abnormal behaviors in surveillance videos, the automatic analyses of objects in a scene can be performed. These analyses need to locate and track peculiar targets, investigating their suspicious actions and trajectories, for example.
- Augmented reality: the employment of MOT techniques can improve, for instance, the user experience in situations like video conferences or visualization of 3D virtual objects.
- Human-computer interface: useful in the extraction of visual information related to facial expression, eye tracking, gesture, and others, making the task more natural and intelligent.
- Sports expertise: the tracking of players on the court or sporting objects in the scene can produce important statistical and strategical information about a game or a team in a league. Big companies are adopting this application due to its immense practical return.
- Robotics: the real-time analysis of tracked objects, for instance, is very useful to guide a robot in collision-free locomotion through crowded and dynamic environments.

All these applications, among others, have aroused significant interest in the MOT research area. However, compared with single object tracking (SOT), which primarily focuses on designing sophisticated appearance and motion models, MOT additionally requires maintaining the identities among multiple objects (LUO; ZHAO; KIM, 2014). It is also regarded as a natural extension of SOT, sharing some tracking challenges like object scale changes, rotation, translation, illumination variations, camera distortion, and information loss because of the projection from 3D to 2D. Despite these traditional challenges, some critical issues make MOT a more complicated task. They are strengthened by crowded environments, especially because many applications need to track targets with similar appearance while facing object singular pose variation, shape deformation, frequent occlusions, initialization and termination of tracks, the small size of objects (BETKE et al., 2007), interaction among multiple targets, and dynamic, cluttered backgrounds (LUO; ZHAO; KIM, 2014; FAN et al., 2016). In order to solve one or more of these issues, several algorithms have been proposed. The primary methods for doing so, involve either building a detector or exploiting the tendency for objects to look the same over time, and to move coherently. The majority of these methods can be categorized by their high-level strategies according to the following aspects, as described in Figure 3.1.

Tracking strategy There are two general strategies to track objects in a video or a sequence of images. In the first, named herein as *tracking by detection* (TBD), objects are identified in each frame by a robust model describing them and linked into trajectories based on their similar features. In the second, *tracking by matching* (TBM), objects in the previous frame are already

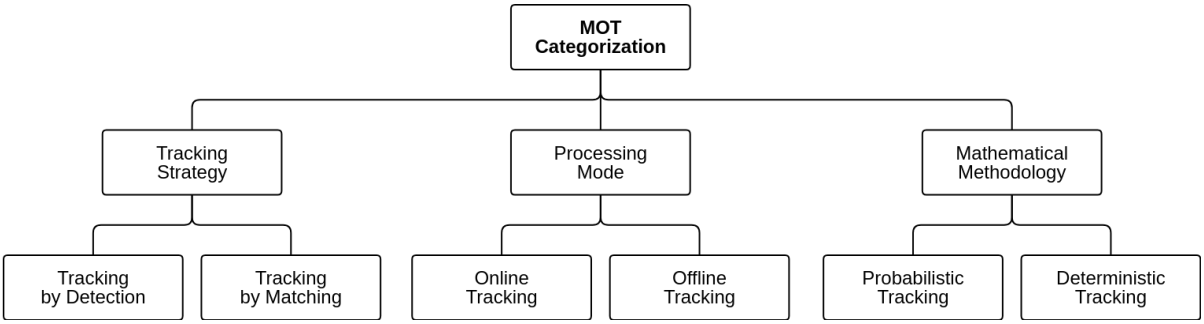


Figure 3.1: Multiple object tracking categorization.

known, and then a model describing how they move is created. In other words, objects in the last frame could serve as a template if we have a motion and domain models to link them in the current frame. TBD is often used when the number of objects varies over the video analysis, i.e., when new objects appear or disappear in the scene. However, in most cases, the object detection procedure is not the focus of TBD methods since they use pre-trained object detectors for specific kinds of targets (FELZENSZWALB et al., 2010; SUN; BEBIS; MILLER, 2006). Although TBD initialization is automatic, its performance highly depends on object detection. TBM, on the other hand, requires manual identification of a fixed number of targets in the first frame, but it does not rely on object detectors to provide object observation (ZHANG; MAATEN, 2014, 2013; HU et al., 2012).

Processing mode Regarding the way methods process the data, MOT algorithms can be categorized into *online* (sequential tracking) and *offline* (non-sequential tracking). The online tracking methods use object observations only from previous frames to estimate the current object state, i.e., the image sequence is handled in a stepwise way, which is very useful in real-time applications. Offline tracking methods (QIN; SHELTON, 2012; YANG; NEVATIA, 2012), on the other hand, use information from both the past and future frames to conduct the object state estimation, which facilitates a globally optimal solution but can have a delay in outputting the final results.

Mathematical methodology Another conventional way to categorize MOT algorithms is related to the scientific methodology adopted. In this case, MOT can be classified into probabilistic and deterministic tracking approaches. Methods that use a probabilistic framework to solve the tracking problem are classified as *probabilistic tracking* methods. They are based on Bayesian theory and use this formulation to estimate all the object trajectories in the video. Approaches based on *deterministic optimization* framework (or data association) cast the tracking task as an optimization problem, in which the observation of targets from all the frames, or part

of them, are associated with the trajectories based on their affinities.

Besides being a handy way to categorize tracking methods, the categories above put algorithms into big groups, and sometimes they are not capable of describing all the main approaches and their characteristics. The following section presents in more detail the primary components of different MOT methods.

3.2 MOT taxonomy

As suggested by Luo et al. (LUO; ZHAO; KIM, 2014) and Fan et al. (FAN et al., 2016), MOT methods can be divided into two primary components, the *observation model* and the *dynamic model*. The former measures the similarity of objects observation between the video frames, and the latter finds the object matchings based on the similarity measurements. The taxonomy of MOT methods is illustrated in Figure 3.2. All components of the MOT methods are described in the following subsections.

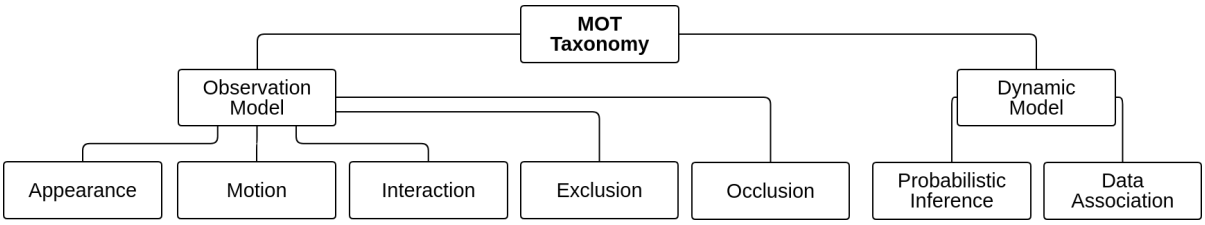


Figure 3.2: Multiple object tracking taxonomy, adapted from Luo's work (LUO; ZHAO; KIM, 2014).

3.2.1 Observation model

The observation model is related to the object's property, like appearance, location, and velocity. It is generally based on feature extraction to identify the objects in the sequence of images accurately. In other words, an observation model includes modeling of appearance, motion, interaction, exclusion, and occlusion of the objects in the scene.

3.2.1.1 Appearance model

Different from SOT, where sophisticated models are built for a single object, appearance in MOT is not the primary focus of the algorithms because multiple objects in real applications can hardly be discriminated only by this information (LUO; ZHAO; KIM, 2014). The model describes the objects according to several features. The most common ones used for both SOT and MOT are listed in Table 3.1.

Table 3.1: Common features used in MOT.

Feature	Representation	Advantages	Disadvantages	References
Point	Harris corner, SURF, Kanade-Lucas-Tomasi (KLT)	Fast, simple	Sensitive to occlusions and out-of-plane rotations	(TOMASI; KANADE, 1991; SHI; TOMASI, 1994) (ZHAO; GONG; MEDIONI, 2012; CHOI; SAVARESE, 2010)
Raw pixel	Normalized Cross Correlation (NCC)	Fast, simple	Suffers from changes in illumination and occlusion	(YAMAGUCHI et al., 2011)
Color	Color histogram (RGB, HSV, CIELuv, CIELab)	Efficient, can differentiate multiple targets in a crowded scene	Sensitive to lightning changes and noise, and it ignores the spatial position of the object	(RIAHI; BILODEAU, 2015; HU et al., 2015) (IZADINIA et al., 2012; KUO; HUANG; NEVATIA, 2010)
Motion	Optical flow, background subtraction	Produce a proper object moving information and handle occlusion	Computation expensive, struggles to identify non-moving objects	(HU et al., 2015; JIA et al., 2015) (IZADINIA et al., 2012; RODRIGUEZ et al., 2011)
Gradient	Histogram of Oriented Gradients (HOG), Scale-Invariant Gradients, Scale-Invariant Feature Transform (SIFT)	Suit human detection, can describe the shape of object, robust to illumination changes	Cannot handle occlusion and deformation	(KUO; HUANG; NEVATIA, 2010; DALAL; TRIGGS, 2005) (IZADINIA et al., 2012; LOWE, 1999) (ANDRIYENKO; SCHINDLER, 2011; MITZEL et al., 2010)
Texture	Gray-Level Co-Ocorrence Matrix (GLCM), Local Binary Pattern (LBP), Run Length	Good accuracy in the presence of distinct texture appearance	Computation expensive	(HU et al., 2015) (JIA et al., 2015)
Contour	Edges	Simple and less sensitive to illumination changes	Sensitive to noise	(HU et al., 2015) (SUWANNATAT; CHINNASARN; INDRA-PAYOONG, 2015)
Depth	Level sets, 3D	Make the computation of affinity more accurate	Requires multiple views of the same scenery and/or additional algorithm	(CHEN et al., 2015; MITZEL et al., 2010) (GAVRILA; MUNDER, 2007)
Frequency	Frequency domain analysis, Phase Congruency (PC)	Explores the frequency and phase information	Computation expensive	(SUGIMURA et al., 2009) (ELISA DE SOUZA et al., 2016)
Others	Region covariance matrix, biological features, Probabilistic Occupancy Map (POM), spatiotemporal (super-pixel)	Explores different features	Generally computational expensive	(FLEURET et al., 2008) (BERCLAZ et al., 2011)
Multi-feature	Multiple cues	Can aggregate several information	Computation expensive	(KUO; HUANG; NEVATIA, 2010; MITZEL et al., 2010) (GAVRILA; MUNDER, 2007)

Some applications involve simple object observations, and many times a single feature is sufficient to perform tracking (ZHAO; GONG; MEDIONI, 2012; KRATZ; NISHINO, 2010). However, a multi-feature model is preferred in most cases since a combination of cues could improve the robustness of the method. Meantime, to formulate a multi-feature based appearance model is necessary to use a fusion strategy. Luo et al. 2014 (LUO; ZHAO; KIM, 2014) summarized multi-feature based appearance models into five kinds of fusion strategies:

- Boosting: this strategy usually selects a portion of features from a feature pool sequentially via a Boosting based algorithm. For example, for the color histogram, HOG and covariance matrix descriptors, AdaBoost, RealBoost, and a HybridBoost algorithm are respectively employed to choose the most representative features to discriminate pairs of trajectory segments of the same object from those of different objects (FENG et al., 2015; WU et al., 2012; HUANG; WU; NEVATIA, 2008).
- Concatenation: different kinds of features can be concatenated for computation. In (BRENDEL; AMER; TODOROVIC, 2011), color, HOG, and optical flow are concatenated for appearance modeling.
- Summation: this strategy takes affinity values from different features and balances them with weight coefficients (MITZEL et al., 2010; LIU; LIN; ACTON, 2012).
- Product: different from the strategy above, values are multiplied to produce the integrated affinity (YANG et al., 2009; BERCLAZ; FLEURET; FUA, 2006). Note that the independence assumption is usually considered when applying this strategy.
- Cascading: this is a cascade manner of using various types of visual representation to either narrow the search space (GAVRILA; MUNDER, 2007) or model the appearance in a coarse-to-fine manner (IZADINIA et al., 2012).

Other options for the multi-features approach are related to feature selection or dimensionality reduction, such as principal component analysis (PCA) and its variations, for example.

3.2.1.2 Motion model

The motion model focuses on describing how objects move and, in addition to SOT methods, they are essential to MOT approaches because they might reduce the search space when predicting the object's position in future video frames. Popular motion models employed in MOT approaches can be divided into two groups: linear motion model and non-linear motion model.

- Linear motion model: in this model, it is assumed the velocity of objects is linear and constant, i.e., the object's speed in the next frame is the same as in the current one (COLLINS, 2012; ANDRIYENKO; SCHINDLER, 2011).
- Non-linear motion model: when targets move freely, the linear motion model cannot deal with motion prediction properly. In these cases, the non-linear motion model is used to produce more accurate results. This model can be used to distinguish targets with similar appearance in data association frameworks (DICLE; CAMPS; SZNAIER, 2013; YANG; NEVATIA, 2012), and to analyze the effects of unexpected camera motion (YOON et al., 2015).

3.2.1.3 Interaction model

Interaction models consider the motion influence between the objects. In a crowded scenario, objects can suffer from the force of others and environmental factors. Popularly, interaction models are divided into social force models and crowd motion pattern models. In the first, the objects are evaluated by their own forces, which considers the destination and velocity of objects will not change, and by group force, in which the attraction, repulsion, and coherence between the individuals are also considered (QIN; SHELTON, 2012; YAMAGUCHI et al., 2011; CHOI; SAVARESE, 2010). In the second, an over-crowded scenario is generally assumed. Indeed, when the density of objects is considerably high, targets are usually quite small, and the motion pattern from the crowd could be considered (ZHAO; GONG; MEDIONI, 2012; KRATZ; NISHINO, 2010).

In cell tracking, the blood flow primarily drives the cells movement. However, due to the three-dimensional characteristic of vessels and the attraction force caused by the leukocyte-endothelial interactions, it is difficult to measure the influence of leukocytes or erythrocytes on other cells.

3.2.1.4 Exclusion model

Exclusion models usually work as a constraint when objects collision occurs in MOT. To solve this problem, Milan et al. (MILAN; SCHINDLER; ROTH, 2013) considered two constraints: detection-level exclusion and trajectory-level exclusion. The former considers that two different detection responses in the same video frame cannot be assigned to the same trajectory hypothesis, while the latter accounts for the case where two different trajectories cannot occupy the same detection response. Another practical solution is the analysis of spatial-temporal relationships to interpret the collision (CHEN et al., 2016; KUMAR; VLEESCHOUWER, 2013).

3.2.1.5 Occlusion model

As already stated in Chapter 2, occlusion is a problematic issue in tracking applications. Some works developed different strategies to handle this problem. One popular approach is the assumption that at least a part of the object is visible when an occlusion occurs. Thus, the whole object position is inferred by using the information provided by the visible parts (HU et al., 2012; IZADINIA et al., 2012). Another strategy is based on the buffering of observations when occlusion happens, remembering the states of objects before occlusion. Thus, at the end of the occlusion, the states are recovered according to the stored observations and states (MITZEL et al., 2010). Other works handle occlusion by creating hypotheses and testing them between the objects observation (TANG; ANDRILUKA; SCHIELE, 2014; TANG et al., 2013).

3.2.2 Dynamic model

A dynamic model is closely related to the way observations are analyzed over time, i.e., the idea of inferring data association, target states, or both, acting as a tracking strategy. As the name indicates, it takes into account the object's dynamic by linking the targets correctly using their previous observations. There are two main approaches to do so: *probabilistic inference-based* and *data association* (or deterministic optimization-based), depending on the type of linking strategy.

3.2.2.1 Probabilistic inference

Probabilistic-based approaches often use targets' observations to estimate a probabilistic distribution. By using their states, such as size, position, and velocity, for example, they can create a tracking algorithm that works with only the current information, i.e., past and present observations to estimate the next states. This characteristic makes it a good option for online applications. Over the years, several algorithms started to arise in MOT applications, such as Kalman filter (RODRIGUEZ et al., 2011; BERCLAZ et al., 2011), Extended Kalman filter (MITZEL; LEIBE, 2011), Particle filter (HU et al., 2012), Bayesian framework (YOON et al., 2015), and Probability Hypothesis Density (PHD) filter (FENG et al., 2015).

3.2.2.2 Data association

Data association based approaches firstly select the observations (usually the detection candidates) in some frames and then search for similarities between them. This process is generally imposed as a deterministic optimization problem since a global optimum solution is searched

using carefully designed cost functions. These kinds of algorithms can be divided into local and global optimization frameworks, depending on the number of frames the method applies to solve the association problem. The most famous approaches in the literature are those related to Bipartite Graph Matching (SHU et al., 2012), Dynamic Programming (CHOI; SAVARESE, 2012), Network Flows (WALIA; KAPOOR, 2016; XI et al., 2015), Conditional Random Field (MILAN; SCHINDLER; ROTH, 2013) and Maximum-Weight Independent Set (BRENDEL; AMER; TODOROVIC, 2011). Although data association models outperform the probabilistic inference ones by their capacity to analyze multiple frames, especially in the occlusion occurrences, they suffer from the consumption of time and space, which restrict their applications in online tracking scenarios.

3.3 Cell tracking

In simple image sequences, where cells present high contrast and the movement is absent or imperceptible, the detection and tracking tasks are facilitated. Usually, a mere global threshold technique can separate the cells from the background when controlled studies like that are conducted. These image sequences are generally associated with in vitro experiments, where the analysis is performed in cell cultures. However, thresholding algorithms mostly fail in the presence of visual issues, such as photobleaching effect, severe image noise, poor contrast, or the appearance of undesirable objects. In this case, a more sophisticated technique is required. Some of the most traditional detection techniques are (1) template matching (KACHOUIE et al., 2006; GONZALEZ; WOODS, 1992), in which predetermined cell intensity profiles are fitted to the images, but may fail if cells appearance change over time or are different between themselves in the case of MOT; (2) watershed transformation (ZHOU et al., 2009; WäHLBY et al., 2004), where the images are divided into sub-regions according to a topographic relief, but is not robust to noise and may cause over-segmentation; and (3) deformable models (PADFIELD et al., 2009), also defined as parametric contours (snakes) and level-set functions, where, starting from a rough initial segmentation, they try to minimize an energy functional, but may fail in the presence of nearby cells.

A conventional approach to solving the data association problem or the tracking itself is based on the most straightforward analysis of spatial positions of the cells from frame to frame. Generally speaking, it tries to link each cell centroid in a current frame to the spatially nearest centroid in the next frame. However, this tracking approach may easily drift when images suffer from clutter, or the frame rate is considerably low with cells moving fast. As a result, new methods started to be used in order to overcome these problems. They extend the characteristic

of "nearest" by including new features, such as intensity similarity, morphology, volume, orientation, among others, and consequently help to reduce the chances of ambiguity. An example of a similar approach is the use of these features with the well-known mean-shift method (CHEN et al., 2009; DEBEIR et al., 2005), which is an iterative method for locating the maxima of a probability density function of the data.

Many of the cited approaches for cell detection can also be extended to the idea of online tracking. For instance, centroid trackers (GHOSH; WEBB, 1994; DIVIETRO et al., 2001), TM trackers, and deformable model trackers, basically apply their detection step in one frame and then use this information to initialize the detection in the next frame. However, when the ideal conditions to perform these steps are not satisfied, more sophisticated approaches are needed. Among others, we can cite those using gradient-vector flow (RAY; ACTON, 2004; ZIMMER et al., 2002), estimative of cell dynamics (SHEN et al., 2006; DEBEIR et al., 2004), probabilistic schemes (LI et al., 2008; CUI; ACTON; LIN, 2006), and others (XIE; KHAN; SHAH, 2009; DUFOUR et al., 2005).

Current cell tracking approaches using fluorescent microscopy as a standard imaging technique mostly participate in cell tracking challenges (CTC) (SOLÓRZANO et al., 2014) organized to attract the interest of researchers to this relevant research theme. Although these challenges[1] have quite different image datasets from ours, we suggest reading the surveys (ULMAN et al., 2017; MASKA et al., 2014; MEIJERING; DZYUBACHYK; SMAL, 2012; LI et al., 2013) for a much more complete discussion and comparison among the participants' algorithms. As stated in the previous section, these methods are formulated either as a two-step process or using a model-based representation of cell appearances and shapes (TüRETKEN et al., 2017; MASKA et al., 2014; LI et al., 2008).

3.4 Related work

In this section, we present a literature review of the approaches directly related to IVM applications since this kind of imaging presents particular challenges when compared with other cell tracking solutions. We can mention, for example, the animal's movement under the microscope, the multiple protocols on image acquisition, and the blood flow constraints as complex aspects to be considered.

Sato et al. (SATO et al., 1997) (submitted in 1994), for instance, proposed an automated system for the extraction and measurement of the velocities of adhered leukocytes in blood ves-

[1] http://www.celltrackingchallenge.net/

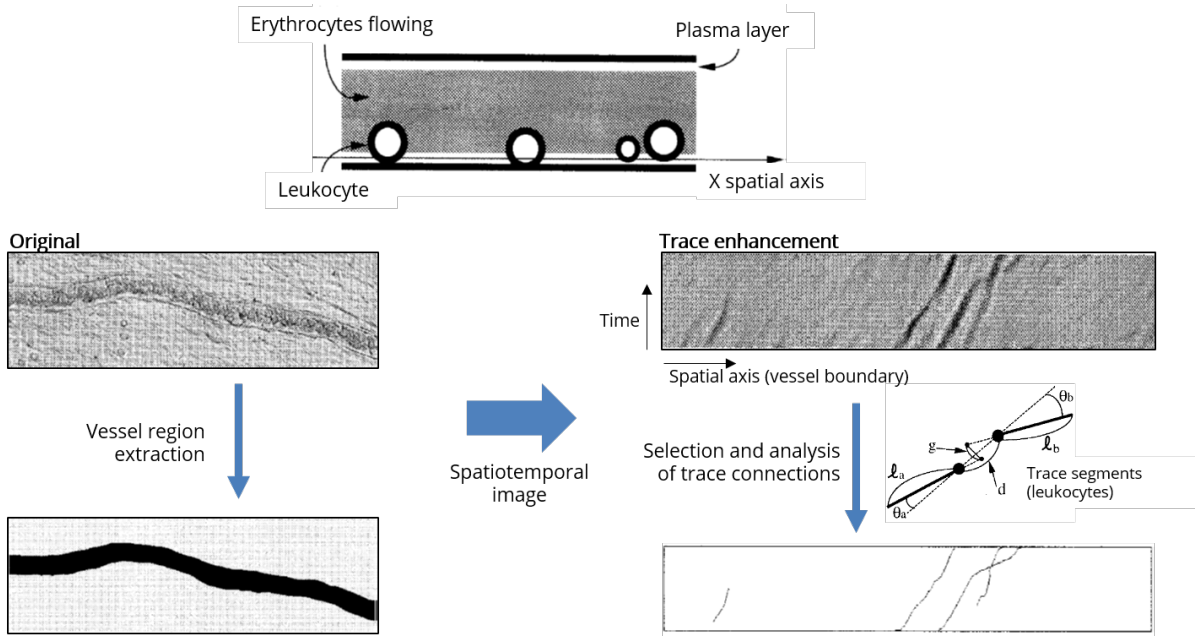


Figure 3.3: Stages of the method proposed by Sato et al. in (SATO et al., 1997).

sel walls (plasma layer) of rats' mesentery. Their method is based on the information of lines projection in a spatiotemporal framework. For that, the plasma layer was segmented, and its axes used as spatial axes in the two-dimensional spatiotemporal images. From these created images, the leukocyte movements were defined as visible traces (curved lines) in the axes. The problem of crossing traces in the spatiotemporal images was formulated as a combinatorial optimization problem and solved by using a Hopfield-based net (HOPFIELD, 1982). A pipeline of this work can be seen in Figure 3.3. As a result, it was presented a chart constructed from the different parameter values of the algorithm, where the values of true-positives (TPs) and false-positives (FPs) were mostly higher than 70% and lower than 5%, respectively. A correlation coefficient value was also calculated between the leukocyte velocities found manually and automatically by the method, resulting in a value of 0.929. Even exploring only the cells located at the vessel borders, Sato's work was able to assess the leukocyte movements separately in the plasma layer, which can be useful in the analysis of cell migration.

In 1995, Sato et al. published a paper (SATO et al., 1995) that aimed to improve some limitations of their previous work (SATO et al., 1997). For that, the authors created more elaborated filters and analyzed 2D+t spatiotemporal images, in which the axes were parallel and vertical to the vessels. With this new approach, it was possible to extend the leukocyte analysis to the center of vessels and not only to the plasma layer regions. Gabor filters were used in this work to detect and segment moving objects. After applying the filters, they used region growing algorithms to improve the identification of each leukocyte trace. The steps followed by Sato et al. are summarized in Figure 3.4. In addition to the images used in the previous work,

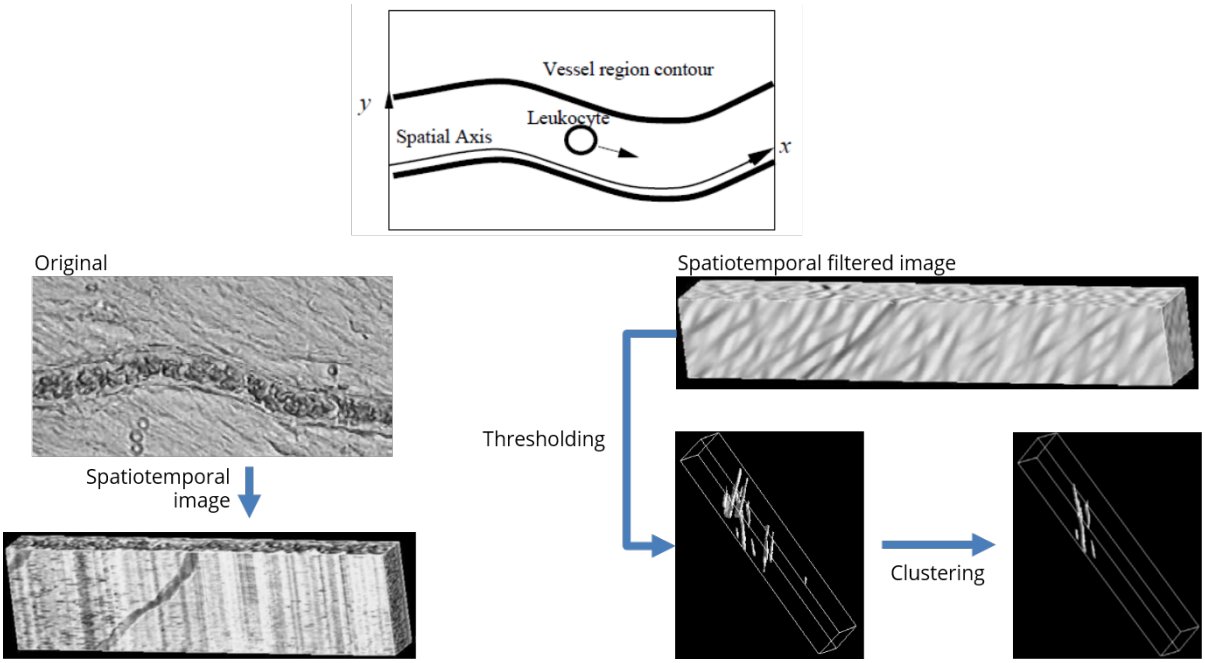


Figure 3.4: Stages of the method proposed by Sato et al. in (SATO et al., 1995).

the authors also incorporated images from human retina and fluorescent angiography. However, the results presented were only visual (qualitative).

Tang et al. (TANG et al., 2002), for instance, conducted experiments in 100 videos of intravital microscopy using well-settled techniques for in vitro analyses: the correlation tracker, the centroids tracker, and the snakes-based tracker. They assessed these methods by using the average number of frames tracked (44.1 frames for correlation tracker, 9.7 for centroids tracker, and 80 for snakes) and the root mean square error (RMSE) calculated in micrometers (25 μm for correlation tracker, 43.3 μm for centroids tracker, and 2.8 μm for snakes) over the distance between the detected cell and its correspondent position in the manual annotation. The authors found that snakes-based tracker overperformed the other methods when applying the approach in real IVM images.

The same techniques used by Tang et al. (TANG et al., 2002) were also applied by Goobic et al. (GOOBIC et al., 2001) in videos from in vivo analyses with TNF-α[2] treatment performed in the cremaster muscle of rats. The authors also used super-centroid and super-correlation trackers, which are modifications of the traditional ones. Their results, summarized in Table 3.2, include the average percentage value of frames tracked and the average RMSE value for each one of the five evaluated techniques.

The idea of adaptive template matching (ATM), already addressed in the super-centroid

[2] *Tumor Necrosis Factor* α: treatment of venules that is capable of increasing the inflammatory response and slowing down the rolling cells.

Table 3.2: Results obtained by Goobic et al. (GOOBIC et al., 2001).

Technique	Average of frames tracked (%)	Average RMSE (μm)
Centroid tracker	39.01	6.22
Correlation tracker	71.11	3.41
Super-centroid tracker	61.23	4.87
Super-correlation tracker	90.41	1.94
Active contours	100.00	0.33

and super-correlation methods by Goobic et al. (GOOBIC et al., 2001), was coarsely applied by Acton et al. (ACTON; WETHMAR; LEY, 2002) in the task of tracking leukocytes in IVM studies performed in the cremaster muscle of rats. For that, they worked on experiments with and without TNF-α treatment. In addition to the adaptive templates, the proposed approach involved frames registration using edges information, noise reduction by morphological filtering, and the application of the Kalman filter algorithm to handle occlusion. Firstly, they drew a window around the manually selected point for the creation of an initial template. Templates in the next frames (adapted) were then updated according to their predecessors. The results obtained can be seen in Table 3.3.

Table 3.3: Results obtained by Acton et al. (ACTON; WETHMAR; LEY, 2002).

Evaluation measure	With TNF-α	Without TNF-α
Average rolling velocity (μm/s)	5.6$\pm$0.4	20.3$\pm$0.4
Deviation of average velocity (%)	6.9	5.6
Mean absolute difference between the centroids (μm)	1.2	4
RMSE velocity (%)	12	8

According to Acton, these results were better than those from the centroids tracker. Although the proposed method recognized possible morphological changes that leukocytes may have during an IVM experiment, it did not handle the manual initialization (undesirable task) and the proper template update in the cases where occlusion occurred.

Ray et al. (RAY; ACTON; LEY, 2002) used active contours for the detection and tracking of leukocytes. The energy functional of the model was formulated to incorporate the cells shape and scale constraints, while for the external energy, they chose the generalized gradient vector flow (GGVF) field (XU; PRINCE, 1998b, 1998a). The results obtained were compared with manual annotations and with the results of centroid and correlation trackers. The average RMSE values related to the detected leukocytes and those manually annotated for the images with and without TNF-α treatment were, respectively, 0.5 μm and 4.6 μm, with standard deviations of 0.2 μm and 6.2 μm. The average percentage of frames tracked reached 99.9$\pm$0.3% in treated

images and 74.4±33.4% in the untreated ones. Regarding the method drawbacks, we can cite the high computational cost, the instabilities in the initialization of boundary control points, the manual initialization limited to a margin of error from 3 to 4 pixels in leukocyte centroid, and, finally, the inability of the method to handle appearing and leaving cells in the microscope FOV.

Ray et al. (RAY; ACTON, 2004) also compared the gradient vector flow (GVF) and motion gradient vector flow (MGVF) techniques. The latter was proposed in replacement of GVF, that proved inadequate when leukocyte displacements (frame-by-frame) exceed the value of a cell radius. Tests were performed for both approaches using four different temporal resolutions that simulated the variation of cell displacements. Table 3.4 shows the results found by the authors.

Table 3.4: Results obtained by Ray et al. (RAY; ACTON, 2004).

	Average RMSE value (μm)			
Field	**30 *fps***	**15 *fps***	**10 *fps***	**7,5 *fps***
GVF	2.5	2.7	4.3	6.0
MGVF	1.6	1.5	1.8	3.9

After evaluating the results of Table 3.4, it was verified that MGVF outperformed GVF mainly in the cases where the cell displacement was higher than its radius size. However, it is necessary to know in advance the direction of blood flow to apply this technique.

Dong et al. (DONG; RAY; ACTON, 2005) chose an active contours model for the automated detection of leukocytes. The proposed model used the B-spline (MENET; MARC; MEDIONI, 1990) technique as a continuous parametric representation, and a function named gradient inverse coefficient of variation (GICOV) as a restriction to the energy functional, which assigns a score to each estimated contour. The GICOV measure can be interpreted as the ratio of the mean and the standard deviation of directional image derivatives over an entire closed contour fitted to a leukocyte boundary. Initially, to coarsely identify the leukocytes, an ellipse matching algorithm was used. Next, a B-spline snake was evolved to refine the estimated leukocyte boundaries, followed by a thresholding technique applied in the final GICOV scores. As a result, the method presented an accuracy of 78.6%, with an FP rate of 13.1% on the leukocytes detection. By using the same methodology, Sahoo et al. (SAHOO; RAY; ACTON, 2006) proposed to detect leukocytes by adopting a standard teardrop shape. The results were analyzed qualitatively and indicated a better model agreement.

In 2004, Mukherjee et al. (MUKHERJEE; RAY; ACTON, 2004) proposed the detection and segmentation of leukocytes using image-level sets. The idea was to minimize an energy functional that quantifies the quality of a given curve delineating a cell. Cell tracking was cast as a maximization problem using the similarity measure between level sets in consecutive

frames. As results, the average number of frames with cells tracked by both the proposed and correlation methods were 90% and 73%, respectively, as well as the average values of 5.45 μm and 11.47 μm for the RMSE between the centroids' positions found by both techniques and the manual annotation.

Cui et al. (CUI; ACTON; LIN, 2006) used the Monte Carlo (MC) technique for the tracking of a single rolling leukocyte in IVM studies of the cremaster muscle of rats. The approach started with the alignment of frames by using the TM technique, where the employed template was obtained from the first video frame. The initial position of the leukocyte centroid was manually defined in the first three video frames since the method performs the movement prediction using two antecedent frames. Sample points around the centroid, weighted according to the cell local intensity, were then generated from this prediction. Next, the points are arranged to form radial lines around the centroid, allowing the use of a one-dimensional operator for edge detector in each segment of line created. Finally, the position of the leukocyte centroid in the next frame is defined as the sample with the highest weighted value. The results were compared with the centroid, correlation, and active contours-based (using GVF) trackers. When working with 99 microscopy videos, the proposed technique achieved an average RMSE value of 0.47 μm, with an average of 82.77% of frames tracked[3], a number of videos with the last frame tracked[4] being equal to 57, and 53 videos in which the cell was tracked in all the frames[5]. Although this approach requires initial manual annotations, the method was less sensitive to the presence of the blood vessel wall, and to additive Gaussian and salt-and-pepper noises when compared to active contours.

Ray (RAY, 2010) used a grid of points to start his detection procedure by analyzing the directional image gradient. He then applied a curve fitting as a concave cost minimization problem, followed by a calculation of the GICOV score and a simple mean shift clustering algorithm. His method demonstrated to better deal with outliers when compared with his previous techniques. One significant point to emphasize is that Ray's work concern only to the detection of cells and not to the tracking of them.

Liu et al. (LIU; LIN; ACTON, 2012) introduced a target motion prediction model based on a Bayesian approach to predict cell positions. This prediction was made by griding an ellipsoid around the target cell and generating weighted samples using their distances and visual features. Unlike the Monte Carlo tracker (CUI; ACTON; LIN, 2006), where samples are randomly

[3]Percentage of frames tracked: the number of frames that a cell is tracked divided by the number of frames in the video.

[4]Last frame tracked: if the last frame in the video sequence is tracked, the authors regard the sequence as "last frame tracked".

[5]All frames tracked: if all the frames in the video sequence were tracked, the authors considered the sequence as one with "100% frames tracked".

created, in this grid-based Bayesian (GBA) approach, samples are generated in a predicted position. Their results in a single cell tracking process revealed that the proposed approach is much faster than snake (RAY; ACTON; LEY, 2002) and MC (CUI; ACTON; LIN, 2006) trackers and, at the same time, is significantly more accurate and more robust, as illustrated by the resulting values in Table 3.5. The authors performed their experiments in 98 videos of IVM, each of which consisting of 91 frames. For these videos, they tested a registration framework based on the TM technique and showed the results for both cases, with and without registration.

Table 3.5: Results obtained by Liu et al. (LIU; LIN; ACTON, 2012) in experiments using a single cell.

	Videos registered			**Videos non-registered**		
Evaluation measure	**Snake**	**MC**	**GBA**	**Snake**	**MC**	**GBA**
Frames tracked (%)	71.7	82.7	96.4	70	55.5	93.6
Videos with all frames tracked	44	57	82	41	32	75
Average RMSE (pixels)	1.84	1.64	1.33	1.9	2.26	1.37

Although their method partially handled occlusions via analysis of the past frames, all cells in the first frame must be manually annotated before executing the algorithm. Another limitation of their method is that if a new cell appears in the FOV, it will not be considered in the analysis.

In 2013, Huang et al. (HUANG et al., 2013) proposed a method to analyze the dynamic behavior of single lymphocytes by combining shape features, cell deformation, and intracellular motion. For that, they segmented and tracked cell boundaries by using active contour models (LI; ACTON, 2007). However, in their work, the dynamic analysis was performed using only a single cell at a time, which might be a problem if a high number of cells with different trajectories is considered in the analysis.

3.4.1 The use of artificial neural networks

Because the aforementioned techniques heavily rely on data representations and make strong assumptions of cell features, they are conditioned to a specific scenario and can not be applied to multiple types of cells or imaging modalities (LECUN; BENGIO; HINTON, 2015). Artificial neural networks (ANN), on the other hand, have the ability to learn task-specific feature representations from raw data and are generally superior to handcrafted features.

Egmont-Petersen et al. (EGMONT-PETERSEN et al., 2000) adopted ANN to detect and track leukocytes in IVM studies of the mesentery of mice and rats. In this work, the authors compared the application of an ANN using two training datasets collected from real and syn-

thetic images of cells. A stochastic model created the synthetic images to mimic the intensity distribution of the leukocytes, as illustrated in Figure 3.5.

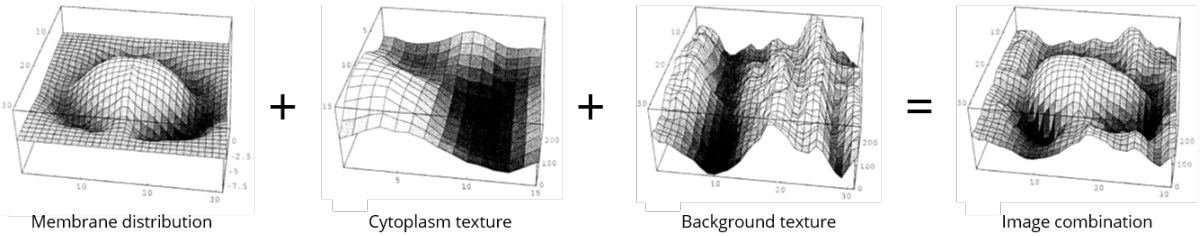


Figure 3.5: Model developed in (EGMONT-PETERSEN et al., 2000) to fashion the intensity distribution of the leukocytes and create synthetic images for training the ANN.

A classification process indicated whether an image region of 13×13 pixels belonged to a cell or not. The average of the areas under receiver operating characteristic (ROC) curves using the proposed method was 0.9 for the synthetic images and 0.71 for the real ones. However, the authors needed to create a training dataset for the ANNs, and the cell tracking procedure was not handled in their work.

Eden et al. (EDEN et al., 2005) also resorted to the use of ANNs for the detection and tracking of leukocytes in IVM. The proposed cell detection approach started using a motion detection algorithm based on the subtraction of the image background. After this rough detection, they selected only the cells inside the vessel region and used an ANN for the cells classification (as a target or a non-target), which have afterward their points analyzed by a clustering strategy. To overcome the problem of cell occlusion in the video images, the authors used a method proposed by Chetverikov et al. (CHETVERIKOV; VERESTÓI, 1999; CHETVERIKOV, 2001). With these techniques, it was possible to achieve the detection and tracking of leukocytes in 9 colored videos of rats' mesentery. In addition to an analysis of vessel segmentation, the work resulted in a correlation coefficient measure of 0.85 (BALDI; BRUNAK, 2001) when applied in the training dataset for the detection of leukocytes, and in a value of compatibility over 97% between manual and computer extracted statistics in the average velocity of cells found in each video frame. The algorithm was also compared with another method of motion correspondence (IPAN, (VERESTÓY; CHETVERIKOV, 1998)) and tested with the approach of virtual flow enabled and disabled (accuracies of 88% and 94%, respectively).

The use of these shallow ANN architectures, however, may not represent complex features, resulting in a low level of generalization and weak learning of data representations. On the other hand, deep neural networks (DNN) have demonstrated a remarkable ability to learn abstract feature representations hierarchically while requiring less human interventions and expertise (CRUZ-ROA et al., 2013; XIE et al., 2018; GOODFELLOW; BENGIO; COURVILLE, 2016). Among different architectures of DNNs, the convolutional neural networks (CNN) are the most

popular ones in numerous tasks of image analysis (GREENSPAN; GINNEKEN; SUMMERS, 2016; LECUN; BENGIO; HINTON, 2015; LECUN et al., 1998).

Although not dealing with IVM images directly, Akram et al. (AKRAM et al., 2017) proposed a joint cell detection and tracking method based on a CNN for cell candidate bounding boxes and another CNN for cell segmentation masks. The network structure in (AKRAM et al., 2017) has a similar shape as U-Net (RONNEBERGER; FISCHER; BROX, 2015) but with fewer layers and a fewer number of feature maps. Ronneberger et al. (FALK et al., 2019; RONNEBERGER; FISCHER; BROX, 2015), inspired by (LONG; SHELHAMER; DARRELL, 2015), proposed a network named U-Net that can be trained end-to-end for the detection and segmentation of cells in microscopy images. The U-Net uses a fully convolutional neural network (FCN) and incorporates high-resolution features from contracting layers to the upsampled output, allowing the network to propagate context information. Between their main contributions, Ronneberger et al. introduced the use of elastic deformations for data augmentation in microscopy imaging, which is useful to simulate local spatial variations in the images.

Software packages like Aivia[6], FindMyCells[7] (SULEYMANOVA et al., 2018), and CellProfiler[8] also use deep learning models, but they often require a training process from scratch or a related dataset. Except for U-Net, all these models rely on basic geometric and intensity transformations for data augmentation. FindMyCells, for example, is based on DetectNet (TAO; BARKER; SARATHY, 2016) augmentation method, which applies image crop, shift, scale, flip, rotation, and desaturation techniques, while CellProfiler utilizes part of the U-Net approach, with image rotations, shifts, and drop-out layers. We refer the interested reader to more comprehensive surveys (XIE et al., 2018; XING et al., 2018) for further details about general cell detection, classification, segmentation, and tracking using deep learning in microscopy images.

3.4.2 Our previous works

In our previous works, we proposed several methods for leukocytes detection in IVM. We started with the work (FREIRE et al., 2012), where we tested the Hough transform technique (ATIQUZZAMAN, 1999; HIERKEGAARD, 1992) to identify circular-like structures in 15 images of mice brain. This work compared the Hough transform application using three different strategies: (1) after the application of Sobel filter and Canny edge detector (GONZALEZ; WOODS, 1992), (2) after the application of Laplace filter and Canny edge detector, and (3) in

[6] https://www.drvtechnologies.com/aivia
[7] http://www.findmycells.org
[8] https://cellprofiler.org

the raw intensity values. The resulting accuracy values for the three strategies were 16.31%, 30.32%, and 84.2% on average, respectively, showing a better performance of the Hough transform in the raw images.

In (PINTO et al., 2015), we applied blind deconvolutional techniques for image deblurring and tested two methods for leukocytes detection, the conventional TM, and a technique based on the local phase symmetry (LPS). The results reached F_1-score values of 0.72 and 0.80 for the TM and LPS techniques, respectively, considering 15 images extracted from the mice's brain after deblurring.

In the works (ELISA DE SOUZA et al., 2015, 2016), we used second-order momentum matrices obtained by using the phase congruency technique to detect leukocytes also in the brain images of mice. The best result achieved for this technique was a F_1-score value of 0.79.

Still using images from IVM studies in the brain of mice, we proposed an approach based on the local analysis of eigenvalues obtained from Hessian matrices (GREGÓRIO DA SILVA; CARVALHO-TAVARES; FERRARI, 2015, 2016, 2019). Firstly, we applied a method for leukocytes detection using a frame-based approach, where blob-like structures were enhanced and detected by Frangi's algorithm (FRANGI et al., 1998). Next, we used a modified version of the same algorithm to enhance 3D tubular-like structures representing our cell trajectories directly in spatiotemporal images (GREGÓRIO DA SILVA; CARVALHO-TAVARES; FERRARI, 2019). This work was not only able to detect leukocytes but also track them over the video frames. Regarding the main results, our methods achieved F_1-score values of 0.84 and 0.88 for the 2D and 3D approaches, respectively.

Despite promising results, our previous approaches were developed to enhance and detect circular-like cells in IVM images from the CNS of mice. Therefore, these methods mostly fail when either the cells have distinct appearances or came from different image acquisition protocols. For this reason, we tested different methods invariant to such cell changes in this work. A better discussion about our findings is made in Chapter 7.

3.4.3 Comparative summary

A comparative summary of the works directly related to IVM applications analyzed in this subsection is presented in Table 3.6. This summary table considers the capability of methods in (1) detecting and tracking single or multiple cells, (2) performing cell detection, (3) performing cell tracking, and (4) handling cell occlusions. Also, it shows the organs used in each study.

Table 3.6: Comparative summary of related works. The algorithm type is represented as U: unicellular, and M: multicellular. Symbol × denotes whether method performs the tasks of detection, tracking and its ability to deal with cell occlusions.

Reference	Approach	Type	Detection	Tracking	Occlusion	Organ
(SATO et al., 1997)	Line projection in a spatiotemporal framework (2D)	M	×	×	×	Mesentery
(SATO et al., 1995)	Gabor filters in a spatiotemporal framework (3D)	M	×	×	×	Mesentery
(EGMONT-PETERSEN et al., 2000)	Artificial neural networks and synthetic images	U	×			Mesentery
(GOOBIC et al., 2001)	Centroid, super-centroid, correlation, super-correlation, and snake trackers	M	×	×		Cremaster
(TANG et al., 2002)	Centroid, correlation and snake tracker	M	×	×		Cremaster
(ACTON; WETHMAR; LEY, 2002)	Adaptive template matching and Kalman filter	U	×	×	×	Cremaster
(RAY; ACTON; LEY, 2002)	Active contours (shape and scale constraints) and GGVF	U	×	×	×	Cremaster
(RAY; ACTON, 2004)	Motion gradient vector flow and gradient vector flow	U	×	×		Cremaster
(MUKHERJEE; RAY; ACTON, 2004)	Image-level sets	M	×	×		Cremaster
(DONG; RAY; ACTON, 2005)	Active contours (B-spline) and GICOV	M	×			Cremaster
(EDEN et al., 2005)	Background subtraction and artificial neural networks	M	×	×	×	Mesentery
(CUI; ACTON; LIN, 2006)	Monte Carlo	U	×	×		Cremaster
(SAHOO; RAY; ACTON, 2006)	Active contours (B-spline) and GICOV (teardrop shape)	M	×			Cremaster
(RAY, 2010)	Parametric curve fitting and GICOV score	M	×			Cremaster
(LIU; LIN; ACTON, 2012)	Grid-based Bayesian model	M	×	×	×	Cremaster
(FREIRE et al., 2012)	Hough transform	M	×			Brain
(HUANG et al., 2013)	Active contours	U	×	×		Skin
(PINTO et al., 2015)	Template matching and local phase symmetry	M	×			Brain
(ELISA DE SOUZA et al., 2015, 2016)	Template matching and phase congruency	M	×			Brain
(GREGÓRIO DA SILVA et al., 2015, 2016)	Hessian-based 2D	M	×			CNS
(GREGÓRIO DA SILVA et al., 2019)	Hessian-based 2D+t	M	×	×	×	CNS

3.5 Final considerations

In this chapter, we presented a theoretical background for the understanding of the main strategies addressed in multiple object tracking, as well as the explanation of some particular techniques to detect and track cells in microscopy image sequences.

Based on our review of state of the art for IVM automated algorithms, we did not find any related study using IVM images from the CNS (animal's brain or spinal cord) beyond ours or from more than one organ, which could add numerous challenges to the algorithms. Also, the 2D+t spatiotemporal idea was only explored by Sato et al. (SATO et al., 1995) for the tracking of leukocytes in IVM studies. However, their work limited the results to a visual inspection and used a technique with a high computational cost for the detection. Finally, many studies are concerned with the processing of only one cell at a time, while others do not handle occlusion, which is a significant issue in real scenarios.

In this sense, we believe the use of multiple image features and a combination of 2D and 2D+t processing can significantly improve the current results for the multiple cell tracking approach. In the next chapter, we start to describe our methodology in this research work.

Chapter 4

PREPROCESSING

This chapter details all the techniques employed in our preprocessing pipeline to improve the quality of data to the next processes. It also describes the metrics used to evaluate the preprocessing outputs visually and quantitatively.

4.1 Preprocessing pipeline

The preprocessing stage is the first step of our automatic computational pipeline. It is composed of particular techniques that intend to solve the main inherent problems related to the IVM image acquisition and, consequently, improve the quality of the data for the next stages. It follows a logical sequence of image corrections that include the application of conventional techniques for blurred frames removal, noise reduction, contrast improvement, video stabilization, and vessel segmentation. The sequence of techniques is illustrated in Figure 4.1.

Figure 4.1: Steps of our preprocessing stage.

Except for the techniques used to remove blurred frames and to stabilize the videos, all the remaining parameters were visually adjusted. The following sections describe all these essential steps to achieve a better cell detection and tracking.

4.2 Blurred frames removal

In the first stage, all frames strongly affected by the animal movement were detected and removed by a technique previously developed for this purpose (FERRARI et al., 2015). Al-

though image restoration techniques can be used to recover the motion-blurred frames, in this work, we have only extracted them for further analysis.

The method proposed by our research group (FERRARI et al., 2015) uses directional statistics of local energy maps obtained from the convolution of each video frame with a bank of log-Gabor filters specially designed to detect motion blur. The bank of filters was built using three scales and six orientations. The maximum response of the filters for each spatial image position (at a particular orientation along with different scales) was used to generate an angular distribution of the responses. From this distribution, a set of directional statistics was extracted and analyzed for deciding whether the motion in the image was excessive or not. The work hypothesizes that the motion blurring introduces local changes in the image texture, inserting a large number of directional information in the spectral bands that are neither high nor low, and this can be measured using directional statistics of the filters' responses. However, it is critical to notice that the removal of blurred frames from our videos was only possible because the video sample rates were sufficiently high to guarantee that, i.e., even after removing degraded images, the continuity of the cell movement is preserved and do not affect the tracking process.

4.3 Noise reduction

In the next stage, we applied the bilateral filter technique (TOMASI; MANDUCHI, 1998) to reduce noise without introducing noticeable blurring in the images and to improve the signal-to-noise ratio of the video frames. This technique replaces the value of each image pixel, $I(\boldsymbol{x})$, by the weighted average of its neighbors inside a neighborhood defined as Ω. Generally, the weights are defined using a Gaussian distribution (as in the case of this work) and are dependent not only on the Euclidean distance of pixels but also on radiometric differences, such as range differences, depth distance, and others. In addition to the systematic scanning of pixels, this technique preserves image edges and allows the adjustment of weights according to the neighborhood. The filtered image $I_{BF}(\boldsymbol{x})$ is defined as:

$$I_{BF}(\boldsymbol{x}) = \frac{1}{W_p} \sum_{\boldsymbol{x}_i \in \Omega} I(\boldsymbol{x}_i) G_{\sigma_s}(\|\boldsymbol{x} - \boldsymbol{x}_i\|) G_{\sigma_r}(|I(\boldsymbol{x}) - I(\boldsymbol{x}_i)|), \tag{4.1}$$

where $I(\boldsymbol{x})$ is the pixel intensity, and $\boldsymbol{x}$ represents the pixel coordinates in the neighbors Ω being considered. G_{σ_s} corresponds to a Gaussian kernel (spatial) with a standard deviation σ_s that smooths the differences in the pixel coordinates, and G_{σ_r} a Gaussian kernel (range) with a standard deviation σ_r that decreases the influence of pixels $\boldsymbol{x}_i$ when their intensity values differ from $I(\boldsymbol{x})$. W_p is a normalization factor which ensures that the sum of the pixel weights equals

one, i.e.,

$$W_p = \sum_{\boldsymbol{x}_i \in \Omega} G_{\sigma_s}(\|\boldsymbol{x} - \boldsymbol{x}_i\|) G_{\sigma_r}(|I(\boldsymbol{x}) - I(\boldsymbol{x}_i)|). \tag{4.2}$$

The filter parameters were experimentally adjusted to provide the best trade-off between noise reduction and low blurring effect. Consequently, the diameter of filter neighborhood and the range and spatial parameters of Gaussian kernels were set to $d = 9$, $\sigma_r = 10$ and $\sigma_s = 10$, respectively.

4.4 Contrast standardization

To diminish the photobleaching effect (ANDRESEN et al., 2012) and, consequently, to improve our frame-to-frame image registration method (described in the next subsection), the histogram matching technique proposed by Nyúl et al. (NYÚL; UDUPA; ZHANG, 2000) was applied to each pair of consecutive frames in the videos. The idea behind this approach is the pixel intensity standardization of the video frames by using the image histograms.

Given a pair of images, I_r and I_s, with their corresponding intensity ranges $[r_1, r_2]$ and $[s_1, s_2]$, the algorithm estimates two sets of l reference points defined as $\{q_{r,k} \mid 1 \leq k \leq l\}$ for the histogram of the reference image I_r, and $\{q_{s,k} \mid 1 \leq k \leq l\}$ for the histogram of image I_s. Next, the algorithm applies a sequence of linear mappings in the intervals between the reference points $\{q_{s,k}\}$ and $\{q_{r,k}\}$, i.e., mappings from $[q_{s,k}, q_{s,k+1}]$ to $[q_{r,k}, q_{r,k+1}]$, for all $k = 1, \ldots, l-1$, and also from $[s_1, q_{s,1}]$ to $[r_1, q_{r,1}]$, and from $[q_{s,l}, s_2]$ to $[q_{r,l}, r_2]$. In practice, these reference points are defined as the histogram percentiles according to the number of points chosen. For example, for $l = 3$, they would be the quartiles of the histograms.

In this work, the parameter l was experimentally set to 7, and the first frame of all pairs of consecutive frames was used as the reference image for the histogram matching.

4.5 Video stabilization

The video stabilization or temporal image registration framework developed in this work to correct for small specimen movements is comprised of four modules: metric, optimizer, interpolator, and transformation. The method, as shown in Figure 4.2, consists of finding a transformation $T(\boldsymbol{x}|\boldsymbol{p})$ to correct the misalignment between consecutive pairs of frames in the video. The set of parameters $\boldsymbol{p}$ of the transformation T is obtained iteratively by mapping all pixels from the moving frame $M(\boldsymbol{x}) = F(\boldsymbol{x}, t+1)$ to their corresponding pixels in the fixed frame

$F(\boldsymbol{x})$ so that the similarity metric $S(\boldsymbol{p}|F,M,T)$ is minimized.

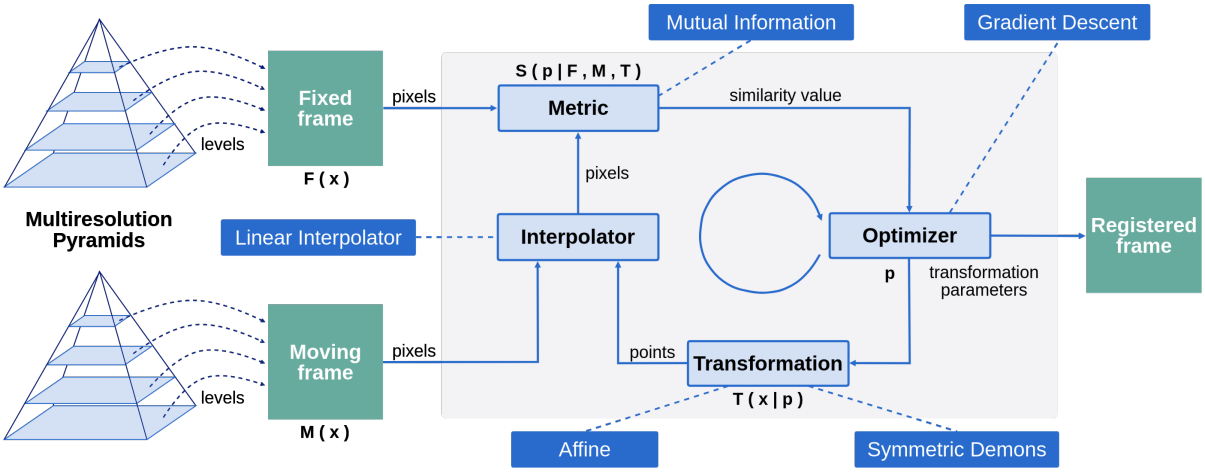


Figure 4.2: Temporal image registration framework developed to stabilize video motion due to small animal movements.

This framework uses a multiresolution approach in which the levels of a Gaussian pyramid represent images with different resolutions, as illustrated in Figure 4.2. The estimation procedure for the parameters $\boldsymbol{p}$ starts using the lowest resolution images in the top level of the pyramids, and the estimated values at this level are used as a first start to the algorithm on the next lower level (higher resolution images). This procedure is repeated until the pyramid bases (full resolution images) are reached. Each step illustrated in Figure 4.2 is described in the following subsections.

4.5.1 Metric

The metric chosen in this work was the mutual information (MI) (PLUIM; MAINTZ; VIERGEVER, 2003), which measures the statistical dependency between two data sets (fixed and moving images) by taking into account the amount of information that one random variable has over another. MI is defined in terms of entropy in the following way (PLUIM; MAINTZ; VIERGEVER, 2003):

$$\begin{aligned} S(\boldsymbol{p}|F,M,T) &= MI(F,T) \\ &= H(F)+H(M|T)-H(F,M|T), \end{aligned} \tag{4.3}$$

where $H(\cdot)$ is the entropy of a random variable (in this case, images F or M), which can be calculated from the marginal probability (normalized intensity histogram), $P(\cdot)$, of the images as:

$$H(F) = -\sum_{f\in F} P(f)\log P(f), \tag{4.4}$$

$$H(M|T) = -\sum_{m \in M} P(m|T) \log P(m|T). \tag{4.5}$$

In Equation (4.4) and (4.5), f and m represent, respectively, pixel intensities in the images F and M. The joint entropy of images F and M, which is the last term in Equation (4.3), is calculated from their joint probability distribution (joint normalized intensity histogram), $P(f,m|T)$, as:

$$H(F,M|T) = -\sum_{f \in F, m \in M} P(f,m|T) \log P(f,m|T). \tag{4.6}$$

4.5.2 Optimizer

An optimizer based on the gradient descent method (KLEIN; PLUIM; STARING, 2009) was used to search for the best set of parameters $\boldsymbol{p}$ that minimizes the similarity function S between images F and M. As a consequence of the multiresolution approach, the algorithm processing time is reduced, and the method's stability is improved since the coarser details from the top levels of the pyramids increase the chances of the gradient descent method to converge to a global minimum, providing, therefore, the estimation of a proper set of parameters in each iteration.

The choice of using the gradient descent algorithm as an optimizer in our registration framework was made because it is a low computational complexity technique, which is an important feature when processing vast amounts of data. Also, as mentioned previously, the optimization of the parameters is performed using a multiresolution framework, which increases the convergence speed and minimizes the chances of the gradient descent algorithm getting trapped in a local minimum.

4.5.3 Interpolator

Similarly to the optimizer, a linear interpolator was used in our image registration framework because of its low computational complexity concerning the number of image pixels. This module is necessary because the mapping of points from one image into another is performed in the physical coordinate system. Therefore, an interpolator method is required to put these points back in their corresponding places in the image pixel grid.

4.5.4 Transformation

Mathematically, geometrical transformations represent mappings of points from a space X of one view (moving image) to a space Y of a second view (fixed image). As indicated in Figure

4.2, our proposed framework uses two types of geometrical transformations to correct for undesirable frames misalignment caused by the animal movements. First, an affine transformation T, representing a linear combination of rotations, translations, scaling and shearing operations was applied to each position $\boldsymbol{x}$ of the moving image (herein, a point in X is represented by the column vector $\boldsymbol{x}$) to produces a transformed point $\boldsymbol{x}'$ given the set of transformation parameters $\boldsymbol{p}$,

$$\boldsymbol{x}' = T(\boldsymbol{x}|\boldsymbol{p}). \tag{4.7}$$

This transformation results in a coarse alignment between the moving and fixed images.

After this affine registration, the deformable transformation technique proposed by Thirion (THIRION, 1998) was applied to the moving image to refine the previously computed alignment. A deformable transformation consists of finding a mapping of an image $M(\boldsymbol{y})$ to an image $F(\boldsymbol{x})$ using a deformation field $u(\boldsymbol{x})$ (AVANTS; TUSTISON; SONG, 2009). The deformation is defined in the physical image space and provides the positional difference between two given images. In this way, if a feature defined in $F(\boldsymbol{x})$ has its equivalent in $M(\boldsymbol{y})$, the deformation field u in $\boldsymbol{x}$ is computed as

$$u(\boldsymbol{x}) = \boldsymbol{y} - \boldsymbol{x}, \tag{4.8}$$

and, therefore, it can be applied to deform an image M into an image F by

$$M_{deformed} = M(\boldsymbol{x} + u(\boldsymbol{x})). \tag{4.9}$$

The idea of the deformable transformation technique (THIRION, 1998) to compute the deformation field is that a regular grid of forces deforms an image by pushing the contours to the normal direction of each grid point. The orientation and magnitude of the displacement vectors are derived from the instantaneous optical flow equation (HORN; SCHUNCK, 1981). In this case, the conservation of gray level intensity of the moving points is assumed to be constant, i.e., $I(\boldsymbol{x}(t),t) = const$, with $\boldsymbol{x}(t)$ representing the coordinates of the point at time t.

In our case, two consecutive pairs of frames (the fixed frame denoted by $F(\boldsymbol{x})$ and the moving frame $M(\boldsymbol{x})$) are compared to allow the computation of a displacement vector $u(\boldsymbol{x})$ that let $M(\boldsymbol{x})$ closer to $F(\boldsymbol{x})$. Then, giving that $F(\boldsymbol{x})$ and $M(\boldsymbol{x})$ are separated by one time unit, $\partial I/\partial t = M(\boldsymbol{x}) - F(\boldsymbol{x})$ and $u(\boldsymbol{x}) = d\boldsymbol{x}/dt$ is the instantaneous velocity of $F(\boldsymbol{x})$ to $M(\boldsymbol{x})$, thus

$$u(\boldsymbol{x}) \cdot \nabla F(\boldsymbol{x}) = -\left(M(\boldsymbol{x}) - F(\boldsymbol{x})\right). \tag{4.10}$$

In this case, $u(\boldsymbol{x})$ is considered the velocity because the images are two consecutive frames, i.e., $u(\boldsymbol{x})$ is the displacement during the time interval between the two image frames (THIRION,

1998). It is well known in optical flow literature that Equation (4.10) is not sufficient to define the velocity $u(\boldsymbol{x})$ locally and, in this case, it is usually determined using some form of regularization. For registration, the projection of the vector on the direction of the intensity gradient is used as:

$$u(\boldsymbol{x}) = -\frac{(M(\boldsymbol{x}) - F(\boldsymbol{x}))\,\nabla F(\boldsymbol{x})}{\|\nabla F\|^2 + (M(\boldsymbol{x}) - F(\boldsymbol{x}))^2/K}, \tag{4.11}$$

where $K > 0$ is a normalization factor that accounts for the units imbalance between intensities and gradients. This factor is computed as the mean squared value of the pixel spacings. The addition of K makes the force computation to be invariant to pixel scaling in the images. In order to provide a level of symmetry in the force calculation, a variation of the Equation (4.11) was used. In this case, the gradient of the deformed moving image is also involved, so that

$$u(\boldsymbol{x}) = -\frac{2 \cdot (M(\boldsymbol{x}) - F(\boldsymbol{x}))\,(\nabla F(\boldsymbol{x}) + \nabla M(\boldsymbol{x}))}{\|\nabla F + \nabla M\|^2 + (M(\boldsymbol{x}) - F(\boldsymbol{x}))^2/K}. \tag{4.12}$$

An elastic-like behavior, smoothing the deformation field with a Gaussian filter between iterations, was included in the implemented algorithm to make it more natural.

4.6 Vessel segmentation

Another essential step in our preprocessing stage is the segmentation of the region where the leukocytes' events are occurring. With this segmented region, we can reduce the algorithm processing time and the number of false-positives (cells wrongly detected). The vessel region in the images was extracted by assessing the temporal variance of each pixel, as proposed in (SATO et al., 1997). The rationale, in this case, is that the gray level of each pixel in a vessel region, where blood cells are flowing continuously, will vary significantly within frames while the gray level in other regions will be almost constant, i.e., the gray level variance tends to be large in the vessel region and small in other regions. However, we observed that the Sato's approach alone was not sufficient to segment the venule's region correctly since his technique also responds to motion in thinner capillaries in the IVM images. Therefore, we added other steps to Sato's approach to overcome this problem.

After the computation of temporal variance for the video, we blurred the resulting image with a Gaussian kernel. Next, we created a binary image using a global thresholding technique defined empirically for each video, and then we applied a morphological opening operation with a circular structuring element. All the parameters were visually defined according to the image contrast and the thickness of small vessels in the image so that to discard thinner structures

that are not useful for the detection and tracking processes. As a final step, we selected the larger region in the image as the one to be processed, i.e., the blood vessel region in which the leukocyte recruitment will be analyzed. In the cases of colored images, we transformed the video frames into the HSV color representation and used only channel V for the process as it presents a better response to vessel regions. The resulting image is used for further analysis to reduce the number of false-positive cells automatically detected in the videos.

The outputs for each step of the vessel segmentation process can be seen in Figure 4.3.

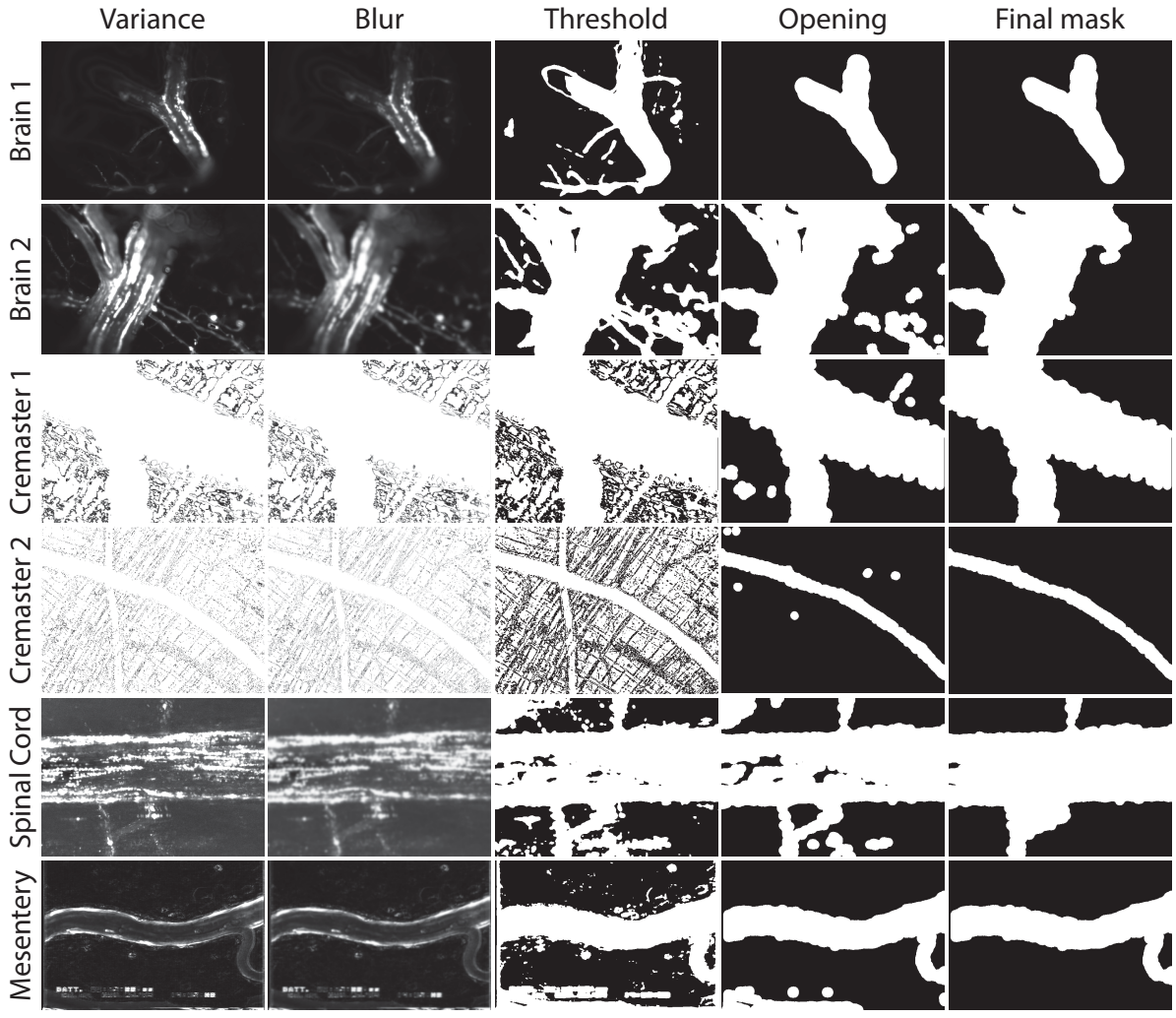


Figure 4.3: The outputs for each vessel segmentation step of our proposed approach. Each line of the figure corresponds to an IVM video used. Each column of the figure shows a particular processing step, such as, from left to right: 1st) the variance image, 2nd) blurred image, 3rd) binary image, 4th) morphological opening output, and 5th) the final mask obtained by selecting the larger binarized region of the image.

4.7 Methods of evaluation

Since the results of leukocytes detection and tracking are directly affected by the video stabilization process applied in our preprocessing stage, we evaluated the registration pipeline visually and quantitatively using the following methods.

4.7.1 Line projection

The line projection technique allows visual analysis of the whole video. This technique creates a two-dimensional image by stacking all the central lines of video frames, i.e., each row of the image created corresponds to the central line profile extracted from each frame over the entire video. As a result, we have an image of size $w \times n_f$, where w is the frame width, and n_f is the total number of frames in the video. In this sense, when analyzing the intensity profiles stacked, we can observe how aligned (or misaligned) the edges of image objects are.

4.7.2 Peak signal-to-noise ratio – PSNR

We can define the PSNR term as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation in the video frames. It is vulnerable to the distortions caused by the pixels misalignment, like the spatial changes, rotation, and resizing (KORHONEN; JUNYONG, 2012). The PSNR measure is calculated as:

$$PSNR = 20\log_{10}\left(\frac{MAX_I}{RMSE}\right), \tag{4.13}$$

$$RMSE = \sqrt{\frac{1}{mn}\sum_{1}^{m}\sum_{1}^{n} \parallel I_{t+1}(x,y) - I_t(x,y) \parallel^2}, \tag{4.14}$$

where MAX_I represents the maximum signal value existing in the image I, and $I_t(x,y)$ represents the image pixel at moment t and position (x,y). The number of rows and columns are given, respectively, as m and n. In this work, the PSNR measure was calculated for the residual images resulted from the subtraction of consecutive pairs of frames. In this case, if the residue is low, then the PSNR value will be high, indicating a proper alignment between the pair of frames. Otherwise, if the residue is high, meaning a high level of misalignment, then the PSNR value will be low.

4.8 Final considerations

In this chapter, we described the main problems that may affect the quality of IVM images and provided details of the techniques developed and used to correct them. These techniques, which includes blurred frame detection, noise reduction, contrast standardization, video stabilization, and vessel segmentation, were used as the first stage of our automatic computational pipeline. We also presented the methods used for quantitative assessment of some of these techniques.

Chapter 5

DETECTION – 2D PROCESSING

This chapter presents the three approaches used in our 2D processing stage for leukocytes detection as well as the metrics used for their evaluation. The application of different methods for cell detection in this work shows not only the variability of approaches existing in the literature but also the reliability of our proposed pipeline by allowing the use of any detection technique as a plug-in style. We named our developed approaches as MTM-PCA, MTM-DCNN, and DCNN. The basic concepts of each one of them are, respectively, the application of multiple template matching technique in hand-crafted feature images after the application of principal component analysis, the application of multiple template matching technique in the outputs of pre-trained convolutional neural network layers, and the use of a deep convolutional neural network after a fine-tuning process.

5.1 MTM-PCA: Multiple Template Matching with Principal Component Analysis

Our first detection approach named multiple template matching with principal component analysis (MTM-PCA) was developed to be a simple and straightforward method that uses a set of hand-crafted image features. Figure 5.1 illustrates a flowchart of the framework utilized for this approach.

The processing starts with the computation of image features for all video frames, resulting in a set of videos, each one obtained for a particular feature. To reduce the number of feature dimensions, we extracted the first frame of each feature-video and used this data as an input to the PCA technique. With the PCA eigenvectors calculated, we then project the remaining frames into the new PCA basis. A set of ROIs containing leukocytes of fairly different appearance is then manually extracted from the first video frames, creating a vector of leukocyte-templates

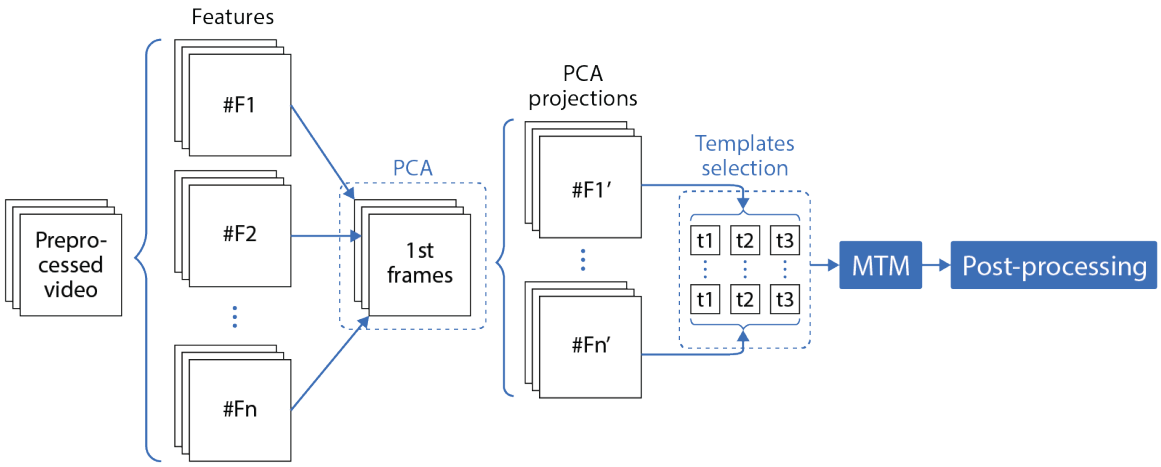


Figure 5.1: Pipeline of the first approach for leukocytes detection.

that will be used later in a multiple template matching technique (MTM). Each module of Figure 5.1 is detailed in the following subsections.

5.1.1 Features

To better contribute to the detection process, we chose image features that could address different aspects of the cells. In addition to the original gray-level intensity, we selected features related to the object contour, texture, geometry, scale, and intensity variations. The methods employed to obtain these features are described as follows.

Gray level intensity The gray-level intensity of the pixels was used as the original feature in the next processing steps. In the cases of colored images (I_{RGB}), they were converted to a grayscale representation (I_{gray}) using the following transformation:

$$I_{gray}(\boldsymbol{x}) = 0.299 \times I_R(\boldsymbol{x}) + 0.587 \times I_G(\boldsymbol{x}) + 0.114 \times I_B(\boldsymbol{x}). \tag{5.1}$$

This image transformation is known as luma or luminance (POYNTON, 1997). It is commonly used in standard color TV and video systems and is related to signal brightness.

Edges The image feature based on the edges was created using the Sobel operator (SOBEL, 1990). This edge operator is typically used to find the approximate absolute gradient magnitude at each point in a grayscale image. It consists of a pair of 3×3 convolution kernels, as illustrated in Figure 5.2, designed to respond maximally to edges running vertically and horizontally, i.e.,

1	2	1
0	0	0
-1	-2	-1

(a) Q_y

1	0	-1
2	0	-2
1	0	-1

(b) Q_x

Figure 5.2: Examples of the kernels used in Sobel operator technique to enhance image edges.

to generate vertical (Q_y) and horizontal (Q_x) derivatives.

The final image is produced by combining the two derivatives to find the absolute magnitude of the gradient at each point, which is given by:

$$Q = \sqrt{Q_x^2 + Q_y^2}. \tag{5.2}$$

Texture In this work, image texture features were obtained using the gray level co-occurrence matrix (GLCM) algorithm. This technique can be defined as "a tabulation of how often different combinations of pixel brightness values (gray-levels) occur in an image" (CONNERS; TRIVEDI; HARLOW, 1984; CONNERS; HARLOW, 1980; HARALICK, 1979; HARALICK; SHANMUGAM; DINSTEIN, 1973). The algorithm's idea is to create a matrix containing the frequency of gray-level intensity variations between a reference pixel and its neighbors. In our case, the texture feature images were all obtained using a one-pixel offset, i.e., the GLCM algorithm considers the relationship between two pixels at a time (a reference pixel and one of its immediate neighbors). Also, in this work, the gray-level co-occurrence matrices were first computed considering the pixel relationships in four different spatial directions (0, 45, 90, and 135 degrees) and then averaged to create a final matrix from which the texture features were obtained.

After transforming the GLCM matrix into a symmetric and normalized form, we can analyze it using different measures that summarize the local texture information into a single value. The metrics used in this work to characterize image texture information were chosen over several other metrics by visual analysis. They are described below, where $g(i, j)$ is the element in cell (i, j) of the normalized GLCM.

- Difference moment:

$$\sum_i \sum_j \frac{g(i,j)}{1+(i-j)^2}; \tag{5.3}$$

- Inertia (or contrast):

$$\sum_i \sum_j (i-j)^2 g(i,j); \tag{5.4}$$

- Haralick's correlation:

$$\frac{\sum_i \sum_j (ij) g(i,j) - \mu_t^2}{\sigma_t^2}. \tag{5.5}$$

Above, μ_t and σ_t are the mean and standard deviation of the row (or column, due to symmetry) sums.

Blobness We can easily incorporate the cell shape information as a feature by treating the problem of leukocyte detection as a Hessian eigenvalue analysis (see Appendix A). In this way, prior information can be used as a consistency check to discard structures present in the dataset with a different polarity than the one sought. Isotropic structures, for instance, are associated with eigenvalues having a similar non-zero magnitude. Accordingly, we shall look for structures whose λ_1 and λ_2 are both, simultaneously, high and negative. By considering this, a blobness measure function $B_\sigma(\boldsymbol{\lambda})$ (FRANGI et al., 1998; GREGÓRIO DA SILVA; CARVALHO-TAVARES; FERRARI, 2015), defined as

$$B_\sigma(\boldsymbol{\lambda}) = \begin{cases} \left(1 - \exp\left(-\frac{R_A^2}{2\alpha^2}\right)\right)\left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right), & \text{if } \lambda_1 < 0 \text{ and } \lambda_2 < 0, \\ 0, & \text{otherwise,} \end{cases} \tag{5.6}$$

was created by using the ratio and magnitude strength of the eigenvalues and used to enhance blob-like structures representing the leukocytes in the images. In Equation (5.6), $R_A = |\lambda_1|/|\lambda_2|$ helps to distinguish between plate-like and line-like patterns. Besides, the measure $S = \sqrt{\lambda_1^2 + \lambda_2^2}$ helps to reduce the influence of noisy background pixels in the blobness measure function since they present low eigenvalues and, therefore, will result in a low value of the second term in Equation (5.6). Parameters α and c can be adjusted to control the sensitivity of the filter components and, in this work, they were set, respectively, to 0.5 and one-tenth of the maximum value of the Laplacian of the image, as suggested in (DZYUBAK; RITMAN, 2011). The σ footer in B_σ indicates that the blobness measure is computed on a smoothed version of the image and, therefore, it is representative of the variations of image intensity at the spatial scale σ. The function was evaluated at a range of spatial scales (σ), varying between 1 and 8. This range was based on the size of the observed leukocytes in the images. The maximum response at every pixel was taken as

$$B(\boldsymbol{\lambda}) = \max_{\sigma \in [\sigma_{min}, \sigma_{max}]} B_\sigma(\boldsymbol{\lambda}). \tag{5.7}$$

As a result, we have an image sequence containing all possible blob-like structures enhanced by the algorithm. As mentioned previously, because the leukocytes may be positioned above and below the microscope focal plane, their apparent size and, especially their contrast,

can significantly change in the images. For this reason, the multiscale blob enhancement will produce real-valued responses at our feature video images.

5.1.2 Principal component analysis – PCA

To use only the relevant information provided by the computed video features, we tested our approach using a popular method in machine learning for dimensionality reduction, the principal component analysis (PCA) (GONZALEZ; WOODS, 1992).

The goal of the PCA method is to seek the most accurate data representation in a lower-dimensional space by using the variance of the data. Thus, PCA can be thought of as finding a new orthogonal basis by rotating the old axes until the directions of maximum variance are found. It is composed of u principal components that are orthogonal, uncorrelated, and represent the direction of the maximum variance of the data. By choosing a number for u lower than the original dimensionality (k), we are reducing the data to u dimensions. This process for two-dimensional data is illustrated in Figure 5.3.

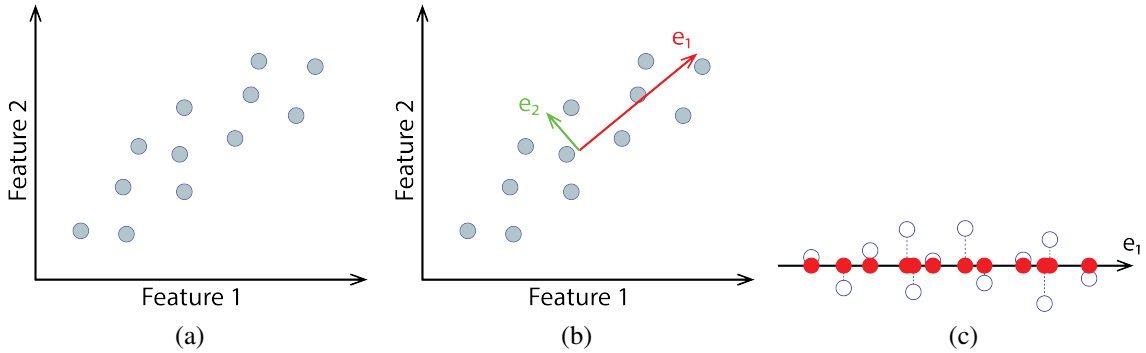


Figure 5.3: Processes performed by PCA technique. (a) Original data distribution for features 1 and 2, (b) principal components found by the PCA algorithm, and (c) data projection.

The first principal component of the PCA space is the one representing the maximum variance of the data, the second component is perpendicular to the direction of the first one and has the second largest variance, and so on. They are calculated as the eigenvectors of the covariance matrix of the data.

Below are the steps for calculating PCA in three-dimensional data, but the same idea applies to any number of dimensions.

1. *Assemble a data matrix:* the first step is to assemble all data points into a matrix where each row represents one normalized feature-frame of our approach, and each column corresponds to one data point in the three-dimensional space or the corresponding pixels

in the feature-frames.

$$\Phi = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ z_1 & z_2 & \dots & z_n \end{bmatrix} \tag{5.8}$$

2. *Calculate mean:* next, we calculate the mean (μ_x, μ_y, μ_z) of all data points as:

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i,\ \mu_y = \frac{1}{n}\sum_{i=1}^{n} y_i, \text{ and } \mu_z = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{5.9}$$

3. *Subtract mean from the data matrix:* a new matrix W is created by subtracting the mean values from every data point of Φ:

$$W = \begin{bmatrix} (x_1 - \mu_x) & (x_2 - \mu_x) & \dots & (x_n - \mu_x) \\ (y_1 - \mu_y) & (y_2 - \mu_y) & \dots & (y_n - \mu_y) \\ (z_1 - \mu_z) & (z_2 - \mu_z) & \dots & (z_n - \mu_z) \end{bmatrix} \tag{5.10}$$

4. *Calculate the covariance matrix:* the covariance matrix captures the data spread information. The diagonal elements of a covariance matrix are the variances along the x, y, and z-axes. The off-diagonal elements represent the covariance between two dimensions (x and y, y and z, z and x). The covariance matrix C is calculated using the following product:

$$C = WW^T \tag{5.11}$$

5. *Calculate the eigenvectors and eigenvalues of the covariance matrix:* the principal components are the eigenvectors of the covariance matrix. The first principal component is the eigenvector corresponding to the largest eigenvalue; the second component is the eigenvector corresponding to the second largest eigenvalue and so on and so forth.

One approach to select the number of principal components (u, with $1 \leq u \leq k$) is usually looking at the "percentage of retained variance" for different values of u.

More generally, let $\lambda_1, \lambda_2, \dots, \lambda_k$ be the eigenvalues of C (sorted in decreasing order) so that λ_j is the eigenvalue corresponding to the eigenvector e_j. Then, if we keep u principal components, the percentage of retained variance is given by:

$$U = \frac{\sum_{j=1}^{u} \lambda_j}{\sum_{j=1}^{k} \lambda_j}. \tag{5.12}$$

In the case of images, one common heuristic is to choose u to retain 99% of the variance, or if we are willing to incur some additional information, values in the 90-98% range are also

sometimes used. In this work, we chose u to retain 95% of the variance, so we pick the smallest value of u that satisfies

$$U \geq 0.95, \tag{5.13}$$

which is a much more easily interpretable description than saying that we are retaining two or more components.

After selecting the subset of eigenvectors, we project the remaining data to the new basis, i.e., all the frames of our feature-videos are projected so that the MTM technique can correctly use them.

5.1.3 Multiple template matching – MTM

To perform our cell detection step and consequently test our proposed framework, we used the template matching technique (KHOSRAVI; SCHAFER, 1996; LEWIS, 1995; GONZALEZ; WOODS, 1992) with multiple templates. The normalized cross-correlation (NCC) based TM is an algorithm of pattern recognition field that performs the detection of similar objects in an image $I(x,y)$, taking as input the image itself and a template (sub-image) $T(x,y)$ to be detected. The TM algorithm used in this work has the NCC coefficient as its similarity measure, computed as:

$$\rho(x,y) = \frac{\sum_r \sum_s \left[T(r,s) - \overline{T}\right] \cdot \left[I(x+r,y+s) - \bar{I}_T\right]}{\sqrt{\sum_r \sum_s \left[T(r,s) - \overline{T}\right]^2 \cdot \sum_r \sum_s \left[I(x+r,y+s) - \bar{I}_T\right]^2}}, \tag{5.14}$$

where $\overline{T}$ is the average value of pixel intensities in $T(x,y)$, $\bar{I}_T$ is the average value of I in the coincident region with the current position of T, and the sums are only realized over the common coordinates of $I(x,y)$ and $T(x,y)$, delimited by the variables r and s of the summations.

The coefficient of correlation ρ indicates the level of similarity between the template T and the current image region. Its scale varies in the range of $[-1,1]$ and is, therefore, normalized by the amplitudes of T and I, wherein $\rho = 1$ means the total correlation between T and the sub-region of I, $\rho = 0$ means that there is no correlation, and $\rho = -1$ means inverse correlation.

As already stated at the beginning of this section, our approach used a set of leukocyte-templates as input for the MTM algorithm. As a result, we have image maps with the coefficient values corresponding to the similarity of our selected templates with the video frames. In the cases where the number of templates is higher than one, we sum the final maps found to have only one output for the algorithm.

5.1.4 Post-processing

As a result of the MTM algorithm, we have a sequence of image maps where we shall find the leukocyte positions after analyzing their local maxima responses. Because some feature images did not enhance all the cells as we needed, we applied an adaptive thresholding technique in the final MTM maps to also capture the low responses corresponding to cells. The adaptive thresholding technique has the form:

$$I_R(\boldsymbol{x}) = \begin{cases} 1, & \text{if } I(\boldsymbol{x}) > \Gamma(\boldsymbol{x}), \\ 0, & \text{otherwise,} \end{cases} \tag{5.15}$$

where the local thresholding function $\Gamma(\boldsymbol{x})$ is calculated as a weighted sum (cross-correlation with a Gaussian window) of a $b \times b$ neighborhood of position $\boldsymbol{x}$ in the image plus a constant c. The default standard deviation of the Gaussian window is defined by the specified neighborhood size b. In this approach, we set b by considering the radius of the largest manually selected template and the constant c as 10% of the image gray-level intensity, i.e., if the image pixel values are varying in the range [0, 255], then c equals to 25.5.

As a consequence of the image binarization by the adaptive thresholding technique, we have a set of detected regions corresponding to our cell candidates. For each region, we computed the centroid coordinate to be used as a detection result, which is compared with the manual annotations to create an evaluation measure. However, to reduce the number of false positives in this approach, two additional steps were included as a post-processing stage.

Firstly, the mask image (segmented vessel) calculated in the preprocessing stage was applied to all video frames, allowing the analysis of the detected cells only inside the region of interest. Finally, we performed a circularity analysis for each remaining candidate cell in the MTM resulting maps.

The circularity analysis was employed to review the cell candidates and, consequently, decrease the number of wrong detections. These wrong detections mostly correspond to the MTM map regions that have a non-circular shape, which is mainly observed when two cells are very close to each other, or there is a remaining motion blur in the images.

To perform the analysis, we extracted the local maximum of each binarized region in the resulting MTM map and analyzed four radial lines centered at this point. We then extracted the intensity profiles of these lines (of size $\kappa = 2 \times \lfloor b/2 \rfloor + 1$) and accumulated them, as illustrated in Figure 5.4.

Since the resulting MTM maps present the most reliable candidate cells as blob-like struc-

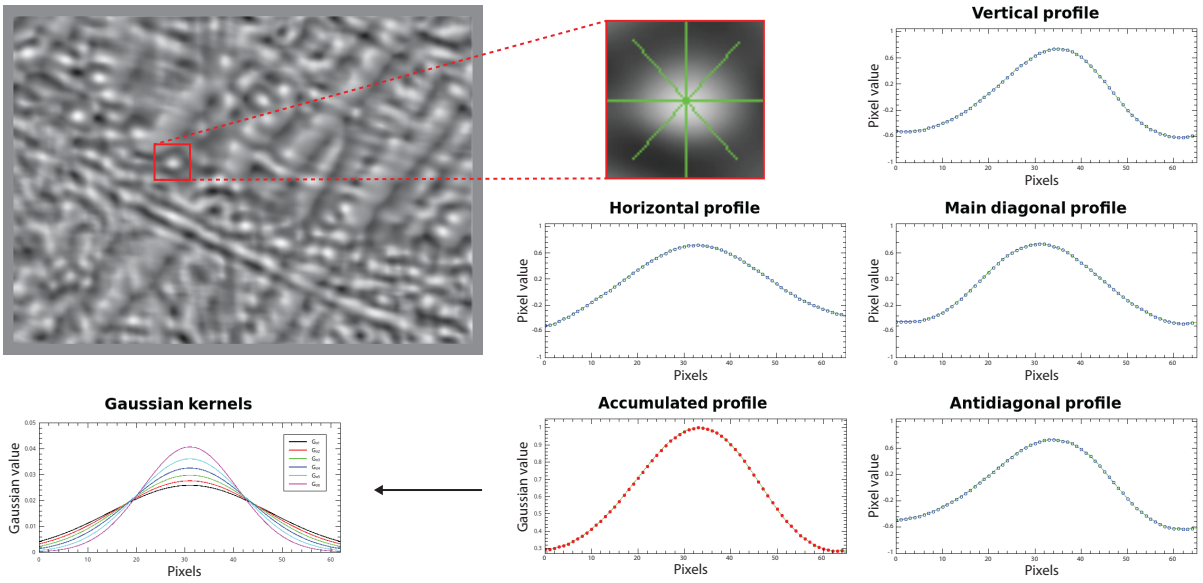


Figure 5.4: Example of the post-processing step for circularity analysis.

tures, we can compare the distribution of the accumulated line profiles with Gaussian kernels of different standard deviation values (σ). The bank of Gaussian kernels was created by varying the σ value according to κ and a sigma step. The initial sigma value is defined as:

$$\sigma_{min} = 0.3 \times ((\kappa - 1) \times 0.5 - 1) + 0.8, \tag{5.16}$$

with the sigma step as:

$$\Delta\sigma = 0.25 \times \frac{r}{N_k}, \tag{5.17}$$

where $r = {}^{(\kappa-1)}/_{2}$, and $N_k = 6$ is the number of kernels used. The comparison between the accumulated distribution and the Gaussian kernels is computed by using the Pearson's correlation coefficient:

$$\Upsilon_\sigma(\ell, \omega) = \frac{\sum_i (\ell_i - \overline{\ell})(\omega_i - \overline{\omega})}{\sqrt{\sum_i (\ell_i - \overline{\ell})^2 \sum_i (\omega_i - \overline{\omega})^2}}, \tag{5.18}$$

where ℓ and ω are the accumulated line profiles and the kernel distributions, respectively. The $\overline{\ell}$ and $\overline{\omega}$ are the averages of ℓ and ω.

The score value used to select the final centroids is then calculated as the highest correlation coefficient found for each accumulated distribution weighted by the central pixel value, that is:

$$Z = \left[\max_{\sigma \in [\sigma_{min}, \sigma_{max}]} \Upsilon_\sigma(\ell, \omega) \right] \times \ell_c, \tag{5.19}$$

where σ_{min} and σ_{max} are the minimum and maximum sigma values of the kernels used in the comparison, and ℓ_c is the value of the centered pixel of distribution ℓ (the center point or local maximum). This final score varies in the range [-1,1] and, consequently, the higher it is, the

higher the chance its central point belongs to a cell.

5.2 MTM-DCNN: Multiple Template Matching with Deep Convolutional Neural Networks

Although conventional machine learning techniques have presented great results as in our first detection approach, they still rely on domain/business knowledge, demanding high expertise to create handcrafted features capable of describing the object of interest.

In the last few years the use of artificial neural networks or, more specifically, the CNNs have attracted considerable attention because of its ability to learn data representations automatically while dealing with raw data. The use of shallow CNN models, however, may not represent complex features, resulting in a low level of generalization and weak learning of data representations.

In order to have a CNN model with a high level of generalization and without incurring in the overfitting problem, a high number of images with labeled objects is required for training it properly. As this condition is not always satisfied, other options should be considered.

It is well-known that the first layers of deep CNNs (DCNN) trained on natural images learn more general features that can be similar to the ones obtained via convolution of the input data with Gabor filters and color blobs (YOSINSKI et al., 2014). This important statement suggests we can use the output of these layers as feature extractors in a process called transfer learning. Transfer learning is a popular approach in deep learning where a network developed for a specific task may have its weights from early layers used as a feature extractor or as a starting point for training a new model and adapted in response to a new problem. This procedure can exploit the generalization of a previously well-trained architecture in another model setting (YOSINSKI et al., 2014).

In this detection approach (MTM-DCNN), we explore the transfer learning strategy by using different DCNN models trained on the ImageNet dataset (RUSSAKOVSKY et al., 2015) as feature extractors. For that, we used the output of their first convolutional layers in our problem, i.e., in a task entirely different from the original. These output maps are then selected and used as input for the MTM technique. Figure 5.5 illustrates the pipeline of this approach.

Microscopy frame images generally have a large matrix size, which can limit the processing of a CNN model or make it take a long time to train and predict. It is not reasonable to directly resize the image frames into smaller ones (e.g., 256×256 or 512×512 pixels) as the

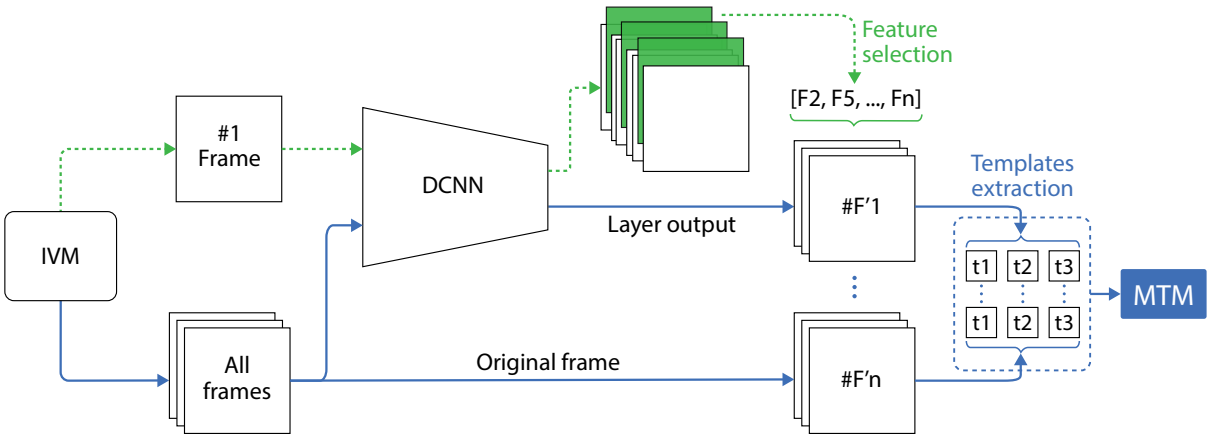


Figure 5.5: Pipeline of the second approach for leukocytes detection.

massive information contained in this kind of image is quite significant for cellular morphology characterization, for example. One common approach to handle large images in deep models is to separate the image into a set of smaller patches. However, it is quite cumbersome to design an effective and efficient patch stitching method. In this approach, therefore, we decided to rescale our input images into the fixed range of 1400×1000 pixels since our largest images have a matrix size of 1392×1040 pixels.

With all the images preprocessed and rescaled, we started our detection pipeline by extracting the first frame of each video and passing it forward into the DCNN model until the selected layer. In this approach, each selected layer was chosen by visual inspection of its output feature images. Since our image frames present relevant information in small regions, we decided to analyze only the first convolutional layers of each DCNN.

As a consequence of transfer learning, not all output images present relevant characteristics that could help in a detection process. For this reason, we performed a feature image selection capable of separating only the best set of features to be used next. To accomplish that, we extracted a small ROI previously selected and used it as a template for the template matching technique. We then get the corresponding output maps for each feature image and applied a thresholding technique on each one of then. In this case, the threshold value was set to 0.9, which results in a map of detection candidates with a high probability of being indeed cells. The accuracy of each resulting map was evaluated following the metrics described in Section 5.4, but to choose the best set of features, we sorted and normalized the evaluation results (so that the sum of all elements is one) in order to select only those top features whose accumulated value (or retained score) was higher than 0.1. Figure 5.6 illustrates the process of feature selection.

After the first passage through our pipeline, illustrated by the green dashed arrows in Figure 5.6, we have our best set of image features from the DCNN layer and can now apply our

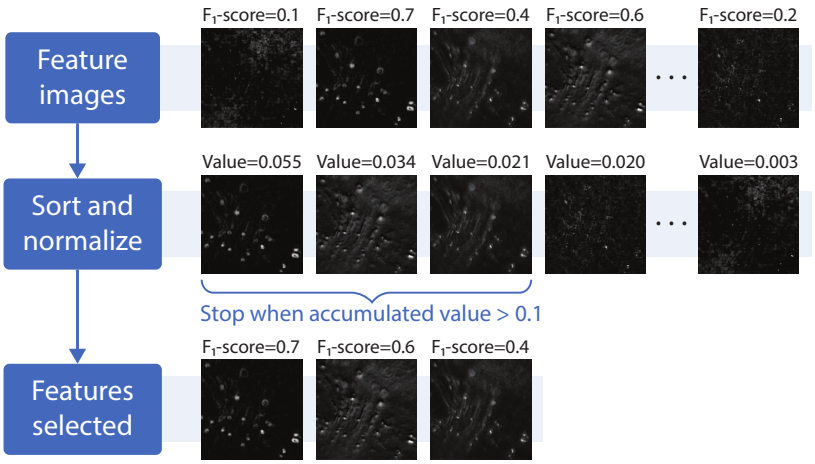


Figure 5.6: Steps of feature selection for the second detection approach.

approach to all the video frames. The blue arrows in Figure 5.6 show the remaining steps in our pipeline. They are similar to the previous steps, except that we now know what the best feature set is and can finally apply the MTM algorithm to identify the cell candidates. At this step, we also include the original frame image into the vector of selected features.

As in the previous detection approach, we used a set of leukocyte-templates as input for the MTM algorithm. As a result, we have intensity maps with the highest coefficient values indicating the spatial locations in the video frames with high similarity with our selected templates. In the cases where the number of templates is higher than one, a fusion step is employed by summing and normalizing all the MTM output maps.

Finally, instead of applying the post-processing strategy in the resulting MTM maps as before, we got our cell candidates by setting different threshold values in a simple thresholding technique since our new feature extraction method showed more robust to enhance the cells in our video frames. These values were defined in the range of [0.7, 0.95], with a step of 0.5.

5.3 DCNN: Deep Convolutional Neural Network

As stated before, one typical deep architecture for target detection and classification is the convolutional neural network, which generates hierarchical data representations given images and targets annotations (LECUN; KAVUKCUOGLU; FARABET, 2010). However, a considerable amount of data is still required in order to train a CNN model from the ground up without overfitting. Among other strategies to reduce overfitting, data augmentation approaches are often used to overcome this problem by artificially inflating the training dataset under the assumption that more information can be extracted from them through augmentations (SHORTEN;

KHOSHGOFTAAR, 2019). There is a vast number of techniques to augment images in recent literature, although many works in object detection only apply basic geometric and intensity operations to the images. Although significant, the generation of augmented images by transformations such as rotation, shift, scale, and flip, might be highly correlated and insufficient to provide a robust method that works well for the variability found in IVM data, for example. Moreover, augmenting data with only basic image transformations could cause overfitting on the minority object class since the biases present in this class are more prevalent post-sampling with these simple techniques (SHORTEN; KHOSHGOFTAAR, 2019).

Already cited in the previous section, the transfer learning technique is another exciting paradigm to prevent overfitting, as also stated by Yosinski et al.:

> *"Initializing a network with transferred features from almost any number of layers, even from distant tasks, can produce a boost to the generalization that lingers even after fine-tuning to the target dataset and proved to be better than using random initialization features"* (YOSINSKI et al., 2014).

When combining the power of CNNs with data augmentation and transfer learning approaches, one can accelerate the training step and improve the performance of a new model (YOSINSKI et al., 2014). Therefore, our third detection approach explores an adaptation of the RetinaNet (LIN et al., 2017b) model to detect leukocytes in IVM images, where we analyzed the use of different backbones, feature pyramid levels, image input scales, and the impact of not using frames from the same video in the training and test datasets.

To accomplish that, we have designed a suite of augmentation techniques for detecting leukocyte recruitment in IVM data and applied the transfer learning approach by fine-tuning the weights of ResNet (HE et al., 2016a) backbones pre-trained with the ImageNet dataset (RUSSAKOVSKY et al., 2015). This strategy not just enables training without overfitting, but also boosts generalization performance.

5.3.1 Model architecture

RetinaNet (LIN et al., 2017b) is a FCN created by the Facebook AI Research group[1] that uses a feature pyramid network (FPN) (LIN et al., 2017a) coupled on top of a CNN as its backbone and attaches two subnets for each feature pyramid level, one for classification, and one for regression of anchor boxes to ground-truth object boxes, see Figure 5.7.

[1] `https://ai.facebook.com/`

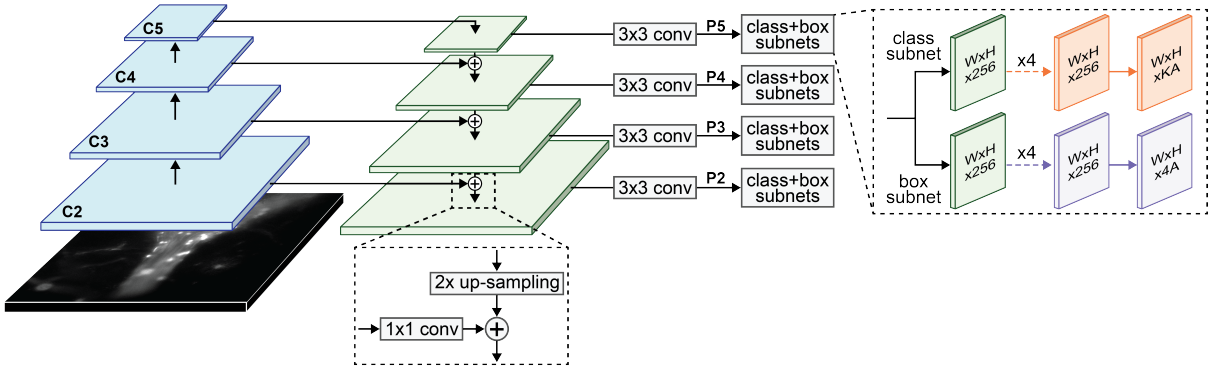


Figure 5.7: RetinaNet architecture (adapted from (LIN et al., 2017b)).

This model is a one-stage detector that uses a dynamically scaled cross-entropy loss for class balance and has demonstrated significant results in small object detection compared to other common methods in the literature. Its design highlights an efficient use of anchor boxes and a feature pyramid that is responsible for the computation of convolutional feature maps.

In this detection approach, we tested the base model ResNet (HE et al., 2016a) pre-trained on ImageNet1k (RUSSAKOVSKY et al., 2015) with different depths (50, 101, and 152) to perform transfer learning. All parameters and convolutional (conv) layers were initialized as in (LIN et al., 2017b) when not specified. Unlike (LIN et al., 2017a), we added the feature activation output from ResNet residual stage conv2 (HE et al., 2016a) to the FPN on top of our backbone. This minor modification improves small object detection by pairing a higher-resolution pyramid level. Connections between the base model and the FPN are made by combining the low-resolution, semantically strong features of ResNet with high-resolution, semantically weak features of FPN via a top-down pathway and skip connections (LIN et al., 2017a). This process results in a feature pyramid with all levels having rich semantic information and a model that can be used to predict objects at several scales from a single input image scale without increasing the predicting time.

To cover cells of different shapes and sizes, a set of anchors (REN et al., 2017) was assigned to ground-truth boxes in the training process. These anchors are predefined reference boxes tiled across the image that allow the network to evaluate all object predictions at once, eliminating the need to scan the image with a sliding window that computes a separate prediction at every potential position. They are defined to capture the scale and aspect ratio of our targets in the training dataset, and, as a consequence, we have the network predicting the probabilities and refinements corresponding to the tiled anchors instead of directly predicting bounding boxes.

In our ablation experiments, we tested anchors with different aspect ratios and sizes at each pyramid level. Assuming n_{sc} is the number of anchor scales, we added anchors of sizes

$\{2^0, 2^{1/n_{sc}}, \ldots, 2^{(n_{sc}-1)/n_{sc}}\}$ to the original set of n_{ar} aspect ratios. Thus, the number of anchors at each spatial position was $A = n_{ar} \times n_{sc}$. To improve speed and accuracy in the training process, we only decode the 300 top-scoring predictions per FPN level, after thresholding the confidence score at 0.05 and the intersection over union (defined later) at 0.5. We then merged the top predictions from all FPN levels by applying a non-maximum suppression (NMS) algorithm with a threshold of 0.5.

The cell/non-cell classification and bounding box regression were defined by the two subnets linked at each feature pyramid level. Both subnets start with four 3×3 conv layers, each with 256 filters and each followed by ReLU activations. The difference between them is at the end of their design and in the fact that they do not share parameter values. While the classification subnet terminates in a 3×3 conv layer with KA filters (here, $K = 1$ is the number of object classes) followed by a sigmoid activation, the regression subnet terminates in a 3×3 conv layer with $4A$ linear outputs representing the relative offset between the anchor and the ground-truth box. Also, the loss for box regression is calculated by the standard smooth L_1 loss (GIRSHICK, 2015), while the focal loss is used at the output of the classification subnet. The focal loss was introduced in (LIN et al., 2017b) to address the class imbalance during training in one-stage detectors. It is defined as following:

$$\mathrm{FL}(p_t) = -\alpha (1 - p_t)^{\gamma} \log(p_t), \tag{5.20}$$

where $\alpha \in [0, 1]$ is a weighting factor and p_t is

$$p_t = \begin{cases} p, & \text{if } y = 1, \\ 1 - p, & \text{otherwise.} \end{cases} \tag{5.21}$$

In the above, $y \in \{\pm 1\}$ specifies the ground-truth class, and $\gamma \geq 0$ is the adjustable parameter for the modulating factor $(1 - p_t)^{\gamma}$. This modulating factor aims to decrease the loss contribution from easy examples and extends the range in which an example receives a low loss, focusing the training step on a sparse set of hard examples. In our case, this is a significant addition since our images have lots of easy negatives that could impair the learning process.

For training the model in our dataset, we used the Adam stochastic optimization (KINGMA; BA, 2015) over 2 GPUs[2] with the number of images per batch being set according to the available memory size. In RetinaNet, the training loss is the sum of the losses from both subnets.

[2]NVIDIA GeForce GTX 1080 Ti (11GB).

5.3.2 Data augmentation

Because manually annotating cells is a tedious and time-consuming task, the annotated datasets available are generally small and require an extensive data augmentation for training a DNN model consistently. The augmentation process inflates the training data and empowers the model to learn invariance to such transformations without the need of having them in the source dataset. Reproducing these transformations efficiently is essential when working with microscopy images from different studies since the cells can change their shape and appearance, and the imaging modality is entirely dependent on the biological experiments. Accordingly, we have designed a suite of augmentation techniques that enables our network to learn without overfitting and could represent most of our image variations during IVM experiments. All these techniques were combined and applied on the fly (online data augmentation) to save memory. They are described bellow.

5.3.2.1 Photometric distortions

Photometric distortions, similar to those used in the original Caffe implementation of single shot multibox detector (SSD[3]), were applied to our images in order to simulate the color and contrast variations found in microscopy imaging.

We started by converting the input images to the RGB color scheme and then applying random transformations in brightness and contrast. Next, we converted the images to the HSV color scheme and applied random transformations in hue and saturation. Finally, we returned the images to the RGB scheme and randomly swapped their channels. All these transformations had their parameters randomly varying between defined upper and lower values, and were applied in a random but logic sequence with a probability of 50% to occur.

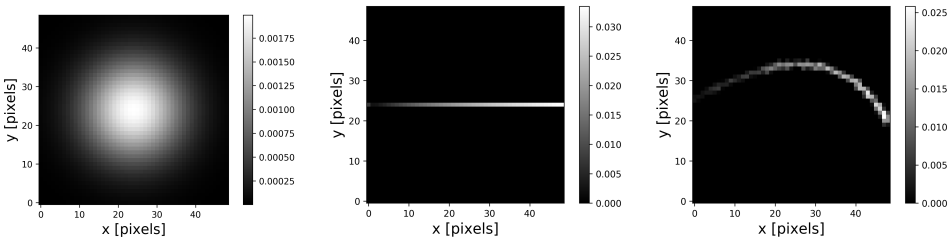
5.3.2.2 Motion kernels

We simulated the sample motion and light diffraction patterns by convolving kernels created from different PSF to our images. As the input image scale may vary between a defined and restricted range, small kernels can actively modify smaller images, but the outcome in the bigger ones are not always noticeable. On the other hand, big kernels can significantly change the bigger images and cause extreme and undesirable changes in the smaller ones. To deal with this problem while keeping the use of random values for better generalizability, we set the kernel sizes in the range of 0% and 1% of minimal image size. As a consequence of our input images

[3] `https://github.com/weiliu89/caffe/tree/ssd`

with different scales, we had kernels varying between the sizes {0, 3, 5, 7, 9}, in which zero means that we did not apply the convolution to the image.

Movements caused by peristaltic motion or by the animal's breath and heartbeat can result in a combination of vertical and horizontal displacements of the organ under analysis. This combination of movements is frequent in IVM experiments and has as its main consequence the momentary loss of microscope focus. Even using the focal auto-adjustment option, the microscope is unable to correct for its focal plane in time, creating blurred and tremulous images. Therefore, besides the Gaussian kernels for scale invariance, we created kernels with line and curved patterns, which were randomly rotated to encompass different directions of movement. Some of these kernels can be visualized in Figure 5.8.



(a) Gaussian kernel. (b) Line kernel. (c) Curve line kernel.

Figure 5.8: Examples of motion kernels.

To simulated the Airy disk pattern present in fluorescent microscopy images (see Chapter 2), we randomly created different sized kernels with the Airy PSF, as the one illustrated in Figure 5.9.

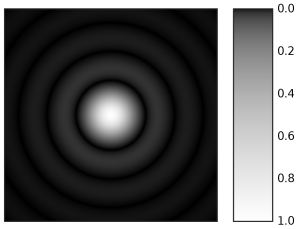


Figure 5.9: Example of an Airy disk kernel.

The Airy PSF is defined in terms of the Bessel function (McClarren, 2018) of the first kind, J_1, as follows:

$$f(r) = A\left[\frac{2J_1\left(\frac{\pi r}{R/R_z}\right)}{\frac{\pi r}{R/R_z}}\right]^2, \tag{5.22}$$

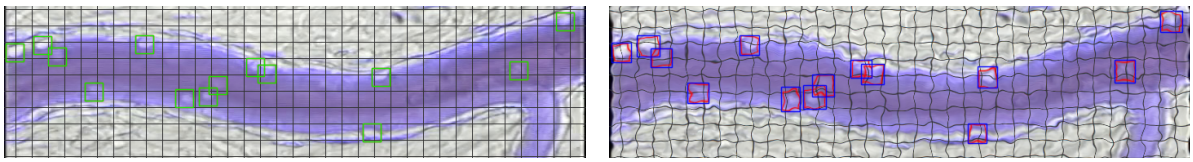
where $A = 1$ is the amplitude of the Airy function, r is the radial distance from the function maximum ($r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$), R is the radius of Airy disk defined as one third of the kernel size, and $R_z \approx 1.22$. The point (x_0, y_0) in Equation (5.22) is the coordinate position of

the maximum of the Airy disk or the central point of the created kernel, and the constant R_z is defined according to the first dark ring in the diffraction pattern (MERCHANT et al., 2005).

5.3.2.3 Geometric transformations

In addition to the previous techniques, we applied basic geometric transformations to the images such as uniform scaling, horizontal/vertical flips, rotation, and shift. The level of each transformation was randomly defined in a fixed range of small values to avoid extrapolations of image distortions. The uniform scaling in our case works as "zoom-in" and "zoom-out" operations, creating more small and large training examples and improving the model performance for small object detection.

Deformable transformations were also used in this approach to make our model robust to such variations. As in the U-Net (FALK et al., 2019) approach, we applied smooth elastic deformations (SIMARD; STEINKRAUS; PLATT, 2003) using displacement vectors (on a 1×1 grid) sampled from Gaussian distributions with standard deviation values varying randomly in the range [4, 7], and a scaling factor that controls the intensity of the deformation varying in the range [1, 200]. We then proceeded to the per-pixel displacements by using the spline interpolation of order one. Empirical and visual analyses were used to define all parameter ranges. Figure 5.10 illustrates an example of an image with its deformable grid when the values for the standard deviation and scaling factors are maximum in terms of severe deformation.



(a) Original image. (b) Severe deformation.

Figure 5.10: Example of a severe deformation in an image from the ME video.

5.3.3 Cyclical learning rate

The most common practice in training DNNs is to set the model learning rate to a constant value and decrease it by an order of magnitude once the accuracy has plateaued. However, instead of defining schedules that monotonically decrease the LR values, we applied a cyclical learning rate (CLR) strategy to be used in our model after the setting of hyperparameters. In practice, this procedure prevents the exhaustive search for a reasonable initial value, which would require many experiments, and the uncertainty that lowering the LR will make our model descend into areas of low loss.

Initially proposed by Leslie Smith in (SMITH, 2017), the CLR schedule varies between a lower and an upper bound (*base_lr* and *max_lr*). These periodic changes in LR values help to avoid saddle points or local minima, which, consequently, accelerate the training process. To derive the optimal bounds for CLR initialization, we let the model run for a few epochs while the LR increased linearly from a minimum value of $1e-10$ to a maximum value of $1e+1$ that we deem fit to observe all three zones limited by vertical lines of the plot in Figure 5.11.

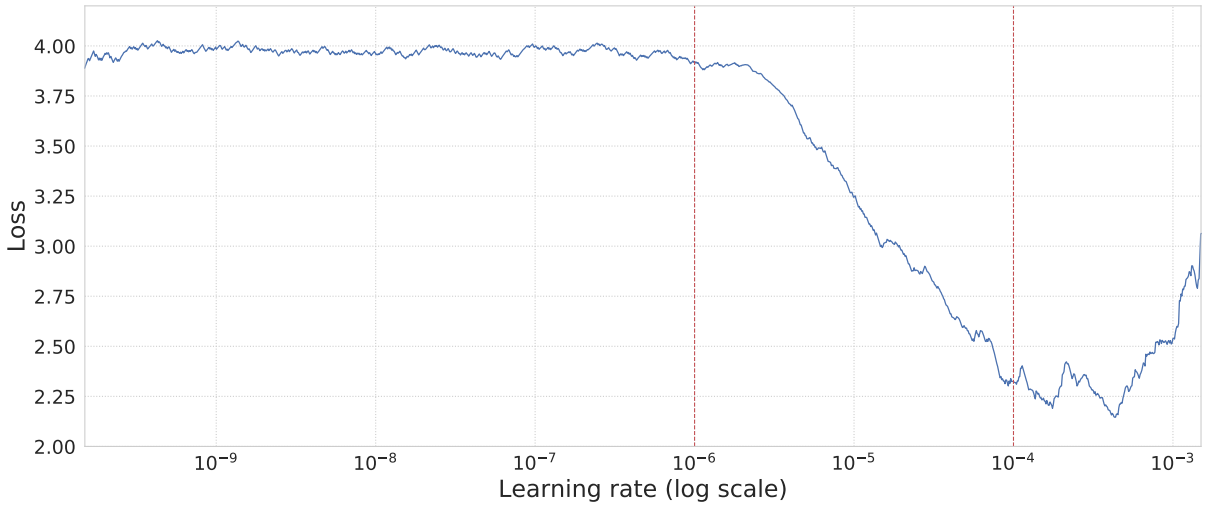


Figure 5.11: Analysis of the LR optimal range for cyclical learning rate.

According to the plot, when the LR is too low, the loss function does not improve the model any further. When entering the zone between 10e-6 (*base_lr*) and 10e-4 (*max_lr*), one can observe the optimal range for the LR since we are looking for those values associated with the steepest drop in the loss. Increasing the LR further will provoke an increase in the loss as the parameter updates, causing the loss to "bounce around" and even diverge from the minima. After finding the initial LR range, we set the upper bound to follow an exponential decay, giving us a more fine-tuned control in the rate of decline in *max_lr*.

5.4 Evaluation methods and metrics

The results of our detection stage were assessed by applying two different approaches. They were defined based on the spatial coordinates of 1) leukocytes' centroids for all the detection approaches, and 2) bounding boxes' points for the evaluation of our DCNN model.

In the first case, we compared the spatial coordinates of the leukocytes' centroids that were manually identified and annotated by an expert (ground truth) with those automatically detected. For the outputs of our DCNN model, we calculated the spatial points centered at each bounding box. In this sense, we defined the detection as true when the distance between a manually

annotated centroid and an automatically detect one is less or equal than k pixels. This distance value k was estimated according to the average radius of the observed cells. It means that in the first two detection approaches, we only have the information about the template images selected, so the value k was set to $k = {}^{\max(\text{template\_size})}/_{2}$. On the other hand, the algorithm knows all the cell sizes in the last detection approach, so the value k was set to $k = {}^{\text{avg\_cell\_size}}/_{2}$, according to the values in Table 2.2.

Accordingly, we defined the true positives (TPs) as the accumulated amount of leukocyte positions that were correctly detected by the algorithms, the false positives (FPs) as the accumulated amount of leukocytes automatically detected without correspondence to those manually annotated, and the false negatives (FNs) as the accumulated amount of leukocytes that the algorithms could not identify.

The measures of precision (P), recall (R) and F_ω-score (GOUTTE; GAUSSIER, 2005) were then used to evaluate the overall performance of our approaches. These measures are based on the accumulated TP, FP, and FN numbers over the sequence of video frames. They are defined as follows:

$$P = \frac{TP}{(TP+FP)}, \tag{5.23}$$

$$R = \frac{TP}{(TP+FN)}, \tag{5.24}$$

$$F = \frac{1}{\psi\frac{1}{P} + (1-\psi)\frac{1}{R}} = \left(1+\omega^2\right) \cdot \frac{P \cdot R}{\omega^2 P + R}. \tag{5.25}$$

The precision of the system depicts the level of effective success among all detections (TP+FP), while the recall measure represents the proportion of what was correctly detected among the real positive instances (TP+FN). The F_ω-score can be considered a compensation measure between the precision and recall evaluations through a weighted harmonic mean. It measures the effectiveness of the results by assigning ω times more significance to the recall rate than to the precision rate. The most widely used measure, F_1, involves the same weighting for both rates, i.e., $\psi = 0.5$ and, consequently, $\omega = 1$. Other two commonly used measures for F_ω are the F_2 and $F_{0.5}$, which gives more control to the recall and precision rates, respectively. For all these measures, the closer they are to the maximum value 1, the better the effectiveness of the detection system.

Based on the above definitions, we can also measure the counting and localization accuracies of our centroids by applying the following metrics. (1) The mean (μ_c) and standard

deviation (σ_c) of the counting error. Particularly, given N testing images, we have

$$\mu_c = \frac{1}{N}\sum_{i=1}^{N}\hat{c}_i, \text{ and } \sigma_c = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(\hat{c}_i - \mu_c)^2}, \tag{5.26}$$

where $\hat{c}_i$ represents the absolute difference between the total number of predicted cells and the manual annotations for the ith image. (2) The mean (μ_d) and standard deviation (σ_d) of the prediction distance error. For N testing images, we have

$$\mu_d = \frac{1}{N}\sum_{i=1}^{N}\hat{d}_i, \text{ and } \sigma_d = \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left(\hat{d}_i - \mu_d\right)^2}, \tag{5.27}$$

where $\hat{d}_i$ refers to the average Euclidean distance between manually annotated centroid and the corresponding matched TP prediction for the ith image.

However, to evaluate the performance of our DCNN model regarding its original output, we computed the agreement between the predicted bounding boxes and the manual annotations. A conventional metric, called intersection over union (IoU), was used to measure how much our predicted boxes overlap with the ground truth. Its formulation is as follows:

$$\text{IoU} = \frac{\text{area of overlap}}{\text{area of union}}. \tag{5.28}$$

In this case, we applied IoU thresholds of 0.25 and 0.5 to classify a prediction. The choice for shallow IoU threshold values was based on the considerable influence of small objects when their dimensions are compared, which means that even tiny differences between the objects' bounding boxes can significantly decrease the IoU responses. This problem is potentially increased in microscopy images since the cell boundaries are sometimes misdefined because of the poor image contrast or microscope defocus.

Accordingly, TP values were defined as the predicted boxes which had an IoU value higher than the IoU threshold for some manual annotation; FP values as the predicted boxes with no correspondence to any manual annotation; and the FN values as those manual annotations that the model could not predict. The average precision (AP) metric was also used to measure the accuracy of our DCNN model. AP computes the average precision value for recall values over 0 to 1 after sorting the predictions by their confidence scores. It can also be calculated by finding the area under the precision-recall curve. Note that, although these metrics result in values ranging from 0 to 1, in this work, we sometimes used them as percentage values in order to facilitate their comparison with other methods in the literature.

5.5 Final considerations

In this chapter, we described the 2D processing stage of our automatic computational pipeline. It is comprised of three different detection approaches (MTM-PCA, MTM-DCNN, and DCNN) that were developed exclusively for our IVM data. We also detailed the methods and metrics used to evaluate the performance of our approaches. In the next chapter, we present the subsequent stage in our computational pipeline, which is composed of the 2D+t processing or tracking procedure.

Chapter 6

TRACKING – 2D+T PROCESSING

This chapter presents all the methodology used to perform the cell tracking in the automatic computational pipeline proposed. We detail all the techniques applied to the spatiotemporal images as well as the final statistical measures extracted for the leukocyte recruitment analysis.

6.1 Pipeline overview

With all candidates to cell centroids detected by our detection approaches, we can start the 2D+t processing stage by identifying the cell trajectories in the spatiotemporal images. Our initial steps, however, comprise some adjustments in the output points and images of 2D processing.

Firstly, we downsampled the output images and the corresponding centroid points by a scale factor of $\lfloor k_{avg}/5 \rfloor$ in both x and y dimensions, where k_{avg} is the average cell size in the video. This is a crucial step to avoid wrong statistical measures in the final analyses caused by the high displacements of large cells. In other words, the displacement of a cell between two consecutive frames is highly dependent on its size and video sampling rate, which could affect our spatiotemporal analysis. Therefore, in order to correctly identify the cell trajectories in our algorithms, we made the displacements of large cells seem more continuous over the videos. The scale factor was defined so that only the points from images with the average cell size larger than 10 pixels were modified.

Next, we computed the convolution between the images containing the centroids and a Gaussian kernel of size 4×4 pixels and sigma values equal to 9. This operation transformed the final centroid points in the images into blob-like structures of the same size so that to facilitate the subsequent analyses, as detailed in the next sections.

After all these adjustments, we created a spatiotemporal image by stacking all frames from the previous step. Consequently, we have a volumetric image containing the cell candidates as tubular-like structures representing our cell trajectories, as illustrated in the next section. The task of tracking was then changed to a three-dimensional detection problem, where the objects of interest have tubular shapes. Figure 6.1 shows all the steps of the 2D+t processing applied in our computational pipeline and the following subsections describe how the whole process is executed in order to ensure a good performance of the method.

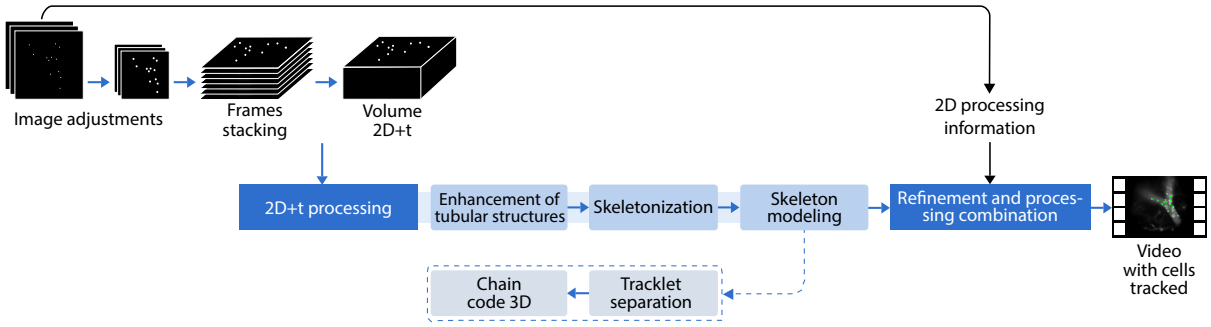


Figure 6.1: Steps of our 2D+t processing stage or tracking.

6.2 Enhancement of tubular-like structures

In our spatiotemporal images (volumes 2D+t), we have bright tubular-like structures corresponding to tracings of the leukocyte trajectories (or paths). Given that, we can use the object shape information provided by the local Hessian matrices (see Appendix A) to build a function that enhances the leukocyte trajectories. Figure 6.2 presents the local pattern of a bright tubular-like structure in three-dimensional images through eigenvectors representation.

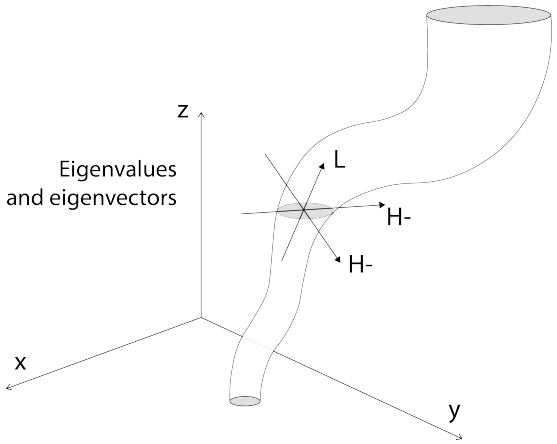


Figure 6.2: Characteristics of a tubular-like structure in a three-dimensional image with a dark background. The eigenvector corresponding to the eigenvalue with the smallest magnitude gives the longitudinal direction of the structure.

In the 2D+t processing case, the desired eigenvalues ($\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]$) for enhancing the leukocyte paths at a specific point $\boldsymbol{x}_0$ must have the following relations of values: $|\lambda_1| \approx 0$, $|\lambda_1| \ll |\lambda_2|$ and $\lambda_2 \approx \lambda_3$, where λ_2 and λ_3 are negatives and have high magnitude values, according to Table A.1. Thus, the maximum response at every voxel is taken as

$$V(\boldsymbol{\lambda}) = \max_{\sigma \in [\sigma_{min}, \sigma_{max}]} V_\sigma(\boldsymbol{\lambda}), \tag{6.1}$$

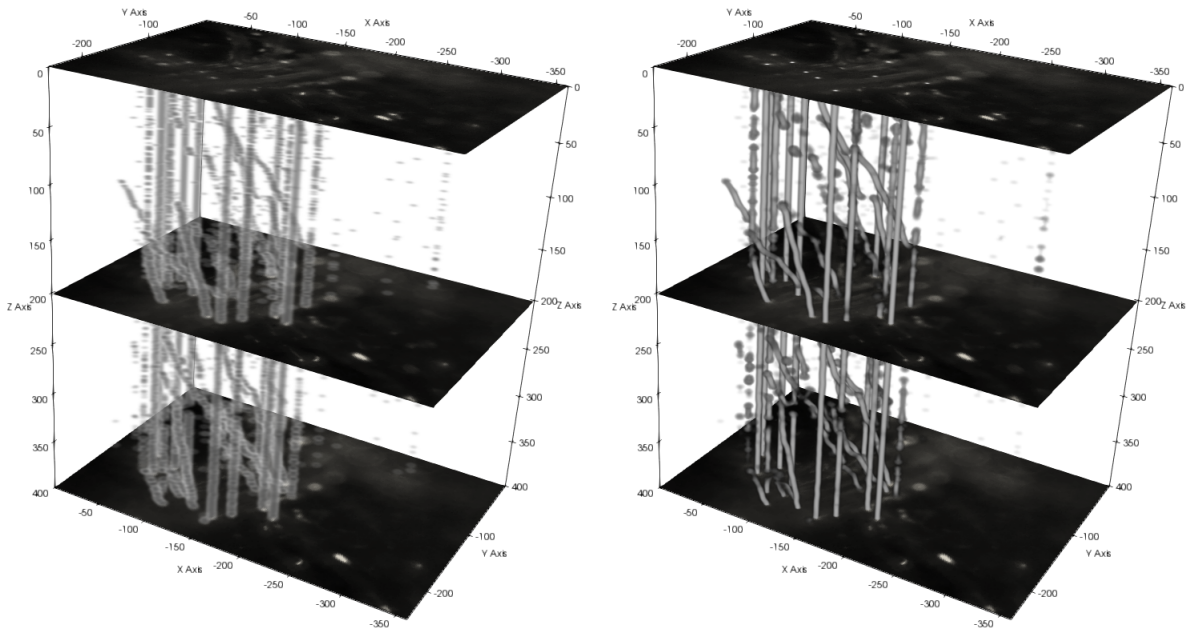
where the vesselness[1] measure function $V_\sigma(\boldsymbol{\lambda})$, is defined as

$$V_\sigma(\boldsymbol{\lambda}) = \begin{cases} 0, & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0, \\ \left(1 - \exp\left(-\frac{R_A^2}{2\alpha^2}\right)\right) \times \exp\left(-\frac{R_B^2}{2\beta^2}\right) \times \left(1 - \exp\left(-\frac{S_N^2}{2c^2}\right)\right), & \text{otherwise.} \end{cases} \tag{6.2}$$

In Equation (6.2), which provides the relationships between the local Hessian eigenvalues accordingly with the object shape desired, $R_A = |\lambda_2|/|\lambda_3|$ helps to distinguish between the plate- and line-like patterns, and $R_B = |\lambda_1|/\sqrt{|\lambda_2 \lambda_3|}$ accounts for the deviation from the blob-like structure, but it cannot distinguish between plate- and line-like patterns. In order to reduce the effect of noisy voxels in the image background, the component $S_N = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}$ (Frobenius norm) was used. For this measure, the response is low when no structure is present in that image position as the local Hessian eigenvalues are small for the lack of contrast (FRANGI et al., 1998). Component sensitivities can be regulated by the parameters α, β and c, which were defined, respectively, as 0.5, 0.5, and one-tenth of the maximum image Laplacian value, as suggested by Dzyubak and Ritman (DZYUBAK; RITMAN, 2011). Since the analysis is now performed over our generated spatiotemporal image, we defined a small range of scales to be analyzed by the vesselness function. A range from 1 to 3 for the σ values was used to cover the leukocyte paths from the volumetric image created.

As a result of this step, we have a volume containing all tubular structures enhanced by the algorithm, which produces real-valued responses close to 1 if the local structure is similar to a tube. An example of an initial spatiotemporal image and the corresponding 3D enhancement by the Hessian algorithm can be seen in Figure 6.3. As detailed in the next sections, a set of techniques was developed to improve the detection responses and to isolate the structures of interest, thus allowing their combination to refine cell detection and tracking.

[1] Although the name vesselness makes little sense in this study, we decided to use the same name as proposed by Frangi et al. (FRANGI et al., 1998) to define the measure responsible to enhance the leukocyte trajectories.



(a) (b)

Figure 6.3: Sub-output images of our framework applied to B2 video. (a) Initial spatiotemporal image, and (b) 3D Hessian enhancement output for the same input video.

6.3 Skeletonization

In 3D Euclidean space, the skeleton of a geometric object is the locus of the centers of all inscribed maximal spheres of the object where these spheres touch the boundary at more than one point (LEE; KASHYAP; CHU, 1994). It can be considered a simplistic characterization of an object, used to reduce the search space of feature points in a geometric model. In this work, we use a skeletonization technique to obtain continuity in the cell trajectories of the created tubular-like structures.

One of the most used approaches to creating a skeleton of an object is performing thinning techniques. The main characteristic of this approach is the repeated deletion of points in the object boundary, respecting the topological (preserving the number of connected objects, cavities, and holes of the object's original shape (MORGENTHALER, 1980)) and geometrical (condition that is used to ensure the desired width and location of the skeleton) restrictions until it reaches a small set of connected points. The technique of symmetric erosion has been widely used to obtain the central lines of objects and, consequently, ensure its connectivity. Figure 6.4 illustrates an example of the skeletonization process applied to a circular object.

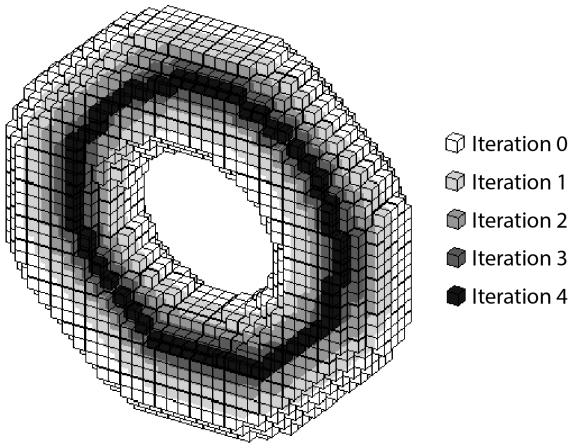


Figure 6.4: Image of a circular object containing the iteration steps of the erosion (PALÁGYI, 2015).

There are two main approaches to performing the thinning: a) kernel-based filters, and b) decision trees. Kernel-based filters rely on the application of a structuring element to an image and, usually, can be extended to dimensions larger than 3D (JONKER, 2000). Methods based on decision trees are restricted to 2D and 3D spaces, but, if adequately projected, they can be potentially faster than morphological filters, finding more points to be deleted in each iteration.

As an initial step of the skeletonization process in this work, the real-valued spatiotemporal images were binarized by using a global thresholding technique (OTSU, 1979). Then, the skeletonization method proposed by (LEE; KASHYAP; CHU, 1994) was slightly modified and used to detect the central region of the tubular-like structures. It is based on a parallel thinning approach that uses an octree data structure to determine the connectivity between voxels more efficiently. Tests in the image voxels are performed for the objects erosion until no more change occurs in the objects. Thus, a voxel was deleted if:

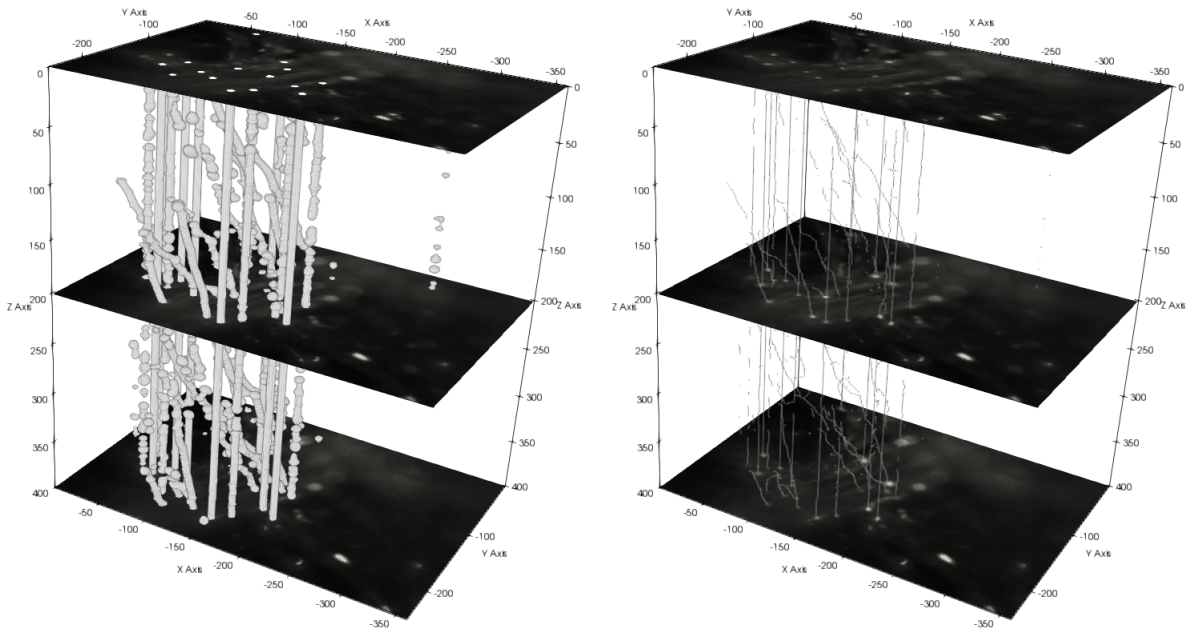
1. It was a surface voxel;

2. It was not the end of a line;

3. Its deletion would not change the Euler characteristic (MORGENTHALER, 1980), i.e., if no holes are created when deleting the voxel;

4. Its deletion did not change the number of connected objects.

The algorithm, which uses a 26-neighborhood connectivity for the object, guarantees no change in the object connectedness and no creation of holes or cavities. Consequently, the connectivity considered herein uses three consecutive frames, which means that $3\times3\times3$ cubic voxels are



(a) (b)

Figure 6.5: Skeletonization output from the B2 video. (a) binary output image, and (b) skeleton output image.

being analyzed for the neighborhood. The resulting output is a binary volume that contains only the skeleton of tubular-like structures enhanced. An example can be seen in Figure 6.5.

6.4 Skeleton modeling

After the skeletonization step, the skeleton tracklets[2] are separated into individual paths and post-processed to eliminate spurious elements, which are undesired skeleton branches created due to noisy boundaries, small holes, and cavities mainly from the remaining motion artifacts. This post-processing step is necessary to better perform the dynamic cell analysis in the final process.

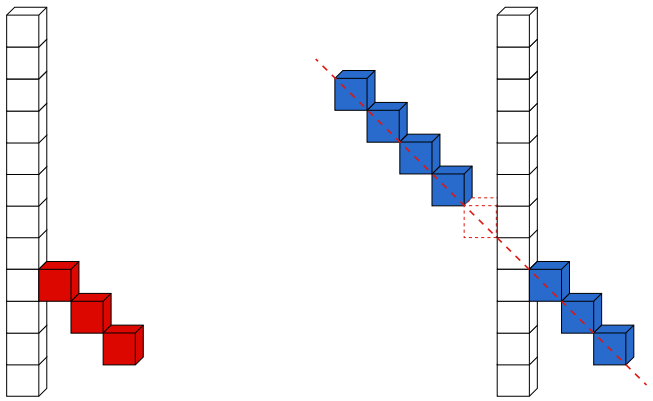
The chain code technique (FREEMAN, 1961) was used to model each tracklet found after skeletonization. Besides being very efficient, this technique can preserve object information and allows considerable data reduction. Also, chain codes are the standard input format for many shape analysis and pattern recognition algorithms (BOSE, 2000). However, before using the chain code algorithm to model our tracklets, bifurcation points in the skeleton image must be determined and separated for the definition of path directions accurately.

[2]Tracklets are defined herein as the parts of cell trajectories in the spatiotemporal images, i.e., the fragments of the track.

6.4.1 Tracklets separation

The primary goal of tracklet separation is to isolate each leukocyte track (or fragments of it) in the spatiotemporal image by identifying their initial and final route positions. This step is mandatory since many tracks end up connecting to each other due to their proximity after the binarization and thinning processes or because of residual movements are still present in the video, even after the stabilization process.

Moreover, the residual movement in the videos or the sudden leap of some cells can cause the appearance of horizontal traces (small branches) in the skeleton, referred here as spurious elements. The most straightforward and widely used approach to eliminate these elements is by thresholding the size of the extracted tracklet so that to remove the small parts found in junction points, such as the one highlighted in red in Figure 6.6(a). However, since spatiotemporal images are built from moving cells, these fragments may belong to a track of another cell, as exemplified in blue in Figure 6.6(b). In this case, thresholding the cell track to remove small fragments will probably be inadequate. For this reason, the connection between points belonging to the same cell trajectory is a task to be performed next.



(a) (b)

Figure 6.6: Examples of cell track fragments. A tracklet with a bifurcation is not necessarily a spurious element (see (a)), it may contain part of another track that intersects it at some point (see (b)).

Junction points must be identified and eliminated in order to separate the connected tracklets. They are defined herein as voxels in which the number of connected points in a 26-neighborhood is higher than two. Consequently, the procedure for track separation segregates all tracklets with bifurcation points and creates isolated line segments.

The algorithm starts by searching for junctions or isolated points in the skeleton image, and then it creates a cumulative matrix (CM) of zeros of the same size as the input image. Next, for each identified junction point, the corresponding cell position in the CM is iteratively incre-

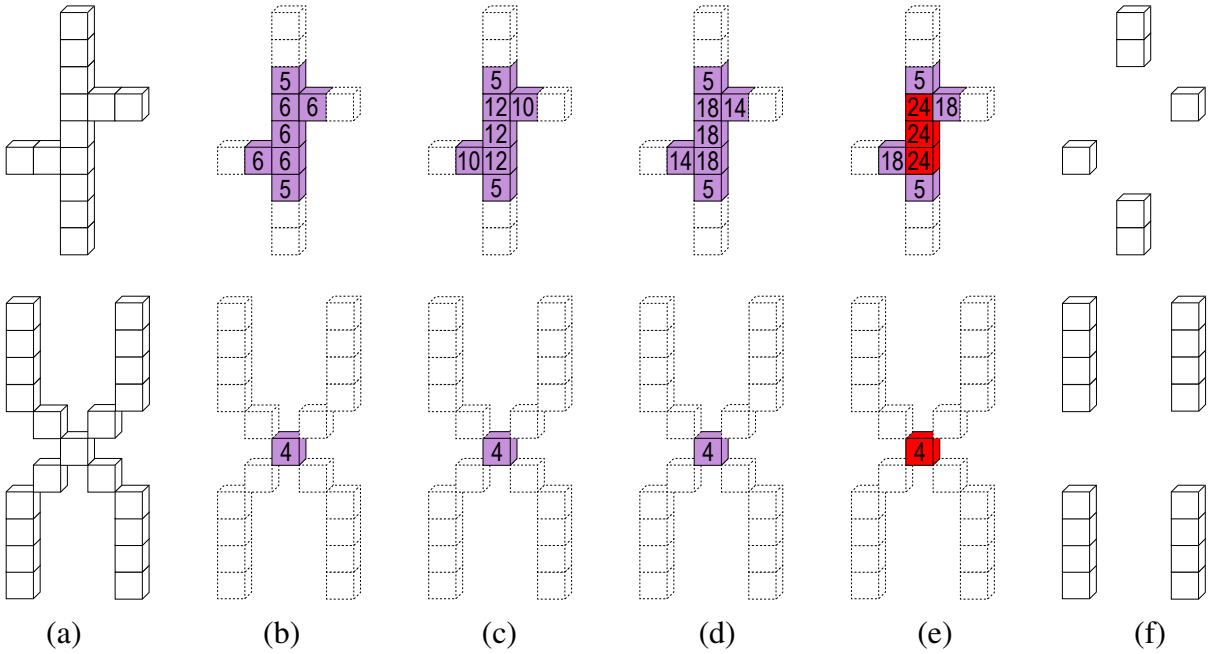


Figure 6.7: Steps of the tracklets separation algorithm applied to two examples of connected paths. (a) Originally connected paths, (b) 1st, (c) 2nd, (d) 3rd, and (e) 4th iterations of the algorithm, in which local maxima are identified as red voxels (darker cubes in the black-and-white version), and (f) neighbors removal for tracklets separation. Dashed cubes are not considered in the algorithm after the first iteration.

mented by using the following weight value configuration: 2 for the 6 face-neighbors connected to the junction voxel, and 1 for the other 20 (12 edge- and 8 point-neighbors) remaining connected neighbors[3]. The algorithm is then repeated three more times over the voxels included in the CM. In the last iteration, the central region of each set of junction points will have higher values. Finally, local maxima are determined in each region, and their neighborhood ($3\times3\times3$ cubic voxel) is eliminated, resulting in track separations.

Figure 6.7 illustrates two examples of connected tracklets and the corresponding iterations of the algorithm for tracklets separation. In the first column, Figure 6.7(a), each example simulates the connection between two different trajectories. Figures 6.7(b-e), show all four iterations of the algorithm in the CM approach. As a final step, Figure 6.7(f), local maxima are identified and eliminated along with their neighbors. This procedure discards the same number of voxels as the number of initial junction points found in the first example. However, in the second example, all neighbors of the single junction point are eliminated, preventing wrong connections in the following steps. The processes described earlier can be better understood by following the Algorithm 6.1.

As the tracks have no more junction points, the identification of the tracklets can be initiated

[3]Each voxel $\boldsymbol{x}$ has three types of neighbors among its 26 closest neighbors; 6 face-, 12 edge-, and 8 point-neighbors, that share a face, an edge, and a point with $\boldsymbol{x}$, respectively.

Algorithm 6.1 Tracklets separation algorithm for the spatiotemporal images.

Input: skeleton image of leukocyte tracks $I(\boldsymbol{x})$

$C \leftarrow I$ ▷ copy of input image
$M \leftarrow \{\}$ ▷ initializes cumulative matrix with zeros
$i \leftarrow 1$
while $i \leq Niterations$ **do**
  for each voxel $\boldsymbol{x} \in object$ **do**
    if $C(\boldsymbol{x}) \geq i$ **then**
      if $Nneighbors = 0$ or $Nneighbors > 2$ **then**
        $M(\boldsymbol{x}) \leftarrow M(\boldsymbol{x}) + \text{CALCWEIGHT}(\boldsymbol{x})$
      end if
    end if
  end for
  $C \leftarrow M$
  $i \leftarrow i + 1$
end while
$I \leftarrow \text{LOCALMAXIMUM}(M)$
VOXELSREMOVAL(I)

Output: images $I(\boldsymbol{x})$ with tracklets separated

by the chain code technique, which is described in the next subsection.

6.4.2 Chain code 3D

The chain code 3D technique proposed by Bose (BOSE, 2000) was adapted and used in this work to create structures describing the tracks of leukocytes simply and efficiently. Its development is based on Freeman's work (FREEMAN, 1961), who introduced the technique in the literature. In his work, the set of neighbor voxels (26-neighborhood connectivity) of a point P is denoted by $N(P)$ and can be seen in Figure 6.8.

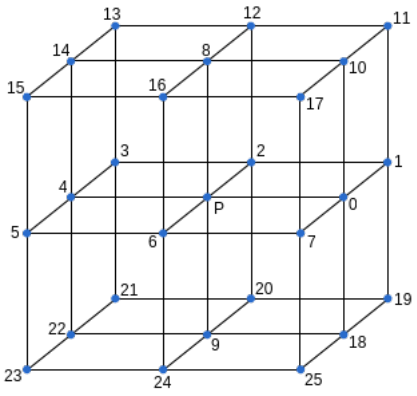


Figure 6.8: The 26-neighborhood of point P.

Thus, if P has the coordinates (i, j, k), then

$$N(P) = \{(x,y,t) : 0 < [(i-x)^2 + (j-y)^2 + (k-t)^2]^{\frac{1}{2}} \leq \sqrt{3}\}, \tag{6.3}$$

and $x, y, t \in \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers.

For each point $P(x,y,t)$ in a three-dimensional image, the algorithm uses the codes presented in Table 6.1, in which the coordinates represent the directions from point P.

Table 6.1: Algorithm codes and its respective directions.

Code	Direction	Code	Direction
0	$(x+1,y,t)$	13	$(x-1,y-1,t-1)$
1	$(x+1,y-1,t)$	14	$(x-1,y,t-1)$
2	$(x,y-1,t)$	15	$(x-1,y+1,t-1)$
3	$(x-1,y-1,t)$	16	$(x+1,y+1,t-1)$
4	$(x-1,y,t)$	17	$(x,y+1,t-1)$
5	$(x-1,y+1,t)$	18	$(x+1,y,t+1)$
6	$(x,y+1,t)$	19	$(x+1,y-1,t+1)$
7	$(x+1,y+1,t)$	20	$(x,y-1,t+1)$
8	$(x,y,t-1)$	21	$(x-1,y-1,t+1)$
9	$(x,y,t+1)$	22	$(x-1,y,t+1)$
10	$(x+1,y,t-1)$	23	$(x-1,y+1,t+1)$
11	$(x+1,y-1,t-1)$	24	$(x,y+1,t+1)$
12	$(x,y-1,t-1)$	25	$(x+1,y+1,t+1)$

However, the algorithm was modified to consider only positive time directions (t or $t+1$), i.e., only the set of directions $\{0,1,2,3,4,5,6,7,9,18,19,20,21,22,23,24,25\}$, as can be seen in Figure 6.9. The reason for this modification is based on the fact that displacements in $t-1$ directions would imply on back in time since the algorithm always initializes from the highest points of a track (first points on time).

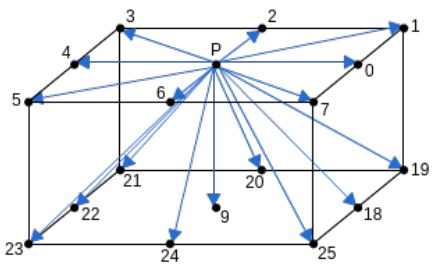


Figure 6.9: Directions considered in the chain code algorithm adapted.

According to the algorithm proposed by Bose, the order of search for these directions is as follows:

4,3,2,1,0,7,6,5,

9,22,21,20,19,18,25,24,23.

In the case of the spatiotemporal images, we have a set of tracklets with initial and final points. For the definition of such points, named endpoints, we search for the voxels whose number of 26-connected neighbors is equal to 1. The algorithm begins with each endpoint found, following the temporal order of the image $(t, t+1, t+2, \ldots)$. Hence, starting from an endpoint (initial point), the cell trajectory is encoded until a new endpoint (final point) is found along the path. Finally, the coding process stops and saves the tracklet chain code. This procedure is performed until all the endpoints found are defined as initial or final points, ensuring that all tracklets (already separated in the previous step) are checked.

By representing the leukocyte trajectories via chain code, the leukocyte tracking, and the computation of quantitative measures were facilitated. To compute the traveled distance of a cell in a spatiotemporal image, for instance, we need to know only the spatial image resolution and add all the cell displacements over time (axis t), according to the information provided in Figure 6.10.

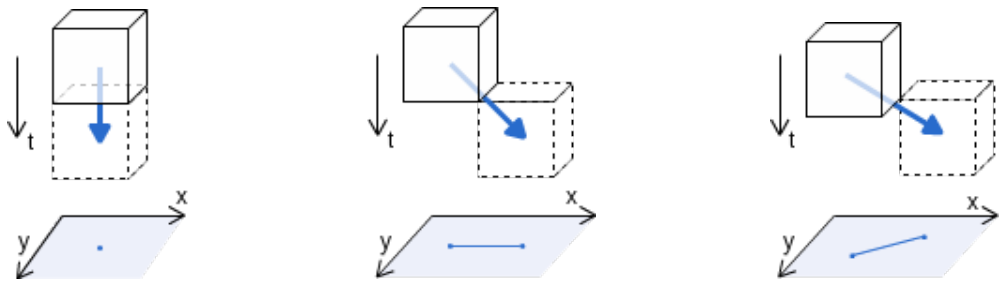


Figure 6.10: Examples of displacement vectors for the calculation of the traveled distance of cells. (a) Vertical displacement indicates that the cell is at rest; (b) horizontal displacement indicates that the cell has traveled the distance of 1x or 1y; and (c) diagonal displacement indicates that the cell has traveled the distance of $\sqrt{x^2+y^2}$.

In other words, the traveled distance of a cell is defined as:

$$Dist = N_{h_x} R_x + N_{h_y} R_y + N_d \sqrt{R_x^2 + R_y^2}, \tag{6.4}$$

where the measures R_x, R_y and N_{h_x}, N_{h_y} and N_d represent, respectively, the resolutions (μm) in spatial axes x and y, and the number of cell displacements in the corresponding x, y, and diagonal directions.

6.5 Refinement and processing combination

During this step, all tracklets detected and separated in previous stages are processed to create continuous and coherent leukocyte trajectories. Results from the 2D processing stage are used to help to decide on how the isolated tracklets should be connected to form the final leukocyte trajectories. The following subsection describes the algorithm developed to refine the leukocyte tracking.

6.5.1 Track linking

The algorithm for track linking, or tracklets connection, first locates the final position (a_{end}) of the ith tracklet in the spatiotemporal image. Then, it searches for the initial position of other tracklets situated inside a 3D right circular cone-shaped region, as illustrated in Figure 6.11(b). The searching region is defined by the height h, the angular aperture θ, and the directional vector $\vec{A}_i$ of the cone, which points to the temporal axis direction ($\vec{t}$) and is parallel to the centerline of the cone, i.e., the dashed line defined by the vertex and the center position of the cone's base.

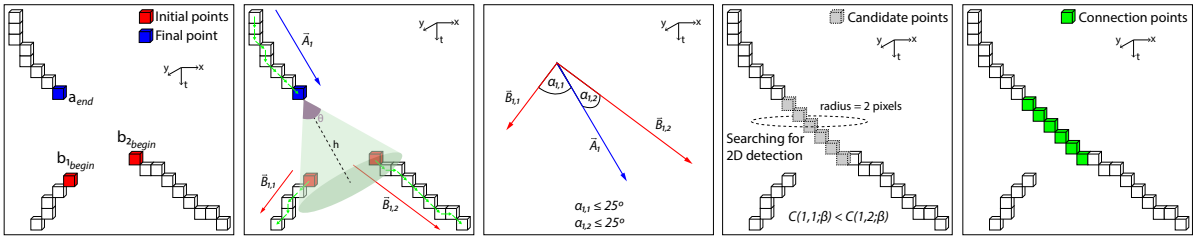


Figure 6.11: Steps followed in the track linking algorithm. (a) Initial tracklets; (b) searching for initial points inside the cone spatial region and definition of directional vectors; (c) angular comparison between the directional vectors of the candidates; (d) searching for detected points in 2D processing; and (e) connected segments.

The directional vector $\vec{A}_i$ is determined by the vector addition of n vectors (green vectors in Figure 6.11(b)) formed from each previous position (a_{end-n}, ..., a_{end-2}, a_{end-1}) and the final position (a_{end}) of the tracklet. The same is done for each jth track fragment inside the cone-shaped region to create directional vector $\vec{B}_j$, but now starting from the initial point (b_{begin}) and adding the next positions of its voxels ($b_{begin+1}$, $b_{begin+2}$, ..., $b_{begin+n}$). Accordingly, we have:

$$\vec{A}_i = \vec{a}_{i_{(end-n,end-n+1)}} + \ldots + \vec{a}_{i_{(end-2,end-1)}} + \vec{a}_{i_{(end-1,end)}}, \tag{6.5}$$

$$\vec{B}_j = \vec{b}_{j_{(begin,begin+1)}} + \vec{b}_{j_{(begin+1,begin+2)}} + \ldots + \vec{b}_{j_{(begin+n-1,begin+n)}}. \tag{6.6}$$

In this study, the value n was set to 20 or the total number of voxels of the tracklet if it had

less than 20 voxels in its composition. The angle between directional vectors $\vec{A}_i$ and $\vec{B}_j$ is then computed as

$$\alpha_{i,j} = \arccos \frac{\vec{A}_i \cdot \vec{B}_j}{\| \vec{A}_i \| \cdot \| \vec{B}_j \|}. \tag{6.7}$$

In this case, a connection between the ith and jth tracklets will be only considered when the angle $\alpha_{i,j}$ does not exceed 25°. The smaller the angle, the higher the chance these two tracklets will be connected.

In addition to the angular difference $\alpha_{i,j}$, the spatial positions of detected leukocytes from the 2D processing stage were also taken into account by the proposed algorithm. It was implemented to mitigate the effect of residual motion artifacts, which appear in the spatiotemporal image as false track fragments. For that, the spatial position (x,y,t) of each voxel in the straight-line connecting a_{end_i} and b_{begin_j} (created by using the Bresenham (JOY, 1999) algorithm) was compared with the same position in the corresponding output frame of the 2D processing stage. In this case, the number of mutual points, confirmed by the assessment of a circular region of 2-pixel radius around the corresponding (x,y) position in the 2D resulting image, was recorded $(nLeukocytes_{2D}(i,j))$ along the entire time evolution (t) for each ith-jth tracklet connection candidate, and compared with the total number of points in the connection line $(nPoints_{2D+t}(i,j))$ as

$$C_1(i,j) = \frac{nLeukocytes_{2D}(i,j)}{nPoints_{2D+t}(i,j)}. \tag{6.8}$$

Since the number of leukocytes detected during the 2D processing stage is always less than or equal to the number of points in the connection line, then the C_1 measure will remain in the $[0,1]$ range.

To determine to which jth tracklet the endpoint $a_{i_{end}}$ should be connected to, a weighting function ranging from zero to unity was devised as

$$C(i,j;\beta) = \beta C_1(i,j) + (1-\beta)C_2(i,j), \tag{6.9}$$

where $C_2(i,j) = \cos(\alpha_{i,j})$ and $0 \leq \beta \leq 1$. In this study, the parameter β was set to 0.5. In this case, a $C(i,j;\beta)$ value close to one means a high probability of a proper connection to happen. After that, each candidate to the connection was stored in a vector of candidates containing its respective initial and final points, followed by its $C(i,j;\beta)$ value. After evaluating and storing all possible candidates, we search for the best connections in the vector created. Therefore, the segment with the highest $C(i,j;\beta)$ value will be the one selected for connection on each case.

The cone-shaped search region was gradually modified to allow the algorithm to correctly reach and connect tracklets that were far apart from each other. In this case, the parameters (h,θ)

of the cone shape were iteratively changed. The first parameter h varied from 5 to 30 voxels in steps of 5, while the second, θ, varied from 90° to 15° in steps of $-15°$. This procedure decreases the lateral region of the search and increases the extension that the algorithm can reach along the iterations. Figure 6.12 shows this parameter variation graphically and makes it clear that when considering only one value for each parameter in the cone initialization (dashed line in the figure), it would cover a vast and probably unnecessary image region, causing a large number of wrong connections.

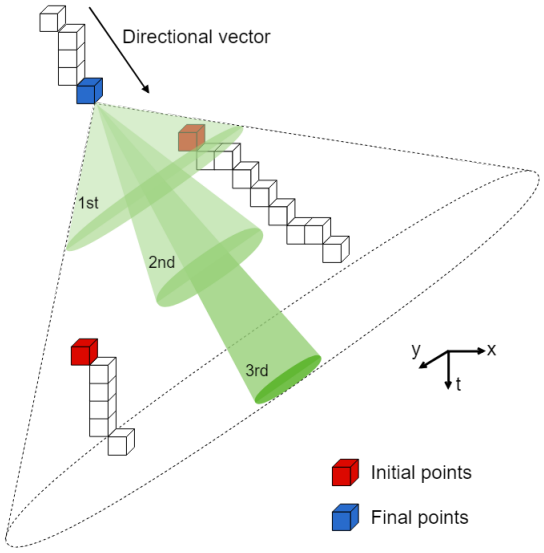


Figure 6.12: Iterations of the algorithm to create the cone-shaped regions

In summary, the algorithm for track linking follows the steps below to identify a fragment to be connected. These steps can also be observed in the illustrations of Figure 6.11.

1. Select the final point a_{end} of a tracklet i;
2. Compute the directional vector $\vec{A}_i$ using the 20 previous voxels of the tracklet;
3. Parameters of the cone are adjusted according to the iteration number and vector direction computed in the previous step;
4. For each initial point $b_{j_{begin}}$, check whether it lies within the cone spatial region;
5. In an affirmative case, compute the directional vector $\vec{B}_j$ and the angular difference α_{ij};
6. If the angular difference α_{ij} between the analyzed vectors was less or equal than 25°, then compute $C(i, j; \beta)$ measure and store the connection information in a vector of candidates;

7. After all candidates were stored, start a search for the best ones, i.e., select those whose C value was higher than the others;

8. A connection is then performed by creating a line (using the Bresenham (JOY, 1999) technique) linking the final and initial points of the selected candidates.

As a consequence of this process, we have our cell trajectories correctly connected, and a final step needs to be applied before the extraction of cell statistical measures. This step consists of upsampling the images and trajectory points back to their original sizes. Accordingly, we applied a bilinear interpolation process with the same scale factor used for image downsampling in our first step. After that, the leukocyte recruitment analysis was finally performed.

6.6 Tracking evaluation

Similarly to the 2D processing evaluation, we assessed our 2D+t approach by comparing the automatic tracking with the leukocytes' manual annotations. The only difference in this process is that the cell annotations are now labeled. Thereby, for each track detected by the proposed approach, an initial coordinate in the ground truth is searched within a radius value of r voxels in the same frame number, where $r = \lfloor k_{avg}/2 \rfloor$, i.e., the cells average radius for each video. However, if a valid annotation is found, then the analysis continues to other frames by searching for the positions belonging to the same labeled cell. Final measures of precision, recall, and F_1-score are also computed for this tracking evaluation.

6.7 Final considerations

We detailed in this chapter all the methodology used in our 2D+t processing stage. Although our cell tracking strategy has been interpreted as a detection of three-dimensional structures, many other techniques were employed to guarantee its robustness and reliability. All these techniques played an essential role in the proposed automatic computational pipeline, and the results of their application can be seen in the next chapter.

Chapter 7

RESULTS AND DISCUSSIONS

This chapter presents the results and discussions about several experiments projected and conducted to prove the primary concept of this research. The results, analyses, and discussions presented here follow the progress of the proposed pipeline, starting with the preprocessing stage, then presenting the results of detection approaches, and finally, showing the tracking strategy outcomes and its ability to handle occlusion and trajectory gaps in the spatiotemporal images.

7.1 Preprocessing evaluation

The evaluation of our preprocessing stage started by assessing the algorithm developed to remove frames with excessive motion blur. It is worth noting that only two (B1 and SC) of the six videos were processed at this stage as the other videos do not present significant motion to justify the removal of frames.

Figure 7.1 shows four examples of frames removed from videos B1 and SC. When comparing them with the images in Section 2.5, we can observe a reasonable amount of motion artifacts caused by the animal movement and microscope defocus. As these frames can severely hamper the subsequent registration, detection, and tracking procedures, we removed them from the videos before further analysis.

The stabilization process on the remaining video frames was analyzed as a final step of our preprocessing stage. In this sense, we evaluated our registration framework through the visual analysis of line projections, as described in Subsection 4.5. The results obtained for this technique are illustrated in Figure 7.2. Video ME is not present in this analysis because it did not exhibit significant motion for a visual inspection.

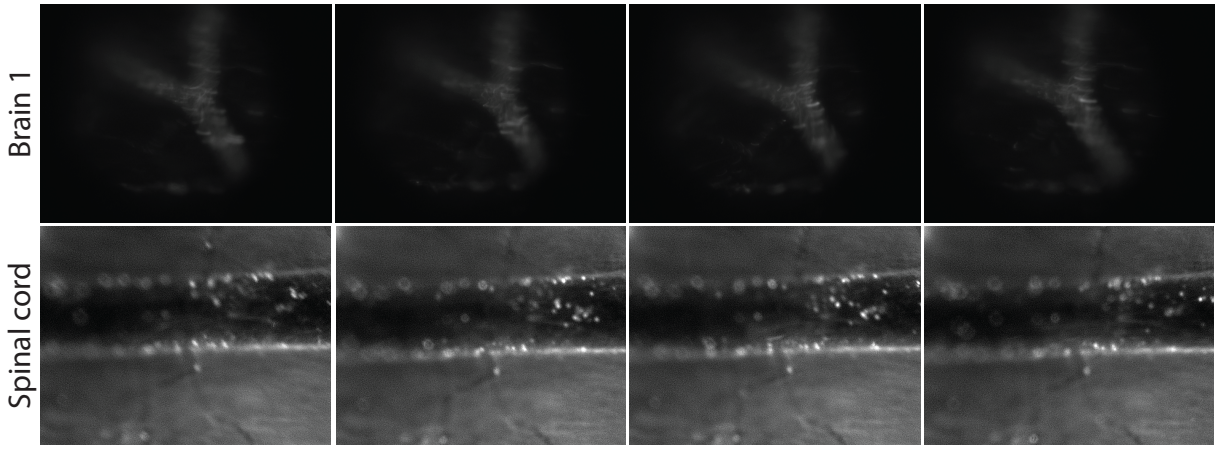


Figure 7.1: Examples of frames removed from the original videos in the first step of our pipeline. Images in the first row were removed from video B1, while images in the second row were removed from video SC.

By comparing the video projections in Figure 7.2, we can notice a sawtooth pattern in the leftmost images. This visual pattern indicates a notable misalignment between video frames before preprocessing. On the other hand, in the rightmost images, this pattern is significantly softened, showing the presence of more continuous lines and, consequently, better alignment between video frames after the frames registration process.

A quantitative evaluation of the video frames registration was also performed in this work. For that, we calculated the PSNR measure for the residual images resulted from the subtraction module of consecutive pairs of video frames. Besides being a measure to evaluate the denoising process, we can also check the alignment between consecutive frames when applying it in residual images. However, the resulting values must not be analyzed alone since we expect that the residual images also depict the movement of the cells over the videos. It means that the average of PSNR values can vary for videos in which cells are mostly stationary and for videos with a cluttered environment, where the movement of cells can create responses in the residual images and consequently decrease the PSNR values. The results for this metric can be seen in Table 7.1 and Figure 7.3.

In this analysis, we did not evaluate the video from the mesentery since mechanical devices can easily stabilize this organ, and consequently, motion blur may not be an issue. We realized that videos from the CNS, for instance, are those more challenging to stabilize, as confirmed by the resulting measures, where the values of variance remained high or higher than before registration techniques. It makes sense considering the nature of the organs, which are located in critical regions of the animal, and the mechanical stabilization is laborious. Consequently, the more significant is the apparent motion, the more out-of-focus the images are.

As stated before, another point to be considered in the analysis is the leukocytes movement.

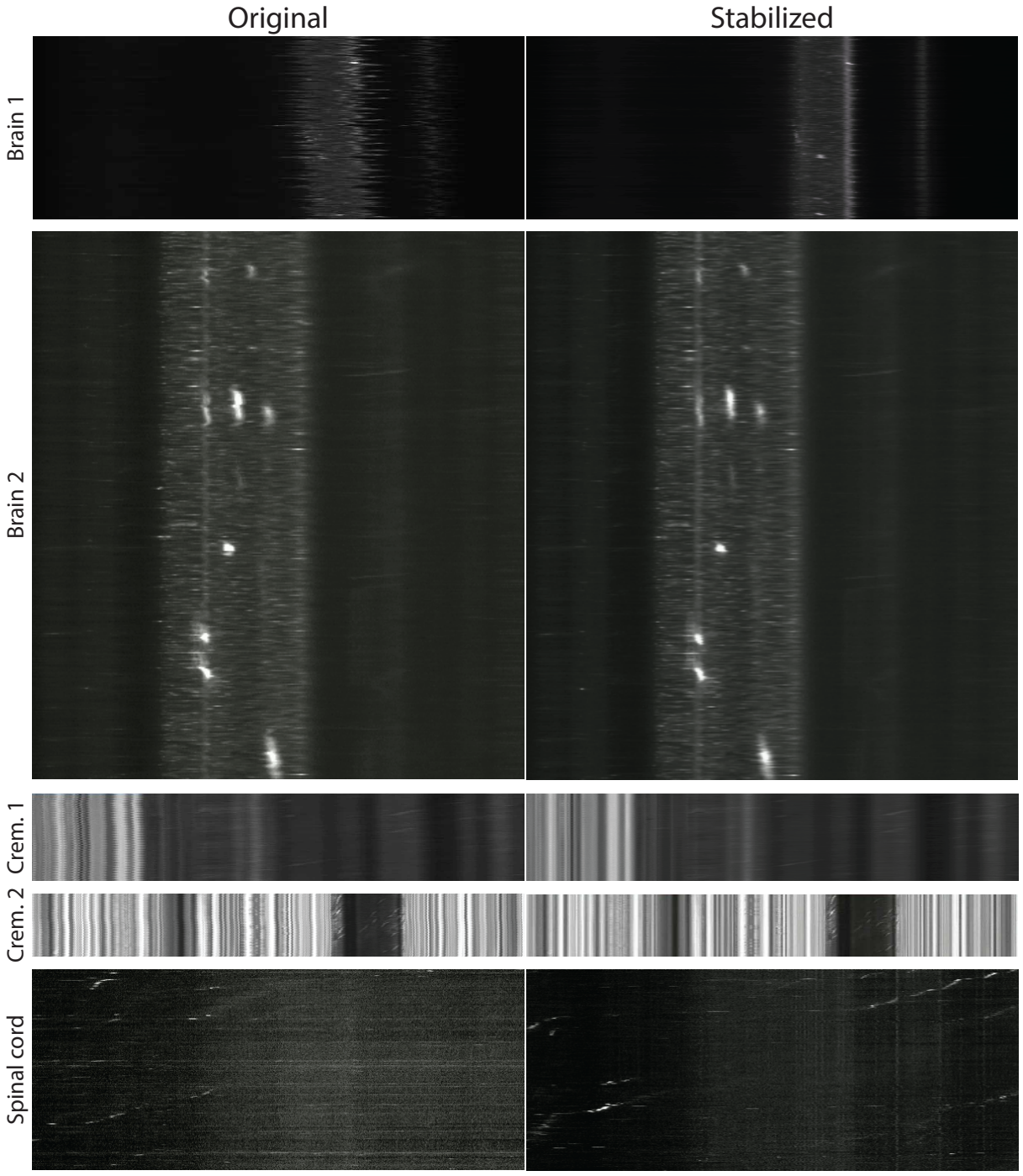


Figure 7.2: Output images from line projection technique. First and second columns show the projections before and after the registration process, respectively.

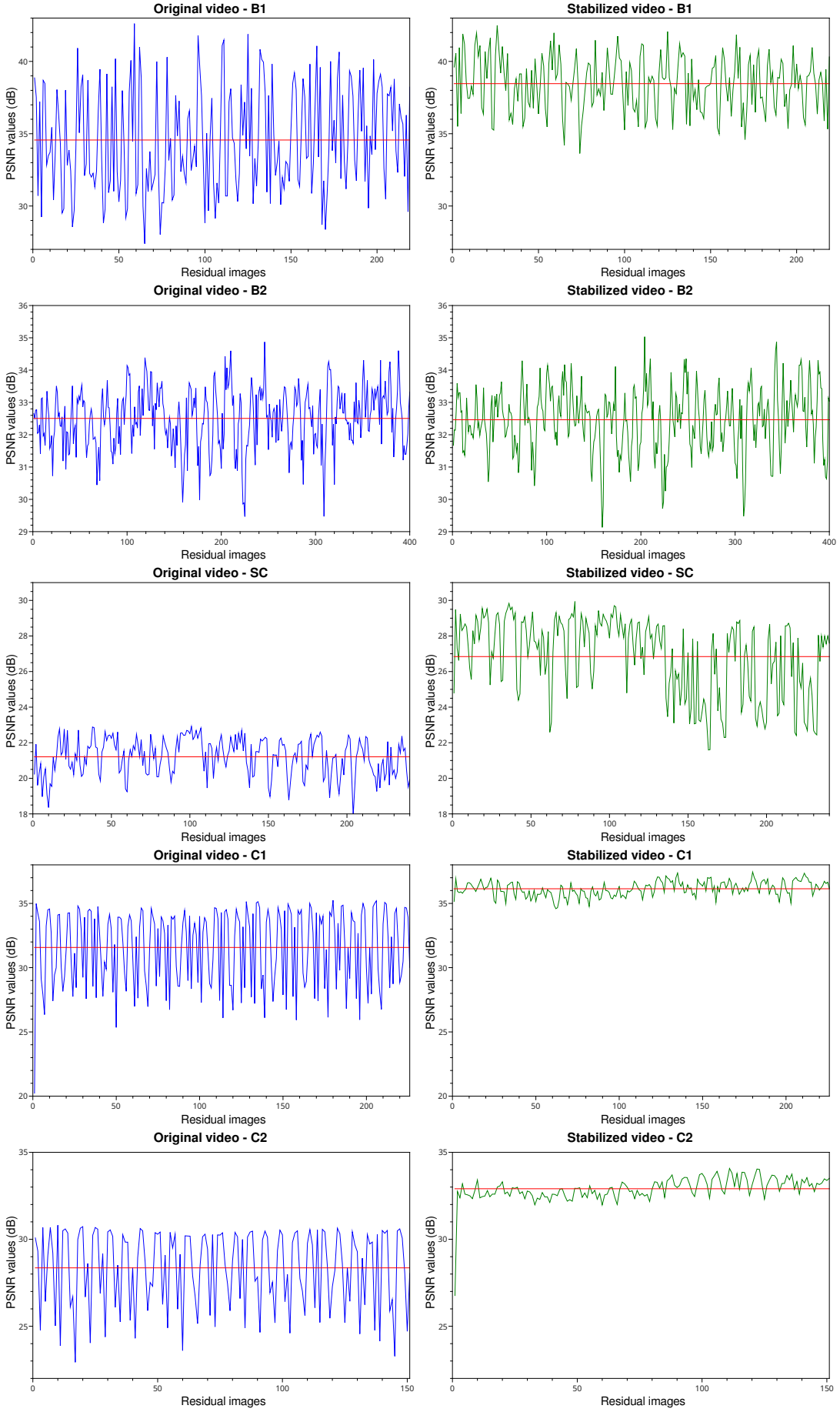


Figure 7.3: PSNR values computed for all the residual images resulting from the subtraction module of consecutive pairs of video frames.

Table 7.1: PSNR values for the IVM videos assessed in this research.

Video		Average (dB)	Variance (dB)
B1	Original	34.56	12.79
	Stabilized	38.46	3.54
B2	Original	32.51	0.85
	Stabilized	32.46	0.93
SC	Original	21.21	1.09
	Stabilized	26.84	4.74
C1	Original	31.57	8.68
	Stabilized	36.14	0.33
C2	Original	28.36	4.94
	Stabilized	32.91	0.49

In clutter environments, if cells are moving fast, the residual image will present high responses, decreasing the PSNR value. We separated two cases to exemplify these problems in the video from the animal's spinal cord. They are shown in Figure 7.4.

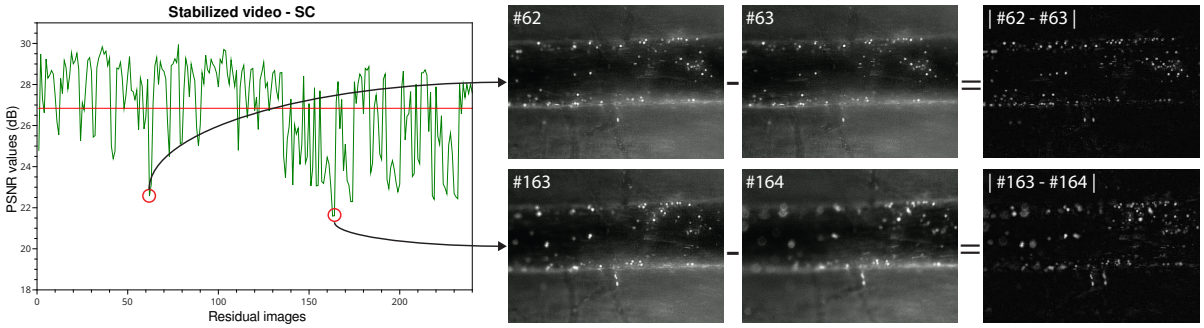


Figure 7.4: Examples of images with low PSNR values.

In the first case, the residual image from frames number 62 and 63 is formed only by the leukocytes movement. Thus, since this organ presents a high number of flowing cells, the PSNR value was low. Also, the process of frames removal contributed to a low PSNR in this case, in which three middle frames were eliminated because of their poor quality.

In the other example of Figure 7.4, corresponding to the frames number 163 and 164, we can notice the consequence of a defocused image in the performance of our approach. The residual image, in this case, is mainly composed of the difference in contrast between two consecutive video frames.

7.2 Detection evaluation

In this section, we present the results obtained for the three approaches used in the 2D processing stage. All these approaches include the method evaluation in different scenarios, as can be observed in the next subsections.

7.2.1 Results for the MTM-PCA

We evaluated our first approach in four different scenarios. They involve the employment of single or multiple features, PCA applied to frames or templates, and the use of single or multiple templates in the template matching technique. In all the experiments, we used the precision, recall, and F_1-score measures to compute the overall method performance. With these measures, we could generate the so-called precision-recall curves as well as calculate the area under them, the AUCPR. Each point in the curves corresponds to an operation point of the applied approach that was computed by setting a threshold value for the circularity score Z in the 2D post-processing stage (see Subsection 5.1.4). We varied this threshold in the range $[0, 0.99]$ and compared the results with the cells manually annotated to obtain the curve points in the following experiments.

7.2.1.1 Experiment 1: simple MTM

In the first experiment, we investigated different numbers of templates for the MTM algorithm applied over only one feature, the raw pixel or gray-level intensity. In this case, we did not use the PCA technique for dimensionality reduction. The evaluation was performed for one, two, and three manually selected templates as input for the MTM algorithm. The precision-recall curves can be seen in Figure 7.5. Together with the generated graphs of each video, we can also see the AUCPR values calculated for each scenario.

By observing the graphs in Figure 7.5 and their corresponding AUCPR values, we notice a slight improvement when using more than one template in the MTM algorithm, except for the SC video. The results seem to be consistent as the method is susceptible to small changes in the target object, i.e., the more templates to search, the more chances we have to find the targets. We concluded the same when consulting the values in Table 7.2, which shows the best F_1-score value found for each curve computed in Figure 7.5. Boldfaced values in the table indicate the best results for each video.

Another important observation is the fact that videos from the CNS presented better re-

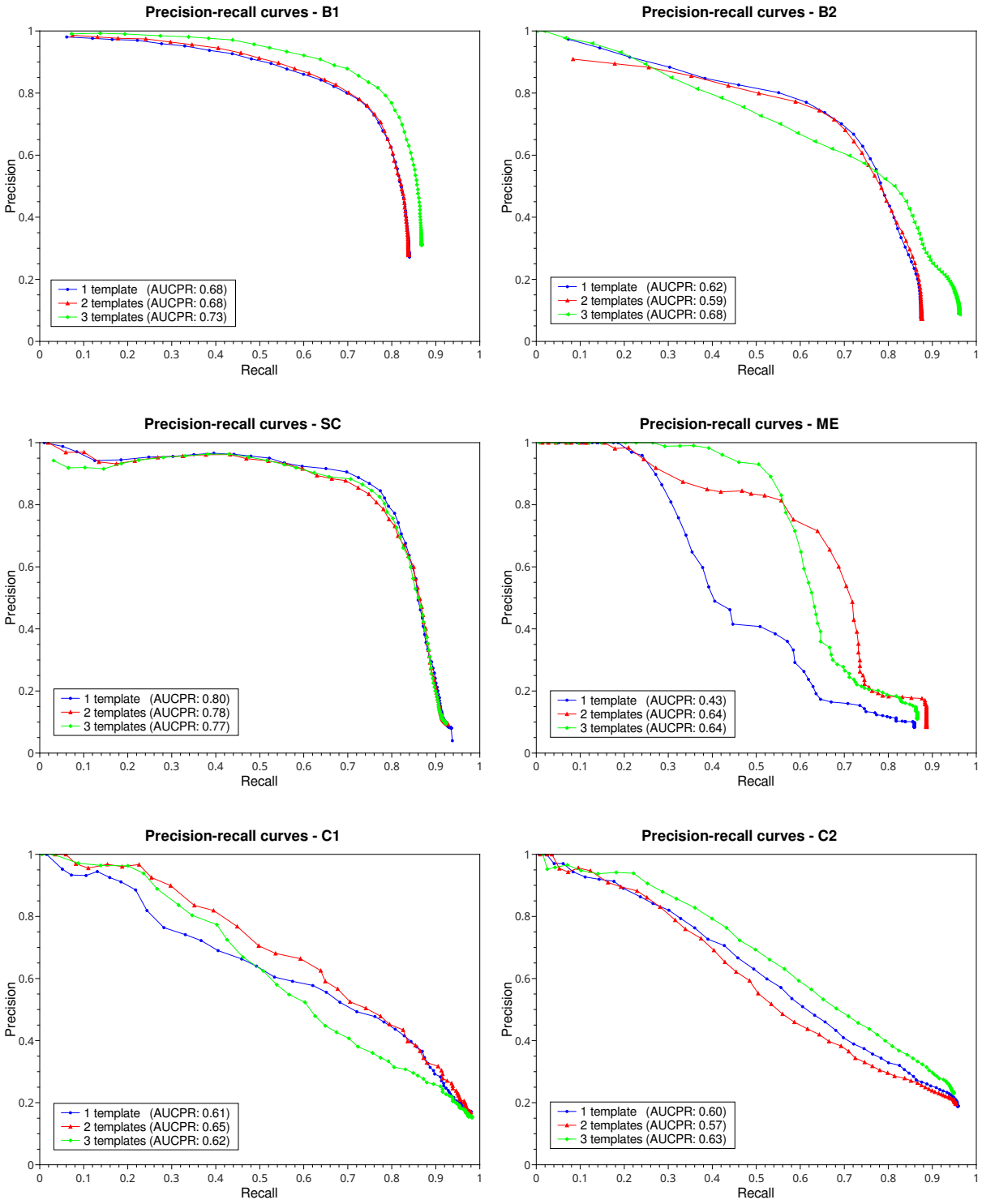


Figure 7.5: Precision-recall curves obtained for the first experiment using all videos in our dataset. For each video, we tested the use of one, two, and three different templates selection.

Table 7.2: Results for the leukocyte detection in experiment 1 according to the best F_1-score values found. The #tmp column indicates the number of templates used.

		Counting					
Video	**#tmp**	**TP**	**FP**	**FN**	**P**	**R**	**F_1**
B1	1	4217	1191	1610	0.78	0.72	0.75
	2	4234	1197	1593	0.78	0.73	0.75
	3	4476	1001	1351	0.82	0.77	**0.79**
B2	1	5581	2377	2467	0.70	0.69	**0.70**
	2	5447	2172	2601	0.71	0.68	0.70
	3	5724	3840	2324	0.60	0.71	0.65
SC	1	1215	223	355	0.84	0.77	**0.81**
	2	1174	233	396	0.83	0.75	0.79
	3	1212	255	358	0.83	0.77	0.80
C1	1	254	203	136	0.56	0.65	0.60
	2	249	149	141	0.63	0.64	**0.63**
	3	235	214	155	0.52	0.60	0.56
C2	1	891	669	712	0.57	0.56	0.56
	2	777	533	826	0.59	0.49	0.53
	3	904	529	699	0.63	0.56	**0.60**
ME	1	110	74	181	0.60	0.38	0.46
	2	186	74	105	0.72	0.64	**0.68**
	3	162	33	129	0.83	0.56	0.67

sponses when compared with others. Although they usually suffer more with inherent imaging problems, these videos have well-defined cells, which can help an appearance-based detection algorithm to achieve better results.

7.2.1.2 Experiment 2: all feature-frames PCA

For the second experiment, we computed all image features described in the Subsection 5.1.1 for each video in our dataset. Next, we extracted the first frame of each feature-video to build the input matrix of the PCA algorithm. After setting the threshold value for the retained variance as 0.95 for the PCA, we projected the remaining data (frames from the feature-videos) into the new PCA bases. With all feature-videos on the new bases, we manually selected the same template images from the last experiment to be used in the MTM algorithm and compared the results with our manual annotations.

The first video frames already projected into the PCA basis are illustrated in Figure 7.6. They are arranged according to the PCA eigendecomposition, i.e., the first image (in the natural reading order) corresponds to the first principal component, the second image corresponds to the second principal component, and so on. From a visual inspection of the feature-images in

Figure 7.6, we can observe that frames from the CNS group presented a good cell contrast for most of the projected images. However, this observation can not be stated for the other image groups, and consequently, their subsequent detection processes will be harmed.

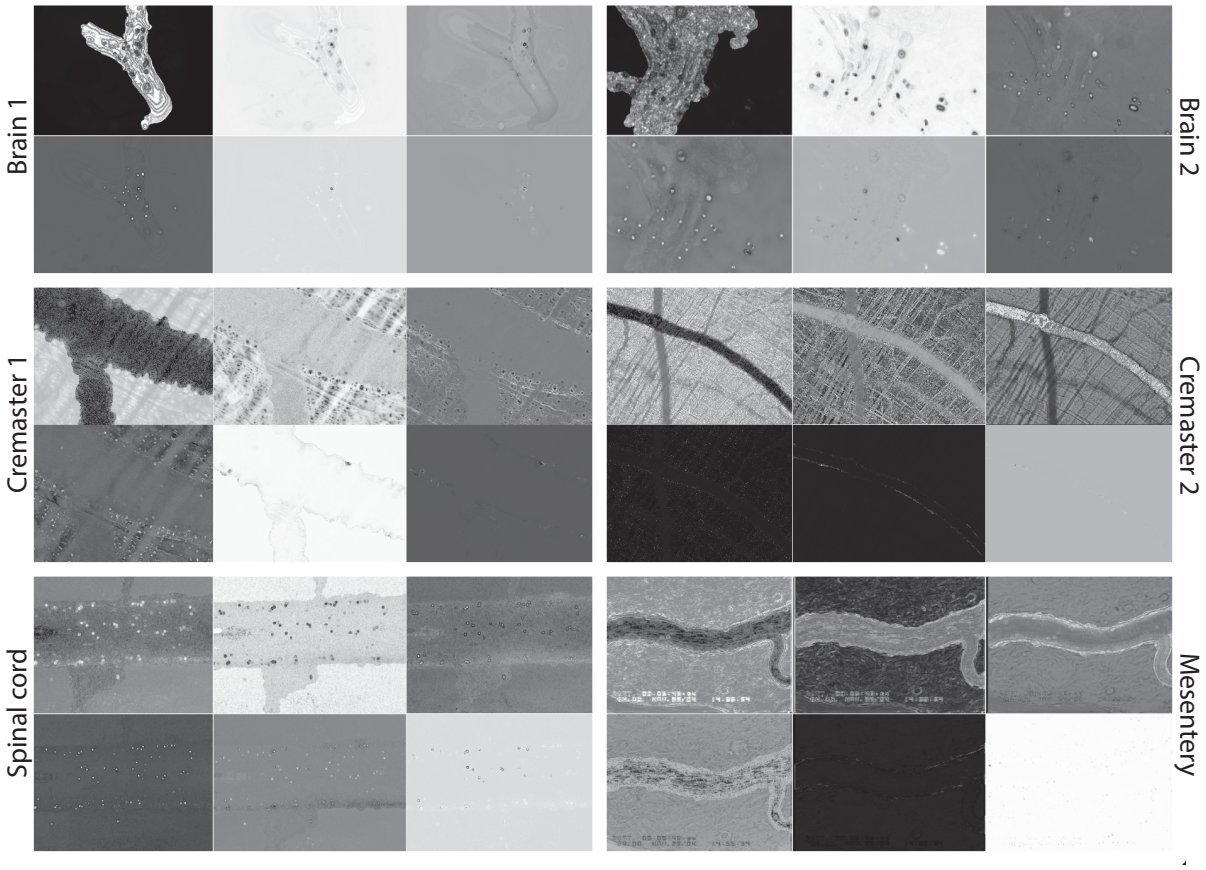


Figure 7.6: First video frames extracted from the feature-videos (already projected into the PCA bases). They are ordered according to the PCA eigendecomposition.

Precision-recall curves were also generated (see Figure 7.7) for this experiment by using one, two, and three template images in each video. According to the graphs in Figure 7.7 and the values in Table 7.3, only the SC video had a better performance if compared with the first experiment. It can be explained if we analyze the template images extracted from this video.

Figure 7.8(a) shows the templates selected in the first experiment for SC video (intensity feature only) and the templates selected for the four PCA components considered in experiment two. In this case, Figure 7.8(b) presents cells with better contrast and less noise when compared to Figure 7.8(a), which probably caused the increase in the number of correct detections.

7.2.1.3 Experiment 3: selected feature-frames PCA

As most results of the second experiment were far from the application of a simple TM technique as in the first experiment, we suspected that some features could not be so useful as

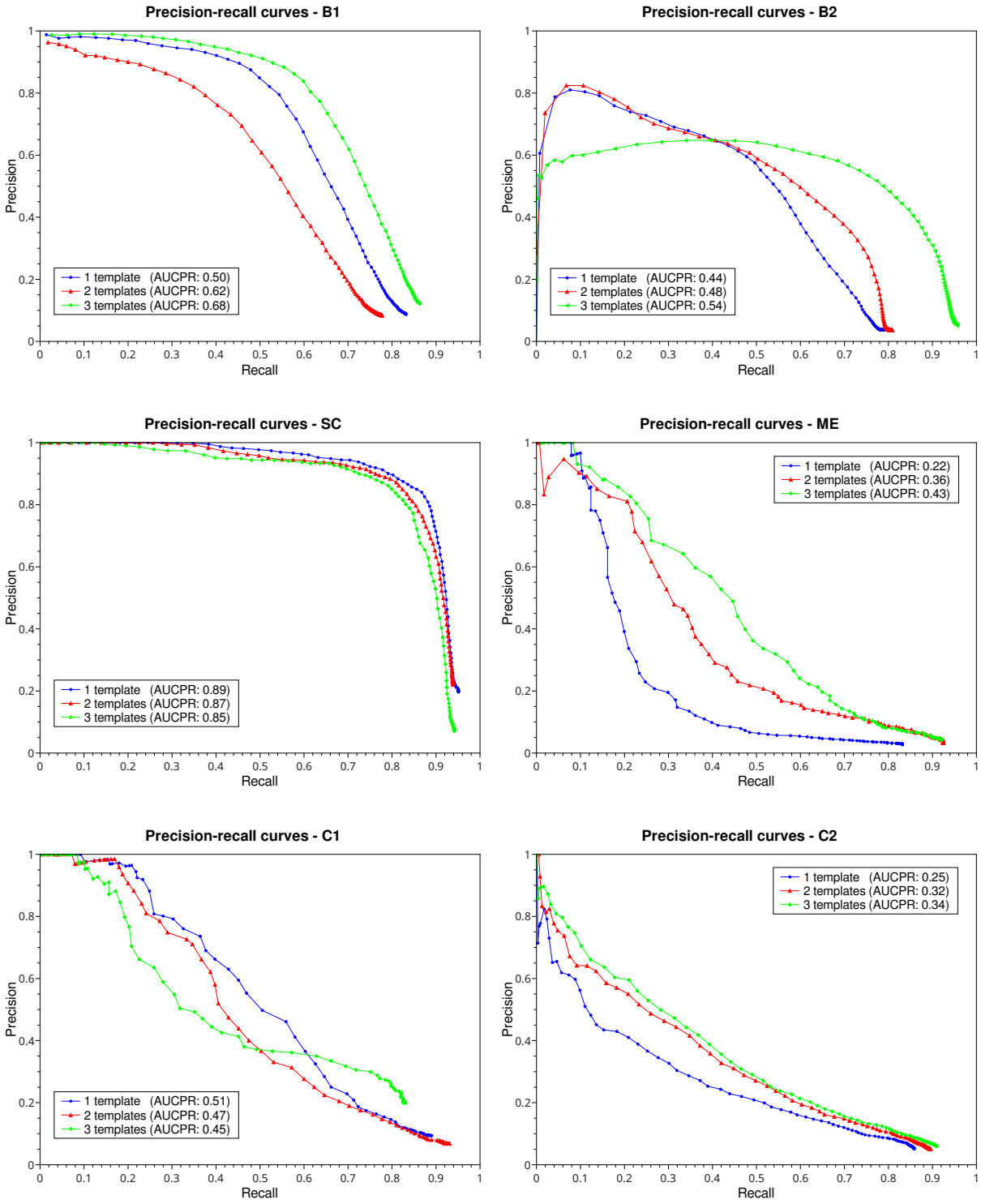
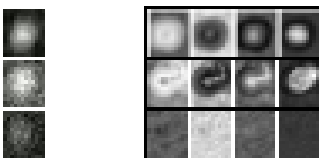


Figure 7.7: Precision-recall curves calculated in the second experiment for all videos in the dataset. For each video, we tested the use of one, two, and three different templates selection.

Table 7.3: Resulting values for the leukocyte detection in experiment 2 according to the best F_1-score values found. The features listed below are as follows: I: intensity, H: Hessian, E: edges, A: inertia, K: Haralick's correlation, and D: Difference Moment. The #tmp and #PC columns indicate the number of templates and the number of principal components used, respectively.

	Features						PCA			Counting					
Video	**I**	**H**	**E**	**A**	**K**	**D**	**#PC**	**Ret.var.**	**#tmp**	**TP**	**FP**	**FN**	**P**	**R**	**F_1**
B1	×	×	×	×	×	×	2	0.9828	1	2809	1535	3018	0.65	0.48	0.55
									2	3167	816	2660	0.80	0.54	0.65
									3	3490	674	2337	0.84	0.60	**0.70**
B2	×	×	×	×	×	×	3	0.9673	1	3996	2948	4052	0.58	0.50	0.53
									2	4511	3849	3537	0.54	0.56	0.55
									3	5696	4344	2352	0.57	0.71	**0.63**
SC	×	×	×	×	×	×	4	0.9754	1	1362	259	208	0.84	0.87	**0.85**
									2	1271	186	299	0.87	0.81	0.84
									3	1242	199	328	0.86	0.79	0.82
C1	×	×	×	×	×	×	3	0.9701	1	176	120	214	0.59	0.45	**0.51**
									2	151	92	239	0.62	0.39	0.48
									3	245	455	145	0.35	0.63	0.45
C2	×	×	×	×	×	×	3	0.9939	1	596	1600	1007	0.27	0.37	0.31
									2	558	784	1045	0.42	0.35	0.38
									3	589	819	1014	0.42	0.37	**0.39**
ME	×	×	×	×	×	×	4	0.9944	1	55	65	236	0.46	0.19	0.27
									2	97	112	194	0.46	0.33	0.39
									3	122	109	169	0.53	0.42	**0.47**



(a) (b)

Figure 7.8: Templates manually selected for the (a) first and (b) second experiments with SC video. Images in (a) are from the intensity feature, and images in (b) are from the PCA features.

Table 7.4: Resulting values for the leukocyte detection in experiment 3 according to the best F_1-score values found. The features listed below correspond to I: intensity, H: Hessian, E: edges, A: inertia, K: Haralick's correlation, and D: Difference Moment. The #tmp and #PC columns indicate the number of templates and the number of principal components used, respectively.

	Features						PCA			Counting					
Video	**I**	**H**	**E**	**A**	**K**	**D**	**#PC**	**Ret.var.**	**#tmp**	**TP**	**FP**	**FN**	**P**	**R**	**F_1**
B1	×	×	×				2	0.9733	1	4582	772	1245	0.86	0.79	0.82
									2	4601	776	1226	0.86	0.79	0.82
									3	4687	628	1140	0.88	0.80	**0.84**
B2	×					×	2	1.0	1	3089	2792	4959	0.53	0.38	0.44
									2	3639	2676	4409	0.58	0.45	0.51
									3	5774	2837	2274	0.67	0.72	**0.69**
SC	×		×				2	1.0	1	1176	169	394	0.87	0.75	**0.81**
									2	1165	203	405	0.85	0.74	0.79
									3	1081	263	489	0.80	0.69	0.74
C1	×	×	×	×	×	×	3	0.9701	1	176	120	214	0.59	0.45	**0.51**
									2	151	92	239	0.62	0.39	0.48
									3	245	455	145	0.35	0.63	0.45
C2	×	×	×		×		2	0.9923	1	826	594	777	0.58	0.52	0.55
									2	888	331	715	0.73	0.55	0.63
									3	979	325	624	0.75	0.61	**0.67**
ME	×	×	×				2	0.9966	1	106	52	185	0.67	0.36	0.47
									2	173	40	118	0.81	0.59	**0.69**
									3	152	17	139	0.90	0.52	0.66

they should be. For this reason, we decided to select only those features that are contributing to the cell detection. The idea behind this approach is trying to determine what type of feature is better with a particular kind of image (or organ). In this sense, we performed the feature selection using the forward-searching method (CHANDRASHEKAR; SAHIN, 2014), which is based on a wrapper selection strategy.

The results for this embedding approach are shown in the graphs of Figure 7.9 and the values of Table 7.4. The same earlier studies were performed in this experiment, but now we used only the image features specified in Table 7.4 to build the input matrix of the PCA algorithm.

By checking the AUCPR values in Figure 7.9, we noticed an improvement in some resulting measures, which confirms the hypothesis that some features are not contributing to the overall performance of the algorithm. Indeed, if we compare the retained variance values and the number of principal components considered in each video, we observe that even with fewer features and fewer PCA components, the algorithm was able to retain more significant data information.

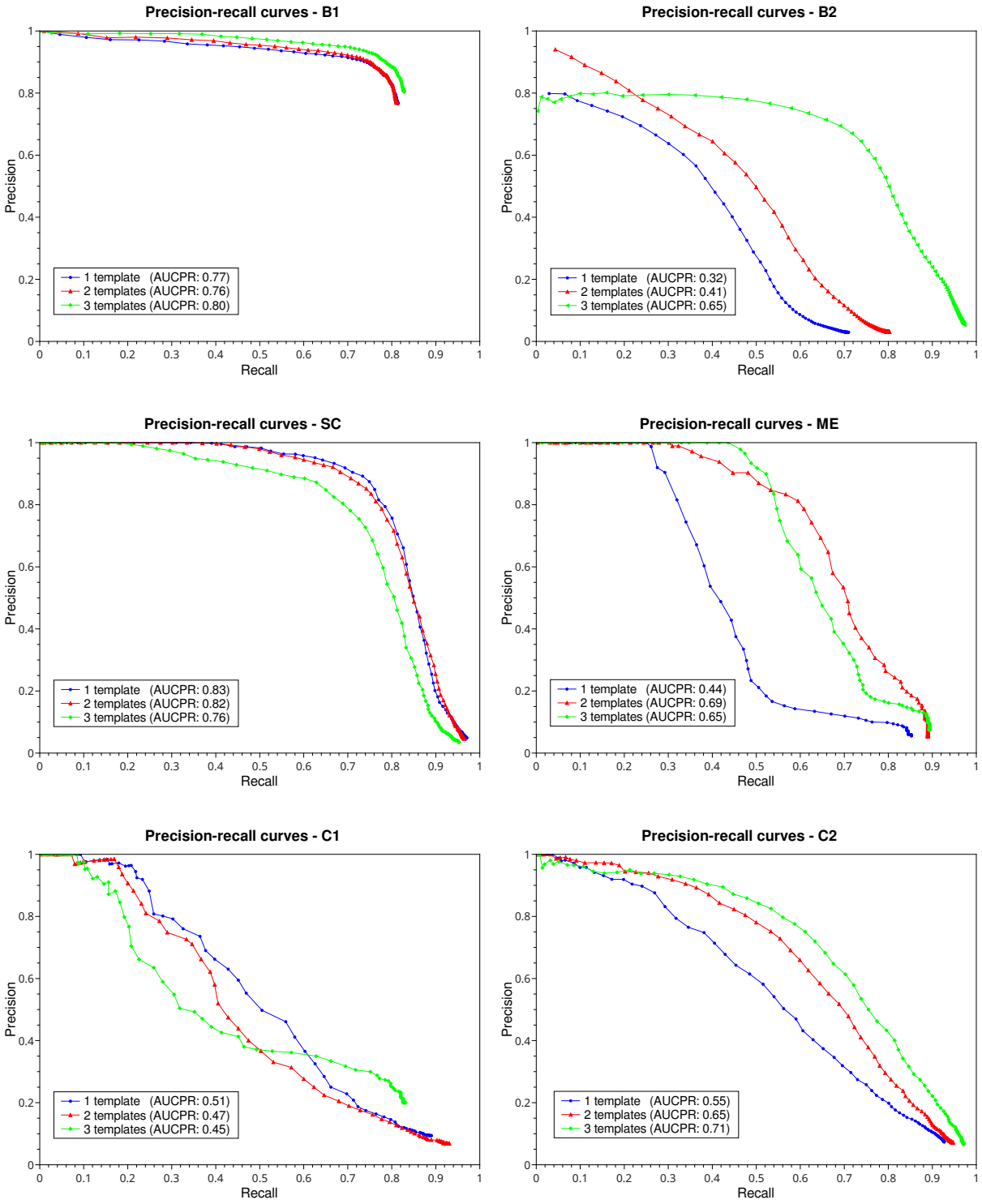


Figure 7.9: Precision-recall curves calculated in the third experiment for all videos in the dataset. For each video, we tested one, two, and three different templates.

7.2.1.4 Experiment 4: feature-templates PCA

To test the behavior of the PCA technique considering only the image template information, we performed our last experiment for the MTM-PCA approach by first selecting the template images and then applying the PCA technique on them. In other words, after selecting and extracting the templates from the first frame of each feature-video, we built the input matrix for the PCA algorithm using the template images.

However, as can be seen in Table 7.5, we used only one template for each video to perform the tests. The reason for that is related to the template images, which have different sizes. This characteristic does not allow the creation of an input matrix for PCA without a proper correction on the size of templates.

Table 7.5: Resulting values for the leukocyte detection in experiment 4 according to the best F_1-score values found. The features listed below are as follows: I: intensity, H: Hessian, E: edges, A: inertia, K: Haralick's correlation, and D: Difference Moment. The #tmp and #PC columns indicate the number of templates and the number of principal components used, respectively.

	Features						PCA			Counting					
Video	**I**	**H**	**E**	**A**	**K**	**D**	**#PC**	**Ret.var.**	**#tmp**	**TP**	**FP**	**FN**	**P**	**R**	**F_1**
B1	×	×	×	×	×	×	2	0.9638	1	3602	460	2225	0.89	0.62	0.73
	×	×		×			1	0.9818	1	4547	990	1280	0.82	0.78	**0.80**
B2	×	×	×	×	×	×	2	0.9595	1	5213	3029	2835	0.63	0.65	0.64
	×					×	2	1.0	1	4864	1485	3184	0.77	0.60	**0.68**
SC	×	×	×	×	×	×	2	0.9668	1	1275	198	295	0.87	0.81	0.84
	×		×				2	1.0	1	1257	173	313	0.88	0.80	**0.84**
C1	×	×	×	×	×	×	2	0.9770	1	286	141	104	0.67	0.73	**0.70**
	×	×	×	×	×	×	2	0.9770	1	286	141	104	0.67	0.73	**0.70**
C2	×	×	×	×	×	×	2	0.9814	1	724	468	879	0.61	0.45	0.52
	×	×	×		×		2	0.9950	1	950	329	653	0.74	0.59	**0.66**
ME	×	×	×	×	×	×	3	0.9740	1	71	77	220	0.48	0.24	0.32
	×	×	×				2	0.9714	1	167	124	124	0.57	0.57	**0.57**

The resulting measures for all image features and those previously selected in the third experiment are shown in the graphs of Figure 7.10 and Table 7.5. These results demonstrated a good performance even when only the template images were used for the PCA algorithm. However, they did not overcome the results from other experiments, except for video C1. As can be observed in the graphs of Figure 7.10, the algorithm becomes very sensitive to changes in the input matrix, i.e., a significant difference is observed in the curves using all features or only a part of them. Probably, this effect was caused by the few information provided by the small image templates employed. This fact also explains why video C1 had a good performance compared to the other experiments, whereas its image matrix size is quite big and its cells are

zoomed in the images, causing the selection of 63×63 templates while in the other videos we have template sizes varying around the average of 19 pixels.

7.2.1.5 MTM-PCA overall analysis

To facilitate the overall analysis of our first detection approach, we created a summary table containing the best results obtained for each video in all experiments described previously. Table 7.6 shows these results considering the F_1-score and AUCPR measures. The values highlighted in the table correspond to the best results for each video, and experiment 4 was divided into tests performed for all features (A) and for the ones previously selected (B).

Table 7.6: Results from all the experiments performed for our first detection approach using the IVM dataset. The best results are highlighted in bold face.

	Exp. 1		Exp. 2		Exp. 3		Exp. 4 (A)		Exp. 4 (B)	
Video	**AUCPR**	**F_1**	**AUCPR**	**F_1**	**AUCPR**	**F_1**	**AUCPR**	**F_1**	**AUCPR**	**F_1**
B1	0.73	0.79	0.68	0.70	**0.80**	**0.84**	0.69	0.73	0.73	0.80
B2	**0.68**	**0.70**	0.54	0.63	0.65	0.69	0.60	0.64	0.64	0.68
SC	0.80	0.81	**0.89**	**0.85**	0.83	0.81	0.87	0.84	0.86	0.84
C1	0.65	0.63	0.51	0.51	0.51	0.51	**0.75**	**0.70**	**0.75**	**0.70**
C2	0.63	0.60	0.34	0.39	**0.71**	**0.67**	0.50	0.52	0.69	0.66
ME	0.64	0.68	0.43	0.47	**0.69**	**0.69**	0.28	0.32	0.56	0.57

In the videos B1, C2, and ME, for instance, we observed a considerable difference in the experiment results, which indicates that some of the features may be causing a decrease in the resulting metrics. Videos B2 and SC, however, presented results very similar over the experiments, indicating that the contribution of each feature is, at least, feasible for this detection approach. Regarding C1 video, we hypothesize that the image background in the features computed is profoundly influencing (in a negative sense) the PCA eigendecomposition. This assumption is highlighted by the fact that C1 was the only video in experiment 3 that used more than two principal components to retain the necessary variance, and used all the features even after applying the feature selection strategy. Thus, by working with only its image templates in the PCA algorithm of experiment 4, we could achieve better results.

Therefore, in conclusion to this overall analysis, we presume that the extraction and selection of essential feature images still require a more thorough investigation to improve the results obtained so far using IVM images from different organs.

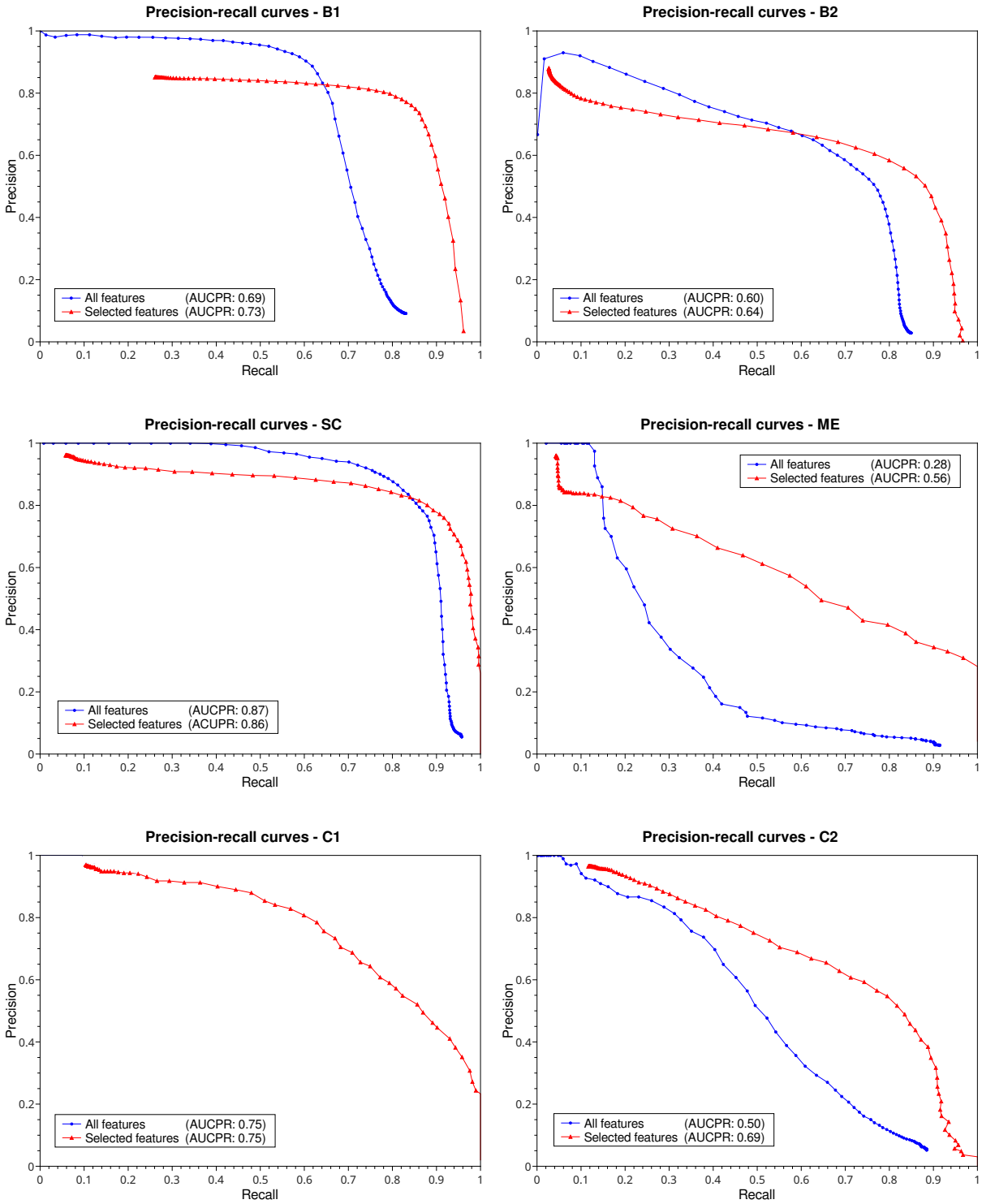


Figure 7.10: Precision-recall curves calculated in experiment 4 for all videos in the dataset. For each video, we tested only one template and the two strategies for feature selection used in experiments 2 and 3.

7.2.2 Results for the MTM-DCNN

The second approach for leukocytes detection was quantitatively evaluated for different pre-trained DCNN models using all the videos from our dataset. A list with the names of all these models and their respective references can be found in Table 7.7.

In Figure 7.11, we can see the resulting F_1-score values for all the videos and models processed. For each one of them, we plotted the best values found in our experiments, considering the set of threshold values tested. Each model exhibits three different bars, colored according to the number of templates used in our experiments with the MTM algorithm.

The plots in Figure 7.11 show the use of selected convolutional layers in CNNs positively contribute as generic feature extractors, even coming from models trained for a completely different task. It is also worth noticing that the use of multiple templates can help to recognize targets that are slightly different, as stated in the results of our first detection approach.

Although some models did not contribute to the MTM technique in most cases, such as the ResNet50V2, ResNet101V2, ResNet152V2, InceptionV3, and InceptionResNetV2, the majority of them achieved reasonable values if compared with the results from the previous approach, which also shows the potential of this detection strategy.

Table 7.7 shows the best set of values found in Figure 7.11 for a better quantitative comparison of the methods. Indeed, when compared with the most common application of MTM, i.e., using only the raw pixel or gray-level information (see Subsection 7.2.1.1), this approach presented a considerable improvement (up to 14%). Videos C1 and C2, however, still exhibited low F_1-score values (0.66 and 0.60, respectively), which is justifiable since they have the most challenging visual aspects, with a cluttered background and cell sizes in the order of 5 pixels.

Examples of output frames for each processed video are shown in Figure 7.12. Each TP point found is illustrated by the green circles in the images, with its respective manual centroid annotation indicated as a cross. The blue circles represent the FP points, while the red squares are the FN ones.

From the images in Figure 7.12, we observe that FN points are often the cells very close to each other or the ones whose appearance is quite different from the rest of them. The FP points, however, mostly correspond to bright regions in the images or to the erythrocytes, which are smaller cells that appear as bright blurred points in non-consecutive frames and are not part of the manual annotations. Even so, the final results for this detection approach were quite promising and indicated that pre-trained DCNN models could also be a good option for generic feature extraction in IVM.

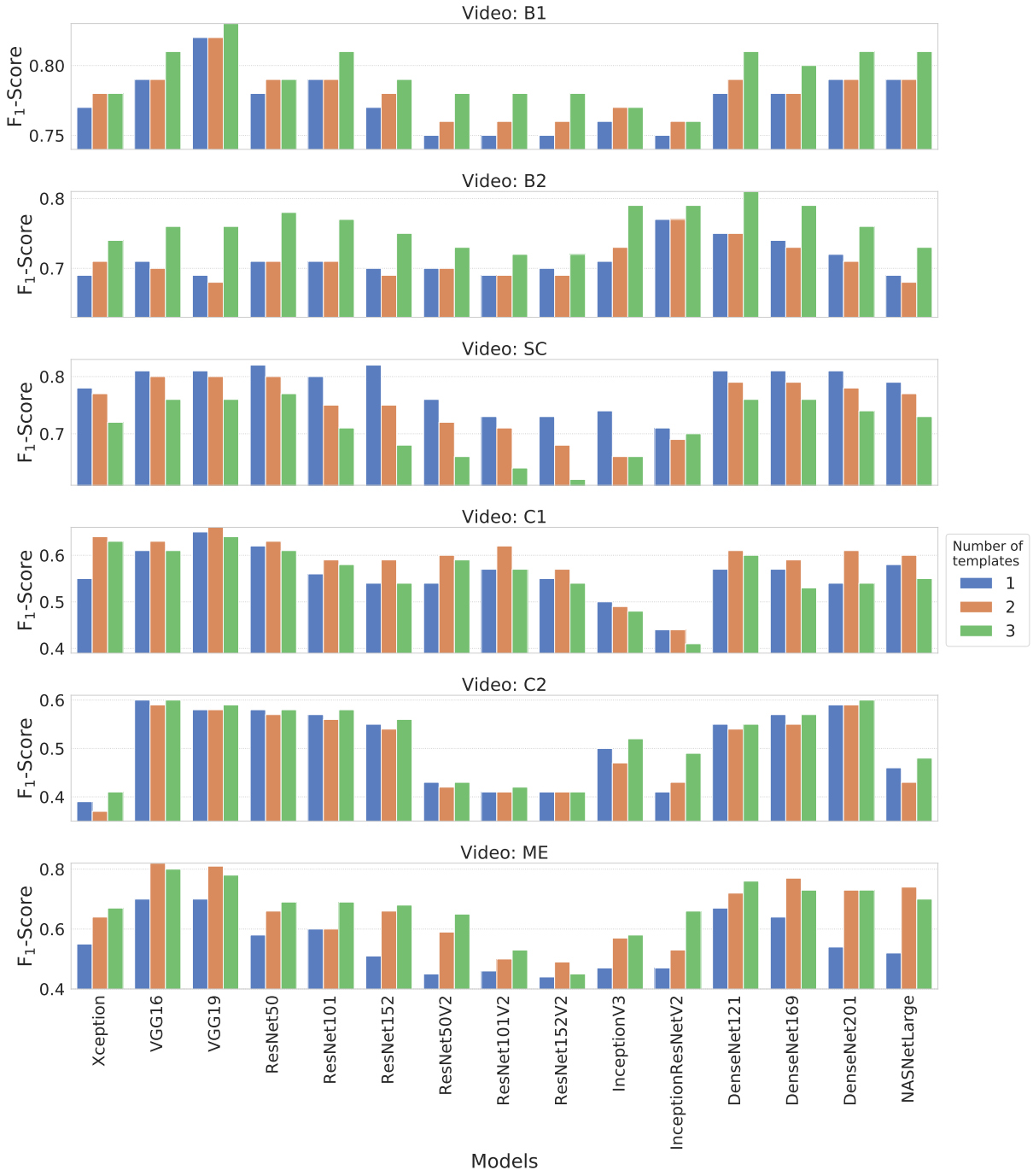


Figure 7.11: Output values for each video and DCNN model tested. The colored bar indicates the number of templates used in the MTM algorithm.

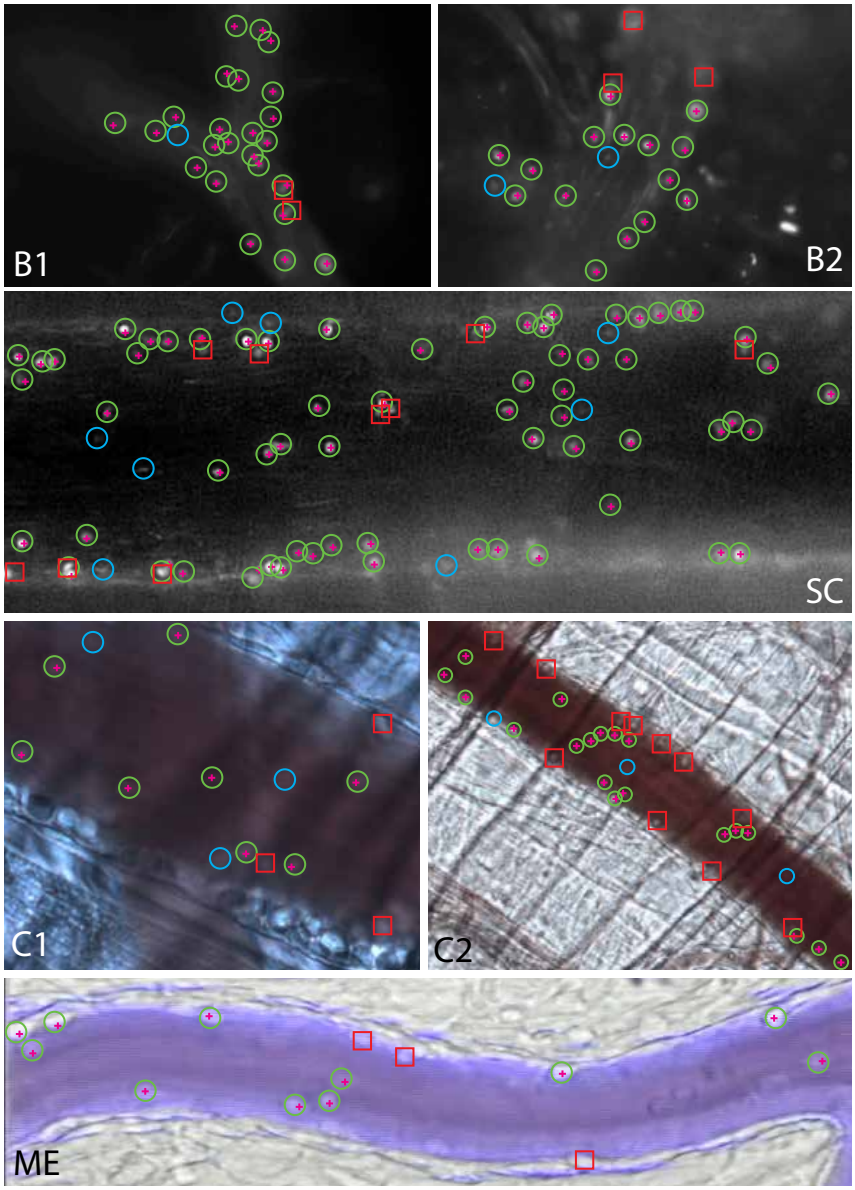


Figure 7.12: Examples of MTM-DCNN outputs for each video in the dataset. Green circles represent the TP points, blue circles the FP points, and red squares the FN points.

Table 7.7: Best F_1-score values found for each video and DCNN model.

DCNN model	B1	B2	SC	C1	C2	ME
Xception (CHOLLET, 2017)	0.78	0.74	0.78	0.64	0.41	0.67
VGG16 (SIMONYAN; ZISSERMAN, 2015)	0.81	0.76	0.81	0.63	**0.60**	**0.82**
VGG19 (SIMONYAN; ZISSERMAN, 2015)	**0.83**	0.76	0.81	**0.66**	0.59	0.81
ResNet50 (HE et al., 2016a)	0.79	0.78	**0.82**	0.63	0.58	0.69
ResNet101 (HE et al., 2016a)	0.81	0.77	0.80	0.59	0.58	0.69
ResNet152 (HE et al., 2016a)	0.79	0.75	0.82	0.59	0.56	0.68
ResNet50V2 (HE et al., 2016b)	0.78	0.73	0.76	0.60	0.43	0.65
ResNet101V2 (HE et al., 2016b)	0.78	0.72	0.73	0.62	0.42	0.53
ResNet152V2 (HE et al., 2016b)	0.78	0.72	0.73	0.57	0.41	0.49
InceptionV3 (SZEGEDY et al., 2016)	0.77	0.79	0.74	0.50	0.52	0.58
InceptionResNetV2 (SZEGEDY et al., 2017)	0.76	0.79	0.71	0.44	0.49	0.66
DenseNet121 (HUANG et al., 2017)	0.81	**0.81**	0.81	0.61	0.55	0.76
DenseNet169 (HUANG et al., 2017)	0.80	0.79	0.81	0.59	0.57	0.77
DenseNet201 (HUANG et al., 2017)	0.81	0.76	0.81	0.61	0.60	0.73
NASNetLarge (ZOPH et al., 2018)	0.81	0.73	0.79	0.60	0.48	0.74

7.2.3 Results for the DCNN

In this subsection, we present the results for our third detection approach, where a modified version of the RetinaNet model was used to detect the leukocytes in the IVM images since this architecture has demonstrated significant results in small object detection compared to other common methods in the literature. We start, however, by showing the results of the model hyperparameters setting and the influence of the data augmentation strategy in our dataset.

Quantitative measures are then presented using four different experiments varying the images used in training to check the robustness of the model with and without data augmentation. All the output measures presented for these four experiments were obtained from a cross-validation procedure with five stratified folds. For each fold, we retrained the model from its initial state and then analyzed the statistical measures for thresholds $\text{IoU}_{0.25}$ and $\text{IoU}_{0.5}$.

The remaining values for NMS and score thresholds were evaluated using the grid search algorithm in the validation datasets with the maximum number of detections per image fixed in 500 objects. Therefore, each outcome presented in the following experiments was assessed in the corresponding test dataset using the best set of hyperparameters found while examining in the validation dataset after model training. The partitions of training, validation and test datasets were created differently for each fold in a stratified way.

After these experiments, we compared our model with other methods by taking the centroids of our detected bounding boxes and analyzing their counting and location precisions. A

better description of each analysis is presented as follows.

7.2.3.1 Hyperparameters setting

In order to find the best set of hyperparameters, we performed some experiments using 20% of all video frames in our dataset randomly for testing, and from the remaining images, we used 80% for training and 20% for validation. All these experiments were evaluated with no data augmentation in 200 epochs. In the first 100 epochs, we trained only the layers after the pre-trained backbone model with a learning rate (LR) set to $1e-5$. For the rest of the epochs, we unfroze all model layers and set the LR to $1e-6$ for fine-tuning. An option of reducing the LR by a factor of 0.1 was also applied if the training loss has stopped improving at least a value of 0.0001 in two epochs.

For the first experiment, we set the input image scale (defined by the smaller image dimension) to 700 pixels and analyzed the anchors' using different feature pyramid levels on top of ResNet-50. In the Table 7.8, we omitted the results for the number of pyramid levels lower than three or higher than four due to their insignificant values or minor improvements. We tested four sets of anchor scales in the scheme $\left\{2^0, 2^{1/n_{sc}}, \ldots, 2^{(n_{sc}-1)/n_{sc}}\right\}$ and two sets of anchor aspect ratios, defined as {1:1} and {1:2, 1:1, 2;1}. Observing Table 7.8, we noticed that using four pyramid levels, four anchor scales, and 1 or 3 aspect ratios yield the best results. Thus, to choose between 1 or 3 aspect ratios, we analyzed the average value between the metrics AP and F_1 and, based on the largest value found, we selected the last option (4-4-3) for our next analyses.

Table 7.8: First experiment varying anchor parameters (number of scales n_{sc} and aspect ratios n_{ar}) and the number of feature pyramid levels, a) 3, and b) 4.

(a) Three pyramid levels.

n_{sc}	n_{ar}	AP	F_1
1	1	69.08	80.39
1	3	71.82	81.95
2	1	80.68	86.85
2	3	80.65	86.79
3	1	80.92	86.82
3	3	81.02	86.56
4	1	80.16	86.05
4	3	81.28	86.73

(b) Four pyramid levels.

n_{sc}	n_{ar}	AP	F_1
1	1	70.25	80.98
1	3	70.37	81.27
2	1	80.05	86.76
2	3	79.18	85.92
3	1	81.14	87.05
3	3	80.55	86.61
4	1	81.48	87.08
4	**3**	**81.52**	**87.06**

Following the previous experiments, we also investigated the performance of RetinaNet in our dataset by varying the base model and the input image scale. For this experiment, we tested 50, 101, and 152 ResNet depths with previous set FPN constructed on top. The image scales

for training and testing were varied between 400 and 1000 pixels in steps of 200. In addition to the network depths and input image scale analyses, we computed the inference time[1] (in milliseconds) for each model and analyzed the influence of using mask images (see Section 4.6) in the test set.

As stated in Table 7.9, the higher the input image scale, the better its results. This property was expected since our targets are small and respond better for high-resolution images. Nonetheless, the improvement caused by depth changes was insignificant when comparing models with the same image scales. In this case, one important characteristic to emphasize is the inference time, which increases according to the depth size. This problem, however, is not observed when using the mask images. In this case, the values of time had a minimum or no increase. Following the same idea of the previous analysis, we selected those parameters with the highest average value between the metrics AP and F_1 (ResNet101-1000).

Table 7.9: Second experiment varying ResNet depths and input image scales. Inference time (in milliseconds) was also evaluated for each model with and without the use of image masks in the test dataset.

		no mask			mask		
depth	**scale**	**AP**	**F_1**	**time**	**AP**	**F_1**	**time**
50	400	72.91	80.59	82	74.75	82.02	89
50	600	78.15	84.24	116	79.45	85.44	116
50	800	80.62	85.48	164	81.03	86.14	164
50	1000	83.99	87.33	219	83.93	87.75	219
101	400	73.85	81.36	89	75.83	83.00	96
101	600	78.62	84.82	130	79.90	85.97	137
101	800	80.51	85.71	185	81.05	86.38	192
101	**1000**	**84.37**	**87.94**	**247**	**84.45**	**88.42**	**253**
152	400	73.53	81.63	103	75.61	83.15	103
152	600	77.42	83.82	151	78.77	85.19	151
152	800	81.70	85.90	219	82.16	86.84	219
152	1000	84.62	87.91	288	84.52	88.27	288

7.2.3.2 Influence of data augmentation

To analyze the outcomes of our model, we used the best set of hyperparameters found in the experiments of the previous subsection. We started by investigating the influence of data augmentation in the training step by plotting the loss curves of the model with and without data augmentation. As stated before, the number of epochs for training without data augmentation was set to 100 when the backbone layers are frozen and more 100 epochs for the fine-tuning

[1] Runtimes are measured on 2 × NVIDIA GeForce GTX 1080 Ti GPUs with 11GB each.

process with all layers unfrozen. The same strategy was used for training the model with data augmentation but using 350 and 100 epochs, respectively.

In Figure 7.13, we can observe that after a few epochs, the model being trained without data augmentation (blue and orange curves) started to overfit. On the other hand, when our augmentation methods are being applied (green and red curves), we can see a consistent learning pattern.

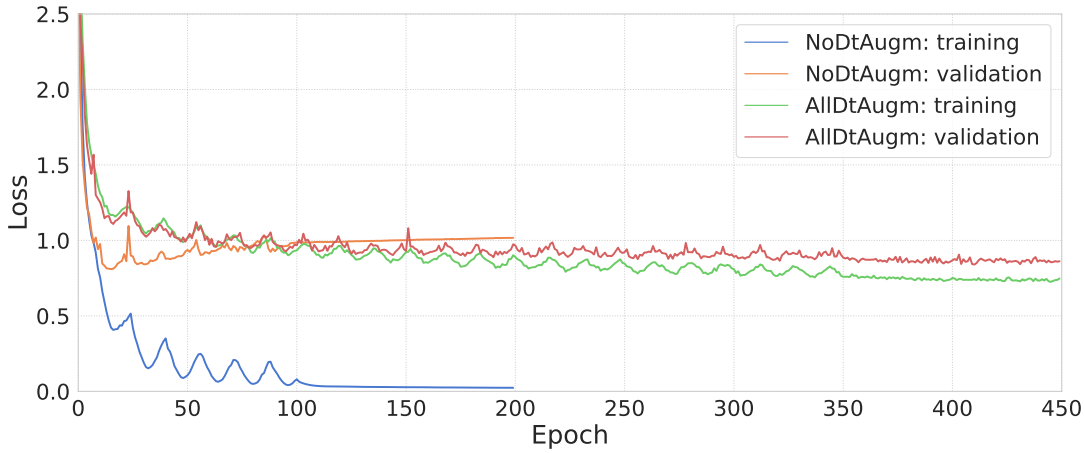
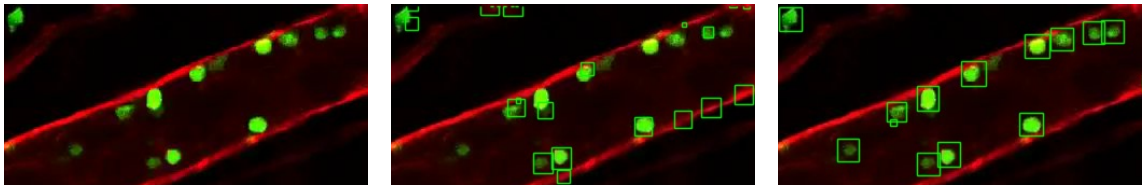


Figure 7.13: Loss curves for the model training with and without data augmentation.

Besides preventing overfitting, the use of data augmentation makes the model invariant to the transformations applied. In Figure 7.14, we verify this premise by providing an image from outside our dataset to the model. This image has cells with different visual aspects that were not seen before by the model. Nevertheless, our model trained with data augmentation was capable to accurately identify many cells in the image, as illustrated in Figure 7.14(c), while mostly failed when augmentation techniques were not applied, see Figure 7.14(b).



(a) Original image. (b) No data augmentation. (c) All data augmentation.

Figure 7.14: Analysis of data augmentation influence in an outside image. (a) Original frame image, (b) model inference without data augmentation, and (c) model inference with data augmentation.

With these qualitative results, we showed that our suite of augmentation techniques not only helps to prevent overfitting but also makes the model invariant to cell variabilities by correctly detecting objects visually different from those used in training. The next subsections show that this kind of cell variability invariance is also valid for the images in our dataset.

Table 7.10: Summary of the statistical measures for all the experiments with and without data augmentation.

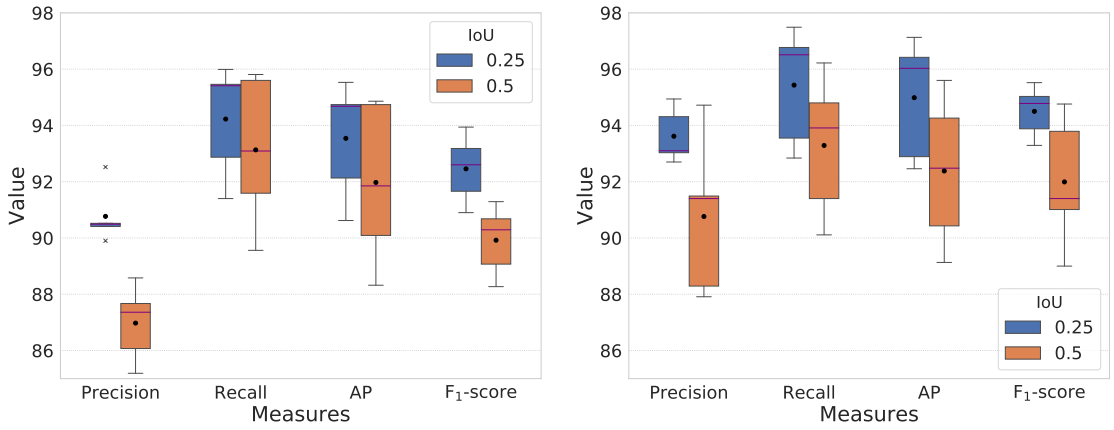
Exp.	Data augm.	IoU	Precision	Recall	F_1-score	AP
1	No	0.25	90.77 ±1.01	94.23 ±1.99	92.46 ±1.20	93.54 ±2.07
		0.50	86.97 ±1.34	93.13 ±2.66	89.92 ±1.23	91.97 ±2.87
	Yes	0.25	93.62 ±0.96	95.43 ±2.09	94.50 ±0.90	94.99 ±2.15
		0.50	90.76 ±2.78	93.29 ±2.50	91.99 ±2.30	92.38 ±2.66
2	No	0.25	72.84 ±5.62	79.16 ±5.26	75.63 ±2.88	73.97 ±4.95
		0.50	64.65 ±4.42	71.64 ±5.02	67.78 ±2.69	62.59 ±5.06
	Yes	0.25	84.14 ±4.25	82.04 ±4.59	82.93 ±2.01	79.79 ±4.58
		0.50	72.17 ±6.36	70.15 ±6.33	71.00 ±5.33	60.99 ±9.02
3	No	0.25	89.57 ±1.07	94.20 ±1.53	91.82 ±0.61	93.34 ±1.57
		0.50	87.21 ±1.98	92.60 ±1.35	89.81 ±1.15	91.32 ±1.46
	Yes	0.25	92.87 ±0.89	95.36 ±1.11	94.09 ±0.73	94.84 ±1.15
		0.50	91.66 ±2.23	93.06 ±2.17	92.36 ±2.14	92.16 ±2.39
4	No	0.25	73.14 ±5.51	79.58 ±6.35	75.91 ±2.48	74.09 ±5.58
		0.50	63.84 ±4.25	72.26 ±6.70	67.48 ±1.83	62.07 ±6.29
	Yes	0.25	83.35 ±5.63	84.02 ±6.67	83.39 ±3.03	81.62 ±6.19
		0.50	72.46 ±2.43	71.17 ±8.23	71.59 ±4.00	62.20 ±10.1

7.2.3.3 Experiment 1: CNS stratified

In our first experiment for this detection approach, we decided to use only images from the mice CNS since they have the most similar cell appearance between then, even in different videos and acquisition protocols. As can be seen in the boxplots of Figure 7.15 and Table 7.10, all statistical measures of the first experiment exhibited excellent values. For instance, the mean and median values of all metrics with data augmentation stayed above 90, and the interquartile ranges were all higher than 88, indicating the outstanding performance of the model when using the CNS images.

Despite overfitting data, the model with no data augmentation showed slightly lower values for the same metrics, which indicates that augmentation techniques may be important to cover objects' diversity.

As expected, the results for $IoU_{0.25}$ were better than those for $IoU_{0.5}$ since our targets are small cells, and minor bounding boxes disagreements created low values for the IoU metric. Indeed, the values for all the metrics had an average improvement of 2.39% when analyzing $IoU_{0.25}$ against $IoU_{0.5}$ for both models.



(a) No data augmentation.

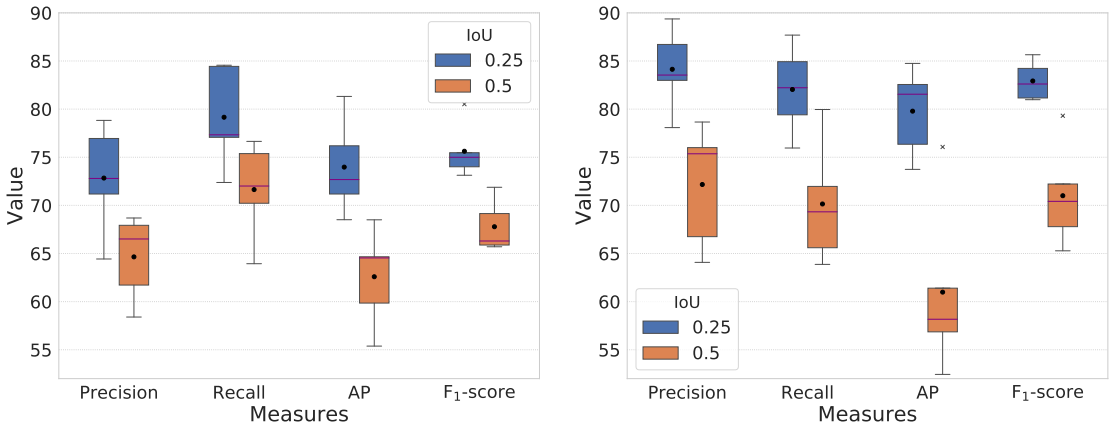
(b) All data augmentation.

Figure 7.15: Statistical measures extracted from experiment 1 with the proposed model with and without data augmentation.

7.2.3.4 Experiment 2: CNS unseen split

For the second experiment, we investigated the influence of images from the same video for training and testing in a small dataset as ours. As the frames of a single video are quite similar to each other, they can bias the model results when used in the training and test datasets, especially when the total number of images is small. To analyze the impact this kind of bias can generate in our model, we separate the three videos (B1, B2, and SC) from the CNS subset, together with 80% of images from group OT, for the *train_val* dataset. As a consequence, only 20% of the images from group OT were put in the test dataset, implying that none of the images used for training and validation belong to the same video used for testing as in the previous experiment.

When analyzing the new outcomes, we observed the same pattern between the IoU thresholds of the first experiment. However, in this case, the disparity of values was higher, presenting an average improvement of 11.19% in the metrics. This difference is mainly caused by the accuracy of bounding box regression and by the lack of images in the test dataset, which could not represent our analysis so well and also cause the slight increase in the recall and AP values when comparing the model with and without data augmentation. Indeed, when observing the boxplot of Figure 7.16(b), we can see an outlier in the metrics AP and F_1-score, meaning that a particular fold in our test dataset is increasing our metrics, or there are images in the other folds that could be decreasing them. When comparing the results for both IoU thresholds in these cases, we can presume that some of the test images present a high number of small cells, which were not present in the outlier fold. A better discussion about these images can be found in the next experiment.



(a) No data augmentation.

(b) All data augmentation.

Figure 7.16: Statistical measures extracted from experiment 2 with the proposed model with and without data augmentation.

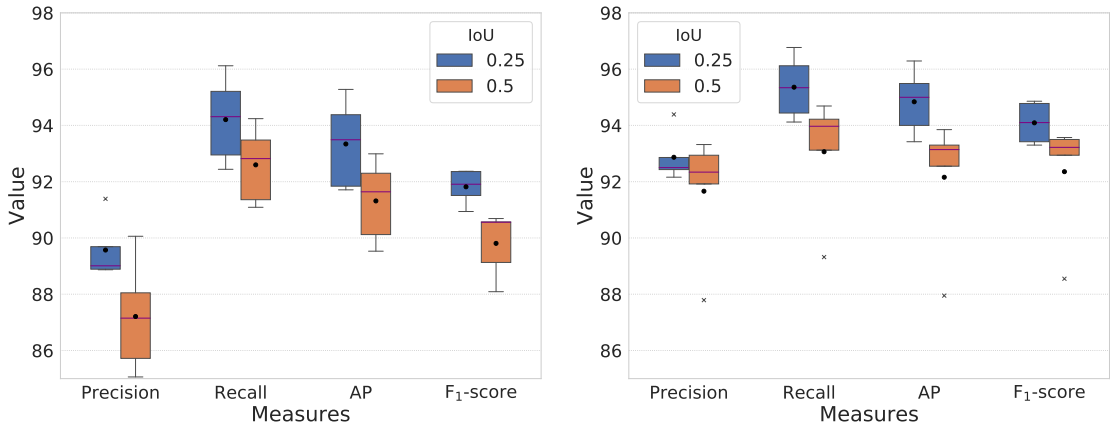
Despite all this, the average results for $IoU_{0.25}$ still kept high metric values (above 79 and 72, with and without data augmentation, respectively), even not using images from the same video in both *train_val* and *test* datasets, showing us an exemplary performance of the model.

7.2.3.5 Experiment 3: All stratified

In our third experiment, we used all available data to check our model behavior when images from different organs and protocols are used. For this challenging setup, we applied the same cross-validation process as the first experiment with five stratified folds, but now using all types of images. Again, the values for the metrics were higher for $IoU_{0.25}$, with an improvement of 1.99% on average. Also, the small standard deviations in Table 7.10 and Figure 7.17 indicated a high level of confidence in the presented values.

Although mean and median values were all above 91 for both IoU thresholds in the model with data augmentation, we can see in Figure 7.17(b) that an outlier fold is present in each metric of the $IoU_{0.50}$ analysis. After careful investigation of the images inside this fold, we noticed that a particular image is decreasing considerably our metrics and also causing problems in other experiments, as stated in the previous subsection. Indeed, this image represents a significant challenge for the model. Its cluttered environment, together with its large amount of cells, complicates even visual analysis by experts, as can be seen in Figure 7.18(a).

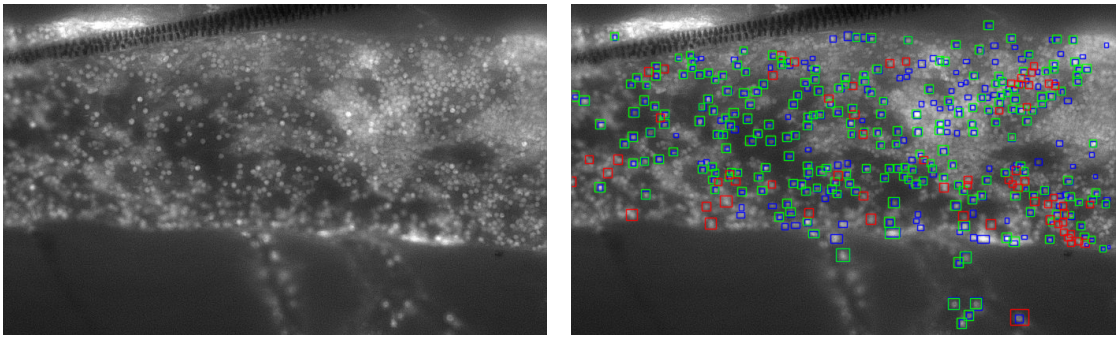
The blue squares in Figure 7.18(b) represent the manual annotations, while red and green squares are the FPs and TPs of the model with data augmentation, respectively. At first sight, we can see several cells without a manual annotation or identified as FPs when they are surely authentic cells. The reason for that is related to the lack of target resolution, which makes visual



(a) No data augmentation.

(b) All data augmentation.

Figure 7.17: Statistical measures extracted from experiment 3 with the proposed model with and without data augmentation.



(a) Original image.

(b) Output image.

Figure 7.18: Example of a challenge image for both manual annotation and algorithm detection. (a) Original image. (b) Output image from the outlier fold of experiment 3 with data augmentation.

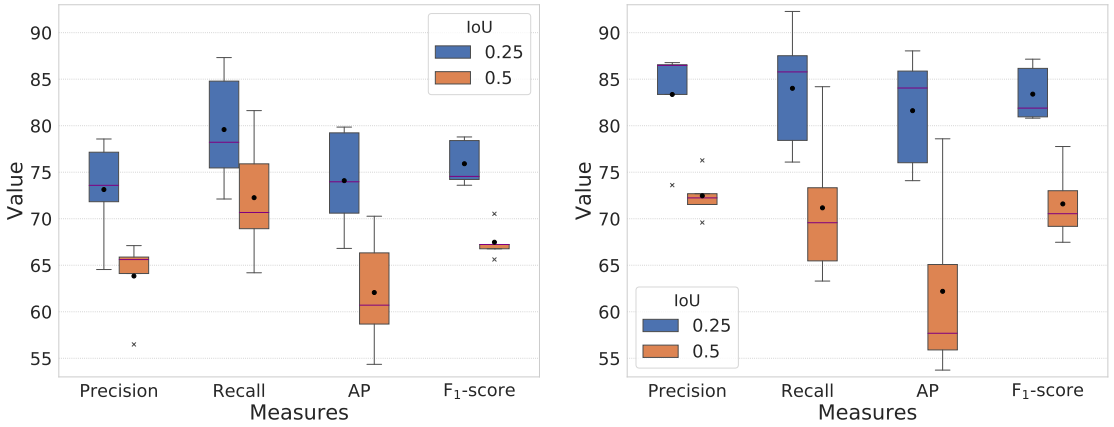
analysis quite tricky and sometimes very subjective. Thus, the combination of poor annotation and small objects eventually affects our statistical measures.

Nevertheless, even with all these difficulties, our model responded satisfactorily for images from different organs and acquisition protocols, which are known to be barriers to conventional techniques in machine learning.

7.2.3.6 Experiment 4: All unseen split

In order to make the same analysis of experiment 2 (subsection 7.2.3.4), but now using all available images, we divided our datasets in a way that images from the same video are not simultaneously allocated to *train_val* and *test* sets. To perform that, we added the images from groups C1, C2, and ME to the *train_val* dataset of each fold in the cross-validation procedure. As expected, the results were considerably low compared to the previous experiment because

of distinct image features appended to *train_val* datasets. Nevertheless, the values obtained for the $IoU_{0.25}$ threshold were all outstanding, with their average values above the baseline of 81 and 73 for the models with and without data augmentation, respectively.



(a) No data augmentation.

(b) All data augmentation.

Figure 7.19: Statistical measures extracted from experiment 4 with the proposed model with and without data augmentation.

7.2.4 Detection overall analysis

In order to perform an overall analysis of our detection approaches and also compare them with other conventional methods for cell detection, we selected the best method configurations presented in the previous sections. Accordingly, for the first approaches based on the MTM technique (MTM-PCA and MTM-DCNN), we used the experiments highlighted in the Tables 7.6 and 7.7, respectively, while for our last detection approach (DCNN), we selected the results of our models with data augmentation from experiment 3, where the images from all organs were used. Furthermore, we extracted the locations of both predicted and manually annotated centroids for each microscopy video.

It is worth noting that the results presented in Subsections 7.2.1 and 7.2.2 were evaluated using a value for k (maximum radius to consider a TP point) different from the DCNN evaluation. Thus, we changed our value from $k = \text{max(template\_size)}/2$ to $k = \text{avg\_cell\_size}/2$ for a fair comparison in this overall analysis. Therefore, some of the results may differ from those presented in previous sections.

One standard method analyzed in our comparison identifies blob-like structures in images based on the analysis of eigenvectors obtained from local Hessian matrices (GREGÓRIO DA SILVA; CARVALHO-TAVARES; FERRARI, 2015). Despite producing consistent results for objects with good contrast, this technique is limited to the object shape. As stated before, an-

other commonly used technique in object detection is the template matching (TM) (BRUNELLI, 2009). It is a well-known and straightforward technique, but its main drawback is the need to pre-select a template image as in our MTM-based approaches. Open-source tools such as Icy[2] and Ilastik (BERG et al., 2019) (using the plugin: *Cell Density Counting*) were also compared with our methods. Ilastik, however, is limited to cell density estimation without presenting their locations in the images.

7.2.4.1 Detection in the central nervous system

Despite the simple cell appearance in the CNS images, a cluttered background with small cells overlapping with each other can make the detection difficult by conventional techniques. The Hessian method, for instance, exhibited reasonable results for this group of images since the cells have a well-defined shape in most video frames. The same can be said about TM-based techniques that achieved values higher than 0.7 for the F_1-score metric. Although its good results, the MTM-DCNN method did not respond so well as other methods for videos B1 and SC. The reason behind this outcome could be related to the transfer learning process of the models tested. As the models were pre-trained in the ImageNet dataset, whose targets are natural objects considerably bigger than our cells, the convolutional layers may give preference for bigger objects or zoomed cells as in video B2.

None of these methods, for instance, outperformed the detection results of the DCNN approach, as can be seen in Table 7.11. The RetinaNet model in our DCNN approach demonstrated excellent accuracy with extremely high values of F_1-score, and low mean and standard deviation values for both counting and distance errors. Only the Hessian and MTM-DCNN techniques applied to the SC video achieved a slightly better result for the centroid distance and counting errors, respectively.

When analyzing the cell counting task for the best approach (DCNN), we observe a high agreement between the number of cells manually annotated and predicted by our model, even for videos with a vast number of cells like SC. These results are shown in the scatter plots of Figures 7.20(a, b, and c) for all the three videos from CNS. Each point in the scatter plots represents an image in the corresponding video. We also illustrated the results of our best model in an example image from each video in Figures 7.20(d, e, and f).

[2] `http://icy.bioimageanalysis.org/`

Table 7.11: Comparative cell detection results for the videos from the CNS group. The values μ_c, and σ_c represent the mean and standard deviation of the counting error; and μ_d, and σ_d represent the mean and standard deviation of centroid distance errors, respectively.

Video	Method	TP	FP	FN	P	R	F_1	$\mu_c \pm \sigma_c$	$\mu_d \pm \sigma_d$
	Ilastik	-	-	-	-	-	-	2.99±1.95	-
	Icy	3281	2769	2546	0.54	0.56	0.55	2.32±2.00	0.49±0.25
	TM	4109	1299	1718	0.76	0.71	0.73	2.78±2.17	0.45±0.24
B1	Hessian	4709	1189	1118	0.80	0.81	0.80	2.26±1.88	0.41±0.24
	MTM-PCA	4514	801	1337	0.85	0.77	0.81	2.87±1.90	0.41±0.23
	MTM-DCNN	4188	1033	1639	0.80	0.72	0.76	2.98±2.11	0.47±0.24
	DCNN	**5316**	**561**	**511**	**0.90**	**0.91**	**0.91**	**0.97±0.98**	**0.41±0.23**
	Ilastik	-	-	-	-	-	-	3.17±2.06	-
	Icy	5505	2034	2534	0.73	0.68	0.71	2.67±1.85	0.34±0.24
	TM	5599	2359	2449	0.70	0.70	0.70	2.15±1.62	0.23±0.19
B2	Hessian	6711	2325	1337	0.74	0.83	0.79	2.77±2.14	0.28±0.23
	MTM-PCA	5599	2359	2449	0.70	0.70	0.70	2.15±1.62	0.23±0.19
	MTM-DCNN	5959	884	2089	0.87	0.74	0.80	3.14±1.96	0.23±0.18
	DCNN	**7992**	**306**	**56**	**0.96**	**0.99**	**0.98**	**0.75±0.87**	**0.15±0.18**
	Ilastik	-	-	-	-	-	-	26.19±4.34	-
	Icy	1321	457	249	0.74	0.84	0.79	9.90±3.99	0.48±0.27
	TM	1117	321	453	0.78	0.71	0.74	6.86±3.98	0.53±0.26
SC	Hessian	1262	670	308	0.65	0.80	0.72	17.24±9.04	**0.45±0.27**
	MTM-PCA	1276	345	294	0.79	0.81	0.80	5.76±6.30	0.51±0.26
	MTM-DCNN	1115	493	455	0.69	0.71	0.70	**4.76±5.48**	0.61±0.25
	DCNN	**1468**	**222**	**102**	**0.87**	**0.94**	**0.90**	5.71±3.38	0.46±0.27

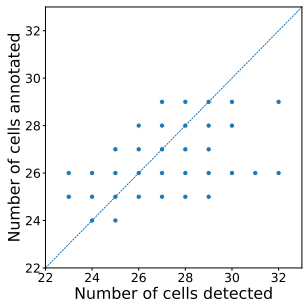
7.2.4.2 Detection in the cremaster muscle

Although coming from the same animal organ, the two videos from cremaster muscle present distinct challenges. While C1 video shows the cells on a larger scale, C2 exhibits a considerable amount of cells on a lower scale flowing in a vessel. All these features must be considered in addition to cell overlapping and shadow artifacts in the images.

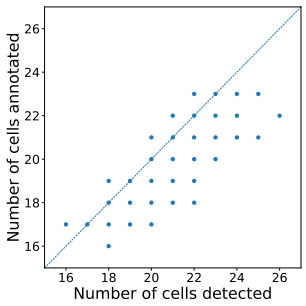
However, even with these challenging characteristics and the small number of video frames to train our model, the DCNN approach achieved the best results when compared with the other methods, as can be visualized in Table 7.12. It is worth noting that our training process in the DCNN approach did not present overfitting since we applied extensive data augmentation (the plots can be seen online[3]).

Another critical information to consider in Table 7.12 is the high number of counting errors in video C2 by the other methods. It means that identify the correct number of cells in these

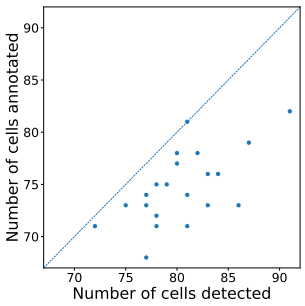
[3] `https://tensorboard.dev/experiment/mbXrauJjQtyVmgr9FznuMA/`



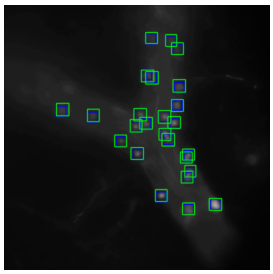
(a) Cell counting: B1.



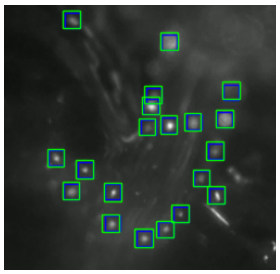
(b) Cell counting: B2.



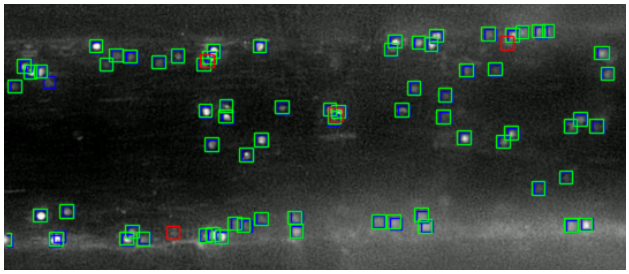
(c) Cell counting: SC.



(d) Output: B1.



(e) Output: B2.



(f) Output: SC.

Figure 7.20: In Figures (a), (b), and (c), we can see a comparison between the number of cells manually annotated and detected by the DCNN approach for videos B1, B2, and SC, respectively. Each point in the scatter plots represents a different image frame. In Figures (d), (e), and (f), we show an example of output frame from each one of the videos B1, B2, and SC, respectively. Blue squares in the images represent manually annotated leukocytes, while green and red squares represent the TPs and FPs, respectively. Note that image frames were cropped for better visualization.

particular frames can be a tough task. Indeed, this is the case where our DCNN approach mostly fails because of the large number of small cells, as can be seen in the scatter plot of Figure 7.21(b) and in the example image of Figure 7.21(d). For the video C1, however, we had a proper correlation in the number of cells, with only a small fraction of FPs, as illustrated by the red square in the image of Figure 7.21(c).

7.2.4.3 Detection in the mesentery

Cell detection in our mice mesentery images has also proved to be a challenging task given the cell's appearance and their deformed shapes. The Hessian, TM, and MTM-PCA methods, for instance, mostly fail to detect these cells because of their shape-changing. The MTM-DCNN method, however, presented a significant result, probably because cells in the ME video have a large size on average, which could help the feature image extraction by the convolutional layers of the pre-trained models. But, it is not surprising that our supervised approach outperformed the unsupervised methods in this dataset, as indicated in Table 7.13.

In this video analysis, our DCNN approach exhibited an outstanding performance, resulting in only 4 FNs, 13 FPs, and, consequently, an F_1-score value much higher than the other methods.

Table 7.12: Comparative cell detection results for the videos from cremaster muscle group. The values μ_c, and σ_c represent the mean and standard deviation of the counting error; and μ_d, and σ_d represent the mean and standard deviation of centroid distance errors, respectively.

Video	Method	TP	FP	FN	P	R	F_1	$\mu_c \pm \sigma_c$	$\mu_d \pm \sigma_d$
C1	Ilastik	-	-	-	-	-	-	38.48±1.82	-
	Icy	208	173	182	0.55	0.53	0.54	**2.05±1.84**	0.27±0.15
	TM	250	207	140	0.55	0.64	0.59	4.71±4.59	0.32±0.18
	Hessian	229	148	161	0.61	0.59	0.60	3.86±4.18	0.26±0.16
	MTM-PCA	298	129	113	0.70	0.73	0.71	3.43±4.08	0.30±0.17
	MTM-DCNN	232	**65**	158	0.78	0.59	0.68	5.19±2.22	0.30±0.15
	DCNN	**381**	70	**9**	**0.84**	**0.98**	**0.91**	3.00±1.98	**0.26±0.15**
C2	Ilastik	-	-	-	-	-	-	75.86±6.14	-
	Icy	938	873	665	0.52	0.59	0.55	10.29±6.04	0.47±0.27
	TM	830	730	773	0.53	0.52	0.52	11.76±10.37	0.49±0.27
	Hessian	1283	1294	320	0.50	0.80	0.61	46.38±17.19	0.48±0.27
	MTM-PCA	930	374	1252	0.71	0.43	0.53	41.81±10.00	0.48±0.27
	MTM-DCNN	964	552	639	0.64	0.60	0.62	**4.81±3.82**	0.52±0.26
	DCNN	**1420**	**293**	**183**	**0.83**	**0.89**	**0.86**	5.81±5.47	**0.46±0.28**

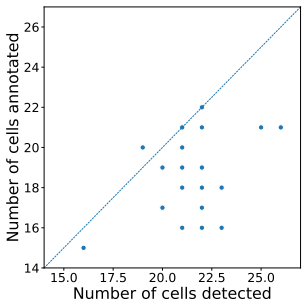
Table 7.13: Comparative cell detection results for the mice mesentery video. The values μ_c, and σ_c represent the mean and standard deviation of the counting error; and μ_d, and σ_d represent the mean and standard deviation of centroid distance errors, respectively.

Video	Method	TP	FP	FN	P	R	F_1	$\mu_c \pm \sigma_c$	$\mu_d \pm \sigma_d$
ME	Ilastik	-	-	-	-	-	-	70.43±1.37	-
	Icy	95	689	196	0.12	0.33	0.18	23.48±2.13	0.45±0.18
	TM	110	74	181	0.60	0.38	0.46	5.10±2.04	**0.38±0.19**
	Hessian	199	116	92	0.63	0.68	0.66	2.67±1.64	0.43±0.21
	MTM-PCA	172	41	119	0.81	0.59	0.68	3.71±2.29	0.38±0.21
	MTM-DCNN	219	29	72	0.88	0.75	0.81	2.24±1.48	0.45±0.21
	DCNN	**287**	**13**	**4**	**0.96**	**0.99**	**0.97**	**0.52±0.66**	0.46±0.28

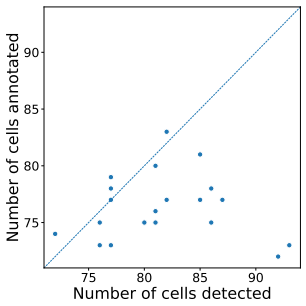
The scatter plot and an example image from the application of DCNN approach are also shown for the ME video in Figures 7.22(a) and 7.22(b), respectively.

7.3 Tracking evaluation

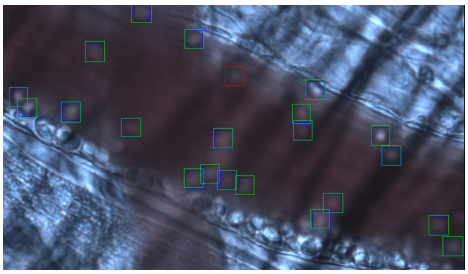
To test and evaluate the performance of our 2D+t approach, we used the best results from the detection stage as input, similarly to the overall detection analysis (see Subsection 7.2.4). Also, our tracking approach was evaluated using two tracklets separation strategies; one using the cumulative matrix (CM), and the other deleting only the junction points (JP) and their neighbors. The next subsections show the maximum F_1-score values obtained for the tracking approach in



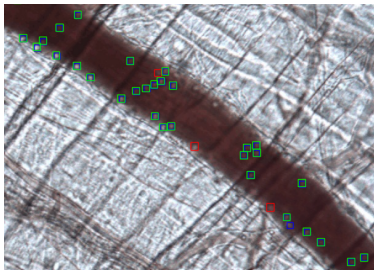
(a) Cell counting: C1.



(b) Cell counting: C2.



(c) Output: C1.



(d) Output: C2.

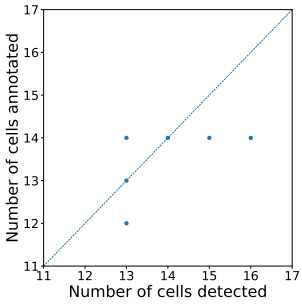
Figure 7.21: In Figures (a) and (b), we can see a comparison between the number of cells manually annotated and detected by the DCNN approach for videos C1 and C2, respectively. Each point in the scatter plots represents a different image frame. In Figures (c) and (d), we show an example of output frame from each one of the videos C1 and C2, respectively. Blue squares in the images represent manually annotated leukocytes, while green and red squares represent the TPs and FPs, respectively. Note that image frames were cropped for better visualization.

each data group.

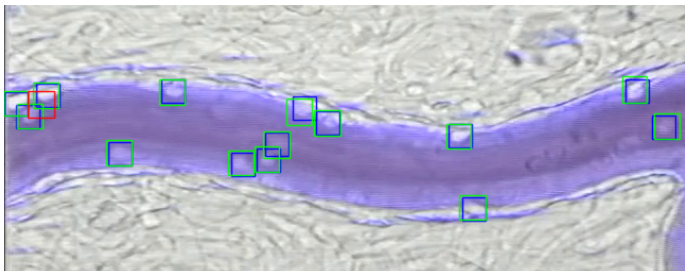
7.3.1 Tracking in the central nervous system

In Table 7.14 we can observe the results obtained for the best configuration of each detection approach in the CNS group. The separation strategies highlighted in boldface indicate the best technique chosen according to the counting measures. Although the CM strategy was slightly better for the CNS group, the results exhibited minimum or zero improvement when compared with the JP strategy. Only in some videos, the change was noticeable. Video B1 for the MTM-PCA approach, for instance, had good improvement (from 0.70 to 0.77 for the F_1-score value) with the use of CM. After a meticulous investigation on this particular case, we noticed the CM strategy helped the tracking method to not shift when cells were very close to each other or when the remaining motion artifacts were still present in the images.

Besides all that, our tracking framework showed excellent performance, mainly with the DCNN approach. It achieved F_1-score values of 0.80, 0.96, and 0.68 for the videos B1, B2, and SC, respectively. The low value for SC video, however, showed us the difficulty in tracking small cells moving fast during a short period (21 video frames), as can be seen in Figure 7.23(c).

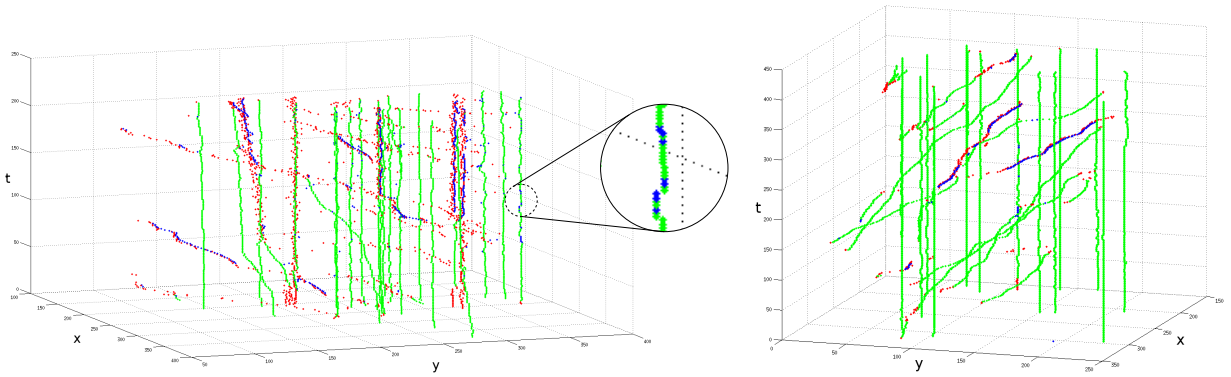


(a) Cell counting: ME.



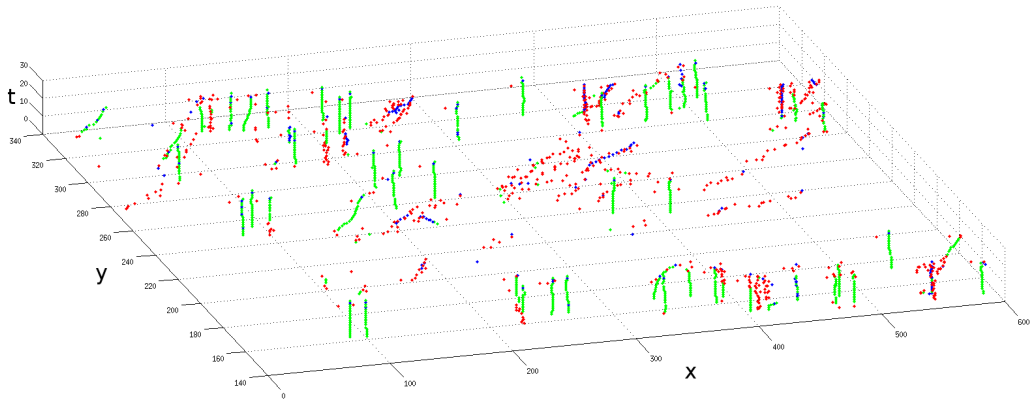
(b) Output: ME.

Figure 7.22: In the scatter plot in (a), we can see a comparison between the number of cells manually annotated and detected by the DCNN approach for the video ME. Each point in the plot represents a different image frame. In (b), we show an example of output frame from the video ME. Blue squares in the image represent manually annotated leukocytes, while green and red squares represent the TPs and FPs, respectively. Note that image frames were cropped for better visualization.



(a) B1

(b) B2



(c) SC

Figure 7.23: Comparison between the outputs of 2D+t processing stage and manual annotations for videos (a) B1, (b) B2, and (c) SC. Green points indicate TP, blue points indicate FP, and red points the FN.

Leukocytes trajectories were plotted in Figure 7.23 for a visual comparison between the outputs of the DCNN detection approach and the manual annotations. In these plots, the vertical

Table 7.14: Best F_1-score values obtained for our tracking algorithm using the JP and CM tracklets separation strategies. The input images were obtained from our best detection results for the CNS group.

Video	Detection approach	Separation strategy	Counting					
			TP	FP	FN	P	R	F_1
B1	MTM-PCA	JP	3704	1047	2123	0.78	0.64	0.70
		CM	4052	663	1775	0.86	0.70	0.77
	MTM-DCNN	JP	4783	2094	3265	0.70	0.59	0.64
		CM	3301	1017	2526	0.76	0.57	0.65
	DCNN	**JP**	4315	640	1512	0.87	0.74	**0.80**
		CM	4297	689	1530	0.86	0.74	0.79
B2	MTM-PCA	JP	4783	2094	3265	0.70	0.59	0.64
		CM	4783	2094	3265	0.70	0.59	0.64
	MTM-DCNN	JP	5052	1328	2996	0.79	0.63	0.70
		CM	5052	1328	2996	0.79	0.63	0.70
	DCNN	JP	7577	237	471	0.97	0.94	0.96
		CM	7579	236	469	0.97	0.94	**0.96**
SC	MTM-PCA	JP	897	343	673	0.72	0.57	0.64
		CM	895	338	675	0.73	0.57	0.64
	MTM-DCNN	JP	785	359	785	0.69	0.50	0.58
		CM	785	358	785	0.69	0.50	0.58
	DCNN	**JP**	926	228	644	0.80	0.59	**0.68**
		CM	914	230	656	0.80	0.58	0.67

axis represents the time (t) while x- and y-axes correspond to the leukocyte spatial positions in video frames. Leukocyte centroids correctly detected (TP) by the DCNN approach form the green lines, while false alarms (FP) and missing detections (FN) form the blue and red lines, respectively, in the images of Figure 7.23.

Visual assessment of the images shows some small blue segments (FP) within the green paths (TP), as illustrated in Figure 7.23(a). The points in these segments provide continuity to the trajectories, despite not being manually annotated as leukocyte centroids within the search region in the video frames. When checking the frames in which these points should be located, we noticed that some residual movements were still present, and this was probably the reason why the experts did not annotate them or made wrong annotations. We isolated a particular case of a sequence of frames where the number of FP points was relatively high to prove that it corresponds to a video frame where the manual tracking of some cells is lost or considerably displaced due to animal movement. It can be seen in Figure 7.24.

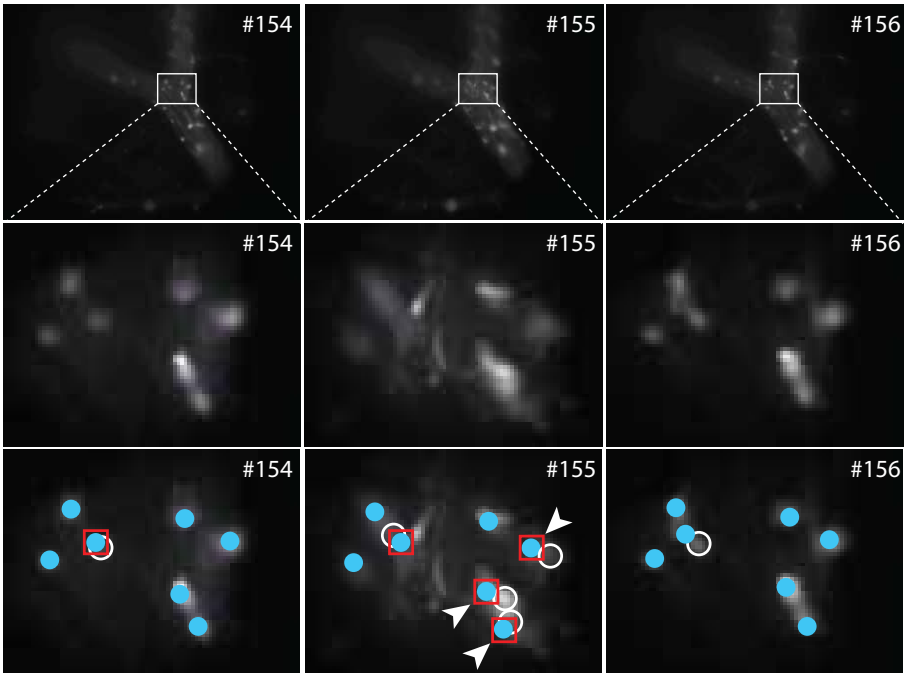


Figure 7.24: Example of a sequence of three consecutive frames from video B1 on the first line, in which a particular region was zoomed and displayed in the second and third lines. Blue circles in the images represent the output cell positions of our algorithm, while red squares show the FP points and white circumferences the FN points. Cells pointed by a white arrow corresponds to the connection segments.

The first line of Figure 7.24 shows three consecutive frames extracted from video B1. Frame number 155 in the first line illustrates an example of an image with motion residues. By comparing these frames with the corresponding outputs of our approach in the third line of Figure 7.24, we noticed that the number of FP points, represented as red squares, increased from 1 to 4 when the residual motion occurred in a particular zoomed region and then decreased to zero after the movement ceased. The same trend is observed for the numbers of FN points, or white circumferences, which increased from 1 to 4 and then decreased to 1 again. In this case, however, the FN changed because the annotations (done on the frame-by-frame basis) presented significant displacements caused by the remaining motion in frame number 155. This annotation issue directly affected 3 of 4 FP points and also created three additional FN points in a small image region. Despite the problems mentioned above, our algorithm was capable of rightly connecting some trajectory segments in those frame regions. These segment points are indicated by a white arrow in the images of the third line. It is also worth noting that all of them belong to the case where the motion artifacts disturbed the results.

Thereby, we argue that the trajectory connections by the proposed technique allow predicting cell movements even when the expert was not capable of annotating the proper location of leukocyte centroids in some individual frames or when the residual motion artifacts are still

present in the video frames. In this case, a large number of FP points could be considered as TP, while the number of FN points could decrease, which would further increase the statistical measures of the proposed approach.

7.3.2 Tracking in the cremaster muscle

In the evaluation performed for videos C1 and C2, results comparing the separation strategies presented an insignificant difference, as can be observed in Table 7.15. This characteristic is justified by the absence of motion artifacts in the video frames and, consequently, less spurious elements in the skeleton images.

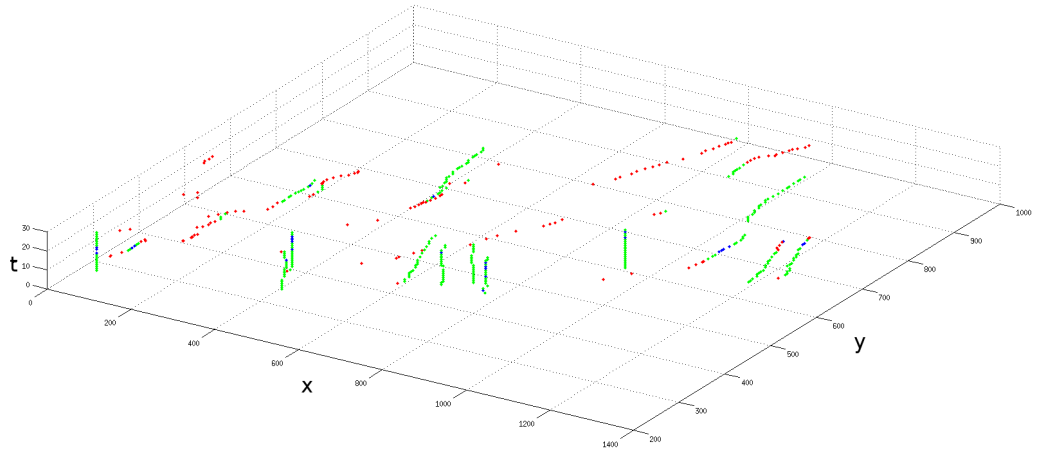
Table 7.15: Best F_1-score values obtained for our tracking algorithm using the JP and CM tracklets separation strategies. The input images were obtained from our best detection results for the cremaster group.

Video	Detection approach	Separation strategy	Counting					
			TP	FP	FN	P	R	F_1
C1	MTM-PCA	JP	155	54	235	0.74	0.40	0.52
		CM	155	54	235	0.74	0.40	0.52
	MTM-DCNN	JP	123	30	267	0.80	0.32	0.45
		CM	123	30	267	0.80	0.32	0.45
	DCNN	JP	267	28	123	0.91	0.68	0.78
		CM	267	28	123	0.91	0.68	**0.78**
C2	MTM-PCA	JP	773	206	830	0.79	0.48	0.60
		CM	773	206	830	0.79	0.48	0.60
	MTM-DCNN	JP	836	365	767	0.70	0.52	0.60
		CM	836	359	767	0.70	0.52	0.60
	DCNN	**JP**	1198	217	405	0.85	0.75	**0.79**
		CM	1190	222	413	0.84	0.74	0.79

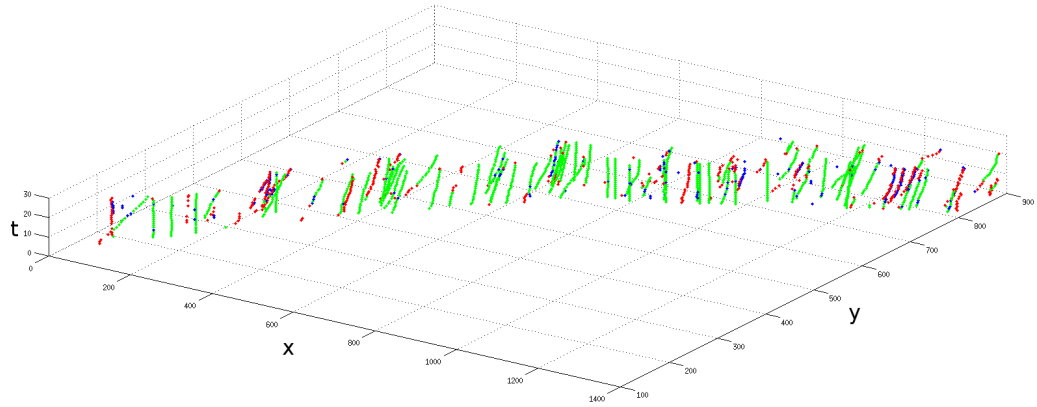
The resulting F_1-score values for both videos were all lower than 0.8, mainly because they presented a large number of FN points compared to the total number of cell annotations. In the case of video C1, we noticed the algorithm lost tracking the cells when they were moving fast, even after applying our downsampling approach to reduce this kind of problem. In the video C2, we had most cell tracks lost due to a low detection result. Indeed, it was very challenging for our 2D processing stage to detect all these small cells, achieving an average F_1-score value of only 0.67, considering our three detection approaches.

A comparative image between our tracking output and the manual annotations can be seen in Figure 7.25 for both videos from the cremaster group. Again, green points in this figure

represent the TP values, while blue and red points indicate the FP and FN, respectively.



(a) C1



(b) C2

Figure 7.25: Comparison between the outputs of 2D+t processing stage and manual annotations for videos (a) C1 and (b) C2. Green points indicate TP, blue points indicate FP, and red points the FN.

7.3.3 Tracking in the mesentery

When analyzing the performance of our 2D+t process in the mesentery video, we also did not observe a significant difference in the separation strategies, as can be seen in the values of Table 7.16. As the cremaster images, this is explained by the absence of motion artifacts.

Although the tracking results were very promising, reaching an F_1-score value of 0.87 for the DCNN approach, we noticed that three particular trackings caused the most number of FN points. By visual analysis of Figure 7.26, we can observe two mostly lost trackings in the central region of the frames, and a third one on the left side. Following a meticulous examination, we found that these cells were very close to each other in the downsampled images, which caused an undesirable effect in the Hessian algorithm to identify tubular-like structures. For these

Table 7.16: Best F_1-score values obtained for our tracking algorithm using the JP and CM techniques to separate the skeleton images.

Video	Detection approach	Separation strategy	Counting TP	FP	FN	P	R	F_1
ME	MTM-PCA	JP	151	13	140	0.92	0.52	0.66
		CM	151	13	140	0.92	0.52	0.66
	MTM-DCNN	JP	205	5	86	0.98	0.70	0.82
		CM	205	5	86	0.98	0.70	0.82
	DCNN	JP	230	9	61	0.96	0.79	0.87
		CM	230	6	61	0.97	0.79	**0.87**

particular cases, therefore, it is necessary a more careful study in the parameters setting of our 2D+t processes.

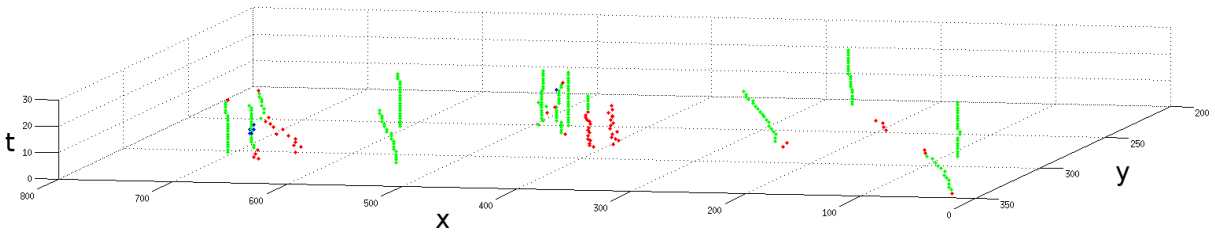


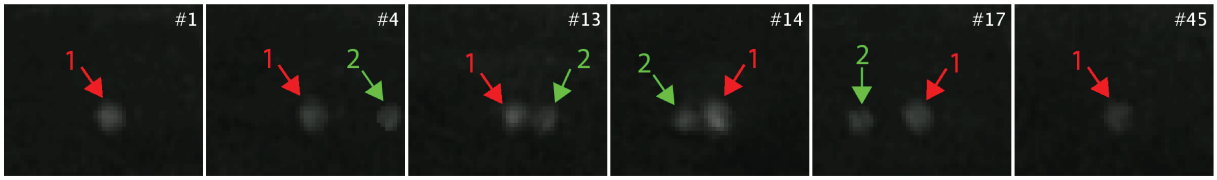
Figure 7.26: Comparison between the outputs of 2D+t processing stage and manual annotations for video ME. Green points indicate TP, blue points indicate FP, and red points the FN.

7.4 Occlusion and trajectory gap

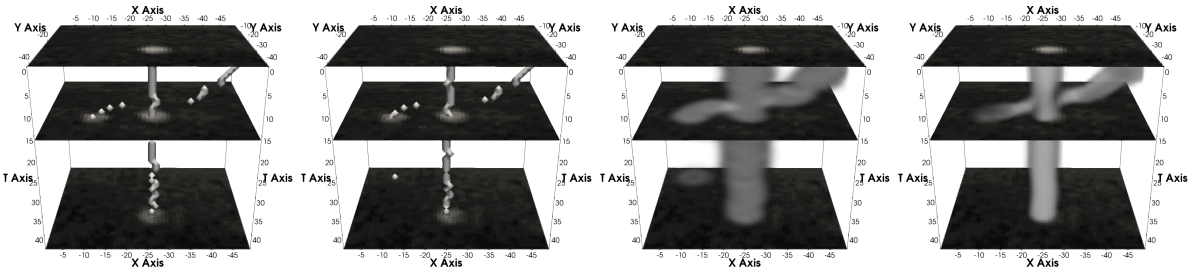
As discussed earlier, IVM may present cases of cell occlusions and trajectory gaps over the recorded videos. In our tracking approach, these cases are treated in a relatively simple manner applying the tracklet linking strategy, which uses a cone searching region to track and link cell trajectories in a spatiotemporal image. To demonstrate the ability of our proposed method to handle cell occlusions and trajectory gaps, we selected an example, as illustrated in Figure 7.27(a), extracted from a sequence of frames from a real IVM. In this frame sequence, one cell remains stationary (cell number 1 pointed by a red arrow in the image) while another (cell number 2 indicated by a green arrow) is heading in its direction from right to left. As can be seen in the frames number 13 and 14, a sudden jump in the cell number 2, possibly caused by low frame rate, resulted in a trajectory gap and, at the same time, illustrates a situation of partial occlusion between two cells. Both cases are treated equally by our proposed method since they are likely to cause a tracking loss.

Manual frame-by-frame annotations of the same cells shown in Figure 7.27(a) are presented

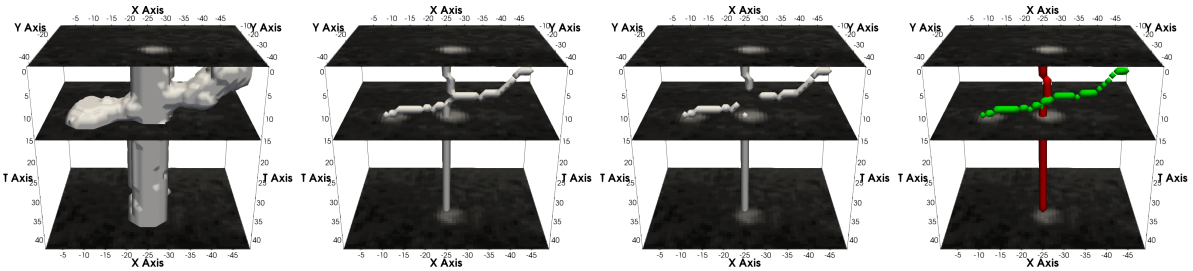
in Figure 7.27(b). Other images in Figure 7.27 show the sequence of results from all processing stages: 7.27(c) output of our detection approach; 7.27(d) output of the 2D processing stage with centroids blurred; 7.27(e) output of 3D Hessian-based processing used to enhance tubular-like structures; 7.27(f) output of the binarization step applied to the enhanced image; 7.27(g) skeletonization output; 7.27(h) tracklets separation using CM; and 7.27(i) track linking output, where each cell trajectory is represented by a different label (colored in red and green). By analyzing the output images, we can conclude that our tracking approach performed adequately and identified the proper cell trajectories, even facing cell occlusion or trajectory gap.



(a) Sequence of frames extract from a real IVM



(b) Ground truth (c) Detection output (d) 2D+t preprocessing (e) 3D Hessian output



(f) Binarization output (g) Skeletonization output (h) Separation output (i) Connection output

Figure 7.27: Example of 2D+t processing results of a real sequence of frames with a cell occlusion. (a) A sequence of frames of a particular IVM region. From left to right, frame numbers: 1, 4, 13, 14, 17, and 45. Arrows numbered as 1 (red) and 2 (green) indicate the movement of two different cells in the extracted region. (b) Cells' centroids manually annotated. (c) Output images from DCNN detection approach. (d) 2D+t preprocessing. (e) Hessian, (f) binarization, (g) skeletonization, (h) tracklets separation, and (i) track linking outputs.

7.5 Final statistical measures

With the proposed 2D+t processing or tracking approach, we were able to compute the statistical measures for the leukocyte recruitment in IVM images. Indeed, measures such as

distance, velocity, and the number of leukocytes were easily computed for each video used in this study. We summarized these measures in Table 7.17 for our best tracking performances in the last section, indicated by the boldfaced numbers in Tables 7.14, 7.15, and 7.16.

Table 7.17: Extracted measures obtained from our best tracking results in each video of our dataset.

Measure	B1	B2	SC	C1	C2	ME
Number of adhered leukocytes	57	25	65	10	40	11
Number of rolling leukocytes	13	16	18	9	63	4
Average velocity (μm/s)	5.31	5.26	10.60	43.04	14.70	14.43
Average traveled distance (μm)	6.31	22.25	3.77	32.53	10.72	8.92

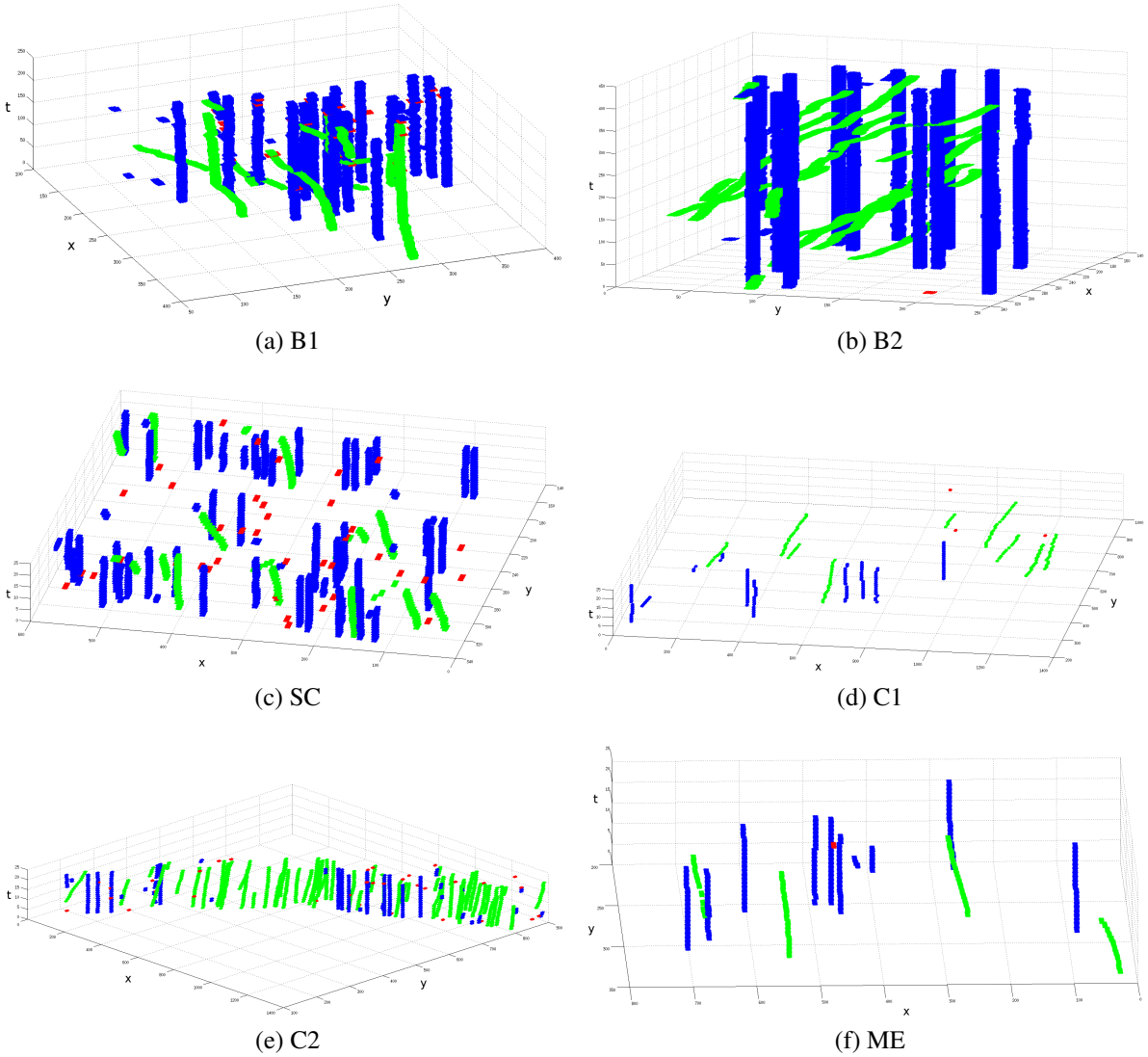
As we do not have the spatial resolution of videos C1, C2, and ME, their measures for cell velocity and traveled distance were acquired using a spatial resolution of 1 pixel/μm.

In order to provide a visual representation of the cell types on each video processed, we plotted our tracking points in Figure 7.28. Adhered cells were plotted using the blue color while the green points represent the rolling cells. The red points scattered across the images correspond to flashing cells, i.e., detections that appear for less than three frames in the videos. Although these flashing cells were considered in the calculation of countings metrics, we did not consider them in the final statistical measures since they do not contribute to an accurate cell recruitment analysis.

As illustrated in Figure 7.28(a), flashing cells mostly appear when the images still have a significant amount of motion artifacts, which is proved by the proximity of their centroids with the proper trackings.

7.6 Final considerations

In this chapter, we presented all the results of this research work. They showed that, despite the inherent problems of IVM, the automated processes for video stabilization, detection, and tracking are capable of assisting the expert in real circumstances. The video stabilization outcomes were evaluated and suggested that the methods used in frames registration can correct most of the motion artifacts arisen from the breathing and heartbeats of the animal, providing images with better quality for the following processes. Moreover, the detection approaches showed that even using different methods, we achieved excellent results. From our experiments on the leukocyte detection, we concluded that DCNN pre-trained models could, indeed, perform very well with the transfer learning and data augmentation strategies. Finally, the 2D+t processing stage proved its potential in correctly identifying and tracking cell trajectories in



(a) B1 (b) B2

(c) SC (d) C1

(e) C2 (f) ME

Figure 7.28: Tracking outputs for each IVM video according to the cell types. Blue segments indicate the adhered leukocytes, while green segments represent the rolling leukocytes. Red points scattered across the images are the flashing cells. The videos are illustrated as follows: (a) B1, (b) B2, (c) SC, (d) C1, (e) C2, and (f) ME.

spatiotemporal images, even when the cell paths were not ideal. In the next chapter, we present a complete discussion about these results, describing what can still be investigated, and some options for future works.

Chapter 8

CONCLUSIONS

This chapter synthesizes the conclusions from the literature review and the main results obtained from our automatic computational pipeline applied to real IVM images in different experiments. Besides, it also indicates further investigations that can be derived from our studies.

8.1 Overview and future investigations

The primary objective of this research was to develop an automatic computational pipeline to aid in the detection and tracking of leukocytes in IVM applied to in vivo experiments of different animal organs. The primary motivation behind this work was the growing demand for more precise measures associated with cell motility since they are essential for quantitative analysis and the understanding of biological mechanisms of many inflammatory diseases. Automatic or semi-automatic techniques, like the ones proposed herein, can provide greater flexibility and reliability to the image analysis tasks given the existing limitations in the purely visual work by the experts. Therefore, a computational system could drastically reduce the number of false statistics related to cell behavior and counting.

Besides all that, we noticed that only a few works in the literature perform tracking in IVM images despite the existence of several kinds of research related to object detection and tracking, as stated in Chapter 3. Also, none of the related studies for in vivo analysis in the literature is applied to images from different organs or even from the animal's CNS. This fact is extremely relevant since the problems found during the CNS analysis, like the animal's stabilization by mechanical devices, for instance, are very specifics and restrictive.

In this sense, we proposed the development of a framework based on image processing and computer vision techniques that combines spatial information from a frame-by-frame detection

(2D processing), and temporal information from the processing of volumetric images created by stacking all video frames (2D+t processing). The results of this research were obtained from qualitative and quantitative evaluations performed over six different IVM videos, where the location of automatically detected cells were compared with the manual annotations of an expert. Our results demonstrated that the combination of these two processing stages minimizes most of the problems involved in the detection and tracking of cells, such as cell occlusion and the proper discrimination of cell trajectories. Moreover, by using this framework as a design pattern, we can opt for the use of different techniques in each processing stage, which makes the code more generic and modularized.

Inherent difficulties from the IVM imaging technique, like the involuntary movement of the animal, were partially overcome through the use of different preprocessing methods specifically developed for this application. In extreme cases involving video frames with excessive motion blur, the images were examined and removed from the videos. However, the analysis and restoration of such images were out of the scope of this research but must be considered in a future study since they were responsible for most of our problems with cell tracking.

For the 2D processing stage, we performed several experiments using different detection approaches. We started with the development of a multiple template matching algorithm to search for cells in handcrafted image features, in which different strategies for feature extraction and templates combination were tested. After determining that some of our image features did not contribute to the detection process, we moved our effort to the automatic extraction of features by using the outputs of first convolutional layers from pre-trained DCNN models. Besides its impressive results, this transfer learning process presented some limitations, mainly when the cells are small objects compared to the whole image. Later, we developed a suite of data augmentation techniques and applied a DCNN model for cell detection using transfer learning and fine-tuning strategies.

After performing several experiments using different backbone networks, pyramid levels, model hyperparameters, and data augmentation techniques, we achieved excellent results. The use of transfer learning also allowed us to train the model using a much fewer number of epochs compared to other standard neural networks trained from scratch. On the other hand, our suite of data augmentation techniques not only helped to prevent overfitting but also made the model invariant to natural cell variabilities. Our results demonstrated the effectiveness and generality of the DCNN approach by achieving excellent values for standard metrics like AP and F_1-score, and low error rates for cell counting and centroid distances.

However, one must consider some particular properties to decide which detection method

to use, such as computational costs, parameter settings, and algorithm limitations. Our methods using the MTM technique, for instance, are more dependent on cell appearances but require few cell annotations and few parameters adjustment. On the other hand, DCNN approaches require a considerable amount of cell annotations while spending too much time on the training phase, but are more accurate and faster predicting the target positions.

As future investigations that can be derived from our cell detection studies, we can include tests with other available backbone networks and trained weights from different datasets. We can also consider the use of temporal information from the videos and the extension of our network to 3D studies. Finally, we can mention the possibility of using trained models to make new cell annotations on unseen images and then retrain them to improve our results even more.

After transforming the results from our leukocyte detection approaches into volumetric images, we started our spatiotemporal approach based on the enhancement of tubular-like structures for the tracking of leukocytes. The main advantage of this approach compared to other methods is the simplicity in dealing with object motions and multiple cells while handling occlusion and trajectory gaps in dynamic scenes. Although the results for this stage heavily depend on the previous processing and the manual annotations, we could prove the usability of our proposed tracking approach in real IVM images. Future works for this processing stage include using different techniques for tubular-like objects detection, skeletonization, and the analysis of cell trajectories individually for better movement predictions, especially when cells are moving fast.

Finally, the proposed automatic computational pipeline showed very promising results while using images from different experiments and animals' organs, especially for leukocytes detection, which achieved an average value of 0.92 for the F_1-score metric when our DCNN approach was used. Besides, our framework searches for leukocytes in the entire region of interest (microvessels) and not only in a vessel section, improving the cell counting and, consequently, the biological studies with more precise statistics.

Appendix A

HESSIAN-BASED LOCAL FEATURE DETECTOR

Initially proposed by Koller et al. (1995) and further developed and improved by Lorenz et al. (1997), Sato et al. (1998), and Frangi et al. (1998), the Hessian-based local feature detector has demonstrated a great potential in the extraction of object shape information in medical images. However, most works using this technique in the literature focus on the enhancement (or segmentation) of tubular-like structures for blood vessel extraction (either 2D or 3D) in angiography images (EBRAHIMDOOST et al., 2011; ÖKSüZ; ÜNAY; KADIPAŞAOĞLU, 2012; JIMENEZ-CARRETERO et al., 2013; DUFOUR et al., 2013).

Since microscopy images are two-dimensional projections of three-dimensional structures (microvessels and cells), leukocytes may be positioned above and below the microscope focal plane and, therefore, their size appearance may be distorted, showing cells in a different range of scales. For this reason, this proposed method was developed based on the analysis of local structures in a multiscale framework (DZYUBAK; RITMAN, 2011). The initial idea of the approach is to generate a family of smoothed images $I(\boldsymbol{x};\sigma)$, computed by convolving the original image $I_0(\boldsymbol{x})$ with Gaussian kernels $G(\boldsymbol{x};\sigma)$ of different standard deviation (σ) values, as

$$I(\boldsymbol{x};\sigma) = I_0(\boldsymbol{x}) * G(\boldsymbol{x};\sigma), \tag{A.1}$$

where $\boldsymbol{x}$ is a vector of dimension k which represents the image spatial position and

$$G(\boldsymbol{x};\sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^k} e^{-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}}. \tag{A.2}$$

In this case, σ represents the scale of analysis. In this sense, we use a strategy to properly select the local scale parameter and build the Hessian matrix. This strategy is based on the response

function $R(\boldsymbol{x};\sigma)$, computed as

$$R(\boldsymbol{x};\sigma) = \frac{\partial^{h+m+\ldots+n} I(\boldsymbol{x};\sigma)}{\partial x_1^h \partial x_2^m \ldots \partial x_k^n}, \tag{A.3}$$

where h, m and n are the orders of partial derivatives, and $x_i \in \boldsymbol{x}$, $i = 1, 2, \ldots, k$. Due to the commutative properties of convolution, the order of operations in Equation (A.1) and (A.3) can be changed, so that Equation (A.3) can be rewritten as:

$$R(\boldsymbol{x};\sigma) = I_0(\boldsymbol{x};\sigma) * \frac{\partial^{h+m+\ldots+n} G(\boldsymbol{x};\sigma)}{\partial x_1^h \partial x_2^m \ldots \partial x_k^n}. \tag{A.4}$$

Then, the local scale is defined as the value of σ (over a range of values) corresponding to the maximum of function $R(\boldsymbol{x};\sigma)$ for each pixel, which indicates the proper Gaussian scale probe (Gaussian observation kernel) with the width value corresponding to object feature size. Because of the amplitude of Gaussian derivative operators in (A.4) tends to decrease with increasing scale (because increasing scale, the response is increasingly smoothed), the so-called γ-parameterized normalized derivatives (LINDEBERG, 1998) were used in this work. They differ from the partial derivatives by the introduction of a normalizing factor $\sigma^{(h+m+\ldots+n)\gamma}$, so that

$$\frac{\partial^{h+m+\ldots+n}}{\partial u_1^h \partial v_2^m \ldots \partial w_k^n} = \sigma^{(h+m+\ldots+n)\gamma} \frac{\partial^{h+m+\ldots+n}}{\partial x_1^h \partial x_2^m \ldots \partial x_k^n}. \tag{A.5}$$

Thus, the increase of smoothing is compensated and, as a consequence, the accuracy of the proper scale selection (performed automatically) is improved. We set $\gamma = 1.25$, which was a value experimentally determined to work well on a variety of intensity structure profiles (MAJER, 2001).

In addition to the scale parameter setting, it is possible to specify the geometry information of local image structures by analyzing their intensity variations. For that, we used a set of second-order partial derivatives applied to the images (smoothed by Gaussian kernels). One of the most well-known partial derivative combinations in the literature is called the Hessian matrix, which is commonly adopted for the analysis of local image features. Therefore, based on the intensity responses of the γ-parameterized normalized derivative filters, a local measure of image structure is devised from the analysis of Hessian matrix eigenvalues of image intensity.

For a given scale σ, the Hessian matrix $H_\sigma(I;\boldsymbol{x})$ of an image I is a square and symmetric

matrix composed of second-order partial derivatives,

$$H_\sigma(I;\boldsymbol{x}) = \begin{bmatrix} I_{\sigma x_1^2} & I_{\sigma x_1 x_2} & \cdots & I_{\sigma x_1 x_k} \\ I_{\sigma x_2 x_1} & I_{\sigma x_2^2} & \cdots & I_{\sigma x_2 x_k} \\ \vdots & \vdots & \ddots & \vdots \\ I_{\sigma x_k x_1} & I_{\sigma x_k x_2} & \cdots & I_{\sigma x_k^2} \end{bmatrix}, \tag{A.6}$$

where

$$I_{\sigma x_i^2} = I_0(\boldsymbol{x}) * \left(\sigma^{2\gamma} \frac{\partial^2}{\partial x_i^2} G(\boldsymbol{x};\sigma) \right), \tag{A.7}$$

$$I_{\sigma x_i x_j} = I_{\sigma x_j x_i} = I_0(\boldsymbol{x}) * \left(\sigma^{2\gamma} \frac{\partial^2}{\partial x_i x_j} G(\boldsymbol{x};\sigma) \right). \tag{A.8}$$

The goal of the eigenvalue analysis is to extract the main directions of the local image structures decomposition. In this case, $\lambda_{\sigma,k}$ is the eigenvalue corresponding to the k-th normalized eigenvector $\hat{\boldsymbol{u}}_{\sigma,k}$ of Hessian matrix H_σ at scale σ. From eigenvector definition:

$$H_\sigma \hat{\boldsymbol{u}}_{\sigma,k} = \lambda_{\sigma,k} \hat{\boldsymbol{u}}_{\sigma,k}, \tag{A.9}$$

and, then

$$\hat{\boldsymbol{u}}_{\sigma,k}^T H_\sigma \hat{\boldsymbol{u}}_{\sigma,k} = \lambda_{\sigma,k}. \tag{A.10}$$

The eigenvalue decomposition extracts the orthonormal directions of the Hessian matrix in the neighborhood of an image point. The mutual magnitude of the Hessian eigenvalues is an indicator of the underlying object shape. Under the assumption that eigenvalues are sorted in order of increasing absolute value ($|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_k|$), the relations that must hold between the Hessian matrix eigenvalues for the detection of different structures are summarized in Table A.1.

Table A.1: Local structure patterns based on the analysis of the Hessian matrix eigenvalues (H=high, L=low, N=noisy, usually small, +/- indicate the sign of the eigenvalue), assuming $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$ (FRANGI et al., 1998).

2D		3D			Local structure pattern
λ_1	λ_2	λ_1	λ_2	λ_3	
N	N	N	N	N	noisy, no preferred direction
		L	L	H–	plate-like structure (bright)
		L	L	H+	plate-like structure (dark)
L	H–	L	H–	H–	tubular structure (bright)
L	H+	L	H+	H+	tubular structure (dark)
H–	H–	H–	H–	H–	blob-like structure (bright)
H+	H+	H+	H+	H+	blob-like structure (dark)

REFERENCES

ACTON, S. T.; RAY, N. Detection and tracking of rolling leukocytes from intravital microscopy. In: *IEEE International Symposium on Biomedical Imaging: Nano to Macro*. Arlington, VA, USA: IEEE, 2004. v. 2, p. 1235–1238.

ACTON, S. T.; WETHMAR, K.; LEY, K. Automatic tracking of rolling leukocytes in vivo. *Microvascular Research*, v. 63, n. 1, p. 139–148, 2002.

ADDISON, W. Experimental and practical researches on the structure and function of blood corpuscles; on inflammation, and on the origin and nature of tubercles in the lungs. In: *Transactions of the Provincial Medical and Surgical Association*. London, United Kingdom: ., 1843. v. 11, p. 223–306.

AKRAM, S. U. et al. Cell tracking via proposal generation and selection. *CoRR*, abs/1705.03386, 2017. Disponível em: <http://arxiv.org/abs/1705.03386>.

AMORNPHIMOLTHAM, P.; MASEDUNSKAS, A.; WEIGERT, R. Intravital microscopy as a tool to study drug delivery in preclinical studies. *Advanced Drug Delivery Reviews*, v. 63, n. 1-2, p. 119–128, 2011.

ANDRESEN, V. et al. High-resolution intravital microscopy. *PLoS ONE*, v. 7, n. 12, p. e50915, 2012.

ANDRIYENKO, A.; SCHINDLER, K. Multi-target tracking by continuous energy minimization. In: *CVPR 2011*. Colorado Springs, CO, USA: IEEE, 2011. p. 1265–1272.

ATIQUZZAMAN, M. Coarse-to-fine search technique to detect circles in images. *International Journal of Advanced Manufacturing Technology*, v. 15, p. 96–102, 1999.

AVANTS, B.; TUSTISON, N.; SONG, G. *Advanced Normalization Tools (ANTS)*. USA: University of Pennsylvania, 2009.

BALDI, P.; BRUNAK, S. *Bioinformatics - The Machine Learning Approach*. Cambridge, MA: MIT Press, 2001.

BERCLAZ, J.; FLEURET, F.; FUA, P. Robust people tracking with global trajectory optimization. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. New York, NY, USA: IEEE, 2006. p. 744–750.

BERCLAZ, J. et al. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 33, n. 9, p. 1806–1819, 2011.

BERG, S. et al. Ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, v. 16, p. 1226–1232, 2019.

BETKE, M.; HARITAOGLU, E.; DAVIS, L. S. Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications*, v. 12, n. 2, p. 69–83, 2000.

BETKE, M. et al. Tracking large variable numbers of objects in clutter. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, MN, USA: IEEE, 2007. p. 1–8.

BLACHNICKI, K. *Fluorescence Filters*. 2008. Disponível em: <https://commons.wikimedia.org/wiki/File:FluorescenceFilters.svg>.

BORN, M.; WOLF, E. *Principles of Optics*. New York: Pergamon Press, 1970.

BOSE, P. *The Encoding and Fourier Descriptors of Arbitrary Curves in 3-Dimensional Space*. Dissertação (Mestrado) — University of Florida, Gainesville, USA, Dec 2000.

BRENDEL, W.; AMER, M.; TODOROVIC, S. Multiobject tracking as maximum weight independent set. In: *CVPR 2011*. Colorado Springs, CO, USA: IEEE, 2011. p. 1273–1280.

BRUNELLI, R. *Template Matching Techniques in Computer Vision: Theory and Practice*. Italy: John Wiley & Sons, 2009.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. *Computers & Electrical Engineering*, v. 40, n. 1, p. 16–28, 2014.

CHEN, J. et al. PSTG-based multi-label optimization for multi-target tracking. *Computer Vision and Image Understanding*, v. 144, n. 1, p. 217–227, 2016.

CHEN, Y. et al. Automated 5-D analysis of cell migration and interaction in the thymic cortex from time-lapse sequences of 3-D multi-channel multi-photon images. *Journal of Immunological Methods*, v. 340, n. 1, p. 65–80, 2009.

CHEN, Y. et al. 3d object tracking via image sets and depth-based occlusion detection. *Signal Processing*, v. 112, n. 1, p. 146–153, 2015.

CHETVERIKOV, D. Particle image velocimetry by feature tracking. In: *Computer Analysis of Images and Patterns - CAIP*. Warsaw, Poland: Springer, 2001. v. 2124, p. 325–332.

CHETVERIKOV, D.; VERESTÓI, J. Feature point tracking for incomplete trajectories. *Computing*, v. 62, n. 4, p. 321–338, 1999.

CHOI, W.; SAVARESE, S. Multiple target tracking in world coordinate with single, minimally calibrated camera. In: *Computer Vision – ECCV 2010*. Heraklion, Crete, Greece: Springer Berlin Heidelberg, 2010. p. 553–567.

CHOI, W.; SAVARESE, S. A unified framework for multi-target tracking and collective activity recognition. In: *Computer Vision – ECCV 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 215–230.

CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, 2017. p. 1800–1807.

COLLINS, R. T. Multitarget data association with higher-order motion models. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA: IEEE, 2012. p. 1744–1751.

CONNERS, R. W.; HARLOW, C. A. A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2, n. 3, p. 204–222, 1980.

CONNERS, R. W.; TRIVEDI, M. M.; HARLOW, C. A. Segmentation of a high-resolution urban scene using texture operators. *Computer Vision, Graphics, and Image Processing*, v. 25, n. 3, p. 273–310, 1984.

CONSTANTINESCU, C. S. et al. Experimental autoimmune encephalomyelitis (EAE) as a model for multiple sclerosis (MS). *British Journal of Pharmacology*, v. 164, n. 4, p. 1079–1106, 2011.

CRUZ-ROA, A. A. et al. A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI*. Nagoya, Japan: Springer, 2013. v. 8150, p. 403–410.

CUI, J.; ACTON, S. T.; LIN, Z. A Monte Carlo approach to rolling leukocyte tracking in vivo. *Medical Image Analysis*, v. 10, n. 4, p. 598–610, 2006.

CUKIERMAN, E. et al. Taking cell-matrix adhesions to the third dimension. *Science*, v. 294, n. 5547, p. 1708–1712, 2001.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. San Diego, CA, USA: IEEE, 2005. p. 886–893 vol. 1.

DAVIDSON, M. W. *Numerical Aperture and Image Resolution*. 2019. Disponível em: <https://www.microscopyu.com/tutorials/imageformation-airyna>.

DEBEIR, O. et al. A model-based approach for automated in vitro cell tracking and chemotaxis analyses. *Cytometry Part A*, v. 60, n. 1, p. 29–40, 2004.

DEBEIR, O. et al. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE Transactions on Medical Imaging*, v. 24, n. 6, p. 697–711, 2005.

DICLE, C.; CAMPS, O. I.; SZNAIER, M. The way they move: Tracking multiple targets with similar appearance. In: *IEEE International Conference on Computer Vision*. Sydney, NSW, Australia: IEEE, 2013. p. 2304–2311.

DIVIETRO, J. A. et al. Immobilized IL-8 triggers progressive activation of neutrophils rolling in vitro on P-selectin and intercellular adhesion molecule-1. *The Journal of Immunology*, v. 167, n. 7, p. 4017–4025, 2001.

DONG, G.; RAY, N.; ACTON, S. T. Intravital leukocyte detection using the gradient inverse coefficient of variation. *IEEE Transaction on Medical Imaging*, v. 24, n. 7, p. 910–924, 2005.

DUFOUR, A. et al. Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE Transactions on Image Processing*, v. 14, n. 9, p. 1396–1410, 2005.

DUFOUR, A. et al. Filtering and segmentation of 3D angiographic data: Advances based on mathematical morphology. *Medical Image Analysis*, v. 17, n. 2, p. 147–164, 2013.

DUTROCHET, H. *Recherches anatomiques et physiologiques sur la structure intime des animaux et des végétaux, et sur leur motilité*. Universidade de Ghent: J.B. Baillière, 1824.

DZYUBAK, O. P.; RITMAN, E. L. Automation of hessian-based tubularity measure response function in 3D biomedical images. *International Journal of Biomedical Imaging*, v. 2011, n. Article ID 920401, p. 16 pages, 2011.

EBRAHIMDOOST, Y. et al. Automatic segmentation of pulmonary artery (PA) in 3D pulmonary CTA images. In: *17th International Conference on Digital Signal Processing (DSP)*. Corfu, Greece: IEEE, 2011. p. 1–5.

EDEN, E. et al. An automated method for analysis of flow characteristics of circulating particles from in vivo video microscopy. *IEEE Transactions on Medical Imaging*, v. 12, n. 8, p. 1011–1024, 2005.

EGMONT-PETERSEN, M. et al. Detection of leukocytes in contact with the vessel wall from in vivo microscope recordings using a neural network. *IEEE Transactions on Biomedical Engineering*, v. 47, n. 7, p. 941–951, 2000.

ELISA DE SOUZA, K. et al. Automatic detection of leukocytes from intravital video microscopy using the phase congruency technique. In: *Proceedigns of XI Workshop de Visão Computacional (WVC)*. São Carlos, SP, Brazil: BDBComp, 2015. p. 387–391.

ELISA DE SOUZA, K. et al. Detection of leukocytes in intravital microscopy video imagesusing the phase congruency technique. *Revista de Informática Teórica e Aplicada*, v. 23, n. 2, p. 33–55, 2016.

ELLINGER, P.; HIRT, A. Mikroskopische untersuchungen an lebenden organen i. mitteilung. methodik: Intravitalmikroskopie. *Zeitschrift für Anatomie und Entwicklungsgeschichte*, v. 90, p. 791–802, 1929.

ELLINGER, P.; HIRT, A. *Handbuch der Biologischen Arbeitsmethoden*. Berlin, Germany: Urban & Schwarzenberg, 1930.

FALK, T. et al. U-Net – deep learning for cell counting, detection, and morphometry. *Nature Methods*, v. 16, p. 67–70, 2019.

FAN, L. et al. A survey on multiple object tracking algorithm. In: *2016 IEEE International Conference on Information and Automation (ICIA)*. Ningbo, China: IEEE, 2016. p. 1855–1862.

FELZENSZWALB, P. F. et al. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 9, p. 1627–1645, 2010.

FENG, P. et al. Variational bayesian phd filter with deep learning network updating for multiple human tracking. In: *Sensor Signal Processing for Defence (SSPD)*. Edinburgh, UK: IEEE, 2015. p. 1–5.

FERRARI, R. J. et al. Automatic detection of motion blur in intravital video microscopy image sequences via directional statistics of log-Gabor energy maps. *Medical & Biological Engineering & Computing*, v. 53, n. 2, p. 151–163, 2015.

FLEURET, F. et al. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 2, p. 267–282, 2008.

FRANGI, A. F. et al. Multiscale vessel enhancement filtering. In: *Medical Image Computing and Computer-Assisted Interventation - MICCAI*. Cambridge, MA, USA: Springer-Verlag, 1998. v. 1496, p. 130–137.

FREEMAN, H. On the encoding of arbitrary geometric configurations. *Institute of Radio Engineers, trans. on Electronic Computers*, EC-10, p. 260–268, 1961.

FREIRE, P. G. L. et al. Detecção de leucócitos em imagens de vídeo de microscopia intravital. In: *XII Workshop de Informática Médica (WIM)*. Curitiba, PR, Brazil: BDBComp, 2012. p. 1–4.

FUKUNAGA, T. et al. Grouptracker: Video tracking system for multiple animals under severe occlusion. *Computational Biology and Chemistry*, v. 57, n. 1, p. 39–45, 2015.

GAVINS, F. N. E. Intravital microscopy: new insights into cellular interactions. *Current Opinion in Pharmacology*, v. 12, n. 5, p. 601–607, 2012.

GAVINS, F. N. E.; CHATTERJEE, B. E. Intravital microscopy for the study of mouse microcirculation in anti-inflammatory drug research: Focus on the mesentery and cremaster preparations. *Journal of Pharmacological and Toxicological Methods*, v. 49, n. 1, p. 1–14, 2004.

GAVRILA, D. M.; MUNDER, S. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, v. 73, n. 1, p. 41–59, 2007.

GHAJAR, C. M.; BISSELL, M. J. Extracellular matrix control of mammary gland morphogenesis and tumorigenesis: insights from imaging. *Histochemistry and Cell Biology*, Springer-Verlag, v. 130, n. 6, p. 1105–1118, 2008.

GHOSH, R. N.; WEBB, W. W. Automated detection and tracking of individual and clustered cell surface low density lipoprotein receptor molecules. *Biophysical Journal*, v. 66, p. 1301–1318, 1994.

GIRSHICK, R. Fast R-CNN. In: *IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, 2015.

GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Boston, MA, USA: Addison-Wesley, 1992.

GOOBIC, A. P. et al. Biomedical application of target tracking in clutter. In: *Conference on Signals, Systems and Computers*. Pacific Grove, CA, USA: IEEE, 2001. v. 1, p. 88–92.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. `http://www.deeplearningbook.org`.

GOUTTE, C.; GAUSSIER, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In: *Advances in Information Retrieval*. Santiago de Compostela, Spain: Springer, 2005. v. 3408, p. 345–359.

GREENSPAN, H.; GINNEKEN, B.; SUMMERS, R. M. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, v. 35, n. 5, p. 1153–1159, 2016.

GREGÓRIO DA SILVA, B. C.; CARVALHO-TAVARES, J.; FERRARI, R. J. Detection of leukocytes in intravital video microscopy based on the analysis of Hessian matrix eigenvalues. In: *28th Conference on Graphics, Patterns and Images*. Salvador, BA, Brazil: IEEE, 2015. p. 345–352.

GREGÓRIO DA SILVA, B. C.; CARVALHO-TAVARES, J.; FERRARI, R. J. Automated technique for in vivo analysis of leukocyte recruitment of mice brain microcirculation. In: *XII Workshop de Visão Computacional - WVC*. Campo Grande, MS, Brazil: WVC, 2016. p. 93–98.

GREGÓRIO DA SILVA, B. C.; CARVALHO-TAVARES, J.; FERRARI, R. J. Detecting and tracking leukocytes in intravital video microscopy using a Hessian-based spatiotemporal approach. *Multidimensional Systems and Signal Processing*, Springer US, v. 30, n. 2, p. 815–839, 2019.

HARALICK, R. M. Statistical and structural approaches to texture. *Proceedings of the IEEE*, v. 67, n. 5, p. 786–804, 1979.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, n. 6, p. 610–621, 1973.

HE, K. et al. Deep residual learning for image recognition. In: *CVPR - IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA: IEEE, 2016.

HE, K. et al. Identity mappings in deep residual networks. In: *European Conference on Computer Vision – ECCV*. Cham: Springer International Publishing, 2016. p. 630–645.

HIERKEGAARD, P. A method for detection of circular arcs based on the hough transform. *Machine Visions and Applications*, v. 5, p. 246–263, 1992.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. In: *Proceedings of the National Academy of Sciences of the USA*. USA: PNAS, 1982. v. 79, n. 8, p. 2554–2558.

HORN, B. K. P.; SCHUNCK, B. G. Determining optical flow. *Artificial Intelligence*, v. 17, n. 1-3, p. 185–203, 1981.

HU, W. et al. Single and multiple object tracking using a multi-feature joint sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 37, n. 4, p. 816–833, 2015.

HU, W. et al. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 34, n. 12, p. 2420–2440, 2012.

HUANG, C.; WU, B.; NEVATIA, R. Robust object tracking by hierarchical association of detection responses. In: *Computer Vision – ECCV 2008*. Marseille, France: Springer Berlin Heidelberg, 2008. p. 788–801.

HUANG, G. et al. Densely connected convolutional networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, 2017. p. 4700–4708.

HUANG, Y. et al. Quantitative analysis of lymphocytes morphology and motion in intravital microscopic images. In: *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Osaka, Japan: IEEE, 2013. p. 3686–3689.

HYNES, R. O.; LANDER, A. D. Contact and adhesive specificities in the associations, migrations, and targeting of cells and axons. *Cell*, v. 68, n. 2, p. 303–322, 1992.

ITSKOVITS, E. et al. A multi-animal tracker for studying complex behaviors. *BMC Biology*, v. 15, n. 29, p. 1–16, 2017.

IZADINIA, H. et al. (MP)2T: Multiple people multiple parts tracker. In: *Computer Vision – ECCV 2012*. Florence, Italy: Springer Berlin Heidelberg, 2012. p. 100–114.

JIA, C. et al. A tracking-learning-detection (TLD) method with local binary pattern improved. In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Zhuhai, China: IEEE, 2015. p. 1625–1630.

JIANG, Z.; HUYNH, D. Q. Multiple pedestrian tracking from monocular videos in an interacting multiple model framework. *IEEE Transactions on Image Processing*, v. 27, n. 3, p. 1361–1375, 2018.

JIANG, Z. et al. Multiple pedestrian tracking using colour and motion models. In: *2010 International Conference on Digital Image Computing: Techniques and Applications*. Sydney, NSW, Australia: IEEE, 2010. p. 328–334.

JIMENEZ-CARRETERO, D. et al. 3D frangi-based lung vessel enhancement filter penalizing airways. In: *IEEE 10th International Symposium on Biomedical Imaging*. San Francisco, CA, USA: IEEE, 2013. p. 926–929.

JONKER, P. P. Morphological operations on 3D and 4D images: From shape primitive detection to skeletonization. *Lecture Notes in Computer Science*, v. 1953, p. 371–391, 2000.

JOY, K. I. Bresenham's algorithm. On-Line Computer Graphics Notes. Computer Science Department, University of California, Davis. 1999.

KACHOUIE, N. N. et al. Probabilistic model-based cell tracking. *International Journal of Biomedical Imaging*, v. 2006, n. ID 12186, p. 10 pages, 2006.

KHOSRAVI, M.; SCHAFER, R. W. Template matching based on a grayscale hit-or-miss transform. *IEEE Transactions on Image Processing*, v. 5, n. 6, p. 1060–1066, 1996.

KILARSKI, W. W. et al. Intravital immunofluorescence for visualizing the microcirculatory and immune microenvironments in the mouse ear dermis. *PLos ONE*, v. 8, n. 2, p. e57135, 2013.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: *3rd International Conference on Learning Representations, ICLR*. San Diego, CA, USA: Ithaca, NY: arXiv.org, 2015. Disponível em: <http://arxiv.org/abs/1412.6980>.

KLEIN, S.; PLUIM, J. P. W.; STARING, M. Adaptive stochastic gradient descent optimisation for image registration. *International Journal of Computer Vision*, v. 81, n. 3, p. 227–239, 2009.

KOLLER, T. M. et al. Multiscale detection of curvilinear structures in 2-D and 3-D image data. In: *Fifth International Conference on Computer Vision*. Cambridge, Massachusetts: IEEE, 1995. p. 864–869.

KORHONEN, J.; JUNYONG, Y. Peak signal-to-noise ratio revisited: Is simple beautiful? In: *Workshop on Quality of Multimedia Experience (QoMEX)*. Melbourne, Australia: IEEE, 2012. p. 37–38.

KRATZ, L.; NISHINO, K. Tracking with local spatio-temporal motion patterns in extremely crowded scenes. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, USA: IEEE, 2010. p. 693–700.

KUMAR, K. C. A.; VLEESCHOUWER, C. D. Discriminative label propagation for multi-object tracking with sporadic appearance features. In: *IEEE International Conference on Computer Vision*. Sydney, NSW, Australia: IEEE, 2013. p. 2000–2007.

KUO, C.; HUANG, C.; NEVATIA, R. Multi-target tracking by on-line learned discriminative appearance models. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, USA: IEEE, 2010. p. 685–692.

KUSUMI, A.; SAKO, Y.; YAMAMOTO, M. Confined lateral diffusion of membrane receptors as studied by single particle tracking (nanovid microscopy). effects of calcium-induced differentiation in cultured endothelial cells. *Biophysical Journal*, v. 65, p. 2021–2040, 1993.

LACKIE, J. M.; CHAABANE, N.; CROCKET, K. V. A critique of the methods used to assess leucocyte behaviour. *Biomedicine & Pharmacotherapy*, v. 41, n. 6, p. 265–278, 1987.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, p. 436–444, 2015.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. In: *ISCAS 2010 - IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*. Paris, France: IEEE, 2010. p. 253–256.

LEE, T. C.; KASHYAP, R. L.; CHU, C. N. Building skeleton models via 3-D medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, v. 56, n. 6, p. 462–478, 1994.

LEEUWENHOEK, A. V.; HOOLE, S. *The Select Works of Antony Van Leeuwenhoek, Containing His Microscopical Discoveries in Many of the Works of Nature*. London: G. Sidney, 1800. v. 1.

LEWIS, J. P. Fast template matching. In: *Vision Interface 95*. Quebec City, Canada: Canadian Image Processing and Pattern Recognition Society, 1995. p. 120–123.

LI, B.; ACTON, S. T. Active contour external force using vector field convolution for image segmentation. *IEEE Transactions on Image Processing*, v. 16, n. 8, p. 2096–2106, 2007.

LI, F. et al. Bioimage informatics for systems pharmacology. *PLoS Comput Biol*, v. 9, n. 4, p. e1003043, 2013.

LI, K. et al. Cell population tracking and lineage construction with spatiotemporal context. *Medical Image Analysis*, v. 12, n. 5, p. 546–566, 2008.

LIN, T.-Y. et al. Feature pyramid networks for object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, 2017. p. 936–944.

LIN, T.-Y. et al. Focal loss for dense object detection. In: *IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy: IEEE, 2017. p. 2999–3007.

LINDEBERG, T. Feature detection with automatic scale selection. *International Journal of Computer Vision*, v. 30, n. 2, p. 79–116, 1998.

LIU, J. et al. Tracking sports players with context-conditioned motion models. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA: IEEE, 2013. p. 1830–1837.

LIU, X.; LIN, Z.; ACTON, S. T. A grid-based Bayesian approach to robust visual tracking. *Digital Signal Processing*, v. 22, n. 1, p. 54–65, 2012.

LOCHNER, M. J.; TRICK, L. M. Multiple-object tracking while driving: the multiple-vehicle tracking task. *Attention, Perception, & Psychophysics*, v. 76, n. 8, p. 2326–2345, 2014.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, 2015. p. 3431–3440.

LORENZ, C. et al. Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2D and 3D medical images. In: *Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery*. London, UK: Springer-Verlag, 1997. p. 233–242.

LOWE, D. G. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Kerkyra, Greece: IEEE, 1999. p. 1150–1157 vol.2.

LUO, W.; ZHAO, X.; KIM, T. Multiple object tracking: A review. *CoRR*, abs/1409.7618, 2014. Disponível em: <http://arxiv.org/abs/1409.7618>.

LUSTER, A. D.; ALON, R. Immune cell migration in inflammation: present and future therapeutic targets. *Nature Immunology*, v. 6, n. 12, p. 1182–90, 2005.

MAJER, P. The influence of the gamma-parameter on feature detection with automatic scale selection. In: *Proceedings of the Third International Conference on Scale-Space and Morphology in Computer Vision*. Vancouver, Canada: Springer, 2001. p. 245–254.

MASEDUNSKAS, A. et al. Intravital microscopy. *BioArchitecture*, v. 2, n. 5, p. 143–157, 2012.

MASKA, M. et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, v. 30, n. 11, p. 1609–1617, 2014.

McClarren, R. G. Open root finding methods. In: *Computational Nuclear Engineering and Radiological Science Using Python*. [S.l.]: Academic Press, 2018. cap. 13, p. 229–249.

MEIJERING, E.; DZYUBACHYK, O.; SMAL, I. Chapter nine - methods for cell and particle tracking. In: CONN, P. M. (Ed.). *Imaging and Spectroscopic Analysis of Living Cells*. [S.l.]: Academic Press, 2012, (Methods in Enzymology, v. 504). cap. 9, p. 183–200.

MENET, S.; MARC, S. P.; MEDIONI, G. B-Snakes: implementation and application to stereo. In: *Proceedings on Image Understanding Workshop*. .: ., 1990. p. 720–726.

MERCHANT, F. A. et al. 10.9 – Confocal Microscopy. In: *Handbook of Image and Video Processing*. Second. [S.l.]: Academic Press, 2005. cap. 10, p. 1291–1309, XL–XLI.

MILAN, A.; SCHINDLER, K.; ROTH, S. Detection- and trajectory-level exclusion in multiple object tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA: IEEE, 2013. p. 3682–3689.

MITZEL, D. et al. Multi-person tracking with sparse detection and continuous segmentation. In: *Computer Vision – ECCV 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 397–410.

MITZEL, D.; LEIBE, B. Real-time multi-person tracking with detector assisted structure propagation. In: *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. Barcelona, Spain: IEEE, 2011. p. 974–981.

MORGENTHALER, D. G. *Three-Dimensional Digital Topology: The Genus*. Maryland University. College Park. Computer Vision Lab. USA, 1980.

MUKHERJEE, D. P.; RAY, N.; ACTON, S. T. Level set analysis for leukocyte detection and tracking. *IEEE Transactions on Image Processing*, v. 13, n. 4, p. 562–572, 2004.

NOBIS, M. et al. Molecular mobility and activity in an intravital imaging setting – implications for cancer progression and targeting. *Journal of Cell Science*, v. 131, n. 5, p. 1–18, 2018.

NYÚL, L. G.; UDUPA, J. K.; ZHANG, X. New variants of a method of MRI scale standardization. *IEEE Transactions on Medical Imaging*, v. 19, n. 2, p. 143–150, 2000.

ÖKSüZ, I.; ÜNAY, D.; KADIPAŞAOĞLU, K. A hybrid method for coronary artery stenoses detection and quantification in CTA images. In: *MICCAI Workshop 3D Cardiovascular Imaging: A MICCAI Segmentation*. Nice, France: [s.n.], 2012.

OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, v. 9, n. 1, p. 62–66, 1979.

PADFIELD, D. et al. Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. *Medical Image Analysis*, v. 13, n. 1, p. 143–155, 2009.

PALÁGYI, K. *Skeletonization*. 2015. Disponível em: <http://www.inf.u-szeged.hu/˜palagyi/skel/skel.html>.

PINHO, V. et al. Intravital microscopy to study leukocyte recruitment in vivo. In: *Light Microscopy - Methods in Molecular Biology*. USA: Humana Press, 2011. v. 689, cap. 6, p. 81–90.

PINTO, C. H. V. et al. Detection of leukocytes in intravital microscopy video images using the phase congruency technique. *Revista de Informática Teórica e Aplicada*, v. 22, n. 1, p. 52–74, 2015.

PITTET, M. J.; WEISSLEDER, R. Intravital imaging. *Cell*, v. 147, n. 5, p. 983–991, 2011.

PLUIM, J. P. W.; MAINTZ, J. B. A.; VIERGEVER, M. A. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, v. 22, n. 8, p. 986–1004, 2003.

POYNTON, C. *Frequently Asked Questions about Color*. 1997. Disponível em: <http://poynton.ca/PDFs/ColorFAQ.pdf>.

PROLONG. *ProLong antifade reagents*. 2015. Disponível em: <http://www.lifetechnologies.com/br/en/home/life-science/cell-analysis/cellular-imaging/fluorescence-microscopy-and-immunofluorescence-if/mounting-medium-antifades/prolong-gold-antifade.html>.

QIN, Z.; SHELTON, C. R. Improving multi-target tracking via social grouping. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA: IEEE, 2012. p. 1972–1978.

RAY, N. A concave cost formulation for parametric curve fitting: Detection of leukocytes from intravital microscopy images. In: *Proceedings of the International Conference on Image Processing*. Hong Kong, China: IEEE, 2010. p. 53–56.

RAY, N.; ACTON, S. T. Motion gradient vector flow: An external force for tracking rolling leukocytes with shape and size constrained active contours. *IEEE Transactions on Medical Imaging*, v. 23, n. 12, p. 1466–1478, 2004.

RAY, N.; ACTON, S. T.; LEY, K. Tracking leukocytes *in vivo* with shape and size constrained active contours. *IEEE Transactions on Medical Imaging*, v. 21, n. 10, p. 1222–1235, 2002.

REN, S. et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, v. 39, n. 06, p. 1137–1149, 2017.

RIAHI, D.; BILODEAU, G. Multiple object tracking based on sparse generative appearance modeling. In: *IEEE International Conference on Image Processing (ICIP)*. Quebec City, QC, Canada: IEEE, 2015. p. 4017–4021.

RODRIGUEZ, A. et al. Toxid: an efficient algorithm to solve occlusions when tracking multiple animals. *Scientific Reports*, v. 7, n. 14774, p. 2045–2322, 2017.

RODRIGUEZ, M. et al. Data-driven crowd analysis in videos. In: *International Conference on Computer Vision*. Barcelona, Spain: IEEE, 2011. p. 1235–1242.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI*. Munich, Germany: Springer, 2015. v. 9351, p. 234–241.

ROSENBLUM, W. I.; ZWEIFACH, W. Cerebral microcirculation in the mouse brain. *Archives of Neurology*, v. 9, p. 414–423, 1963.

RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015.

SAHOO, S.; RAY, N.; ACTON, S. T. Rolling leukocyte detection based on teardrop shape and the gradient inverse coefficient of variation. In: *International Conference on Medical Information Visualisation*. London, United Kingdom: IEEE Computer Society, 2006. p. 29–33.

SANTOS, A. C. D. et al. CCL2 and CCL5 mediate leukocyte adhesion in experimental autoimmune encephalomyelitis an intravital microscopy study. *Journal of Neuroimmunology*, v. 162, n. 1-2, p. 122–129, 2005.

SANTOS, A. C. D. et al. Kinin B2 receptor regulates chemokines CCL2 and CCL5 expression and modulates leukocyte recruitment and pathology in experimental autoimmune encephalomyelitis (EAE) in mice. *Journal of Neuroinflammation*, v. 5, p. 49–58, 2008.

SANTOS, J. C. dos et al. Stingray venom activates IL-33 producing cardiomyocytes, but not mast cell, to promote acute neutrophil-mediated injury. *Scientific Reports*, v. 7, n. 7912, p. 2045–2322, 2017.

SATO, Y. et al. Measuring microcirculation using spatiotemporal image analysis. In: AYACHE, N. (Ed.). *Computer Vision, Virtual Reality and Robotics in Medicine*. Nice, France: Springer, 1995. v. 905, p. 302–308.

SATO, Y. et al. Automatic extraction and measurement of leukocyte motion in microvessels using spatiotemporal image analysis. *IEEE Transactions on Biomedical Engineering*, v. 44, n. 4, p. 225–236, 1997.

SATO, Y. et al. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, v. 2, n. 2, p. 143–168, 1998.

SCHüTZ, G. J.; SCHINDLER, H.; SCHMIDT, T. Single-molecule microscopy on model membranes reveals anomalous diffusion. *Biophysical Journal*, v. 73, n. 2, p. 1073–1080, 1997.

SHEN, H. et al. Automatic tracking of biological cells and compartments using particle filters and active contours. *Chemometrics and Intelligent Laboratory Systems*, v. 82, n. 1-2 SEPC. ISS, p. 276–282, 2006.

SHI, J.; TOMASI. Good features to track. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA: IEEE, 1994. p. 593–600.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, v. 6, n. 60, p. 1–48, 2019.

SHU, G. et al. Part-based multiple-person tracking with partial occlusion handling. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2012. p. 1815–1821.

SIMARD, P. Y.; STEINKRAUS, D.; PLATT, J. C. Best practices for convolutional neural networks applied to visual document analysis. In: *Seventh International Conference on Document Analysis and Recognition*. Edinburgh, UK, UK: IEEE, 2003.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognitions. In: *International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2015.

SMITH, L. Cyclical learning rates for training neural networks. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. Santa Rosa, CA, USA: IEEE, 2017. p. 464–472.

SOBEL, I. *An Isotropic 3x3 Gradient Operator, Machine Vision for Three-Dimensional Scenes*. NY: Academic Press, 1990. 376–379 p.

SOLÓRZANO, C. Ortiz-de et al. *ISBI Cell Tracking Challenge*. 2014. Disponível em: <http://www.codesolorzano.com/Challenges/CTC/Welcome.html>.

STEEBER, D. A.; CAMPBELL, M. A. Optimal selectin-mediated rolling of leukocytes during inflammation in vivo requires intercellular adhesion molecule-1 expression. In: *Proceedings of the National Academy of Sciences*. USA: PNAS, 1998. v. 95, n. 13, p. 7562–7567.

SUGIMURA, D. et al. Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait. In: *IEEE 12th International Conference on Computer Vision*. Kyoto, Japan: IEEE, 2009. p. 1467–1474.

SULEYMANOVA, I. et al. A deep convolutional neural network approach for astrocyte detection. *Scientific Reports*, v. 8, n. 12878, p. 1–7, 2018.

SUN, Z.; BEBIS, G.; MILLER, R. On-road vehicle detection: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 28, n. 5, p. 694–711, 2006.

SUWANNATAT, T.; CHINNASARN, K.; INDRA-PAYOONG, N. Multi-features particle PHD filtering for multiple humans tracking. In: *International Computer Science and Engineering Conference (ICSEC)*. Chiang Mai, Thailand: IEEE, 2015. p. 1–6.

SZEGEDY, C. et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Thirty-first AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2017.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016. p. 2818–2826.

TAKANO, T. et al. Astrocyte-mediated control of cerebral blood flow. *Nature Neuroscience*, v. 9, p. 260–267, 2006.

TANG, J. et al. Evaluation of intravital tracking algorithms. In: *The 2002 45th Midwest Symposium on Circuits and Systems*. Tulsa, Oklahoma: IEEE, 2002. v. 1, p. 220–223.

TANG, S.; ANDRILUKA, M.; SCHIELE, B. Detection and tracking of occluded people. *International Journal of Computer Vision*, v. 110, n. 1, p. 58–69, 2014.

TANG, S. et al. Learning people detectors for tracking in crowded scenes. In: *IEEE International Conference on Computer Vision*. Sydney, NSW, Australia: IEEE, 2013. p. 1049–1056.

TAO, A.; BARKER, J.; SARATHY, S. *DetectNet: Deep Neural Network for Object Detection in DIGITS*. 2016. Disponível em: <https://devblogs.nvidia.com/detectnet-deep-neural-network-object-detection-digits/>.

THIRION, J.-P. Image matching as a diffusion process: An analogy with Maxwell's demons. *Medical Image Analysis*, v. 2, n. 3, p. 243–260, 1998.

TOMASI, C.; KANADE, T. *Detection and Tracking of Point Features*. Carnegie Mellon University, 1991.

TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: *Proceedings of the Sixth International Conference on Computer Vision*. Bombay, India: IEEE Computer Society, 1998. p. 839–846.

TRANTHAM, K.; REECE, T. J. Demonstration of the airy disk using photography and simple light sources. *American Journal of Physics*, v. 83, n. 11, p. 928–934, 2015.

TüRETKEN, e. et al. Network flow integer programming to track elliptical cells in time-lapse sequences. *IEEE Transactions on Medical Imaging*, v. 36, n. 4, p. 942–951, 2017.

ULMAN, V. et al. An objective comparison of cell-tracking algorithms. *Nature Methods*, v. 14, n. 12, p. 1141–1152, 2017.

VERESTÓY, J.; CHETVERIKOV, D. *Feature point Tracking algorithms*. 1998. Disponível em: <http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/CHETVERIKOV/psmweb/psmweb.html>.

WAGNER, R. *Erläuterungstafeln zur Physiologie und Entwicklungsgeschichte*. Leipzig: Leopold Voss, 1839.

WäHLBY, C. et al. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *Journal of Microscopy*, v. 215, n. 1, p. 67–76, 2004.

WALIA, G. S.; KAPOOR, R. Recent advances on multicue object tracking: a survey. *Artificial Intelligence Review*, v. 46, n. 1, p. 1–39, 2016.

WEIGERT, R.; PORAT-SHLIOM, N.; PARENTE AMORNPHIMOLTHAM, P. Imaging cell biology in live animals: Ready for prime time. *The Journal of Cell Biology*, v. 201, n. 7, p. 969–979, 2013.

WEIGERT, R. et al. Intravital microscopy: a novel tool to study cell biology in living animals. *Histochemistry and Cell Biology*, v. 133, n. 5, p. 481–491, 2010.

WU, Z. et al. Coupling detection and data association for multiple object tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA: IEEE, 2012. p. 1948–1955.

XI, Z. et al. A* algorithm with dynamic weights for multiple object tracking in video sequence. *Optik - International Journal for Light and Electron Optics*, v. 126, n. 20, p. 2500–2507, 2015.

XIE, J.; KHAN, S.; SHAH, M. Automatic tracking of escherichia coli in phase-contrast microscopy video. *IEEE Transactions on Biomedical Engineering*, v. 56, n. 2, p. 390–399, 2009.

XIE, Y. et al. Efficient and robust cell detection: A structured regression approach. *Medical Image Analysis*, v. 44, p. 245–254, 2018.

XING, F. et al. Deep learning in microscopy image analysis: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, v. 29, n. 10, p. 4550–4568, 2018.

XING, J. et al. Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling. *IEEE Transactions on Image Processing*, v. 20, n. 6, p. 1652–1667, 2011.

XU, C.; PRINCE, J. L. Generalized gradient vector flow external forces for active contours. *Signal Processing*, v. 71, n. 2, p. 131–139, 1998.

XU, C.; PRINCE, J. L. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, v. 7, n. 3, p. 359–369, 1998.

XU, R.; BOUDREAU, A.; BISSELL, M. J. Tissue architecture and function: dynamic reciprocity via extra-and intra-cellular matrices. *Cancer and Metastasis Reviews*, v. 28, n. 1-2, p. 167–176, 2009.

XU, R.; LIU, Q. Multi-pedestrian tracking for far-infrared pedestrian detection on-board using particle filter. In: *2015 IEEE International Conference on Imaging Systems and Techniques (IST)*. Macau, China: IEEE, 2015. p. 1–5.

YAMAGUCHI, K. et al. Who are you with and where are you going? In: *CVPR 2011*. Colorado Springs, CO, USA: IEEE, 2011. p. 1345–1352.

YANG, B.; NEVATIA, R. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA: IEEE, 2012. p. 1918–1925.

YANG, M. et al. Detection driven adaptive multi-cue integration for multiple human tracking. In: *IEEE 12th International Conference on Computer Vision*. Kyoto, Japan: IEEE, 2009. p. 1554–1561.

YOON, J. H. et al. Bayesian multi-object tracking using motion context from multiple objects. In: *IEEE Winter Conference on Applications of Computer Vision*. Waikoloa, HI, USA: IEEE, 2015. p. 33–40.

YOSINSKI, J. et al. How transferable are features in deep neural networks? In: *Advances in Neural Information Processing Systems 27*. Montreal, Canada: Curran Associates, Inc., 2014. v. 2, p. 3320–3328.

ZENG, Z.; MA, S. An efficient vision system for multiple car tracking. In: *Object recognition supported by user interaction for service robots*. Quebec City, Quebec, Canada: IEEE, 2002. p. 609–612.

ZHANG, L.; MAATEN, L. van der. Structure preserving object tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA: IEEE, 2013. p. 1838–1845.

ZHANG, L.; MAATEN, L. van der. Preserving structure in model-free tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 36, n. 4, p. 756–769, 2014.

ZHAO, X.; GONG, D.; MEDIONI, G. Tracking using motion patterns for very crowded scenes. In: *Computer Vision – ECCV 2012*. Florence, Italy: Springer Berlin Heidelberg, 2012. p. 315–328.

ZHOU, X. et al. A novel cell segmentation method and cell phase identification using markov model. *IEEE Transactions on Information Technology in Biomedicine*, v. 13, n. 2, p. 152–157, 2009.

ZIMMER, C. et al. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing. *IEEE Transactions on Medical Imaging*, v. 21, n. 10, p. 1212–1221, 2002.

ZOPH, B. et al. Learning transferable architectures for scalable image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, USA: IEEE, 2018. p. 8697–8710.