

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Towards Automatic Detection and Quantification of Gait in Arthritic Rats

Ana Filipa Adonias Costa



INTEGRATED MASTER IN BIOENGINEERING

Supervisor: Jaime S. Cardoso, PhD

Co-supervisor: Joana Ferreira-Gomes, PhD

July 30, 2019

Towards Automatic Detection and Quantification of Gait in Arthritic Rats

Ana Filipa Adonias Costa

INTEGRATED MASTER IN BIOENGINEERING

July 30, 2019

Abstract

Rat's gait analysis plays an important role in the assessment of the impact of certain drugs on the treatment against arthritis, a common misunderstood disease that has a significant incidence worldwide, resulting in a considerable economic and social impact. This disorder damages the joints, structures responsible for the movement of the body, causing pain, stiffness and functional impairment, which have a tremendous impact on the patients' routines.

Currently, the search for new treatments for arthritis relies on the use of animal models (mainly rats and mice) to understand the pathophysiological mechanisms underlying this disorder and to test the therapeutic efficacy of targeted drugs. For that purpose, arthritis is induced in the animal, usually in the knee joint, and the symptoms and behavioral signs displayed by the animal throughout the evolution of the disease and following drug administration are quantified along time. Among the different behavioral signs, the animals' gait pattern is analyzed to understand, through a continuous monitoring, the disease impairment as well as the therapeutic efficacy of target drugs. This gait monitoring can be done through video analysis, by a manual or automated procedure. Although there are some systems capable of detecting and quantifying the gait changes in rodents automatically, they still present many limitations and researchers often opt for the development of a customized system for the acquisition of the videos, analyzing them manually. Nonetheless, this manual observation and quantification of the gait changes is a laborious, time-consuming and intensive task that is prone to high user variability, which mainly reveals the need for robust automated gait analysis systems in laboratory animals.

Therefore, and bearing in mind the recent advances in machine learning and computer vision fields, as well as their presence in many processing, tracking and recognition applications, this work is driven by the demand to find a solution to make the detection and quantification of the animals' gait changes an automatic task. Thus, we propose a methodology capable of detecting and segmenting the animal's body, detecting, segmenting and classifying its paws, and of selecting some frames of interest and extracting some gait metrics from them. This tracking-by-detection framework makes the detection and quantification of the animal gait changes task easier, faster, simpler, more robust and less susceptible to user variability. As far as performance is concerned, the adopted methodology proved to be effective, being able to return frames of interest in 87% of the dataset videos and to return final gait metrics similar to the ones obtained by the experts with a global mean difference of $3,44\% \pm 4,25$.

In order to make the proposed framework a free service available to researchers, this was integrated in a web application tool with an user friendly interface.

Results of this work also include the creation of a database with 15 videos of the rat's gait and the manual annotation of 130 frames, as well as an article published in the ninth edition of the Iberian Conference on Pattern Recognition and Image Analysis with the title "Towards Automatic Rat's Gait Analysis Under Suboptimal Illumination Conditions". The data used for the development of this dissertation was provided by the Faculdade de Medicina da Universidade do Porto.

Acknowledgements

First I would like to thank my supervisor, Professor Jaime Cardoso, for the continuous support and availability during my master's thesis development, for all the suggestions and advices and for promoting the collaboration and knowledge exchange between his students. I would also like to thank Professor Joana Ferreira-Gomes and Professor Fani Neto, for the continuous monitoring during the project and for the availability of the material, without which the project would not have been possible to carry out. Thanks also to Raquel, who was always available during my thesis development.

My biggest thank you goes to my parents and my brother who, over the past five years, were always there for me, no matter what, always supporting my ideas and reminding me that there is life behind books. Thank you for letting me make the living room my office.

A heartfelt thanks to João, my person on the cosmos who, through love, makes everything easier and bearable.

To all my family for the unconditional support they have shown. To my friends for always being at Mali ready for a conversation with the timer on (although now they should be more often at Praça). Thank you for helping me keep my sanity whenever I was close to losing it.

To Mariana for being my bioengineering sister. To Paulo Maia for being the father of bio-engineering. To all the people with whom I shared the past five years for understanding how challenging Bio can be. To the Portucheese for giving me the best semester I could ever had. To the BEST people, who are indeed the best, thank you for always challenging me to leave my comfort zone and dare myself to grow.

And finally, to my little star, Snoky for teaching me how infinite the heart can be.

*“Somewhere,
something incredible is waiting to be known”*

Carl Sagan

Contents

1	Introduction	1
1.1	Context	1
1.2	Problem Description and Motivation	1
1.3	Research Goals and Contributions	2
1.4	Dissertation Structure	3
2	Arthritis	5
2.1	Joints	5
2.2	Types of Arthritis	6
2.2.1	Osteoarthritis	6
2.2.2	Rheumatoid Arthritis	8
2.3	Socio-Economic Impact	10
2.4	Summary and Conclusions	10
3	Experimental Arthritis and Rodent Gait	11
3.1	Experimental Arthritis	11
3.2	Rodent Gait Analysis	12
3.2.1	Gait Modifications in an Animal with Arthritis	12
3.3	Detecting Rodent Gait Modifications	13
3.3.1	Manual Video Gait Analysis	13
3.3.2	Video-Based Automatic Systems for Gait Analysis	15
3.4	Summary and Conclusions	18
4	Object Tracking	19
4.1	Tracking-by-detection	20
4.1.1	Object Representation	20
4.1.2	Object Detection	20
4.1.3	Object Classification	30
4.1.4	Object Tracking	31
4.2	Deep Learning-Based Tracking	34
4.3	Tracking Evaluation Metrics	35
4.4	Mouse/Rat Tracking	38
4.5	Summary and Conclusions	38
5	Acquisition Setup and Dataset	41
5.1	Acquisition Setup	41
5.1.1	Animal Model	41
5.1.2	CatWalk Test	41

5.2	Dataset	42
5.2.1	Dataset Heterogeneity	43
5.2.2	Data Annotation	44
5.2.3	Public Rodent Datasets	46
6	Rat Tracking Methodology	47
6.1	Rat Detection and Segmentation	48
6.2	Paws Detection, Segmentation and Classification	51
6.3	Frame Selection	53
6.4	Gait Metrics Extraction	54
7	Rat Tracking Results and Discussion	57
7.1	Rat Detection and Segmentation	57
7.2	Paws Detection, Segmentation and Classification	60
7.3	Frame Selection	62
7.4	Gait Metrics Extraction	63
8	Development of a Web Tool	65
8.1	Django Framework Architecture	66
8.2	Database	67
8.3	Interface and Functionalities	68
8.3.1	Video Upload	68
8.3.2	Video Conversion	69
8.3.3	Video Processing	69
8.3.4	See Results	70
8.3.5	Select Results	71
8.3.6	Download Results	71
8.3.7	Delete Video and Delete Results	72
9	Conclusions and Future Work	73
	References	75

List of Figures

2.1	Anatomical planes.	6
2.2	OA different stages.	7
2.3	OA in finger joints.	8
2.4	Comparison between normal, RA and OA knees.	8
2.5	RA in finger joints.	9
3.1	Typical spatiotemporal gait sequences for walking rodents. (*Fore limbs were removed in spatial characteristics images for clarity).	13
3.2	Set of walking frames example.	14
3.3	Standing still frame example.	14
3.4	DigitGait TM system.	15
3.5	CatWalk XT system	16
3.6	MouseWalker system.	16
3.7	GaitScan interface.	17
4.1	Flow diagram of tracking-by-detection.	20
4.2	Object Representation.	21
4.3	Image segmentation techniques.	22
4.4	Single (left) and Multiple (right) Thresholding.	22
4.5	Neural Networks representation.	24
4.6	Convolutional Neural Networks representation.	25
4.7	Flow Chart Of Background Subtraction Model.	27
4.8	Different tracking approaches.	31
4.9	Different motion constraints. \triangle denotes object position at frame t-2, \circ denotes object position at frame t-1, and x denotes object position at frame t.	32
4.10	TP, FP, TN and FN example.	36
5.1	Principle of the CatWalk setup.	42
5.2	Example of a frame with background noise.	43
5.3	Example of frames with occlusions. A: Full occlusion; B: Partial occlusion. . . .	44
5.4	Example of behaviors that hinder paws segmentation. A: Rat's belly in contact with the platform; B: Scratching.	44
5.5	Annotation process. A. Input frame; B. Frame annotated; C. Ground truth masks.	45
5.6	Process of obtaining masks of paws. A. Input frame; B. Region of the paws manually labeled; C. Ground truth paws mask.	45
5.7	A. Input frame; B. <i>White:</i> Ground truth rat's body; <i>Gray:</i> Ground truth tail; C. Ground truth paws: <i>LH:</i> Left-Hind; <i>RH:</i> Right-Hind; <i>LF:</i> Left-Fore; <i>RF:</i> Right-Fore.	45
6.1	Fluxogram of the framework.	47

6.2	The generic video-based rat tracking framework.	48
6.3	Box linear kernel's (k) effect. A: Input frame; B: Denoised image.	49
6.4	A: Input frame; B: Foreground; C: Output of contrast enhancement operation; D: Result of the sharpening filter.	51
6.5	Rat's elliptical representation.	52
6.6	A: Rat and tail detection; B: Result of the opening operation applied to remove the animal's tail. The gray contour is the convex hull of the result; C: Result of the subtraction of the image A by the convex hull of the image B.	53
6.7	White points: Spatial distribution of the minimum points; Red point: Tail point prediction.	53
6.8	Rat's trajectory. The ellipse represents the rat, the red point the tail point obtained by the first method described and the red points the tail points obtained by the temporal method.	54
7.1	Rat segmentation examples for the proposed methodologies: White: Ground Truth; Blue: K-Means; Yellow: Min BM; Green: Mean BM; Red: RPCA; . . .	57
7.2	Rat segmentation successful plot for the 130 frames manually labelled.	58
7.3	A: Mean Background Model; B: Minimum Background Model; C: RPCA Background Model of the different intervals: C1 - 0 to 1999; C2 - 2000 to 3999; C3 - 4000 to 5999; C4 - 6000 to 7999).	59
7.4	Min background mouse detection examples. A: Frame number 1823; B: Correct mouse position prediction; C: Frame number 5951; D: Incorrect mouse position prediction.	60
7.5	Paws' segmentation examples: White: TP pixels; Black: TN pixels; Blue: FP pixels; Red: FN pixels	60
8.1	Pipeline for the proposed web application.	65
8.2	Django Model-View-Template Architecture.	66
8.3	Django application layout structure.	66
8.4	Simplified diagram of the database for the rat's gait analysis project.	67
8.5	Use Case diagram of the Web Application System.	68
8.6	Initial home page.	69
8.7	Video upload functionality.	69
8.8	Example of a result display.	70
8.9	Main page with videos already processed.	70
8.10	End of the display results page.	71
8.11	Example of a downloaded excel file.	72

List of Tables

3.1	Gait parameters.	12
3.2	Commercially available video-based automatic gait analysis systems.	17
4.1	K-means and Fuzzy C-means Algorithms.	23
5.1	Rodent Behavior Datasets.	46
7.1	Results of paws' segmentation.	61
7.2	Results of paws' classification.	62
7.3	Results of frame selection. Percentage of RH pressure computed: G: for all the frames; W: for the walking frames; SS: for the standing still frames.	62
7.4	Comparison of the obtained results with the GT.	63
7.5	Results of gait metrics extraction.	64

Abbreviations and Symbols

ANN	Artificial Neural Network
BS	Background Subtraction
CB	Codebook
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Networks
EKF	Extended Kalman filter
FD	Frame Differencing
FN	False Negative
FP	False Positive
FPS	Frames Per Second
GA	Gradient Descent
GCBAC	Graph Cuts Based Active Contours
GMM	Gaussian Mixture Model
GT	Ground Truth
KF	Kalman Filter
NN	Neural Network
NO-RMC	Nearly Optimal Robust Matrix Completion
NORST	Nearly Optimal Robust Subspace Tracking
OA	Osteoarthritis
OTA	Object Tracking Accuracy
OTP	Object Tracking Precision
PCP	Principal Component Pursuit
PDE	Partial Differential Equation
RA	Rheumatoid Arthritis
RBM	Restricted Boltzmann Machines
R-CNN	Region Convolutional Neural Network
RNN	Recurrent Neural Network
RPCA	Robust Principal Component Analysis
RST	Robust Subspace Tracking
SIFT	Scale Invariant Feature Transform
SSD	Sum of Squared Differences
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UKF	Unscented Kalman filter

Chapter 1

Introduction

1.1 Context

Technology is growing at a breakneck speed and it is clear, more than ever, its impact on our daily lives, being impossible to live without it. In all fields, technology is present by offering a solution to unanswered questions, and by making procedures easier by turning them more automatic and less susceptible to mistakes. In health-care, the scenario is not different. Medicine is increasingly dependent on technology and automatic processes are being more used to reduce costs and increase efficacy [1].

In medical research, one of the key technologies in high-tech automatic applications is computer vision, an interdisciplinary field of computer science that aims at enabling computers to see, identify and process digital images and videos in the same way that human eyes do. Computer vision tasks include acquiring, processing and understanding digital images or videos [2].

The advances on computer vision, machine learning and imaging processing fields have been contributing to progresses both at the methodological and application levels [3]. Computer vision in medical image applications can be used on many imaging modalities contributing to faster and easier diagnosis [4], and to more effective solutions. With focus on the latter, medical imaging can also be applied in drug development processes for identification of lesions, by detecting changes in tissues, cells or molecules [5], and for therapy monitoring and follow-up by evaluating drug efficacy on animals [6]. The development of medications for arthritis fits into this last example of application, i.e. therapy monitoring.

1.2 Problem Description and Motivation

Arthritis is a very painful disease that affects mainly the joints. It is a condition not easy to live with, and carrying out simple daily tasks, such as walking or climbing stairs, can be challenging.¹ Arthritis precludes patients from having an active and healthy lifestyle making them

¹<https://www.verywellhealth.com/arthritis-affects-daily-living-activities-189933>

more vulnerable to other serious diseases [7]. It can also have a strong impact on patient's performance at work often, making it impossible to accomplish all the tasks. There are many unpleasant symptoms that patients have to live with, which also increases the patients' probability to develop psychiatric-related problems, such as depression.²

A curative treatment for arthritis has not yet been found and, although there are some drugs that aim to relieve its symptoms, these therapeutic strategies remain unsatisfactory once they fail in pain relief or trigger undesirable side-effects [8]. Consequently, scientists have attempted to develop more efficient drugs for the treatment of arthritis by relying on the use of animal experimentation, which remains a key instrument. Among the animal species used in research, mice and rats can be recognized as the most important and widely used models to understand the pathophysiological mechanisms underlying arthritis and to test the therapeutic efficacy of targeted drugs [9]. The experimental procedure consists in inducing the arthritis in the animals (mice or rats), and afterwards administering a drug to the animal and quantifying the changes in the symptoms and behavioral signs of the animal, such as its gait, to understand, through a continuous monitoring, the therapeutic impact of those drugs against arthritis. The monitoring procedure can be done through video analysis. However, the manual observation and quantification of those changes is a laborious and time-consuming intensive task that is prone to high user variability [10]. In addition, the current available automatic solutions are often expensive closed solutions and some of them greatly limit the area where the animal can walk or force it to move at a certain speed on a treadmill, inducing an external pressure which may affect the results [11]. In this way, a demand for robust automated systems of gait analysis in laboratory animals arises primarily, in order to solve problems related not only with the time and cost but also with the reproducibility inherent to the assessment process conducted by the researchers. Moreover, the availability of such systems introduces the possibility to rethink gait analysis itself, since the typical testing time scale can be easily extended, thus diversifying the gait parameters extracted and the context of evaluation.

Therefore, and bearing in mind the recent advances in computer vision field as well as its presence in many tracking and recognition applications, this work is driven by the need to find a solution to make the detection and quantification of the animal gait changes an easier, faster, simpler and more robust process.

1.3 Research Goals and Contributions

According to the description of the aforementioned problem and the behind motivation, this dissertation has as main contribution the development of a framework able to automatically detect and quantify gait changes in the rat as well as the development of a web platform deploying the proposed framework as a free service available to researchers.

To achieve this goal, a review of the fundamentals of arthritis, experimental arthritis and rodent gait analysis is presented, allowing a detailed comprehension of the underlying problem. By

²<https://www.mayoclinic.org/diseases-conditions/arthritis/symptoms-causes/syc-20350772>

gathering the current state-of-the-art of the existent solutions, it is possible to understand the efforts already made to solve it.

The proposed framework was developed by resorting to computer vision techniques. Therefore, an exhaustive study of the current methods of object detection and tracking is also presented in this document. This review allowed to understand which tools are available to achieve the proposed goal and which are the main challenges that current tracking algorithms are facing and extrapolate them to the study in question, raising an awareness of the possible problems that this work may face.

Results of this work also include the creation of a database with 15 videos of rat's gait and the manual annotation of 130 frames, as well as an article published in the ninth edition of the Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) with the title "Towards Automatic Rat's Gait Analysis Under Suboptimal Illumination Conditions" [12].

1.4 Dissertation Structure

Apart from the Introduction (Chapter 1) where both the topic and goals of this dissertation are presented, the following list describes the contents of the remaining chapters:

- *Chapter 2: Arthritis* - The fundamentals of this disorder are presented. Some types of arthritis, symptoms and treatments, as well as the socioeconomic impact are discussed. This chapter aims to provide the basis to understand the need for the development of new treatments for arthritis.
- *Chapter 3: Experimental Arthritis and Rodent Gait* - Some background concepts about experimental arthritis and rodent gait analysis are presented, as well as the actual manual procedure to detect rodent gait modifications. Moreover, an overview of the current existent automatic systems that have been developed to overcome the obstacles of the manual procedure is also done.
- *Chapter 4: Object Tracking* - An exhaustive overview of object tracking methods is presented and some performance metrics evaluation are also introduced. In this chapter, a review of some published work related with the topic of the thesis is also presented. The main goal of this chapter is to allow the knowledge of the current state of existing methodologies for object tracking that can be applied to the described problem in order to achieve the goals of this thesis.
- *Chapter 5: Acquisition Setup and Dataset* - A description of how data is acquired is presented as well as how the dataset is created covering the work done in the annotation task. Some features of the dataset are also mentioned in this chapter.
- *Chapter 6: Rat Tracking Methodology* - In this chapter the proposed methodology and the experiments to perform it are described.

- *Chapter 7: Rat Tracking Results and Discussion* - The obtained results for the different experiments reported in the previous chapter are presented and discussed.
- *Chapter 8: Development of a Web Tool* - This chapter covers the work done to integrate the proposed framework in a web platform making it a free service available to researchers.
- *Chapter 9: Conclusions and Future Work* - The main conclusions of this dissertation are presented, as well as some notes concerning the future work.

Chapter 2

Arthritis

The terminology given to any joint disease is arthropathy.¹ Arthritis (arthro = joint, itis = inflammation)² is no more than a type of arthropathy. In fact, this disease refers to a large group of conditions that primarily affects joints [13] characterized by a variable degree of inflammation (a medical term used to describe a reaction that produces redness, warmth, swelling, pain and loss of function) and, arthralgia (joint pain). When left unchecked, arthritis can lead to a significant damage of the joint, resulting in stiffness and loss of function.

There are more than 100 different types of arthritis [13] and related conditions with different causes and treatment approaches. This illness can affect one or multiple joints being named as monoarthritis, if only a single joint is involved, oligoarthritis, when two to four are affected, or polyarthritis, if it affects five or more joints.

2.1 Joints

Arthrology is the science that studies the structure, function and dysfunction of the joints.³ A joint, which may also be called an articulation, is an area where different bones connect. Joint's main function is to consent different degrees and types of movement of the skeletal system, allowing it to function as a whole [14].

The number of joints in the human body ranges between 200 and 400 depending on the definition and person's age, since joint's definition may vary according to the bibliographic source and, over time, bones tend to fuse together reducing the number of joints.⁴

Joints can be classified in accordance to several characteristics, being the structural and functional classifications the most important. The classification based on the joint's structure concerns the type of tissue present between the bones while the classification concerning its function focus on the degree of movement allowed [15]. Thus, depending on the type of tissue connecting the bones, the joints can be structurally classified as fibrous joints, when fibrous connective tissue

¹<https://www.spine-health.com/glossary/arthropathy>

²<https://www.arthritis.ca/about-arthritis/what-is-arthritis>

³<https://en.wikipedia.org/wiki/Arthrology>

⁴<https://bodytomy.com/joints-in-body>

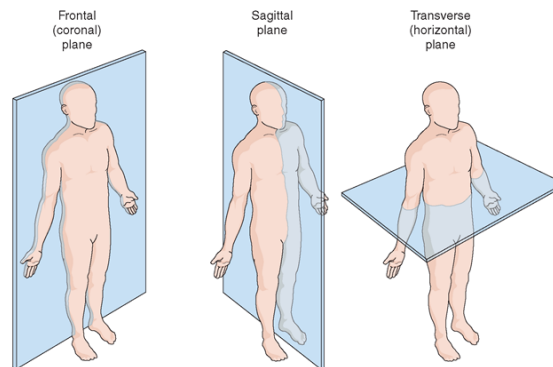


Figure 2.1: Anatomical planes.

unites both bones, cartilaginous joints, if there is hyaline cartilage or fibrocartilage instead, or synovial joints, when the bones are not directly united to each other having a joint cavity filled with a lubricating fluid contained within a fibrous capsule. Synovial joints are the most common type of joints found in the human body. Additionally, joints can be functionally classified as synarthrosis, amphiarthrosis or diarthrosis. A joint is categorized as a synarthrosis when it is immobile or nearly immobile, conferring a strong connection between the articulating bones and being important at locations where they provide protection for organs. An amphiarthrosis joint consents a limited movement while a diarthrosis joint allows free movement, being responsible for the majority of the body movements. All synovial joints are functionally classified as diarthrosis. Depending on the number of anatomical planes where the movement can occur (transverse, coronal, or sagittal; Figure 2.1)⁵, the diarthrosis can be sub-classified as uniaxial (for movement in one plane; e.g. elbow joint), biaxial (for movement in two planes; e.g. knee joint) or multiaxial (for movement in all three planes; e.g. shoulder and hip) [14].

2.2 Types of Arthritis

As previously mentioned, there are several types of arthritis. Here, we will focus on the most common forms: osteoarthritis (OA), which is a more degenerative type of arthritis, and rheumatoid arthritis (RA), which has more inflammation-related characteristics. After OA and RA, fibromyalgia, gout and systemic lupus are the conditions with higher incidence [16]. They all result on different types of joint pain and they have also different etiologies.

2.2.1 Osteoarthritis

OA, also known as degenerative arthritis, is the most common long-lasting joint disorder worldwide. This disorder results from a combination of mechanical and biological processes affecting mainly articular cartilage [8]. Articular cartilage is a rubbery-like tissue present on diarthrodial joints to provide a flat surface for load distribution and a frictionless movement in joint

⁵<http://aspecrowder.blogspot.com/2014/09/movements-of-body.html>

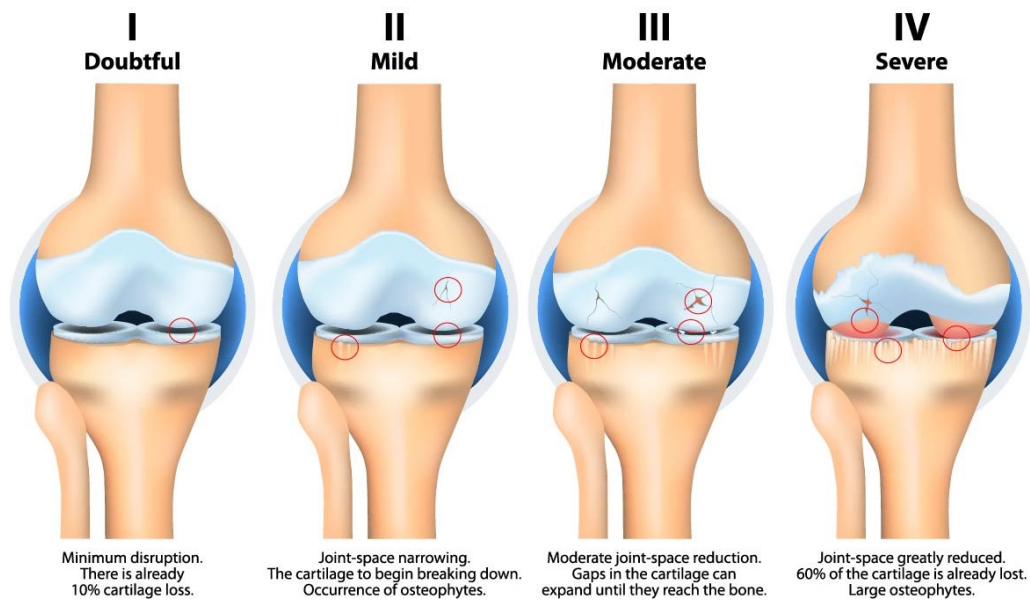


Figure 2.2: OA different stages.

motion [17]. Although OA affects primarily the cartilage, an avascular and aneural tissue that cannot generate pain directly [18], it also affects other structures, leading to an impaired subchondral bone remodelling, formation of osteophytes (bone spurs) and synovitis (inflammation of the synovial membrane), which will trigger pain of the affected joints, hindering their movement [19] (Figure 2.2).⁶

OA usually affects weight-bearing joints,⁷ more frequently the knees, hips and lower back. However, it can also occur on the small joints of the fingers (Figure 2.3)⁸ and the bases of the thumb and big toe.

OA's symptoms may vary according to the affected joint and the stage of the disease. However, the most common symptoms are stiffness and pain that worsens with weight bearing and movement of the affected joint [19]. Other symptoms such as tenderness, limited range of movement and a crackling sensation can be also felt in the injured articulation.⁹

Although OA can occur in people of all ages, it increases with age [20], being more prevalent in women. Other risk factors include obesity (overweight puts an extra-load on the joint and fat tissue produces catabolic proteins),⁶ joint injuries (even injuries that happened a long time ago leave joints more vulnerable to OA development), jobs that involve physical and repetitive labor, genetics and, also, bone deformities.

In OA early stages, the treatment is based on the attempt to regenerate the cartilage [8]. However, most patients seek medical care in a moderate or advanced stage, when cartilage wears out and may break, the subchondral bone thickens and an inflammatory process can arise (Figure 2.2,

⁶<https://www.pthealth.ca/blog/knee-osteoarthritis-my-knees-hurt-do-i-have-oa/>

⁷<https://courses.lumenlearning.com/boundless-biology/chapter/joints-and-skeletal-movement/>

⁸<https://www.arthritis.org/about-arthritis/types/osteoarthritis/what-is-osteoarthritis.php>

⁹<https://www.mayoclinic.org/diseases-conditions/osteoarthritis/symptoms-causes/syc-20351925>



Figure 2.3: OA in finger joints.

IV), causing more pain, joint destruction and eventually, the failure of the joint as a whole [21]. Although the process can not be revoked, symptoms can be managed with some lifestyle changes, therapies, medication and surgery. Pharmacological interventions normally focus on relieving pain, the prominent symptom.

2.2.2 Rheumatoid Arthritis

RA is an autoimmune disease that, unlike OA whose focus is essentially on cartilaginous tissue, mainly affects the joint's synovial lining (Figure 2.4).¹⁰

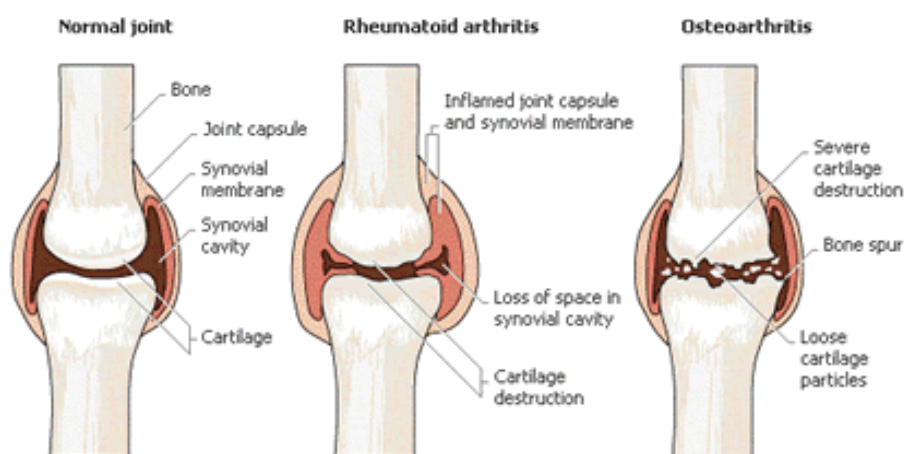


Figure 2.4: Comparison between normal, RA and OA knees.

RA occurs when the body's immune system, which usually functions to defend our body against invaders, mistakenly attacks the coating of the joints leading to a synovial inflammation

¹⁰<https://www.arthritis.org/>

and hyperplasia (swelling) that eventually lead to bone erosion and joint deformity.¹¹ The cause of this immune system's flaw is still unknown, but genetic, environmental and hormonal factors are thought to be involved. In addition to articular involvement, inflammation can also injury other organs such as nerves, eyes, skin, lungs or heart.¹² Thus, once RA affects the entire body, it is considered a systemic chronic disorder [22].

Early RA stages tend to affect primarily the small joints of the hands (Figure 2.5)⁷, feet and wrists, before spreading to the larger joints such as elbows, shoulders, knees and ankles [23]. RA is classified as a polyarthritis, since it spreads to other joints over time, progressing quickly. This disorder is also symmetric, meaning that, if a joint on one side of the body is affected, the other joint on the other side will also be. This feature facilitates its diagnosis.



Figure 2.5: RA in finger joints.

RA signs and symptoms' have waves of incidence, called flares, which are periods when the disease's activity increases and the symptoms become more severe. Flares are alternated with periods of remission.¹³

In early stages, the patients' main symptoms are pain, stiffness and swelling of the joint. Depending on the activity of the disease, some patients may also experience fatigue, a low-grade fever and loss of appetite resulting in weight loss.¹³ As mentioned before, RA also increases the risk of development of problems in other organs, which occurs in around 40 percent of RA's patients.¹⁴

This type of arthritis has its onset typically between the 30 and 55 years old being, like OA, more prone to happen in women [23]. Family history, environmental exposures, smoking and obesity are among the main RA's risk factors.

The goal of RA's treatment is to stop inflammation in order to achieve disease remission. Drugs targeted for RA treatment act to attenuate the symptoms and slow the activity of the disease.

¹¹[https://www.arthritis.ca/about-arthritis/arthritis-types-\(a-z\)/types/rheumatoid-arthritis](https://www.arthritis.ca/about-arthritis/arthritis-types-(a-z)/types/rheumatoid-arthritis)

¹²<https://www.webmd.com/rheumatoid-arthritis/rheumatoid-arthritis-basics>

¹³<https://www.arthritis.org/about-arthritis/types/rheumatoid-arthritis/symptoms.php>

¹⁴<https://www.mayoclinic.org/diseases-conditions/rheumatoid-arthritis/symptoms-causes/syc-20353648>

However, the currently used approaches sometimes flop or yield only a partial response [22]. As a last resort, surgery may be needed to replace the damaged joint.

2.3 Socio-Economic Impact

Arthritis has a tremendous impact on the patient's daily lives, having a relevant socioeconomic impact throughout the world. The burden of arthritis includes economic, social and/or psychological costs to the patient, family and/or society [24]. Arthritis' expenses include costs for adaptive aids and devices, drugs, surgery, and time off at work. In 2015, it was estimated that only the later had an impact of 252 million dollars [16]. For arthritic patients, consequences include living with pain and movement limitations, with consequent reduction of their life quality [25].

OA is one of the most common musculoskeletal disabilities worldwide and, its prevalence and incidence is increasing as the world's population is aging [26], further increasing the need to find effective strategies for this disorder. It is estimated that 9.6% of men and 18.0% of women in the world, aged over 60, have some degree of OA.¹⁵ In accordance with one study of the World Health Organization [27], it is expected that, by 2050, the number of OA patients in the world will be around 130 million.

Considering RA, this type of arthritis affects 1% of the worldwide population [28]. In Portugal, RA is estimated to affect between 0.8%-1.5% of the population, being its occurrence 2 to 4 times higher in women, according to Sociedade Portuguesa de Reumatologia.¹⁶ A study from 2012, authored by Miranda et al. [29], states that the total cost of treating one patient with RA, in Portugal, is about 3415 euros per year, on average.

2.4 Summary and Conclusions

Arthritis is a common misunderstood pathology with no known cure, that has a significant incidence worldwide, yielding a considerable world economic impact [16]. This disorder damages joints, which are structures responsible for the movement of the body, having a tremendous impact on the patients' routines and making daily simple tasks a challenge [21]. In current treatments, there is an evident lack of efficient approaches against arthritis, raising the need for the development of further therapeutic alternatives [22].

¹⁵<http://www.who.int/chp/topics/rheumatic/en/>

¹⁶<http://www.spreumatologia.pt/doencas/artrite-reumatoide>

Chapter 3

Experimental Arthritis and Rodent Gait

This chapter clarifies how rodent gait analysis can be used in the research, diagnosis and treatment of arthritis. Some fundamentals about experimental arthritis and gait parameters will be covered as well as the procedure to detect rodent gait changes.

3.1 Experimental Arthritis

Many of the symptoms of human arthritis may be experimentally mimicked by inducing the disease in animal models. These play an important role in the development of new approaches for the treatment of this disorder and they have been a key to understand the underlying pathophysiology on different types of arthritis and to evaluate potential drugs for clinical use [30]. Therefore, it is important to select the model according to its similarity with the human disease (they should mimic as much as possible the different features of the disease), its capacity to predict the efficacy of agents in humans and its reproducibility. There are several different procedures to induce an experimental animal model of osteoarthritis, ranging from surgical manipulation to the use of chemicals that are injected in the articular space [31]. The injection of monoiodoacetate in the knee is one of these procedures to induce OA, and it results in joint structural deficits and pain symptoms typical of human OA [32]. Although there are many arthritis animal models in the literature, there is not still an ideal one. Each one of them has the ability to mimic different features of the disease and they all have their limitations mainly due to the differences of anatomy, dimensions, biomechanics and underlying physiological processes [33, 34, 35].

Since movement-evoked pain is an early characteristic of arthritis, after the induction of the model, the affected animal modifies its behaviour to decrease pain and to protect the injured limb from load when walking [36]. In this way, the gait's parameters will differ between a normal (not ill) and a diseased rodent, and the analysis of these differences will be an useful tool for scientific research. Rat's gait analysis allows to discriminate between normal and abnormal gait by creating objective quantitative metrics for both, which is important to understand the evolution of the disease and the impact of distinct approaches, namely drugs, on arthritis treatment.

3.2 Rodent Gait Analysis

Gait analysis is the study of the locomotion, a way to assess the dynamic posture and coordination during movement. As already mentioned, the animal's gait analysis is a powerful noninvasive tool used for clinical purposes and research of certain diseases associated with locomotive abnormalities, such as arthritis [37], Parkinson's and Huntington's diseases [38], multiple sclerosis [39], and stroke [40], among others. When performing gait analysis, several gait parameters can be measured and, depending on the field of interest, these parameters may vary. Gait parameters include, but are not limited to, the parameters listed in Table 3.1 (adapted from [41]).

Temporal gait data give information about the timing and synchronicity of foot-strike and toe-off events during the gait cycle, which is defined by two foot-strikes of the left hind limb. Spatial gait data describe the geometric characteristics of the footprint pattern. Other parameters can be representative of forces associated with locomotion [36].

Table 3.1: Gait parameters.

Temporal parameters	Stride velocity
	Cadence or rhythm (number of steps per unit of time)
	Step time
	Swing phase (time interval while each foot is lifted)
	Stance phase (time during which a limb is in contact with the ground)
Spatial parameters	Area of the footprints
	Step length
	Stride length
	Step width (distance between two equivalent points of both feet)
	Toe-out angle
Other parameters	Travelled distance
	Gait phases
	Stride force and momentum
	Pressure

3.2.1 Gait Modifications in an Animal with Arthritis

As mentioned before, gait changes occur when arthritis is induced in an animal which, due to pain, shifts its weight to the non-injured limb, therefore modifying its gait parameters [11]. In healthy rodents, spatiotemporal characteristics are more or less symmetric. Right and left foot-strikes are spaced at approximately 50% of the gait cycle in time and are equidistant in space. On injured animals this is not verified and deviations from the normal pattern occur (Figure 3.1 (adapted from [36])). For an unilateral right limb injury, the left-to-right timing is delayed when compared to the right-to-left, leading to a decrease in the stance time on the injured limb and

consequently, to an asymmetry of the temporal gait pattern (the opposite occurs for a left injury). The spatial gait pattern also becomes asymmetric being verified an increase on step widths and toe-out angles in the injured limb, along with decreased stride lengths. Also, a decrease of the intensity and area of the animal's footprint is reported in the injured limb (indirect measure of pressure) [36].

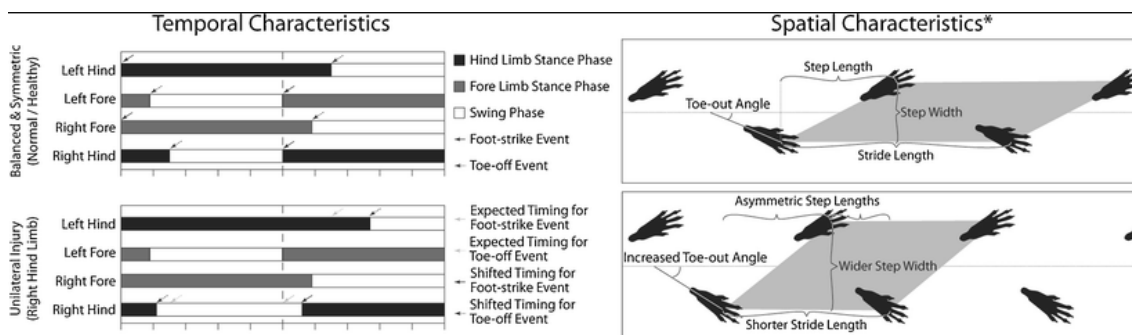


Figure 3.1: Typical spatiotemporal gait sequences for walking rodents. (*Fore limbs were removed in spatial characteristics images for clarity).

3.3 Detecting Rodent Gait Modifications

In the past, when acquiring gait patterns in laboratory rodents was still a challenge, there were manual procedures that consisted, for example, in covering the animal paws with some substance like Vaseline [42] or ink [43], and then letting the animal to walk over a paper to leave its footprints. Nowadays, these approaches are not justified since technological developments have already allowed more practical and automatized solutions to acquire data regarding the animal's gait. Video cameras, pressure and weight sensors are some of the devices currently used to obtain footprint patterns. Regardless of the sensor used, the data can then be processed manually or automatically. There are currently some automatic systems for gait analysis which are not video-based, like the Pressure-sensing WalkwayTM from TekScanTM [44] or the Dynamic Weight Bearing test from Bioseb [45]. However, today's main challenge consists in detecting and quantifying automatically the changes in the walking pattern of the animals recorded in video. Therefore, for the purpose of this dissertation, video-based gait analysis will be the subject matter on the next subsections. These aim to facilitate the understanding of how is a video manually processed to detect rodent gait changes, as well as to review the current state-of-the-art of video-based automated systems for gait analysis.

3.3.1 Manual Video Gait Analysis

The manual video processing involves a large time investment by researchers. They have to choose manually about nine frames where paws are correctly placed at the platform: three pairs of frames of the animal while walking and other three with the animal standing still. The chosen

frames that represent the walking must have only two paws in contact with the surface being almost consecutive frames, and must have the opposite paws placed on the surface meaning that, if the first frame has the right-fore and the left-hind paws, the second must have the left-fore and the right-hind limbs, as it can be seen in Figure 3.2. Standing still frames must have the four paws in contact with the platform (Figure 3.3). In this way, the whole video must be first analyzed becoming a very time-consuming and boredom task.

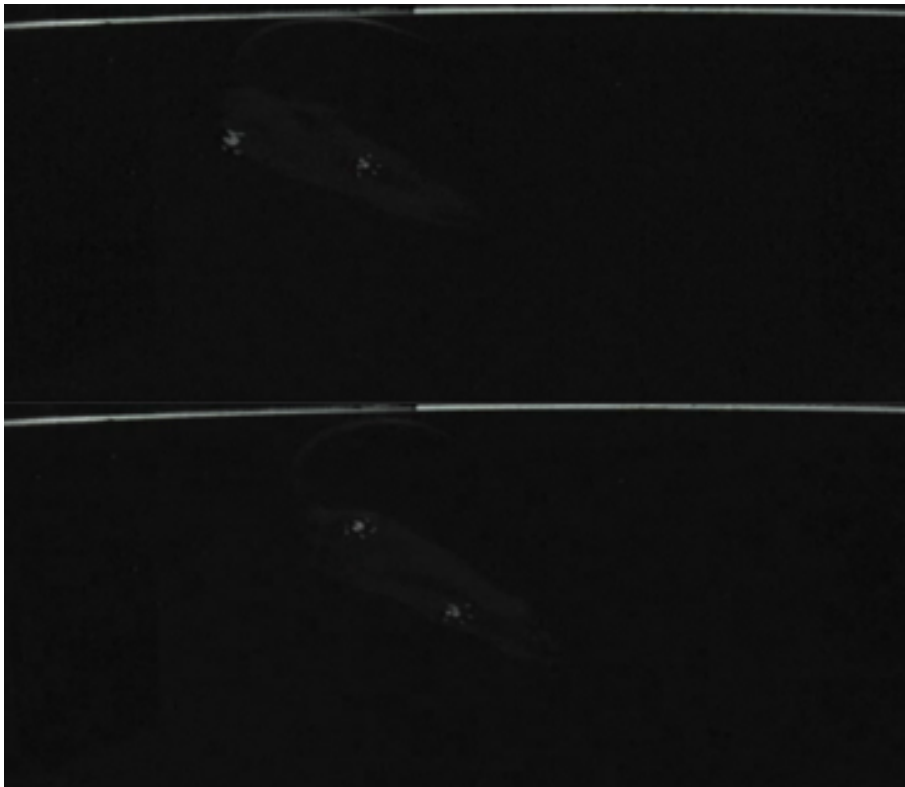


Figure 3.2: Set of walking frames example.



Figure 3.3: Standing still frame example.

After selecting the frames, a software like ImageJ¹ is used to analyze the images. The footprints are manually selected and a threshold value is defined by the user, above which the number and intensity of pixels are quantified, allowing the comparison of the area, mean intensity and pressure applied by each paw. This analysis should preferably be done in a poorly lighted location so that the lighting does not interfere with the selection of the threshold and so the analysis is consistent in between the different videos of the experiment, regardless of the user.

3.3.2 Video-Based Automatic Systems for Gait Analysis

There are already some commercially available automatic video-based systems for gait analysis. These systems resort to video processing, machine learning and computer vision techniques to automatically detect changes in rodent gait pattern. They include, but are not limited to, DigiGaitTM, CatWalk XT, MouseWalker, TreadScan, RunwayScan and GaitScan. Here, their characteristics as well as inherent advantages and limitations will be gathered and reviewed.

The DigiGaitTM system incorporates an automatic software of digital imaging and video processing, developed by MouseSpecifics, that quantifies mice, rats and guinea pigs' gait. A high-speed, 147 frames-per-second digital video camera, placed under the treadmill belt records the animal walking. DigiGaitTM videos are manually cropped and imported to the software to be automatically analyzed [46]. The software is able to identify the paws automatically, determining spatial and temporal parameters, among which force development, loading distribution, strength, balance and coordination. The animal's walking or running can be voluntary or forced by setting a speed for the treadmill (Figure 3.4) [47].

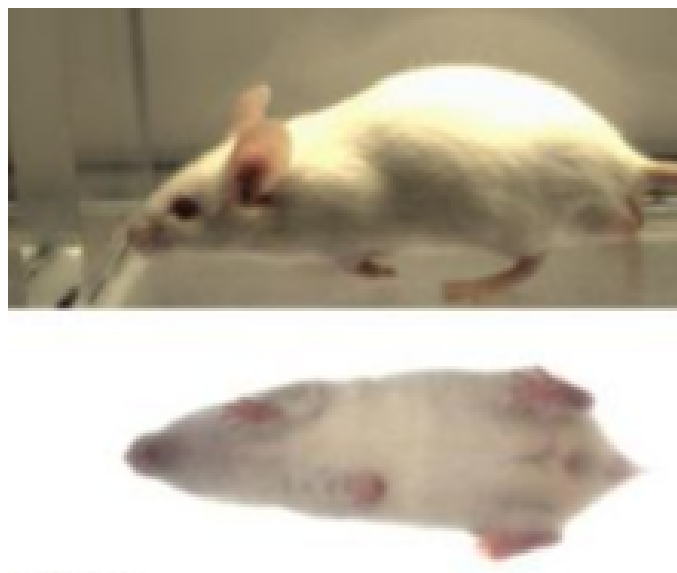


Figure 3.4: DigiGaitTM system.

¹ available at: <https://imagej.nih.gov/ij/>

The CatWalk XT system, developed by Noldus, also records the animal from below while it freely traverses a glass walkway. However, this system requests Illuminated Footprints TechnologyTM, which captures the rodent paw prints where the paws actually touch the glass, therefore reducing the number of falsely detected areas of the paws. The light that illuminates the footprints is internally reflected, with the exception of those areas where the animal contacts the glass plate. The high speed camera captures these illuminated areas and the software analyses the data. The CatWalk XT software is capable of computing several gait parameters, ranging from individual footprints characteristics to distances and time relationships between the footprints (Figure 3.5) [48].



Figure 3.5: CatWalk XT system

The MouseWalker's hardware is also based on the reflection of light within a transparent plate through an optical effect (Figure 3.6) [49]. The software is available as open source and it is written in Matlab. It allows the location of the animal and the extraction of a large set of gait parameters from freely walking rodents, such as step patterns, footprint positioning, inter-leg coordination, and footprint contact parameters [49].

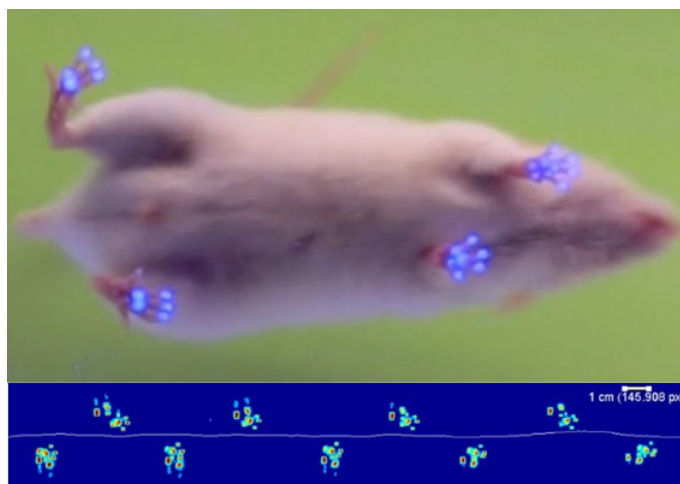


Figure 3.6: MouseWalker system.

The TreadScan [50], RunwayScan [51] and GaitScan [52] are all from CleverSys. The TreadScan hardware is composed of a transparent treadmill belt and a video camera able to record 100 frames-per-second [46]. The video is captured also with an underside view of the animal, which records the animal's footprints as they walk/run on the treadmill. The TreadScan software is able to process the obtained videos and extract some important parameters to analyze the gait, such as stance time, swing time, total stride time, stride length, foot contact area size, and foot pressure, among others. The RunwayScan is very similar to TreadScan but instead of having a treadmill it is composed of a clear walkway. The video is either acquired in the dark or with normal light conditions. The parameters given by RunwayScan are almost the same as the TreadScan. Finally, the GaitScan system combines functionalities of the TreadScan and RunwayScan into a single product (Figure 3.7) [52].

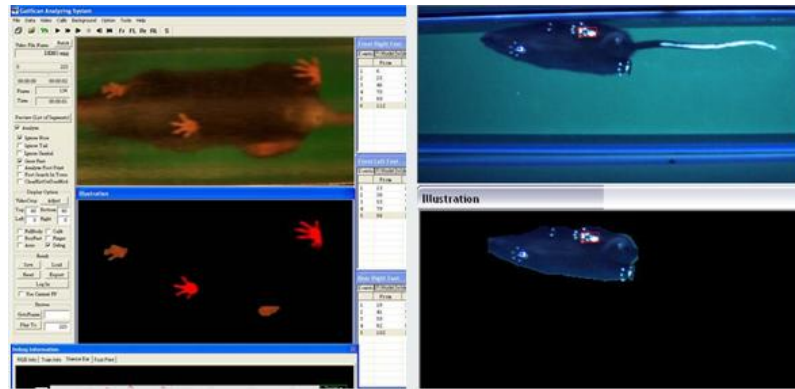


Figure 3.7: GaitScan interface.

Table 3.2: Commercially available video-based automatic gait analysis systems.

System Name	Company	Type of Platform (Trajectory)	Type of Movement (Velocity)
DigiGait TM [47]	MouseSpecifics	Treadmill (forced)	Free or forced
CatWalk XT [48]	Noldus	Walkway (forced)	Free
MouseWalker [49]	Columbia University (NY)	Walkway (forced)	Free
TreadScan [50]	CleverSys	Treadmill (forced)	Forced
RunwayScan [51]	CleverSys	Treadmill/Walkway (forced)	Free
GaitScan [52]	CleverSys	Treadmill (forced)	Free or forced

A summary of the identified gait systems can be found in Table 3.2. However, in spite of the availability of these automated systems, there are still inconsistencies between them, and researchers often select different gait pattern descriptors to report in the literature, according to the

review of gait analysis methods presented by Lakes and Allen [11], opting for the development of a customized system for the acquisition of the videos and analyzing them by the manual procedure described before (Subsection 3.3.1). This happens due to the inflexibility of the software packages presented, which do not allow the users to extract additional and unanticipated metrics. It should also be noted that all the systems have a forced trajectory (presenting a corridor type platform) and some even impose a speed to the animal which can mask subtle spatiotemporal changes by inducing stress to it. Measurements under these conditions vary significantly from open space measurements [11]. Moreover, all mentioned systems are very costly, deterring most individual laboratories from acquiring them.

3.4 Summary and Conclusions

Animal's gait analysis is a powerful noninvasive tool used to understand behavioral alterations in preclinical arthritis models [37]. The process begins by inducing arthritis in the animal, which will mimic some characteristics of the disease, resulting in changes in the animals' gait pattern as animals tend to protect the injured limb by shifting the weight to the non-injured paw [36]. In order to understand the impact of certain therapeutic approaches for arthritis, namely the efficacy of drugs, these changes can be quantified and compared to normal gait parameters and to parameters of arthritic animals prior to the beginning of the treatment.

Rodent gait modifications can be detected through video analysis, by a manual or automated procedure. The manual observation and quantification of the gait changes is a laborious and time-consuming intensive task that is prone to high user variability. Currently, there are some systems capable of detecting and identifying the animal's limbs and that automatically compute spatial, temporal, kinematic and dynamic gait parameters [36]. However, they are often expensive closed solutions [11] and some of them greatly limit the area where the animal can walk and/or force it to move at a certain speed on a treadmill, inducing an external pressure to the animal, which may have influence on the results. In order to make the systems more robust and flexible, improvements at the algorithm level using computer vision techniques for mouse tracking will be necessary.

Chapter 4

Object Tracking

Object tracking is one of the most important tasks in computer vision systems. It fuses many fields like image processing, pattern recognition, artificial intelligence and automatic control [53], and it has a broad area of applications which includes, but is not limited to, motion-based recognition [54], automated surveillance [55], traffic monitoring [56], human-computer interaction [57], medical imaging [58] and vehicle navigation [59].

Tracking can be summarized as the task of estimating the position of a moving object in different frames of a video and segment it from the background. Besides its position, a tracker can also provide other information about the object such as orientation, shape, area and others [60]. Although the ability to track a moving object has been evolving drastically over the last decades, this task remains a challenge, mainly due to the abrupt motion of the object, illumination changes, noise, camera motion, background movements, partial or full object occlusions and non-rigid structures of the object [61]. Beside these challenges, object tracking algorithms have to face the trade-off between speed and reliability, meaning that they have to be robust enough to overcome the aforementioned challenges but, at the same time, they can not be too complex once systems have limited computing power and memory [62].

Concerning object tracking, different scenarios can be found requiring different approaches:

- *Online vs Offline Tracking:* While online trackers only have access to current and past information, offline models can use the entire video, meaning that information from both past and future can be used. It is expected that offline trackers give a better performance since they have access to more information [63]. In this work, it is possible to use both online and offline methods since real time tracking is not a requirement.
- *Single vs Multiple-Object Tracking:* While a single-object tracking identifies only one object in the video sequence, multiple-object tracking identifies several objects. For example, considering a video sequence of the dataset (see Chapter 5), it can be identified only the mouse (single-object tracking) or the paws (multiple-object tracking).
- *Static vs Moving Camera:* In object tracking, moving cameras are a bigger challenge when compared to static ones. This is due to the fact that, in moving cameras, the movement in

both foreground and background in every consecutive frame is larger when compared to static cameras, becoming more difficult to track the object. The focus of this thesis is on static camera where the background is also static existing just small variations due to noise.

4.1 Tracking-by-detection

All tracking algorithms need to detect the target object at some stage [60]. In tracking-by-detection approaches, detection and tracking are performed with a clear distinction. The flow diagram of the tracking-by-detection approach is represented in the Figure 4.1. The general idea of this methodology consists in first detect the target, then classify it and, by lastly, keep it on the tracker [64]. Since a video is no more than a sequence of images, each of which called a frame, displayed with a high frequency, all image processing techniques can be applied to each frame to perform the aforementioned tasks. In a video sequence, there are two main sources of information for object detection and tracking: visual features and motion information [65].



Figure 4.1: Flow diagram of tracking-by-detection.

4.1.1 Object Representation

Some detection and tracking algorithms commonly represent the target object somehow becoming important to understand which object representation can be suitable for the concerned detecting/tracking algorithm and object. As it can be seen in Figure 4.2 [60], objects can be represented by their shapes and appearances in several ways: by a single point, the centroid (a), or a set of points (b); by primitive geometric shapes, such as a simple rectangular (c) or an elliptical (d) shape; another way, is to resort to articulated shape models (e) or to the object skeleton (f); complex nonrigid objects may also be represented through their contours (g, h) which define the boundary of the object limiting its silhouette (i). Typically, the object representation is chosen according to the application domain.

4.1.2 Object Detection

As mentioned before, tracking-by-detection requires a step to detect a specific object in a certain scenario [66]. Over the last years, a large amount of object detection algorithms have been proposed. Some of them use temporal information of the video sequence to distinguish the foreground from the background, others slide a window across the image to try to find the object and, others extract interesting points of the image and classify their neighborhood rather

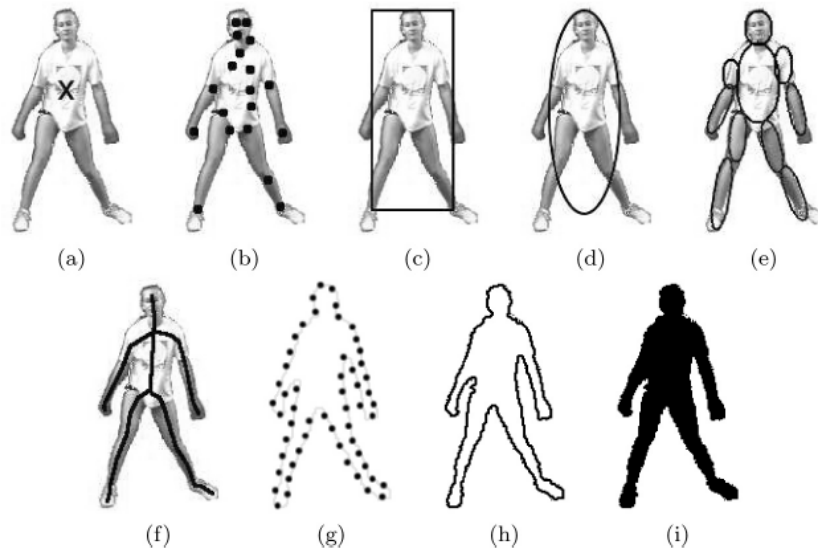


Figure 4.2: Object Representation.

than analyzing all of the possibilities within the image domain [67]. Although this large variety of research in object detection, this task remains a challenge in most of applications, mainly due to the presence of noise, low luminosity, low contrast and intensity inhomogeneity [68]. In this section, some of the object detection algorithms will be covered to better understand the available tools to perform this task.

4.1.2.1 Image Segmentation

Image segmentation consists in the partition of the image into perceptually similar regions with homogeneity characteristics being commonly used to detect the position of an object. This task aims to simplify and modify the representation of an image, making it more meaningful and easier to analyze it, by extracting or contouring regions of interest. Although this is an unconscious activity for the human eye, in computer vision, image segmentation is a logically non-trivial and computationally costly task. In this field, segmentation is one of the most important tasks, being the first step in almost every pattern recognition process [69]. Every segmentation algorithm addresses two problems: first, define the criterion for a good partition and then, find the best method in order to achieve the most efficient partitioning [60]. The most common segmentation techniques include threshold, edge based, region based, clustering, watershed transformation, partial differential equation (PDE), and artificial neural network (ANN) segmentation methods [70] (Figure 4.3).

The simplest way to segment an image is simply to choose a single threshold value (T) above which the intensity values turn to one (or to 255, depending on the image type) and below to zero, transforming the image into binary. However, multiple threshold can be considered instead,

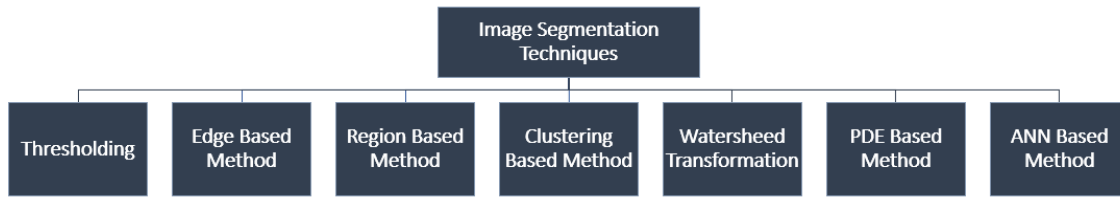


Figure 4.3: Image segmentation techniques.

resulting into the partitioning of the image into multiple regions (Figure 4.4) [71]. There are basically two types of thresholding: global and adaptive. The first is applied to images with a clear distinction in the intensity distribution of objects and background pixels. In this technique a single threshold value is used in the whole image [72] and, one of the most commonly used methods that tries to find the best overall value for T is the Otsu method, first presented by Otsu [73]. However, in images where lightning conditions vary within it, global thresholding fails. In such cases, a better approach is to use an adaptive thresholding where the value of T varies according to the image region, thus using multiple threshold values. Threshold techniques usually achieve good results when there is a good contrast between the object and the background. Currently, threshold methods are combined with others, benefiting from many imaging features such as intensity values, gradients, transform and statistical features.

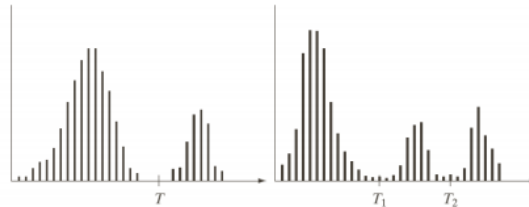


Figure 4.4: Single (left) and Multiple (right) Thresholding.

Edge segmentation methods are based on the abrupt variation of the intensity values. These techniques locate the edges, usually by using a first-order derivative characteristic such as the directional maxima of the gradient magnitude or, by finding zero crossings in the second-order derivative expressions. An example of the later is Laplacian of Gaussian (LoG) [74]. The most commonly used edges detectors are Robert's [75], Sobel [76] and Canny [77] operators.

Region based methods consist on the division of the image into several homogeneous areas. They can be divided in region split and merge or region growing approaches. The first consists in a hierarchical bottom-up segmentation method and it starts by the partition of the image into several regions which are iteratively merged according to similarity constraints [78]. On the other hand, region growing starts by seeding pixels, and examining their neighboring pixels in order to determine if those pixels will be added to the region or not [79]. These methods require the usage of a similarity threshold value.

The method of clustering can be defined as “the process of organizing objects into groups whose members are similar in some way” [80]. A cluster is, therefore, a homogeneous region containing pixels with similar characteristics, differing from pixels belonging to other clusters. The most common clustering algorithms are K-means and Fuzzy C-means [81], described in the Table 4.1. Both of these methods are non-hierarchical partitioning clustering algorithms. K-means was introduced by MacQueen *et al.* [82], and later improved. Fuzzy C-means is no more than a K-mean extension and it was presented by Dunn [83]. Contrarily to K-means, Fuzzy C-means does not need an initial parameter to represent the number of clusters and it gives better results for noisy clustered datasets [81].

Table 4.1: K-means and Fuzzy C-means Algorithms.

Clustering-based Method	Algorithm Description
K-means	<ol style="list-style-type: none"> 1) Centroids of clusters are chosen randomly. 2) Distances between data points and cluster centroids are calculated. 3) Each data point is assigned to the cluster whose centroid is closest to it. 4) Cluster centroids are updated. 5) Distances from the updated cluster centroids are recalculated. 6) If no data point is assigned to a new cluster the run of algorithm is stopped, otherwise the steps from 3 to 5 are repeated.
Fuzzy C-means	<ol style="list-style-type: none"> 1) Initialize the fuzzy partition matrix randomly. 2) Calculate prototype vectors. 3) Calculate membership values. 4) Compare the membership matrix in the one of the previous iteration. 5) If their difference is smaller than a defined convergence value then stop else return to the step 2.

Another segmentation method, is watershed transformation. Watershed framework considers the gradient magnitude of an image a topographic map, where each pixel’s intensity represents a height. Among many, watershed-by-floating algorithm can be used to compute the watershed. This algorithm assumes that a source of water is placed in the darker zones (zones with lower height) and, as the water rises, areas where the floodwater from different basins meet are identified. To perform segmentation by watershed, many concepts of threshold, edge and region based segmentation techniques must be incorporated [84]. Several improvements have been verified in this algorithm [85, 86] becoming a widely implemented algorithm in the medical field: in Hasan *et al.* [87], the traditional watershed is used to perform brain tumor segmentation while in Patel *et al.* [88] is applied to segment the knee cartilage.

PDE based methods are widely used for solving many problems in computer vision, mainly

related to image processing and segmentation. The first is addressed in the work of You and Mostafa [89] by using a fourth order PDE method to remove noise and the second by enhancing the edges, using a second order PDE method [90]. They are appropriate for time critical situations once they are fast segmentation methods. One of the most famous method of PDE used for image segmentation is active contour model, commonly known as snake model. This method delineates an object outline by minimizing the sum of the internal with the external energy associated to the contour, which modifies its shape actively approaching to the desired contour [91].

ANN based methods simulate the learning process of the human brain. The approach transforms the segmentation problem into a classification decision being used to distinguish what belongs to the object and what does not. They are a type of semantic segmentation, a high-level task that paves the way towards complete scene understanding [92].

Standard Neural Network (NN) (Figure 4.5)¹ is a type of learning algorithm which forms the basis of most deep learning methods. A NN is composed by a large number of processors called nodes (circles in the figure) connected to each other with a particular weight (represented by the arrows in the figure). A NN has typically 3 types of layers: the input layer, the hidden layers and the output layer.

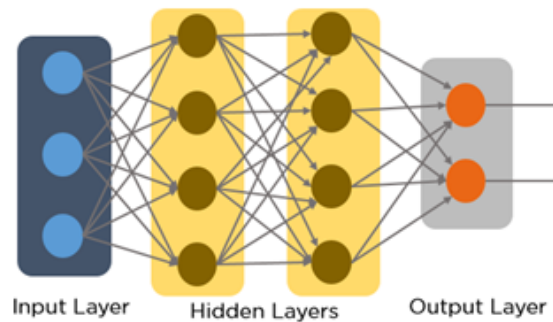


Figure 4.5: Neural Networks representation.

Convolutional neural network (CNN) (Figure 4.6)² is a class of deep, feed-forward ANN and it is one of the most successful type of models in image analysis.

A convolved feature map results from the convolution operation between the input data (feature map) and the convolution kernel (filter). After having the convolved feature map, non-linearity is introduced to the network, resulting in a rectified feature map. A dimensionality reduction is then performed obtaining a pooled feature map which is flattened into a long linear vector. This flattened matrix is the input of the fully connected layer being the output, the object classification [93]. CNNs have been widely used object segmentation, classification and recognition. Algorithms such as YOLO (You Only Look Once) [94], region CNN (R-CNN) [95], fast R-CNN [96], faster R-CNN [97] and others, where a single CNN simultaneously predicts multiple bounding boxes and

¹<https://www.quora.com/What-are-artificial-neural-networks-1>

²<https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks>

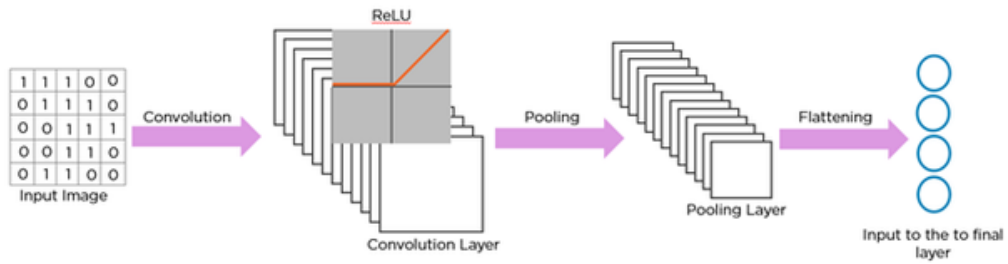


Figure 4.6: Convolutional Neural Networks representation.

class probabilities for those boxes, have been proposed. These unified models are faster, simpler and achieve higher accuracy values for object detection.

4.1.2.2 Point Detector

An object can be detected by finding some of its well-defined interesting points. These points can be a line ending, a maximum on a curve, a local intensity maximum or minimum, or even a corner, a point where two lines intersect. This subsection meets the edge based methods used for image segmentation. In literature, sometimes the term corner is interchanged with the term interest point, therefore, some methods classified as corner detectors can, in practice, be applied to detect interest points in general [61]. Those so-called interesting points should be resistant to illumination changes and camera viewpoint.

A wide variety of interest point detectors exist in the literature [98]. One of the oldest and best known is Moravec's Detector which calculates the difference between the grayscale intensity of a window centered in a pixel and windows shifted in several directions. This algorithm detects points where a large intensity variation occurs between those windows by using a defined threshold [99]. For that, the sum of squared differences (SSD) can be used as similarity measure. Another point detector, Harris detector or Harris-Stephens detector, was presented by Harris *et al.* [100], where the authors exposed the problems of Moravec's operator: it has an anisotropic response because the algorithm only considers shifts at every 45 degree, it outcomes a noisy response due to be binary and rectangular shape of the window and it responds too readily because it only considers the minimum of the difference between windows. Harris detector overcomes these problems by using a local auto-correlation matrix. This method aims to find the direction of the fastest and lowest variation for feature orientation using a Gaussian function to weight the derivatives inside the window. A point of interest is characterized by a large variation in all directions. The points of interest are detected if the auto-correlation matrix has two significant eigenvalues. Later, some variations on the Harris method occurred: the Shi, Tomasi and Kanade detector [101], a faster approach that considers only the minimum eigenvalue; Hessian (Hessian-Affine) point detector [102] combines interest points from several scales in a pyramid, with some iterative selection criteria and a Hessian matrix.

Another method, presented by Lowe [103], is scale invariant feature transform (SIFT). This approach transforms the image into scale invariant coordinates relative to local features. The SIFT algorithm is divided in four-steps: in the first, a scale-space filtering occurs by finding blobs in various sizes within the image. This is performed by using LoG with various σ values (the σ acts as a scaling parameter). In this way, it is possible to find the local maxima across the scale and space. As LoG operation can be computational costly, a Difference of Gaussians (DoG) can be introduced instead. After having the images in the feature domain, the second step is performed. Here, the location of each potential key point is updated by interpolating the intensity values using neighboring pixels. In the third step, the low-contrast key points and edge key points are eliminated. Finally, an orientation is assigned to each remaining key point, based on the scale, magnitude of the gradient and direction of its neighborhood. SIFT method proved to be more resilient to image deformations [104].

4.1.2.3 Supervised Classifiers

The detection of an object can be performed using supervised-learning mechanisms that require prior knowledge of what the output values for the samples should be and map the inputs to those outputs through a learning process. In other words, supervised learning requires a labeled training data set [105]. This type of learning can be grouped into regression and classification problems. In the first, the output is a continuous value and in the later, the output is in the form of a class label. When designing a supervised learning algorithm, the main aspects that should be considered are the complexity of the model and the bias-variance trade-off. In the training process, for an accurate object detection, the selection of the object's features plays an important role, since they must be able to discriminate one class from the other: the object from the background. In addition to the features presented in the subsection 4.1.3, the area, the orientation and the appearance of the object are features commonly used.

Common algorithms in supervised learning used to object detection include, but are not limited to, adaptive boosting and support vector machine (SVM). The former, AdaBoost, is a popular boosting technique which combines multiple “weak classifiers” into a single “strong classifier” [106]. The first step on the training phase is to do an initial distribution of weights. The algorithm searches for the classifier that gives the least error and then, the weights associated with the miss-classified data, are increased, encouraging the search of other classifier that performs better on this data [60]. In the context of object detection, Viola *et al.* [107] used this framework to detect pedestrians using patterns of motion (temporal) and appearance (spatial).

SVM is a linear classifier used to cluster data into classes. Considering two classes, the SVM constructs a hyperplane that maximizes the distance between them. Despite its linearity, SVM can also be used as a nonlinear classifier by projecting the input feature vector into an higher dimensional space which is likely to be separable. This operation is called the kernel trick [108]. In the context of object detection, this algorithm is used to distinguish the object class from the background. An example of application is the detection of pedestrians [109] and faces [110].

4.1.2.4 Background Subtraction

Background subtraction (BS) approaches have been widely used for tracking moving objects in many computer vision applications, mainly in static camera cases once they assume that the observed video sequence is made of a static background in front of which, moving objects, referred to as the foreground, are observed. BS methods are often seen as a "quick and dirty" way of detect objects in motion [111]. BS major challenges include lightning variations, camera shakes, and changes in the background over time [112]. The main distinction between BS techniques relies on the fact of how the background is modeled and which distance metric is applied [111]. The process flow chart of BS techniques is illustrated in Figure 4.7 (adapted from [113]) and their general expression is shown in Equation 4.1.

$$X_t(s) = \begin{cases} 1 & \text{if } d(I_{s,t}, B_s) > \tau \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

where X_t is the motion mask at time t , τ is the threshold to set the maximum distance (d) between the intensity at time t and pixel s ($I_{s,t}$), and B_s , the background model at pixel s .

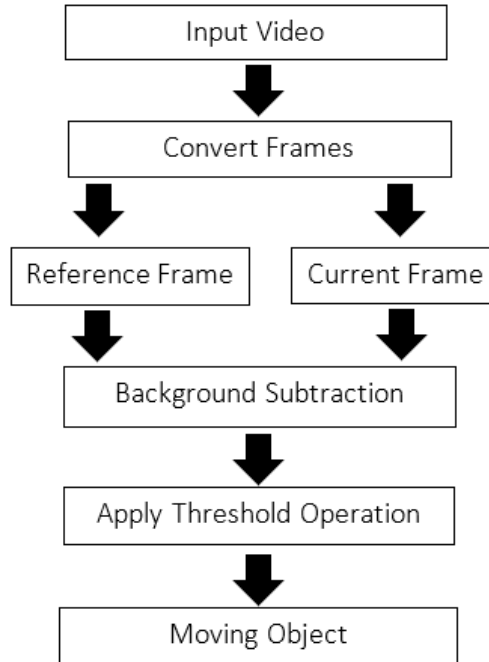


Figure 4.7: Flow Chart Of Background Subtraction Model.

A simple approach to model the background, presented by Heikkila *et al.* [114] in order to cope with lighting fluctuations, is to integrate the new incoming information into the current background image through the following expression:

$$B_{s,t+1} = (1 - \alpha)B_{s,t} + \alpha \times I_{s,t}, \quad (4.2)$$

where α is an adaptation coefficient varying from 0 to 1. The foreground object can be detected by thresholding any distance function which can be the difference between the pixel intensity and the background model in gray-scale or the sum of the differences (or the maximum difference) of each channel red, blue and green individually [111].

Frame differencing (FD) is another approach used to detect an object in motion [115]. In this method, the previously frame is used as the reference frame and a threshold is applied to the difference image obtaining information about every moving pixels. However, only parts of moving objects are detected and, if the object stops, its detection can not occur anymore. In the work presented by Kartika *et al.* [116], FD is applied together with an adaptive threshold method which uses as threshold the average of the intensity of neighboring pixels.

In the study of Wren *et al.* [117], each pixel of the background is modeled following a Gaussian distribution $N(\mu_{s,t}, \Sigma_{s,t})$ where $\mu_{s,t}$ stands for the average background color and $\Sigma_{s,t}$ for the co-variance matrix at pixel s and time t . The noise is assumed to be coming from a zero mean normal distribution and the parameters of the model are updated by using adaptive filters. However, sometimes pixels have values often with more complex distributions needing more elaborated models. For this reason, Stauffer and Grimson [118], proposed a Gaussian Mixture Model (GMM) to model each pixel using, as the name suggests, a mixture of Gaussians. The pixel is considered to belong to the background when at least one Gaussian model comprehends its pixel value with enough consistent and evidence. This is a popular and promising technique to estimate illumination changes and small movements in the background, however, it faces a trade-off problem concerning the learning rate, α , a value that varies from 0 to 1. The speed at which the model adapts to changes is given by the inverse of the learning rate and, when a low value of α is considered, a large amount of false positives is detected once the model is not capable to adapt to the change speed of lighting conditions. On the other hand, if the model adapts to quickly (higher value of α), slowly moving foreground pixels will be seen as background, yielding to a higher false negative rate. To overcome these problems, Elgammal *et al.* [119] presented a non-parametric approach able to handle arbitrary densities by using Gaussian kernels (also termed Parzen-window) for modeling each background pixel. A more sophisticated approach was proposed by Mittal *et al.* [120], which is based on a Variable Bandwidth Kernels.

A different approach, proposed by Haritaoglu *et al.* [121], consists in a bimodal distribution for background variation in which, each pixel is represented by three values: its minimum m_x and maximum M_x intensity values and the maximum intensity difference d_x between consecutive frames obtained from the training period. Once this approach uses an intensity based background model, lightning changes can easily trigger false positives.

The Bayesian approach proposed by Tuzel *et al.* [122], preserves the multimodality of the background model by using a combination of multi-layered Gaussians to model each pixel. The authors use a recursive Bayesian estimation to update the background parameters.

In the work of Kim *et al.* [123], a codebook (CB) BS algorithm is proposed without making parametric assumptions. This algorithm constructs a background model based on a training sequence. For each background pixel, the model assigns a CB composed by one or more codewords

which are representative of the likelihood of a pixel being described by a certain color. For every pixel of the current frame, its distance from codewords is computed to obtain the feature vector and, if it matches with the CB, it means it belongs to background. Other CB approaches differ on how the CB is built. By combining the previous mention algorithm with the shadow suppression model, Doshi *et al.* [124] introduced the “Hybrid Cone-Cylinder” solution. In CB approaches, false positives caused by lightning variations can be reduced by considering shadows brightness shifts and real foreground moving objects chroma shifts. However, this later assumption is too restrictive for some scenes producing a large number of false negatives.

A breakthrough, presented by Candès *et al.* [125], considers the data matrix a sum of a low-rank component with a sparse component, stating that it can be decomposed into them. The low-rank matrix can be then representative of the background, being able of comprising small fluctuations on it. On the other hand, the sparse matrix can formulate the objects in motion which can be seen as outliers. Notice that a third unstructured component is considered only to represent the noise or errors. This study uses a solving convex program, called principal component pursuit (PCP), to solve the robust principal component analysis (RPCA), a pioneer work of De La Torre and Black [126]. Up to nowadays, many are the researches that keep attempting in finding a more efficient RPCA solution [127], with a higher convergence rate, a greater capacity to handle a larger amount of data, with less assumptions and with an optimal outlier tolerance. The first study presenting a non-convex method to recover the low-rank matrix (the background) was one of Netrapalli and Niranjan, [128]. Under the same assumptions as the work presented by Candès *et al.* [125], it was proposed an alternating projection (AltProj) that showed faster results. In Yis *et al.*'s work [129], another solution is presented by using a gradient descent (GD) on the matrix factorized space, known as RPCA-GD. A similar setting called "Nearly Optimal Robust Matrix Completion (NO-RMC)" was introduced by Cherapanamjeris *et al.* [130], however, this approach assumes that the image size is approximately the same as the total number of images which can be a strong assumption for video data. Narayanamurthy and Vaswani [131] presented a recursive projected compressive sensing algorithm, called “Nearly Optimal Robust Subspace Tracking (NORST)”, to solve RST under weakened standard RPCA assumptions. This online algorithm, although having a lower memory complexity, requires two extra assumptions: a lower bound on outlier magnitudes and a slow subspace change. According to Vaswanis *et al.* [127], this later is the one that presents a faster and better performance.

Some researchers have been applying deep neural networks (DNN) to update the background model: while Rafiques *et al.* [132] implement a Restricted Boltzmann Machine (RBM), Braham and Van Droogenbroeck [133] opt to use a CNN approach.

4.1.2.5 Optical Flow

Optical flow is one of the oldest but still most active research domains in computer vision for motion detection. This technique attempts to detect the movement of an object by assigning to each pixel of the current frame a flow vector carrying information about the direction and the

apparent velocity of motion [134]. For a given frame t , a pixel at location (x,y) with intensity $I(x,y)$, will have moved by δx , δy between the two image frames, δt (Equation 4.3).

$$I(x,y,t) = I((x + \delta x), (y + \delta y), (t + \delta t)) \quad (4.3)$$

In order to determine δx and δy , two assumptions are commonly made: 1) the brightness of any object point is constant over time (constant luminance); 2) nearby points move in a similar way (the velocity smoothness constraint) [135]. An overview and comparison of current popular optical flow methods and models can be find in the work of Sun, Roth and Black [136].

This method has a wide range of applications: in medical field it is used to estimate the blood flow [137], organs deformation [138], cell deformation [139] and tracking [140]. Optical flow can be applied in static or moving cameras and in static or moving backgrounds. However, as it has to compute the flow vector of every pixel in every frame, it becomes a very complex and time-consuming approach. This motion detection technique can also be used in background modeling and consequently, in background subtraction (see previous subsection) [141].

4.1.3 Object Classification

Object classification is often mixed or confused with object detection. Object classification is used in the case of two or multiple objects in a scene that need to be distinguished [142]. This step occurs after the detection of the object [64] and it requires the usage of a classifier, an algorithm that takes a set of features that characterize objects and uses these characteristics to determine the class of each object. This classifier receives a list of regions of interest that are likely to contain the object target and classify them aiming in minimizing the number of false positives and false negatives. There are many features that can be used for object classification techniques such as color, shape, texture and motion. Selecting the accurate features plays a critical step in the classification task.

In the work of Singh *et al.* [143], color features are used to perform real-time detecting and tracking. Although color based methods are not always the best choice, once illumination can vary along time, its low computational cost makes color a desirable feature to exploit when appropriate. There are several color spaces: the RGB (red, green and blue space) is not a perceptually uniform meaning that the difference between the colors in the RGB space does not correspond to the color differences perceived by the human eyes unlike the YUV (Y is a luma component and UV two chrominance components) and the LAB (where L stands for lightness and A and B for the green–red and blue–yellow color components) which are both perceptually uniform. HSV (Hue, Saturation, Value) is an approximately uniform color space. However, all of these uniform or approximately uniform color spaces are sensitive to noise.

The texture of an image region can provide an alternative visual cue to indicate the surface material and consequently lead to higher level classification. Texture is a measure of the intensity variation of a region quantifying properties such as smoothness and regularity. Compared to color

features, this characteristic is less sensitive to illumination variations and shadows requiring, however, a processing step to create the descriptor which can be of several types [60]. Gabor [144] first characterized texture as a set of functions described by spatial frequency and orientation. In the work of Leung and Malik [145], a small, finite vocabulary of micro-structures, which the authors called 3D textons, is built. Associated with each texton there is an appearance vector, defined as a discrete representation of a set of linear Gaussian derivative filter kernels that are convolved with the input image. In the work of Ojala *et al.* [146], a multiresolution gray scale and rotation invariant texture classification with local binary patterns is proposed. Local binary pattern is a local texture descriptor based on the signs of differences between neighbour pixels and central pixels [147]. Numerous variants of local binary patterns can be found in the literature [148].

Non-rigid articulated objects can be classified by their motion by analyzing features such as rigidity and periodicity of moving entities. The motion features provide temporal information and can be computed based on the difference between frames or via optical flow which is used to detect moving areas [67].

For a further comprehension about other features that can be extracted and used for object classification, it is suggested to read the work of Bouwmans *et al.* [148] where an intensive review regarding feature concepts is presented.

4.1.4 Object Tracking

The aim of tracking is to find the object target's correspondence position over the sequence frames in order to construct its trajectory [67]. Over the past few years, tracking algorithms have been extensively studied, showing a progressive evolution. In between different tracking approaches, it can be found point correspondence, kernel and silhouette tracking (Figure 4.8 [60], (a), (b) and (c), respectively). In either tracking method, object's representation occurs using shape or appearance models [60].

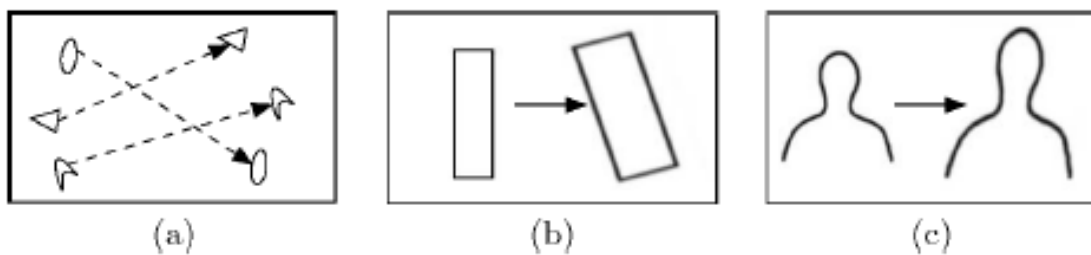


Figure 4.8: Different tracking approaches.

4.1.4.1 Point Tracking

An object can be represented by a point and its tracking can be given by the correspondence of those points in between frames. Point tracking method is based on the use of discrete features

and it can be divided in two categories, namely deterministic and statistical depending on the method used for finding point correspondence [149]. In the first category, the matching of points in different frames uses different motion constraints, as it can be seen in Figure 4.9 [60] such as proximity (a), maximum velocity (b), small velocity-change (c), common motion (d), and rigidity (e). Statistical approaches use, in turn, the state space approach to model its position, velocity and size.

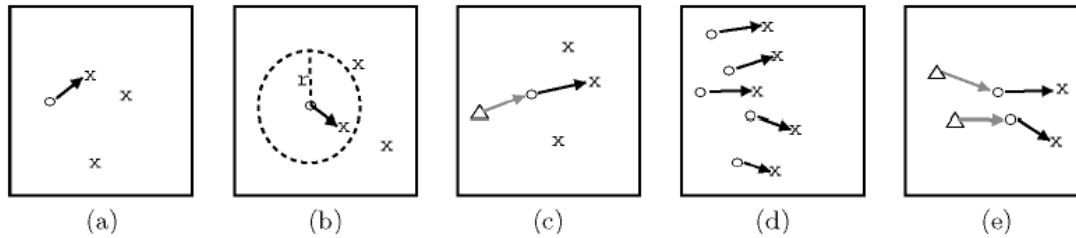


Figure 4.9: Different motion constraints. \triangle denotes object position at frame $t-2$, \circ denotes object position at frame $t-1$, and x denotes object position at frame t .

Kalman filter (KF) is a linear statistical estimator that provides an efficient prediction of the future system state, by integrating visual feature observations. KF calculates the terrain-relative position and velocity based on the previous estimate value and aims to minimize the mean squared error between the actual and estimated data. This method has become a powerful mathematical tool for solving tracking problems [150] and several papers related to it were published emerging many different types of KF [151]. Extended Kalman filter (EKF) is a nonlinear version of KF method that simply linearizes nonlinear models so that, KF can be applied. In Julier and Uhlmann's work [152], an improved version of EKF is presented, the Unscented Kalman filter (UKF). Evensen KF, first presented by Evensen [153], is a recursive filter appropriated for a larger number of variables. These types of KF have been used on many applications such as object tracking [154], image stabilization [155], robot vision [151] and others.

One limitation of KF occurs when the state variables do not follow a Gaussian distribution. The particle filtering (PF), in turn, can overcome this issue producing a good performance when applied to nonlinear and non-Gaussian variables [60]. Like KF, PF is based on previous estimate value and it is divided in four stages: predict, update, estimation, and resampling. The algorithm is first initialized by discretizing the problem into random "particles" and the initial probability density function is estimated from there. Each particle advances independently. After the prediction step, which produces an approximation *a priori*, the weight of each particle is updated, they are redistributed and process restarts again [62]. PF, although having higher computational requirements than KF, is a more flexible approach. Its computational complexity increases with the number of particles. PF can be seen as an alternative to EKF or UKF with the advantage of being more accurate when the number of particles are enough and, when it is not, the PF might suffer from sampling impoverishment [156]. In the same way as KF, PF has many applications in several

fields such as robotics, statistics, automatics, diagnostics and prognostics [157]. When applied for object tracking, particles are usually contours, color features or appearance models [158, 62].

4.1.4.2 Kernel Tracking

The kernel tracking (KT) method uses a representative kernel of the object shape or appearance and may be either a rectangular, an elliptical or a circular shape with an associated histogram. In this tracking process, the motion of the kernel is updated from previous frames. Usually, this motion is assumed in the form of a parametric transformation such as translation, conformal, or affine transform [60]. KT is a demotic approach due to its broad range of convergence and its robustness uncertain spatial deformations. Some surveys, classify KT methods into template model and appearance model.

Template matching method, also called blob matching, searches for the most similar region inside a frame to the object template. The object template is considered to be the reference sub-image of the object and it can be built by using intensity, color or gradient features, histograms or mixture models. This matching processing gives an outcome measure of the degree of similarity which can be measured using the normalized cross correlation [159]. A drawback of template matching is its high computation cost being very time-consuming.

Among appearance model methods, mean shift (MS), first presented by Fukunaga *et al.* [160], is one of the most popular methods. MS is a kernel-based statistical algorithm that locates the maxima in any probability distribution, within the searching window. This iterative process only stops when the algorithm converges after finding the maximum. A probability value is assigned to each pixel representing the likelihood of that pixel belonging to the object. After having each frame represented as a 2D probability distribution, the MS algorithm can be computed. In Cheng's work [161], MS algorithm was first introduced into the domain of computer vision and in Comaniciu and Meer's study [162], this method was applied to solve an object tracking problem. Since MS uses the object's color statistical information, its best performance is ensured with some constraints during the tracking process: the object can not change its color, the illumination must be kept constant, similar objects within the scene can not be present, the background must have a different color from the object and occlusions can not occur [163]. In the work of Allen *et al.* [164], the CAMShift (Continuously Adaptive Mean Shift) algorithm, first presented by Bradski [165] to track heads and faces, is reviewed and improved by using spatial moments to iterate the distribution. To improve the tracking performance, MS has been combined with other algorithms such as double filters [166], Gaussian-cylindroid color model [167], color and shape [168], among others. MS algorithms can also be used to perform smoothing and segmentation [162].

4.1.4.3 Silhouette Tracking

In silhouette tracking, the target is represented by an outline contour and the tracking process is performed by estimating the object region in each frame. These boundary-based approaches, have been widely adopted in object tracking due to their ability of describing accurately complex

shape objects. They can be seen as object segmentation methods applied in the temporal domain. According to Yilmaz *et al.* [60], silhouette tracking can be divided in two categories, namely shape matching and contour tracking.

Shape matching searches for the object silhouette in the current frame. This method is not so different from the template matching method presented in Kernel Tracking subsection and it can be performed in a similar way, by using an object silhouette template. In the work of Huttenlocher *et al.* [169], this method is performed using an edge representation of the object. Here, the Hausdorff distance is used to compare the similarity between the new detected contour and the template. Another way to perform shape matching consists in finding silhouettes correspondence between two consecutive frames [121]. This approach is similar to Point Tracking methods but it considers the complete object region instead using some points. Looking for silhouettes matches in consecutive frames uses essentially spatial information, however, information about the object movement can also be used to track objects' silhouettes by using flow vectors. This was the approach of Sato and Aggarwal [170], where the authors applied the velocity Hough transform to compute voting matrices for the vertical and horizontal flow providing a so-called Temporal Spatio-Velocity image. This matrix encodes the dominant motion of a moving object performing a motion-based matching of the object shapes being less susceptible to appearance changes.

Contour tracking iteratively evolves an contour from the previous frame to its new position in the current frame. This approach requires that a part of the object in the current frame overlay with the object region in the previous frame. There are two different approaches: one that uses state space models and other that evolves the contour by minimizing the contour energy. The latter is analogous to the PDE image segmentation technique previously presented, however the energy is characterized using temporal information instead, such as the temporal gradient (optical flows) or appearance statistics generated from the foreground and background [60]. Contour tracking methods using state space models define the object's state in terms of shape and motion of the contour. This state is updated by maximizing the contour's *a posteriori* probability which is dependent on the previous state and the current likelihood defined, usually by the distance of the contour from the edges. The new state can be predicted using KF [67].

4.2 Deep Learning-Based Tracking

Deep Learning (DL) is, nowadays, the most successful research direction in the field of machine learning since it has been demonstrated to provide higher accuracy and performance [171]. DL has brought significant breakthroughs in many computer vision applications including object detection, recognition and tracking [172].

DL-based techniques for visual tracking can be categorized into three classes according to the network structure, function and training. The latter is an essential step to develop a robust tracker. Considering the network structure, trackers can be categorized in CNN-based, Recurrent Neural Network (RNN)-based or others. While the former has shown to be effective for feature extraction achieving great results on image classification, the later has been shown to be more

suitable for sequence modeling since is able to establish temporal connection [172]. Concerning deep network's function for visual tracking, they can be also divided into two categories: feature extraction and end-to-end networks. While the first uses the network only to extract effective features and then uses traditional methods to learn the appearance model and locate the target, the second tries to simultaneously carry out the detection, classification and tracking tasks by proposing an unified approach. In other words, an end-to-end network unites and optimizes the whole tracking process by using the network to both feature extraction and candidate evaluation of the tracked object [172].

To a more detailed comprehension about DL tracking algorithms, it is suggested to read the work of Li *et al.* [172], where, for each category, an explanation and analysis of the network perspective is made.

4.3 Tracking Evaluation Metrics

The evaluation of tracking algorithms is an imperative task in order to validate them. To allow a better understanding of the proposed tracking metrics, it is first necessary to understand what is expected from a tracker: a tracker should find the correct number of objects present at each frame, it should estimate correctly their position, as precisely as possible and, it must also be consistent, assigning for each object, an unique track ID throughout the whole video sequence. Thus, the evaluation criteria should reflect the tracker's precision in finding the objects and its ability to correctly keep them on track. Additionally, metrics should be adjustable, easily understandable and to allow the comparison between different trackers [173].

To evaluate the performance of the tracker, many metrics have been proposed, being the most common strategy, the comparison between the obtained results with ground truth data which usually requires a large amount of annotation [174]. Some researchers have been proposing evaluation measures without requiring a ground truth. In the work of Erdem *et al.* [175] shape and color differences of the results are used as measures to evaluate quantitatively the performance of video object segmentation and tracking algorithms. However, this is only effective if the segmentation presents reliable results. Other proposals compare the initial and the final positions of object evaluating only the tracking process [175, 176].

Considering a ground truth, metrics for classification and detection problems involve the relative frequency of correct and incorrect decisions. Some terms commonly used in the context of detecting an object are true positive (TP), false positive (FP), true negative (TN) and false negative (FN) (Figure 4.10). These terms are in the base of estimating metrics such as accuracy, sensitivity, specificity, precision, recall and F-score. Accuracy is a measure of the global performance of the algorithm in what concerns correct decisions (Equation 4.4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.4)$$

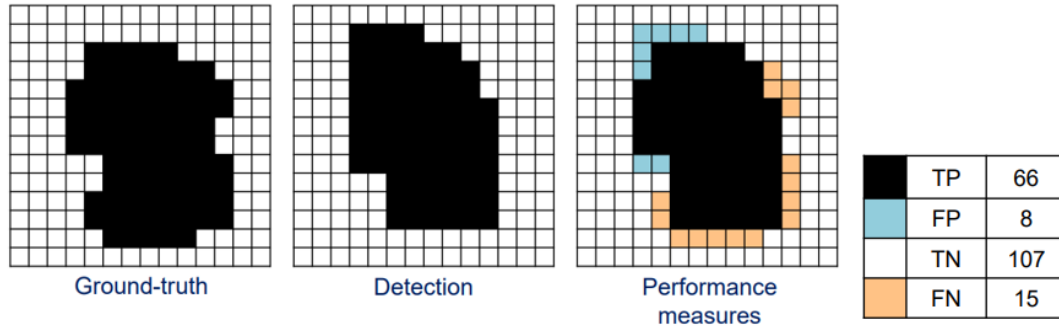


Figure 4.10: TP, FP, TN and FN example.

Sensitivity reports how frequently the algorithm correctly detects the object (Equation 4.5) while specificity refers to the number of times the background is correctly detected (Equation 4.6).

$$Sensitivity = \frac{TP}{TP + FN} \quad (4.5)$$

$$Specificity = \frac{TN}{TN + FP} \quad (4.6)$$

Recall (R) is computed following the sensitivity expression. Precision (P) can be found in Equation 4.7. The F-score combines precision and recall by their harmonic mean. The traditional F-score measure (F) is described in Equation 4.8, where P and R are equally weighted.

$$Precision = \frac{TP}{TP + FP} \quad (4.7)$$

$$F = 2 \times \frac{P \times R}{P + R} \quad (4.8)$$

The area-based F1-score is a variant of the F-score and it can be defined by the expression presented in Equation 4.9 where P^i and R^i represent the precision and the recall in frame i , respectively [174].

$$F1 = \frac{1}{N_{frames}} \sum_i 2 \times \frac{P^i \times R^i}{P^i + R^i} \quad (4.9)$$

Redirecting the metrics to the specific tracking task, FP occurs when the tracker identifies a target which is not a target and, FN, when the tracker misses to identify and locate the target [174]. A common error measure is the deviation also known as error of the center location. This measure is expressed in pixels and it can be representative of the accuracy of location measure. Its expression can be found in Equation 4.10, where $|M_s|$ denotes the number of frames in the video sequence where the tracked bounding box T_i matches with the ground truth bounding box GT_i .

and $d(T^i, GT^i)$ denotes the distance between the centroids of both bounding boxes which can be calculated by using the Euclidean distance. The average of this distance can be calculated over all the frames of the sequence summarizing the overall performance [177]. However, this value may be abruptly influenced by frames where the tracker loses the target, not being totally representative of the center location error. Therefore, to evaluate this measure, a plot precision can be designed, showing the percentage of frames whose estimated location is within the given threshold distance of the ground truth.

$$Deviation = 1 - \frac{\sum_{i \in M_s} d(T^i, GT^i)}{|M_s|} \quad (4.10)$$

An evaluation metric commonly used in tracking algorithms is the bounding box overlap. In a frame k , spatial overlap $A(GT_i, T_i)$ can be defined as the intersection level between the ground truth bounding box GT_i and the tracked bounding box T_i (Equation 4.11).

$$A(GT_i, T_i) = \frac{Area(GT_{ik} \cap T_{ik})}{Area(GT_{ik} \cup T_{ik})} \quad (4.11)$$

A common way to measure the performance on a video sequence is to count the number of successful frames whose $A(GT_i, T_i)$ is larger than a defined threshold, which according to the PASCAL criterion [178], its value is set to 0.5. In the work of Wu *et al.* [177], a success plot showing the ratios of successful frames at the thresholds varied from 0 to 1 is performed since the authors believe that using an unique threshold value may not be representative enough of this measure. The area under curve of the success plot is the metric to evaluate the algorithm.

Another metrics, described in Bernardin *et al.*'s work [173] for multiple object tracking (which can be adapted for a single object tracking [174]) are object tracking accuracy (OTA) and object tracking precision (OTP). OTA gives a very intuitive measure of the tracker's performance (Equation 4.12) while OTP represents the tracker's ability to precisely estimate object positions (Equation 4.13).

$$OTA = 1 - \frac{\sum_i (FN^i, FP^i)}{\sum_i g^i} \quad (4.12)$$

$$OTP = \frac{1}{|M_s|} \sum_{i \in M_s} \frac{|T^i \cap GT^i|}{|T^i \cup GT^i|} \quad (4.13)$$

A slightly variation of the OTP, defined as the average tracking accuracy, is presented in Equation 4.14 [179].

$$ATA = \frac{1}{N_{frames}} \sum_i \frac{|T^i \cap GT^i|}{|T^i \cup GT^i|} \quad (4.14)$$

Taking this into account, it is possible to divide evaluation metrics into two types of measures for precision and recall: one based on the position of objects as a whole such as the F-score or OTA, and other based on the pixels. The later is important for the work in question, since the

quantification of the area of the rodent paws is a very important step. The accuracy, sensitivity, specificity, precision and traditional F-score are common pixel-based measures for classification and detection problems.

While the area-based F1-score, OTA and ATA are important measures to evaluate tracker's ability to hold the tracker, spatial overlap, deviation and OTP metrics measure the capability of a tracker to correctly determine the location of the target object.

4.4 Mouse/Rat Tracking

Sophisticated machine learning and computer vision algorithms have been proposed for mice tracking. However, the majority of the studies found, aims to automatically analyze mice behavior, having a different setup from the one of this study [180, 181, 182, 183]. On these studies, the camera films the animal from above in normal light conditions.

The only similar work found, even with different lighting conditions, was the work of Leroy *et al.* [184], where an automated gait analysis for laboratory mice is presented. In this study, a background subtraction method is applied to segment the mouse and a motion analysis is performed: as the acquisition is done in an open-field test where mouse can move freely, the authors separate running periods from standing and resting periods. For running periods, the speed of the mouse is computed. Unlike the setting of our work, their hardware allows them to detect the footprints by resorting to a color filter, enhancing the pixels that correspond to the region of the paws with a strong red color component.

In between all the commercially available automatic video-based systems for gait analysis listed in the previous chapter, the MouseWalker [49] is the only one which has the code available for the community. Although their work proved to be able to locate the animal and to extract a large set of gait parameters, this system is designed for a much more controlled environment since there are not background movements and the trajectory of the animal is restricted to a walkway, being the mouse always within the frame and walking in the same direction, contrarily to what happens in this dissertation.

4.5 Summary and Conclusions

Object video tracking is one of the most important tasks in computer vision systems, fusing many fields like image processing, pattern recognition, artificial intelligence and automatic control [53]. Video tracking can be summarized as the task of estimating the position of one (or more) moving object(s) throughout the video sequence and segment it/them from the background. Although object tracking remains a challenging task in computer vision field, many attempts have been made to improve tracking algorithms [61]. Traditionally, the tracking flow can be described by the following steps: object detection, object classification and lastly, object tracking. More recent techniques use deep learning techniques to track objects since they have been demonstrated to provide higher accuracy and performance in tasks such as object detection and classification [171].

Some machine learning and computer vision algorithms have been proposed for mice tracking. In fact, the video-based automatic gait analysis systems listed in the [Chapter 3](#) use these methods to extract gait parameters. However, as they are closed solutions, related works of open source algorithms for mice tracking were investigated. No work was found that meets the same degraded luminosity conditions and the same setting. The majority of the works found related to mouse tracking do not concern gait analysis and were developed for different setups, however, they may be an evidence of tracking algorithms application.

Chapter 5

Acquisition Setup and Dataset

5.1 Acquisition Setup

5.1.1 Animal Model

The data used for this study was acquired at Faculdade de Medicina da Universidade do Porto, Porto, Portugal. All experimental procedures were performed in compliance with all ethical norms required [185] and appropriate measures were taken to minimize pain or discomfort of the animals. All the conditions were assured in order to maintain reliable results without external interference. Animals were kept at a constant temperature of 22°C and controlled lighting (12-hour light/12-hour dark cycle) [19]. Under brief anaesthesia, the osteoarthritis monoiodoacetate model was induced on the right hind knee of the animal, as previously described [32]. In order to have a control of the disease, a similar number of animals were injected with saline, under the same conditions.

5.1.2 CatWalk Test

Several behavioral tests were performed in all animals, which were previously accustomed to the researcher and to each test in order to reduce the external interference on the results. Considering the monoiodoacetate model of OA, the behavioral tests were performed at the following time points: 1) At day 0, before monoiodoacetate (OA) or saline (control) injection; 2) After the induction of the model (on days 7 and 14 after monoiodoacetate/saline knee injection); 3) After the surgical procedure, when the molecule of interest was delivered (on day 21 after monoiodoacetate/saline knee injection). It is important to note, however, that this timeline is an example of one experimental design, not being used in all situations, and may vary according to the experimental protocol in use.

This dissertation will focus on data acquired by a custom-made gait analysis system that allows the animal to walk freely on an open field platform located in a darkened room, with no stressors or rewards. The hardware configuration used is based on the CatWalk system (already introduced on Chapter 3), and it is illustrated in Figure 5.1 (adapted from [186]). To sum up: a fluorescent

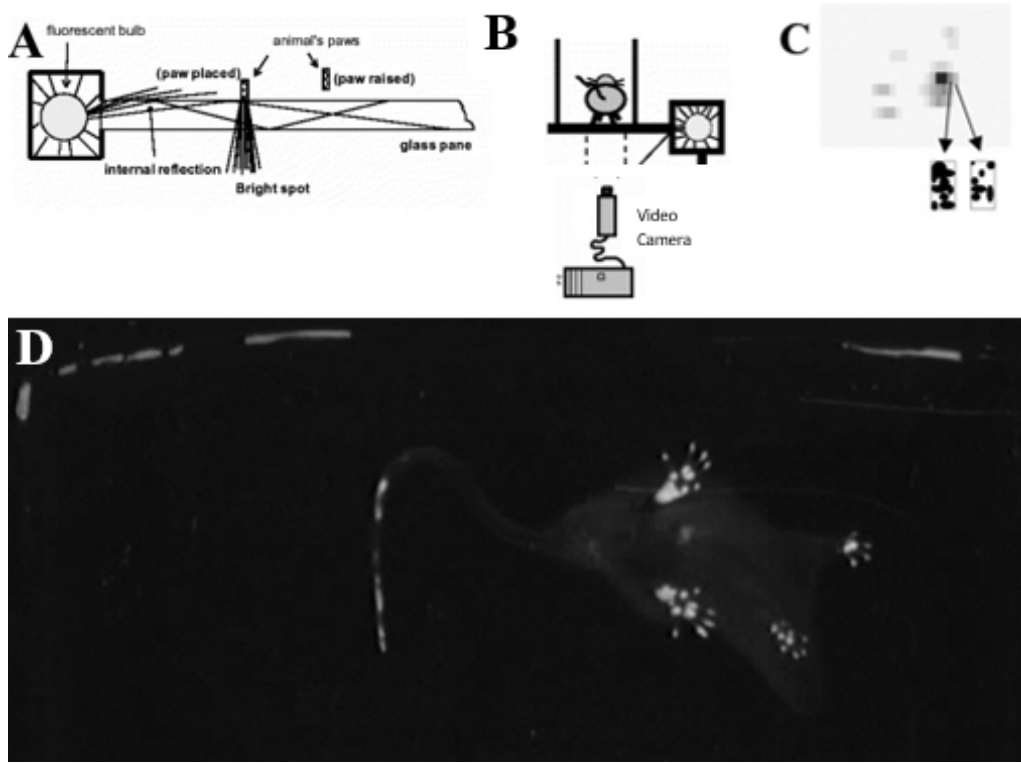


Figure 5.1: Principle of the CatWalk setup.

light beam is sent upwards through a glass platform being completely reflected internally in all the plane with exception of the areas where the paws are placed, where it is refracted (A); images are recorded by a video camera placed under the glass platform in which animals are allowed to walk freely. The video camera is connected to a computer equipped with video acquisition software (Ulead Video Studio, Fremont, CA) (B); the signal intensity depends on the contact area of the paw with the surface and increases with the pressure applied by the paw (C); an example of the obtained images is represented in D.

After recording the animal's gait, the analysis of the CatWalk behavior consists on measuring the mean intensity and the contact area of each paw with the glass platform, to infer the disability induced by the model over time [19]. This analysis is currently done manually, according to the process explained in Chapter 3.

5.2 Dataset

The dataset is composed by 15 videos, each one with different duration, averaging about 5 minutes each. The videos have a resolution of 640 horizontal x 480 vertical pixels and a frequency of 25 frames per second (FPS). The animal features a mean length and width of 220 and 90 pixels, respectively and it presents averagely an area of 14000 pixels, translating into a frame occupation of 5%, approximately. Due the position of the camera relative to the platform, the last 50 rows of

each frame do not correspond to a region of the platform so, this frame dominion never contains the animal on it.

The next subsections will explain some dataset features that hinder the automation of the gait analysis task and the process adopted to obtain the ground truth annotations.

5.2.1 Dataset Heterogeneity

Some artifacts and variations from video to video, and even within each video, makes the automation process more difficult. Since the videos were acquired in the dark, degraded luminosity condition is a feature of the dataset. An example of a frame can be seen in Figure 5.1 (D).

Furthermore, although the data was acquired using a static camera in a slightly controlled environment, some of the problems concerning object tracking reported in Chapter 4 are present in the dataset, such as the presence of noise in the background, whether due to the interference of researchers with the animal who try to make the animal move with the hand or place a box in the middle of the video to confine the animal to the region captured by the camera or simply due to contamination of the platform by the animal (Figure 5.2). Since the platform area is larger than the area captured by the camera, periods in which the animal leaves the field of view, completely or partially, are also present in the dataset (Figure 5.3). In fact, the animal is fully and partially occluded in 17% and 43% of the dataset frames, respectively.

Moreover, as the rat is a non-rigid object, its shape varies during the video sequence, making it more difficult to track. Some natural behaviors such as sitting, lying, scratching and getting up, among others, hinder the detection and segmentation of the paws since these behaviors promote the contact of the body with the platform, enhancing some non-paws pixels (Figure 5.4).

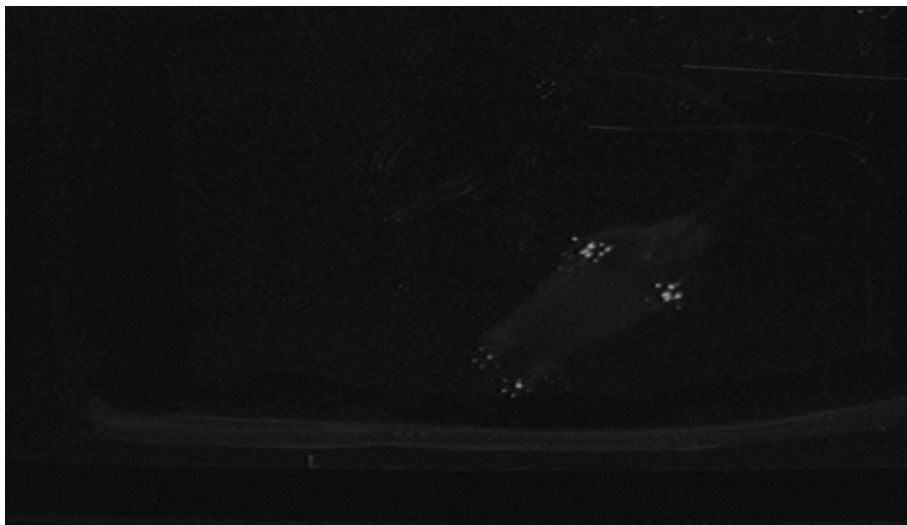


Figure 5.2: Example of a frame with background noise.

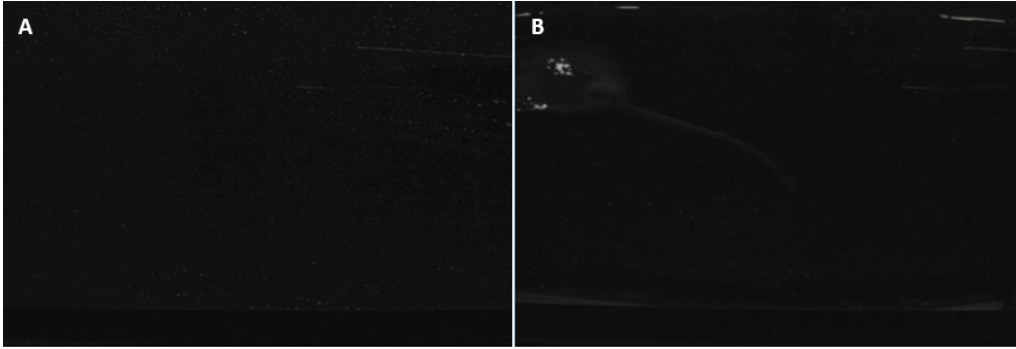


Figure 5.3: Example of frames with occlusions. **A**: Full occlusion; **B**: Partial occlusion.

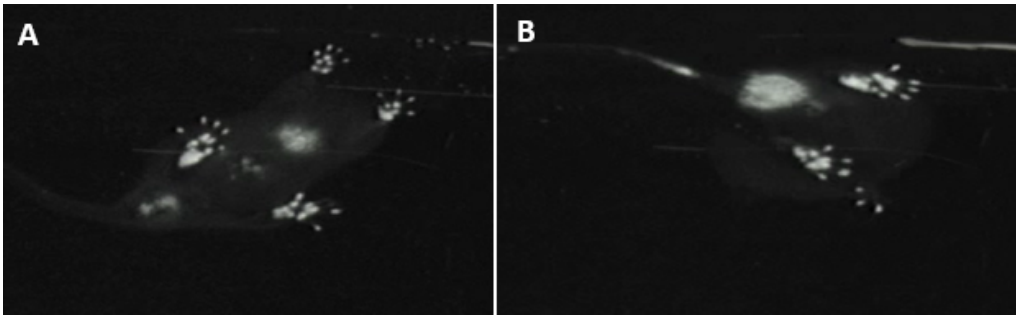


Figure 5.4: Example of behaviors that hinder paws segmentation. **A**: Rat's belly in contact with the platform; **B**: Scratching.

5.2.2 Data Annotation

Data annotation is the task of labeling data, such as images or videos, containing specific objects. Annotation is done manually by humans, being a very tiresome and time consuming task. The most recent techniques of machine learning and computer vision require large amounts of labeled data for supervised learning and quantitative evaluation of the algorithms. Given its significance, the computer vision research community has been trying to design efficient tools and strategies for the annotation of images and videos [187].

All the videos from dataset had already been analyzed by experts who had previously selected some frames of interest (approximately 9 frames per video - 3 pairs of frames walking and 3 frames of the animal standing still - see Subsection 3.3.1) and the associated threshold, determined on ImageJ software, used to compute the area and the mean intensity of all the paws in each frame. In order to evaluate the algorithm's performance, all the frames selected by the experts were annotated using Labelme [188], an open source easy-to-use graphical image polygonal annotation tool inspired by the online annotation tool LabelMe [189]. This software is written in Python and uses Qt User Interface Design Tool for its graphical interface. It allows the representation of the location of the object by a polygon, a bounding-box (rectangle or circle), a point or by the use of keypoints.

Figure 5.5 represents the global process of the annotation step. Polygons were chosen to

represent all the objects of interest (the body of the animal, the tail and all the paws: right-hind, right-fore, left-hind and left-fore). Thus, five labels were used for the annotation vocabulary: mouse (denoting the body of the animal), tail, rh, rf, lh and lf. The output of this process is a JSON file which is used to obtain the ground truth masks.

The detailed masks of the paws were obtained by applying the threshold determined by the experts to the result of the annotated paws' mask applied to the input frame (Figure 5.6). Figure 5.7 shows an example of the obtained masks. Rat's body and tail annotations were thereafter confirmed by an expert.

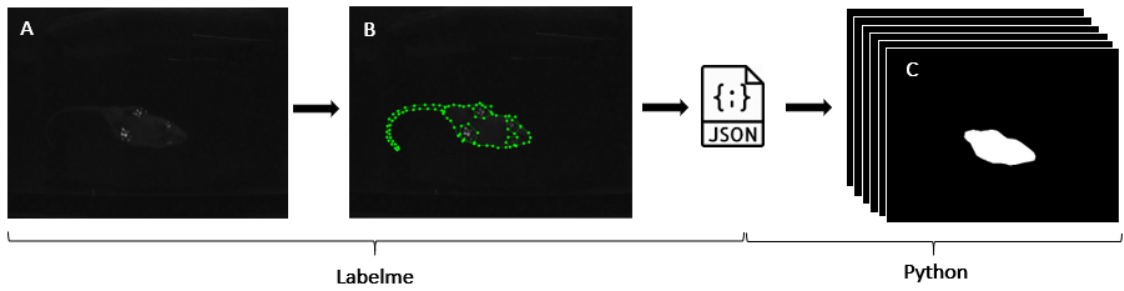


Figure 5.5: Annotation process. **A.** Input frame; **B.** Frame annotated; **C.** Ground truth masks.

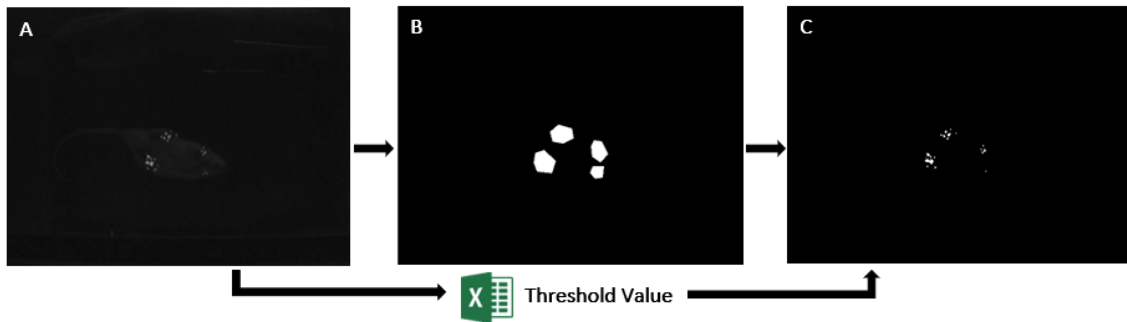


Figure 5.6: Process of obtaining masks of paws. **A.** Input frame; **B.** Region of the paws manually labeled; **C.** Ground truth paws mask.

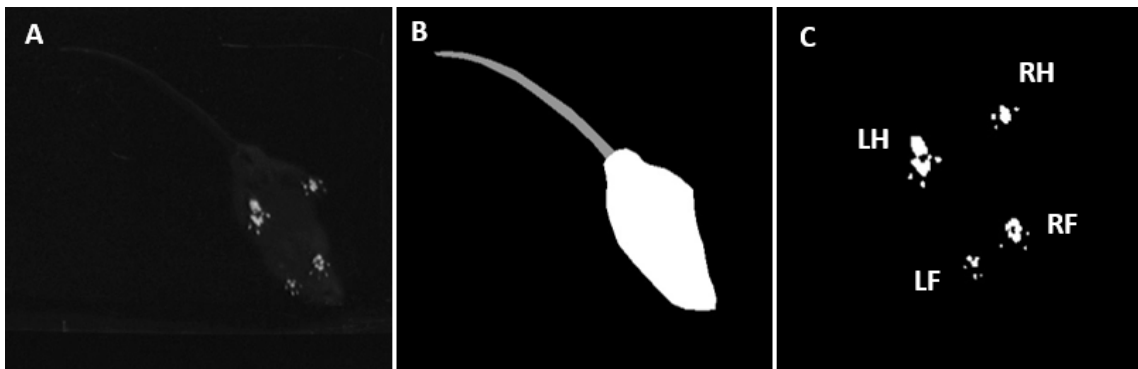


Figure 5.7: **A.** Input frame; **B.** *White*: Ground truth rat's body; *Gray*: Ground truth tail; **C.** Ground truth paws: *LH*: Left-Hind; *RH*: Right-Hind; *LF*: Left-Fore; *RF*: Right-Fore.

5.2.3 Public Rodent Datasets

There are some available public rodent datasets for behavioral analysis, both for social and individual behavior, as it can be seen in Table 5.1. However, to the extent of our knowledge, there are no public datasets for gait analysis, with a setup similar to the one previously presented.

Table 5.1: Rodent Behavior Datasets.

Dataset	Number of Animals	Setup	Annotation Tasks
RatSI [190]	2 rats	Top view	Groom, approach, follow, move away, nape attack, pinning, social nose contact and solitary.
CRIM13 [191]	2 mice	Top and side view	Approach, circle, sniff, walk away, copulation, chase, clean, drink, eat and up.
MBADA [192]	2-3 mice	Top view	Nose2Body, nose2Nose, nose2Genitals, above, following, walk alone, stand together and stand alone.
SCORHE [193]	1-3 mice	Multiple in fish-eye cameras	Not annotated.
P. Dollár [194]	1 mouse	Side View	Drink, eat, explore, groom and sleep.
H. Jhuang [192]	1 mouse	Side View	Eat, drink, groom, hang, micromovement, rear, rest and walk.

Chapter 6

Rat Tracking Methodology

As it can be concluded from the literature review, automated recognition of the animal's gait pattern can be done by using different computer vision and machine learning techniques. In this chapter, the proposed video-based methodologies designed to analyze the rat's gait are presented.

Figure 6.1 shows the fluxogram of the proposed pipeline.

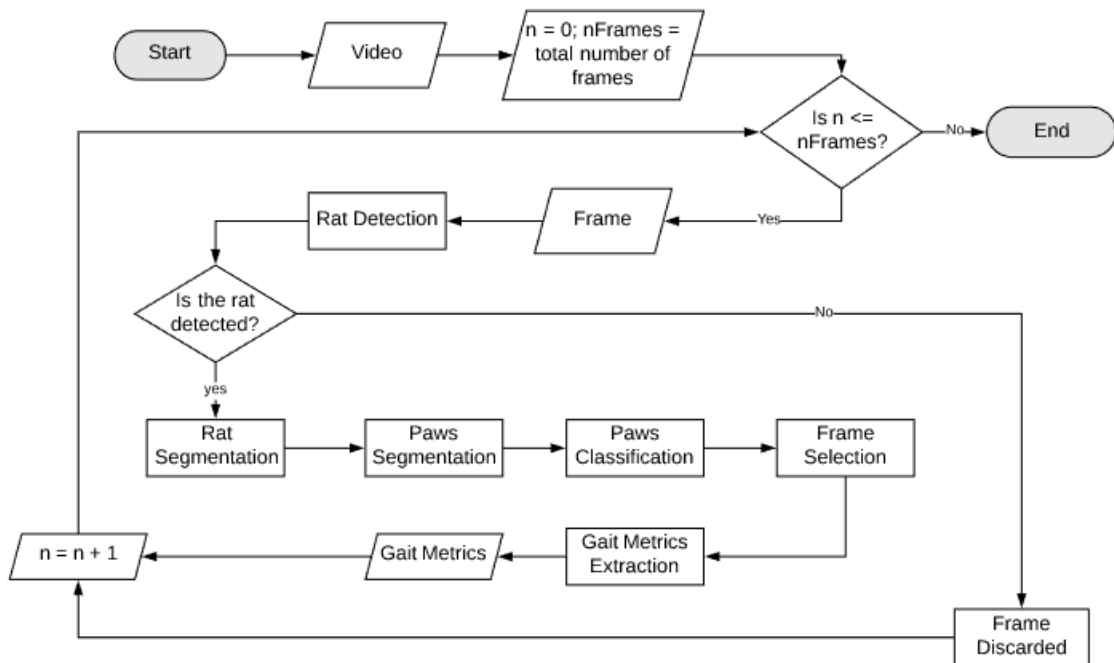


Figure 6.1: Fluxogram of the framework.

The adopted methodology is based on a tracking-by-detection technique where, in each frame, the objects of interest, the rat and its paws are detected and segmented. Since the rat is not present in all the frames, frames where the animal is partially or fully occluded are discarded. In frames where the rat is present, its segmentation occurs as well as its paws detection, segmentation and

classification. After having the paws classified, the framework goes through a frame selection algorithm which aims to select some frames of interest: pairs of frames where the animal is walking and frames where the rat is standing still (see subsection 3.3.1). Lastly, the metrics of interest are extracted.

In this way, taking into account the adopted approach, the generic video-based gait framework may be briefly described as follows: rat detection and segmentation, paws detection, segmentation and classification, frames selection and gait metrics extraction. Figure 6.2 shows a simpler representation of the pipeline as well as the methodologies adopted on each step. Current trend of deep learning was not adopted since the dataset is small, not favouring deep learning solutions, and the strong field knowledge about the setting facilitates the feature-engineering approach.

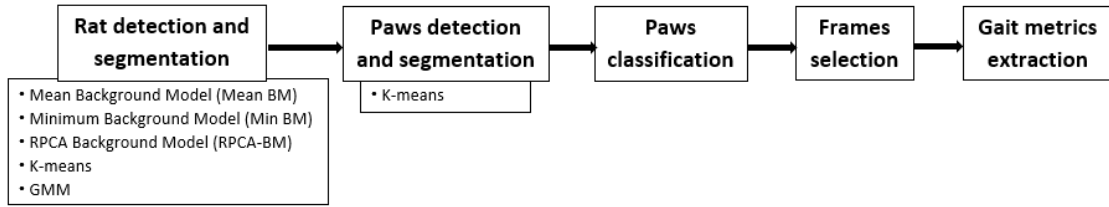


Figure 6.2: The generic video-based rat tracking framework.

6.1 Rat Detection and Segmentation

The goal of this first step is to robustly detect and segment the body of the animal. Since this operation is key in the adopted methodology, once its success will impact the following steps, several segmentation techniques were performed, evaluated and compared to seek accuracy and to reduce the chance of the rat's position mismatch.

Before the segmentation process itself, a pre-processing step was applied to all frames in order to remove noise. This operation is performed by computing a simple convolution between each frame (f) and a 3-by-3 box linear kernel (k), a spatial domain low-pass linear filter in which each pixel results on the average value of its neighboring pixels (Equation 6.1).

$$k = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (6.1)$$

Since the filter's size is small when compared to the image size, the filter's effect, shown in Figure 6.3, is enough to smoothen the image but negligible to not change the image too much.

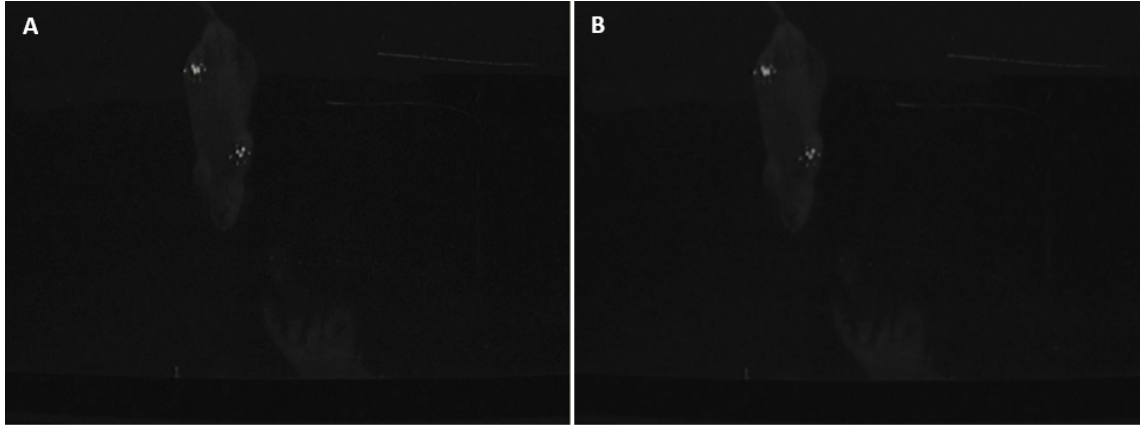


Figure 6.3: Box linear kernel's (k) effect. **A**: Input frame; **B**: Denoised image.

Considering that one of the dataset characteristics is the presence of background fluctuations, four different approaches were tested to model the background. The first approach consisted on averaging the whole video sequence (v) previously denoised following the Equation 6.2.

$$BG(v) = \frac{1}{|v|} \sum_{f \in v} K \otimes f \quad (6.2)$$

In the second, because the animal is ideally the only bright object in motion, the minimum value of each pixel (p_{ij}) in the entire video sequence was computed (Equation 6.3).

$$BG_{ij}(v) = \min_{f \in v} (p_{ijf}) \quad (6.3)$$

Both of these offline approaches result in a static background image which is subtracted to each frame to obtain the foreground, the animal in motion (Equation 6.4). A threshold operation is performed to obtain the mask of the rat and morphological operations are applied to remove some background noise.

$$FG(v, f) = |f - BG(v)| \quad (6.4)$$

The third tested approach to model the background (already mentioned in the Chapter 4) was presented by Candès *et al.* [125] and it considers the data matrix (M) a sum of a low-rank component (L) with a sparse component (S). Each column in M corresponds to a frame of the video, properly vectorized. Assuming that the background changes slowly overtime, the intuition is that the observed data is the sum of the background information and the objects' data. Since the background is essentially static, it should lead to a low-rank matrix being able of comprising only small fluctuations on it; on the other hand, the sparse matrix can formulate the objects in motion, in this case the rat, which tend to be small and it can be seen as an outlier. In this method, the principal

component pursuit is used to solve the robust principal component analysis (RPCA) following the Equation 6.5, where $\|A\|_1$ denotes the vector l_1 norm of the matrix A and $\|A\|_*$ denotes the nuclear norm, the sum of the singular values of A . Morphological operations were applied to the sparse matrix to remove small blobs of the background.

$$\min(\|L\|_* + \lambda \|S\|_1) \quad \text{subject to} \quad M = L + S \quad (6.5)$$

Other tested approach to model the background was the Gaussian Mixture Model (GMM) [195]. In this methodology, previously explained in Chapter 4, each pixel is modeled by a mixture of Gaussian distributions taking into account the pixel's recent history.

K-means was another methodology used to segment both rat body and paws. As explained in Chapter 4, this clustering algorithm aims the partitioning of the image into homogeneous regions containing pixels with similar characteristics, in this case the pixels' intensity. The number of clusters was set to three, where one represents the background, other the body of the rat and the third one the paws. Being an iterative algorithm, the iteration termination criterion was defined to be either when the accuracy was equal to 1 or when the algorithm reached a maximum number of iterations, which was set to 30. This technique was applied to the foreground, using the average of the video sequence as the background model, after a pre-processing algorithm that aims to improve the frames' quality, facilitating the segmentation process. This algorithm was divided in two main steps: a contrast enhancement, achieved by normalizing the frame histogram between zero and its maximum value, followed by a convolution between the enhanced image and a sharpening filter (S) (Equation 6.6) resulting in a well-defined image.

$$S = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & +1 & +1 & +1 & -1 \\ -1 & +1 & +9 & +1 & -1 \\ -1 & +1 & +1 & +1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} \quad (6.6)$$

The result of the contrast enhancement operation is presented in Figure 6.4 C) and the result of the sharpening filter is shown in D).

In all the mentioned methodologies the tail was removed by resorting to an opening morphological operation with a square structuring element of size 13x13. The convex hull of the result of the opening operation was computed, obtaining thus a mask of the body of the rat. The removal of the animal's tail is done to standardize the output of this step since the tail is not always detected and segmented together with the body of the animal.

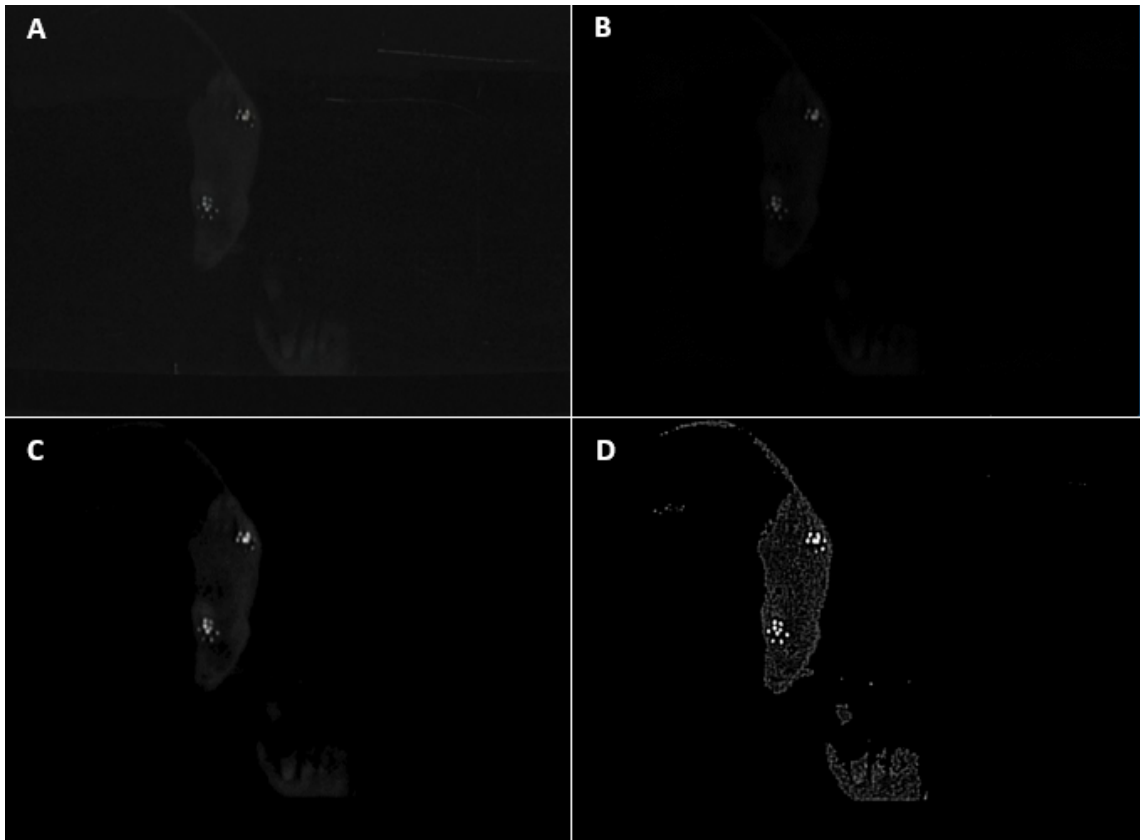


Figure 6.4: **A:** Input frame; **B:** Foreground; **C:** Output of contrast enhancement operation; **D:** Result of the sharpening filter.

6.2 Paws Detection, Segmentation and Classification

The detection, segmentation and classification of the animal's paws is one of the most important steps of this framework since it will be critical in the task of extracting the gait metrics.

K-means was the algorithm used to segment the paws. This unsupervised algorithm was applied to the foreground, with and without the pre-processing algorithm previously described, in order to test its efficacy.

After having the paws segmented, their classification as left-hind (LH), right-hind (RH), left-fore (LF) and right-fore (RF) must occur. For this purpose, the animal's orientation must be first determined. As the body of the animal can be represented by an ellipse, this shape was used to fit the rat's body mask, the output of the rat segmentation step (Figure 6.5).

Not considering temporal information, two approaches were performed to obtain the tail point, the extreme point of the ellipse nearest to the tail. The first consisted on a simple subtraction method between the mask obtained before the tail removal in the rat segmentation stage and after. By this way, when the tail is segmented simultaneously with the body there will exist a considerable difference in one of the ellipse's sides (Figure 6.6).

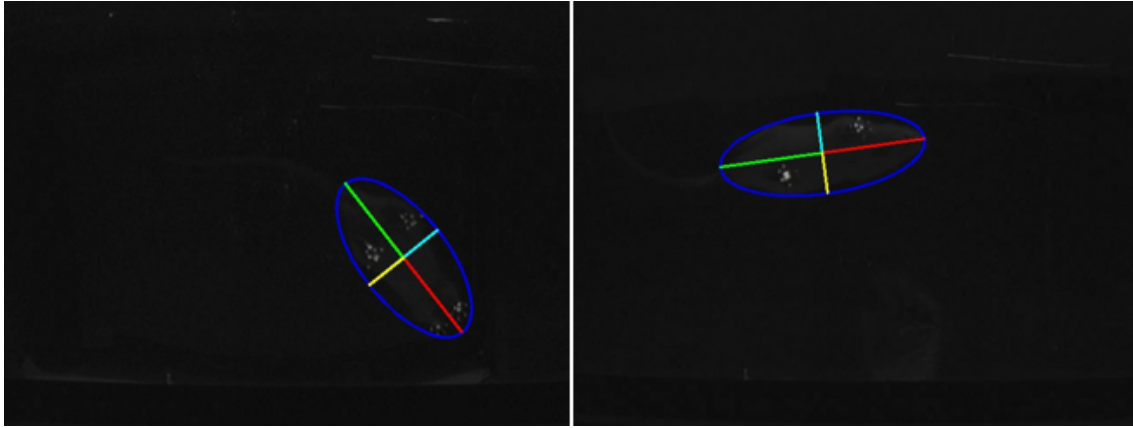


Figure 6.5: Rat's elliptical representation.

However, as already mentioned, sometimes the tail is not visible, not being detected in the first place. In these cases, the difference between both images is not substantial and the tail point is computed by resorting to the second methodology. This second approach computes the minimum difference between the mean of the rat (region inside the ellipse) and its neighborhood in an extension of 250 pixels. A spatial distribution of the minimum points is obtained (Figure 6.7) and each one of the points is associated to the nearest extreme point of the ellipse. The tail point is computed based on the median distance and on the standard deviation between the grouped points and their corresponding extreme point.

These two mentioned approaches are used in frames where there is not information about the tail point of the previous frame (TP_{f-1}), that is, in frames where the animal reappears after an occlusion or when, for some reason, the tail point was not detected in the previous frame. In frames where the TP_{f-1} is known, the tail point (TP_f) is assigned to the extreme point of the ellipse nearest to the TP_{f-1} . Figure 6.8 shows an example of a trajectory obtained by this method.

After determining the tail point, the ellipse is splitted into quadrants which are classified as LH, RH, LF and RF and each paw is assigned to the respective quadrant. However, since the paws are not closed objects, a spatial clustering was performed in order to group the pixels of each paw and thus, obtain the mask of each one of them individually. To classify the paws, the euclidean distance, $d(p, q)$, between the paws' center, p , and each quadrant's midpoint, q , is computed by following the Equation 6.7. By sorting these distances, it is possible to assign, for each quadrant, the nearest paw.

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (6.7)$$



Figure 6.6: **A:** Rat and tail detection; **B:** Result of the opening operation applied to remove the animal's tail. The gray contour is the convex hull of the result; **C:** Result of the subtraction of the image A by the convex hull of the image B.

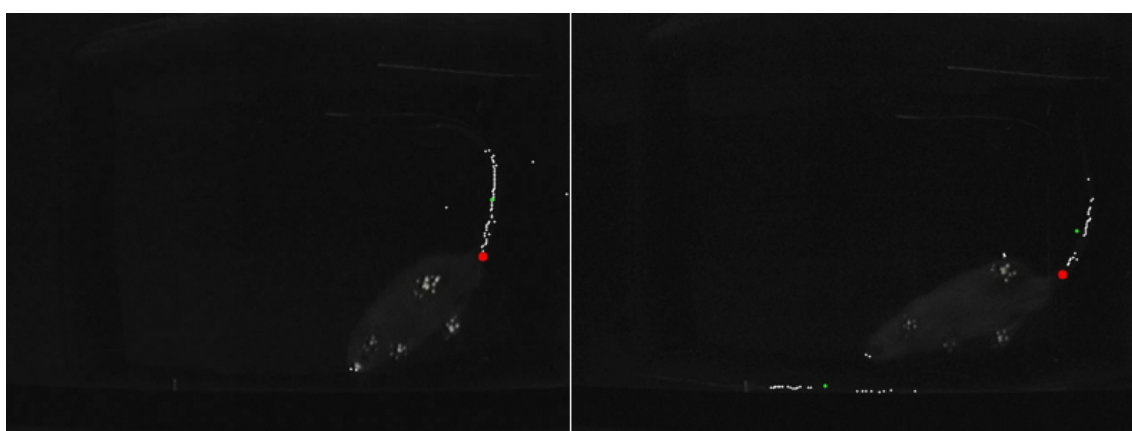


Figure 6.7: **White points:** Spatial distribution of the minimum points; **Red point:** Tail point prediction.

6.3 Frame Selection

The frame selection algorithm aims, as the name suggests, to make a selection of the frames of interest to researchers. As previously explained, these frames include a set of pairs of frames where the animal is walking and a set of frames where the animal is standing still (see Subsection 3.3.1 for more details).

Thus, all the frames where the rat is occluded totally or partially must be detected in order to be discarded. Fully occluded frames are detected by comparing the mean intensity value of the background model with the maximum value of the frame. Since this difference is not significant in frames where the rat is not present, it becomes easier to detect them. Partial occluded frames are detected by resorting to the area of the animal and by its proximity with the image borders.

To choose frames where the animal is standing still, the position of the paws must be considered once sometimes the four paws are detected but the animal is in movement. For this reason, the frame is only considered a standing still frame when all the paws are detected, the paws are in a rectangle position and when there are at least five consecutive frames where all the paws are detected in a rectangle position. In a range of 20 frames where the rat is standing, the algorithm

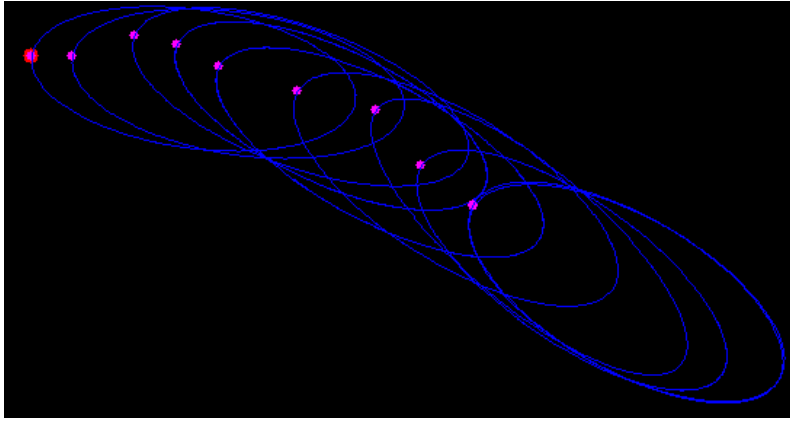


Figure 6.8: Rat's trajectory. The ellipse represents the rat, the red point the tail point obtained by the first method described and the red points the tail points obtained by the temporal method.

will suggest the frame where the difference between the polygon delimited by the paws and its convex-hull is smaller. This will prevent the algorithm from returning consecutive standing still frames which tend to be equal.

To select pairs of frames where the animal is walking, one frame must have only the RH and the LF paws in contact with the platform and the other the LH and the RF. They must be almost consecutive frames and the animal must be in the same direction in the two frames. For this reason, the algorithm will suggest a pair of frames within a maximum range of five frames of difference between both of them and with the ellipse oriented in the same direction.

However, to make sure that the final metrics returned by the algorithm correspond to frames with the paws well-segmented and well-classified, the final frame selection will be performed by the user who will select, in between the proposed frames by the algorithm, the final frames of interest. This semi-automatic task gives the user the power to have the final say on the selection of frames, preventing erroneous results.

6.4 Gait Metrics Extraction

After having the paws segmented and classified and the frames of interest selected, the process of extract gait metrics is straightforward. In order to evaluate the weight applied by each paw, paws' area and mean intensity are the metrics directly extracted from the frames. In this way, a measure of pressure can be computed by multiplying these two extracted measures (Equation 6.8).

$$P = \text{Area} \times \text{Mean Intensity} \quad (6.8)$$

Then, since the affected paw is in this case the RH, the percentage of weight applied by this paw can be given by the Equation 6.9 [19].

$$\%P_{RH} = \frac{P_{RH}}{P_{RH} + P_{LH}} \quad (6.9)$$

This operation is done for the walking frames and for the standing still frames, separately and together, allowing researchers an analysis for each situation.

No other metrics of those presented in the Chapter 3 were considered because they are not of the experts' interest at this moment. However, in the future, there is the possibility to rethink gait analysis itself, diversifying the gait parameters extracted and the context of evaluation.

Chapter 7

Rat Tracking Results and Discussion

On this section, the evaluation metrics and the obtained results concerning the rat detection and segmentation, paws detection, segmentation and classification, frame selection and gait metrics extraction are presented. As reviewed in the literature, segmentation and tracking methodologies are often evaluated by comparison with ground truth annotations. In this work, manual annotations were used to evaluate the results from the performed experiments (see Section 5.2.2 to know how ground truth annotations were obtained).

The results obtained with the chosen metrics provide the necessary data to evaluate and to discuss the pipeline presented in the previous chapter. This also enables the identification of weaknesses that need improvement in order to achieve an optimal functional gait analysis system.

7.1 Rat Detection and Segmentation

The analysis of the performance of the proposed methodologies for rat's detection and segmentation is discussed in this section. As previously mentioned, the purpose of this step is to robustly be able to extract the mask of the animal's body, without the tail. For that, five methodologies were tested to segment the animal's body: mean BM, min BM, RPCA, GMM and K-means. Figure 7.1 shows two examples of the obtained bounding boxes of the animal's body for each one.

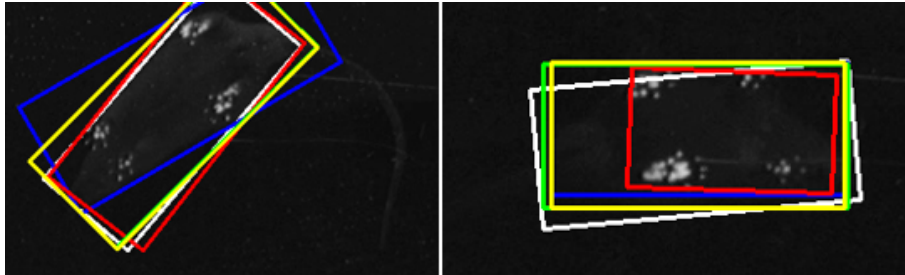


Figure 7.1: Rat segmentation examples for the proposed methodologies: **White:** Ground Truth; **Blue:** K-Means; **Yellow:** Min BM; **Green:** Mean BM; **Red:** RPCA;

In general, all the methodologies are capable of detecting correctly the animal in the majority of the frames. The GMM methodology was not considered since long periods of the animal's immobility lead to the failure of this approach and, once one of our goals is to extract frames where the animal is standing still, this technique is not feasible.

In order to evaluate the different segmentation methods, the intersection level A_f between the ground truth bounding box of the animal's body GT and the tracked bounding box T was computed to each frame (f), according to the Equation 7.1.

$$A_f = \frac{Area(GT_f \cap T_f)}{Area(GT_f \cup T_f)} \quad (7.1)$$

A known measure to evaluate the performance on a video sequence is to count the number of successful frames whose A_f is larger than a defined threshold. According to the PASCAL criterion [178], this threshold value is 0.5, but as the use of an unique threshold value may not be representative enough of this measure, the success plot showing the ratios of successful frames at thresholds ranging from 0 to 1 was computed for each of the methodologies and is presented in Figure 7.2.

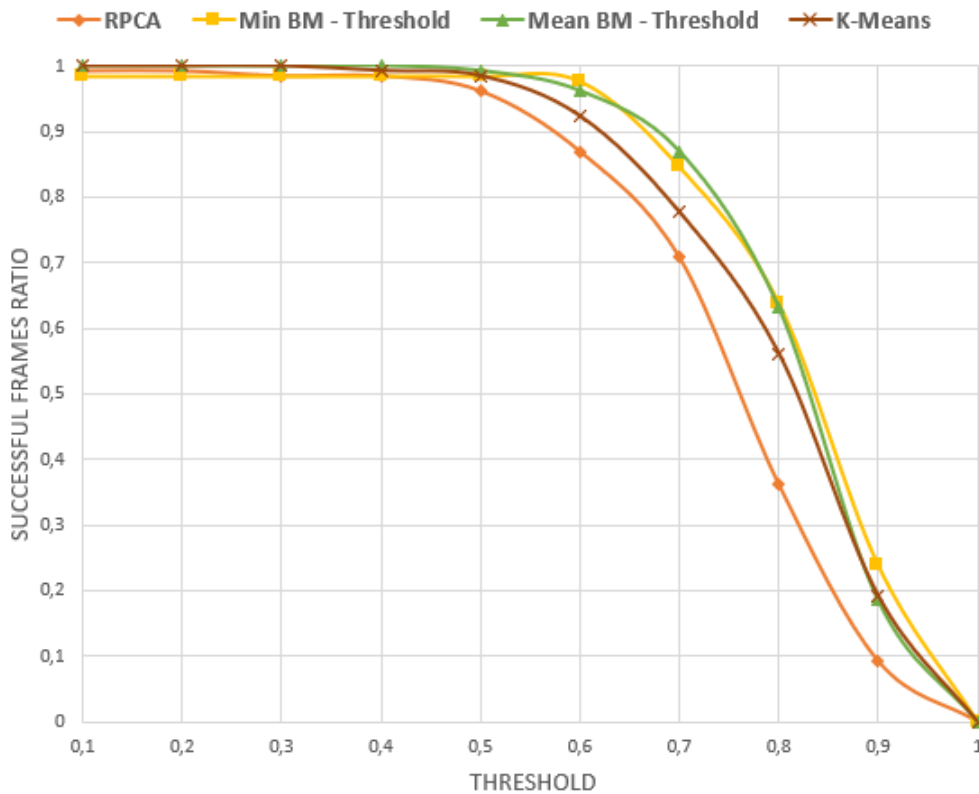


Figure 7.2: Rat segmentation successful plot for the 130 frames manually labelled.

Figure 7.3 shows an example of the video background models obtained using the different methodologies.

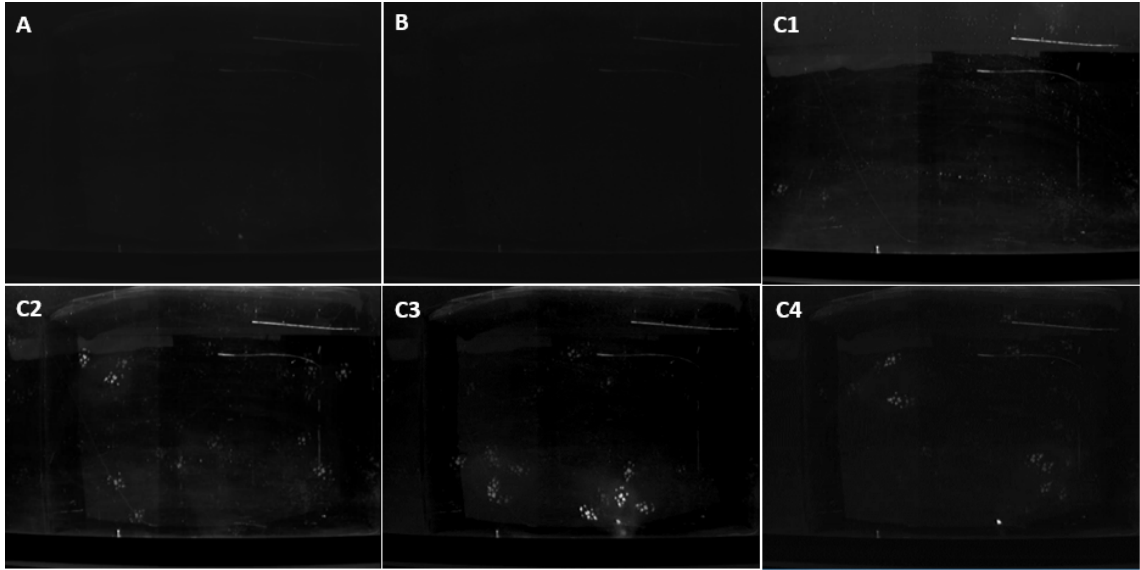


Figure 7.3: **A:** Mean Background Model; **B:** Minimum Background Model; **C:** RPCA Background Model of the different intervals: C1 - 0 to 1999; C2 - 2000 to 3999; C3 - 4000 to 5999; C4 - 6000 to 7999).

The RPCA methodology showed to be the less vigorous due to the presence of the animal in the low-rank matrix. This is explained because this algorithm was computed, without any previous processing of the input video, at intervals of 2000 frames due to computational costs and, if the animal stays in the same place for a while, it will be assumed to belong to the background, hindering its segmentation. Figure 7.3 C shows clearly the presence of the animal in the background model in windows where the animal spends many time in the same place (C3), contrarily to windows where the animal is always walking, not being recognized as a background object (C1).

K-Means algorithm demonstrated to be able to detect correctly the rat in all the frames; however, concerning the segmentation, for higher threshold values, this algorithm demonstrated to be less robust than the Mean BM and Min BM methodologies.

For higher values of thresholds, the Min BM method showed to be the one with results most similar with the ground truth however, this algorithm fails completely in the detection of the animal in frames where there is large background fluctuations. For example, in frames where the researcher interferes with the animal, when there is a contamination of the platform by the animal or even when a box is placed in the middle of the video to confine the animal to the region captured by the camera, this method is not able to detect the rat because it assumes that the animal is the only bright object in movement. Figure 7.4 shows an example where the Minimum Background Model fails, not being able to comprise background fluctuations. From A to C it is possible to see that the background changed, leading to the failure of the algorithm.

Thus, for the dataset characteristics, the Mean BM demonstrated to be the algorithm that presents the most similar results with the ground truth.

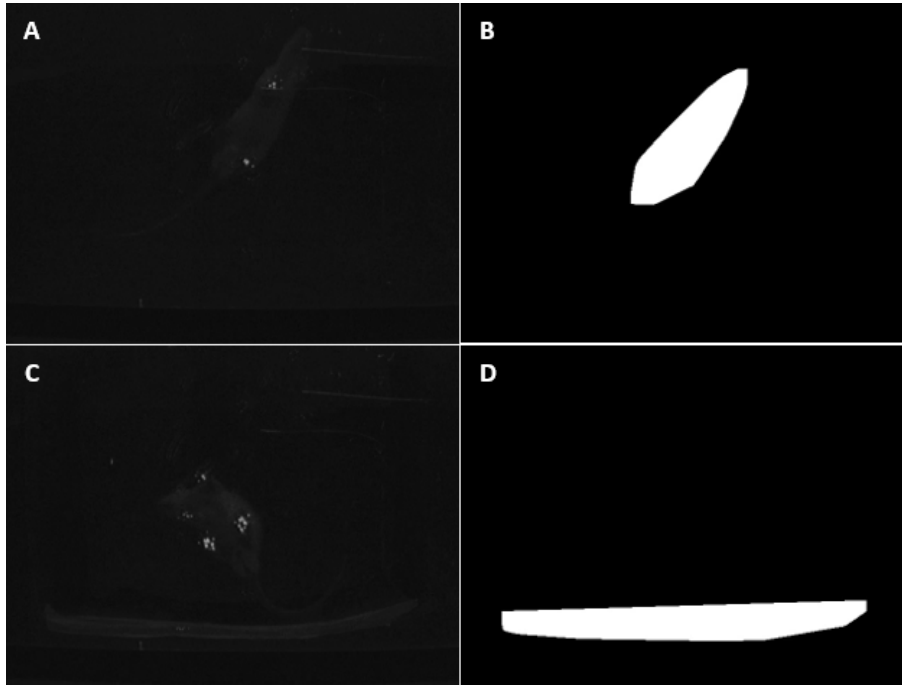


Figure 7.4: Min background mouse detection examples. **A:** Frame number 1823; **B:** Correct mouse position prediction; **C:** Frame number 5951; **D:** Incorrect mouse position prediction.

7.2 Paws Detection, Segmentation and Classification

As the ultimate goal of this work is to quantify the rat's gait metrics, such as the area and the mean intensity of the paws, it is important to evaluate paws segmentation by using pixel-based metrics since the algorithm must be meticulous enough to return the referred gait metrics efficiently. While the true positives (TP) return the number of correctly detected paws pixels, the true negatives (TN) give the number of background pixels correctly detected. In contrast, the false positives (FP) and the false negatives (FN) are pixels that are falsely recognized as foreground and background, respectively. Two examples of these metrics are presented in Figure 7.5.

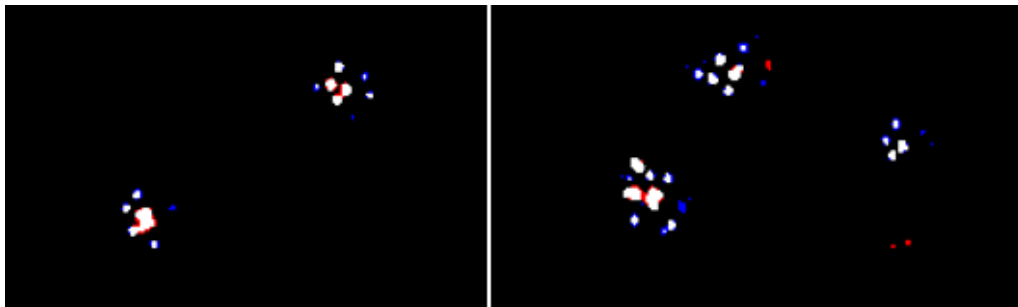


Figure 7.5: Paws' segmentation examples: **White:** TP pixels; **Black:** TN pixels; **Blue:** FP pixels; **Red:** FN pixels

From these metrics, the true positive rate (TPR), false positive rate (FPR), precision (P) and F-score were computed. The TPR reports how frequently the algorithm correctly detects the paws pixels, being given by $TPR = TP/(TP + FN)$. The FPR refers to the number of times the paws are falsely detected and it is given by $FPR = FP/(FP + TN)$. The F-score measure combines TPR and precision, given by $P = TP/(TP+FP)$, by their harmonic mean being a measure of the algorithm's accuracy.

The results of paws' segmentation are presented in Table 7.1.

Table 7.1: Results of paws' segmentation.

Method	TRP	FPR	Precision	F-score
K-Means (with pre-processing)	0,85 \pm 0,06	0,00 \pm 0,00	0,72 \pm 0,12	0,77 \pm 0,09
K-Means (without pre-processing)	0,61 \pm 0,15	0,00 \pm 0,00	0,96 \pm 0,07	0,73 \pm 0,10

In general, one can say that the results for the segmentation of the paws look promising. Comparing both non pre-processed and pre-processed data, it can be concluded that the pre-processing step introduces an improvement in the algorithm's performance. The fact that the precision value is close to 1 in the K-Means without the pre-processing and that the TPR value is 0,61 means that this algorithm, although precised, is not sensitive to the paws' pixels, having a large number of FN. The opposite happens to the K-Means with the pre-processing: the lower precision value, indicates that the algorithm detects a high number of FP being less precise but more sensitive. As far as false positives are concerned, there are two main reasons behind them: they correspond to the pixels of the rat's belly, tail or other body part of the animal when in contact with the platform; or, since the threshold defined by the expert is very specific, some pixels of the paws are not annotated as such.

The results obtained for the paws' classification are presented in Table 7.2. In the same way that the pre-processing improves paws' segmentation, it also improves their classification. A frame is only considered well classified when it detects exactly the same paws as the ones present in the GT image, in the same region of the image. From the 32% of frames that the classification of frames fails, about one-third is because the algorithm fails in detecting the animal's orientation. This happens when the rat is partially occluded, when the rat segmentation is not meticulous, when the tail is not well removed or when the animal is near a bright object. The remaining frames' classification fails when the rat has its belly or other part of the body in contact with the platform, enhancing some non-paws derived pixels, or when the two fore paws are very close to each other and the clustering algorithm assumes that it is only one footprint. When comparing the obtained results for each paw, the RH and LH paws' classification is correct in 82% and 85% of the frames, respectively, while RF and LF are successfully classified in 78% and 76% of the frames, respectively.

Table 7.2: Results of paws' classification.

Method	Frames well classified	Paws well classified
K-Means (with pre processing)	68%	81%
K-Means (without pre processing)	48%	75%

7.3 Frame Selection

In order to evaluate the algorithm's ability to return frames of interest, the global metrics were computed for each video and compared with the ones obtained by the researchers through the manual process, our ground truth. The global metrics used by the scientists are the mean percentage of pressure applied by the injured paw (RH) for all the frames globally (G) and for the walking (W) and the standing still (SS) frames, separately. By comparing these metrics, it is possible to assess the algorithm reliability to return frames of interest.

Table 7.3 shows the mean percentage of pressure applied by the RH paw obtained for each video without and with a manual selection of frames as well as the ground truth (GT) results. While results without selection comprise all frames returned by the algorithm, results with selection encompass only the well segmented and well classified frames which are manually selected.

Table 7.3: Results of frame selection. Percentage of RH pressure computed: **G**: for all the frames; **W**: for the walking frames; **SS**: for the standing still frames.

	Without Selection			With Selection			GT		
Video	G	W	SS	G	W	SS	G	W	SS
Rato 1 T7	41,24	41,27	41,20	41,88	41,74	42,01	34,53	36,14	32,91
Rato 1 T14	43,09	44,16	42,02	42,32	44,09	40,55	38,35	39,94	36,76
Rato 1 T21	41,43	49,11	33,75	37,36	44,94	29,77	36,50	41,52	31,48
Rato 2 BL	53,45	53,93	52,96	55,03	54,50	55,55	46,83	50,42	43,25
Rato 2 T7	24,42	24,88	23,95	18,51*	24,88*	12,13*	16,22	23,58	8,85
Rato 2 T14	41,63	40,90	42,36	40,28	38,20	42,36	39,47	38,58	40,36
Rato 2 T21	29,38	-**	29,38	30,47	-**	30,47	20,93	-***	20,93
Rato 3 BL	50,66	46,62	54,7	48,74	47,00	50,47	46,14	48,26	44,02
Rato 3 T7	40,02	43,54	36,50	38,14	39,59	36,68	35,28	35,54	35,02
Rato 3 T14	44,59	49,79	39,38	42,83	49,79*	35,86	43,03	44,60	41,47
Rato 3 T21	36,88	40,07	33,05	41,35	43,73	38,97	35,59	38,28	32,89
Rato 4 BL	50,13	48,40	51,86	51,41	46,26	56,55	50,69	47,89	53,49
Rato 4 T7	37,81	36,74	38,88	36,00	35,84	36,15	37,01	37,84	36,28
Rato 4 T14	42,12	45,78	38,53	41,51	41,72	41,29	36,74	42,93	30,56
Rato 4 T21	24,23	9,62	38,83	37,81	-**	37,81	31,12	43,82	28,58

* number of frames lower than the GT.

** algorithm was not able to detect pairs of walking frames.

*** researchers were not able to detect pairs of walking frames.

On Table 7.4, the mean absolute difference and the standard deviation between the obtained results without any selection (R_T) and the GT results (R_{GT}) and between results manually selected (R_S) and R_{GT} are presented.

Table 7.4: Comparison of the obtained results with the GT.

	Global	Walking	Standing Still
Mean($R_{GT} - R_T$)	5,89 \pm 4,94	5,67 \pm 8,23	5,86 \pm 4,33
Mean($R_{GT} - R_S$)	4,77 \pm 2,74	3,06 \pm 1,76	5,64 \pm 3,67

By looking into Table 7.4, it is possible to conclude that giving the user the possibility to select the final frames of interest, besides preventing erroneous results, it approaches the global metrics to those obtained in the GT.

In addition to the quantitative analysis, it is important to evaluate this step qualitatively, to understand the algorithm's ability to discard missegmented and misclassified frames, retaining only frames of interest. This analysis was done through a visual inspection of the obtained results for the 15 videos of the dataset presented in the Table 7.4. In between these videos, it was possible to obtain at least the same number of frames used by the experts in the manually way (three pairs of frames walking and three frames of the animal standing still), in 13 videos, meaning that, in 2 videos the algorithm had a global performance unable to analyze the gait of the animal. On these 2 videos, the returned frames had the paws misclassified or missegmented being unable to return reliable metrics of the animal's gait.

In videos where it is possible to analyze the the rat's gait, the algorithm is capable of returning 2 to 3 times more frames than the ones currently used by specialists, increasing the reliability of the results, since 9 frames per video may not be representative enough of the extracted metrics.

7.4 Gait Metrics Extraction

To assess the performance of the proposed methodology to extract the area and the mean intensity of each paw, these metrics were compared to the ones obtained by the experts. This comparison was only made in frames where the paws were correctly detected, segmented and classified in order to understand if, when the algorithm works correctly, it returns metrics similar to those of the researchers. In this way, of the 130 labeled frames, 78 were used to evaluate this task.

The absolute difference between the area (A) and mean intensity (MI) obtained by the proposed framework (A_T and MI_T) and by the researchers (A_{GT} and MI_{GT}) was computed for each one of the paws. In total, 156 paws were used to evaluate this step. The mean and the standard deviation of this difference were computed for the paws, globally, and also for walking and standing still frames, individually.

Since the final metric of researchers' interest is the percentage of pressure applied by the injured hind paw (the RH paw in this case - $\%RH$), this metric was computed for each standing still frame and for each pair of walking frames. Once again, by computing the absolute difference between the value obtained by the algorithm and the value obtained by the experts and calculating the mean and the standard deviation of this difference, it is possible to see if the algorithm returns a final value similar to the obtained by the researchers, thus evaluating its performance. Results are presented in Table 7.5.

Table 7.5: Results of gait metrics extraction.

	Global	Walking	Standing Still
$\text{Mean}(A_{GT} - A_T)$	$22,54 \pm 20,08$	$17,46 \pm 14,20$	$26,89 \pm 23,11$
$\text{Mean}(MI_{GT} - MI_T)$	$7,39 \pm 6,98$	$7,46 \pm 6,27$	$7,33 \pm 7,53$
$\text{Mean}(\%RH_{GT} - \%RH_T)$	$3,44 \pm 4,25$	$2,39 \pm 2,05$	$4,35 \pm 5,31$

Although the area presents a considerable difference with the GT, the final metric (the percentage of pressure applied by the RH paw) does not. This may be due to the fact that the paws segmentation algorithm has a tendency to have a significant number of FP pixels presenting thus a higher value of area. However, since this number is not too high and as it happens to all the paws and the mean intensity is not that different, the final value of RH pressure obtained does not differ too much from the GT value.

Results show that the algorithm's performance for the gait metrics extraction is better in the walking frames than in the standing still frames.

Chapter 8

Development of a Web Tool

As part of this work, the development of a graphic user interface tool was made with the purpose of implementing the framework proposed in Chapter 6 as a free service available to researchers. With the development of this tool, it is expected that the final product is more effective than the current adopted methodology (see subsection 3.3.1) giving reliable results, being easier, faster and user independent. The requirements of this tool also include an user friendly and intuitive interface that allows the user to revisit the results and, ideally, it should be able to receive some user feedback in order to review the results according to the user input.

The implementation of this platform can be done on a web or desktop application and, comparing both, web applications present some advantages over desktop applications once they avoid the burden of deploying the tool on each client machine, they don't have to enforce version check on the client device, updates are easier as well as their support and maintenance, and they can be accessed from anywhere.¹ Taking this into account, a web tool was developed in order to meet the fundamental requirements previously mentioned.

The development of the web tool was achieved by using the Django Framework. The main functionalities of this application include video uploading and processing, visualization of the results, selection of the frames of interest (giving the user the final decision power) and downloading the final gait metrics (area, mean intensity, pressure and the relative pressure of each hind paw as well as the global video metrics) into an excel file (Figure 8.1).

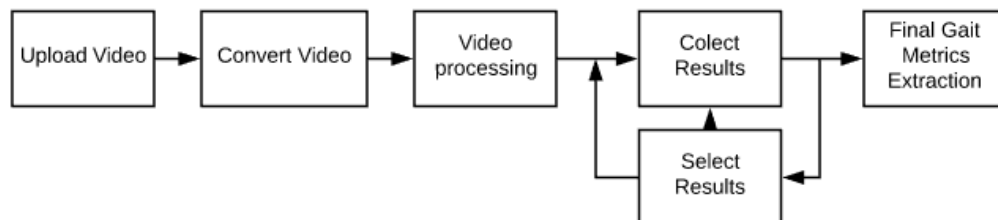


Figure 8.1: Pipeline for the proposed web application.

¹<https://momook.com/web-based-vs-desktop-software-which-is-better/>

8.1 Django Framework Architecture

Django is a high-level open source Web Development Framework based on Python. This framework follows the Model-View-Template (MVT) architectural pattern where the Model determines the data, the Template defines how the data is presented and the View acts as the interface between the Model, the Template and the User.² By looking into Figure 8.2, it is possible to understand how the data flows through a Django framework: the user requests something on a web page, the request is received by the web server being hand off to Django, which uses the defined URL path to pick the view of use. The view has the information of which models are needed and passes it to the template that is used to generate the HTML page, being displayed to the user [196].

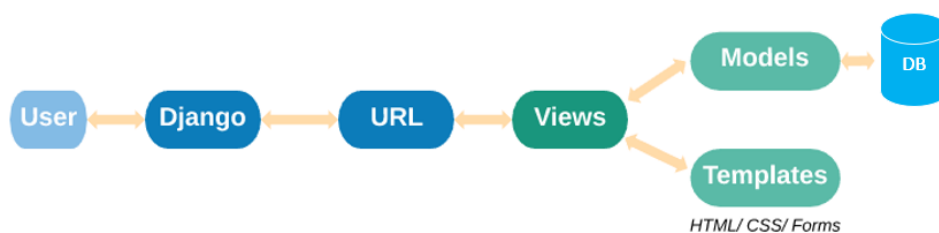


Figure 8.2: Django Model-View-Template Architecture.

Figure 8.3 shows how this web application is structured.

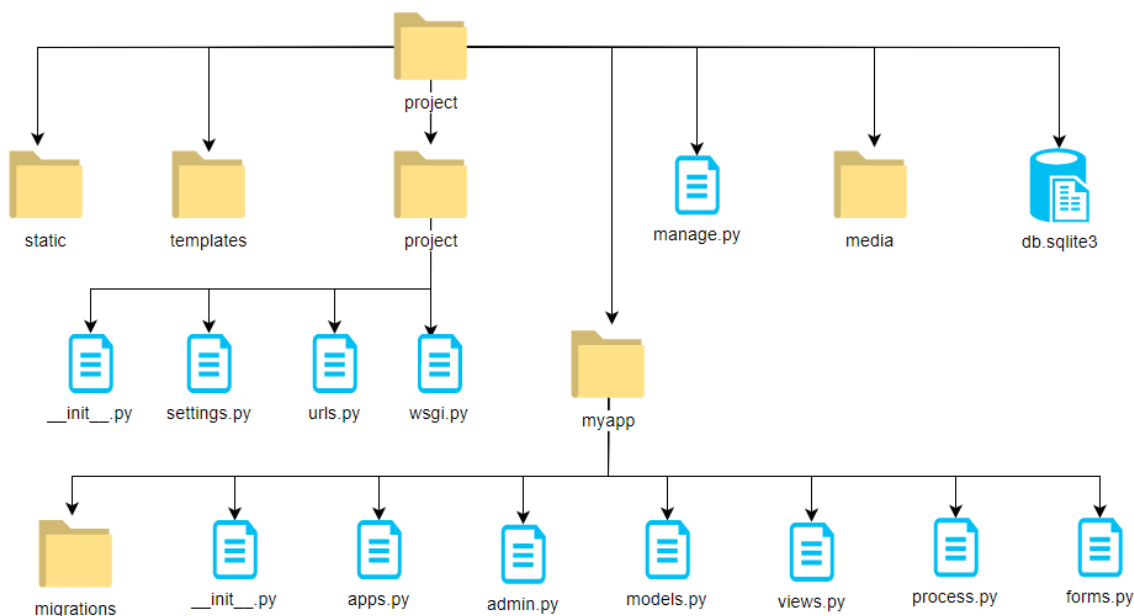


Figure 8.3: Django application layout structure.

²<https://thegeekswatch.blogspot.com>

8.2 Database

The pipeline illustrated in Figure 8.1 requires a database so that information can be permanently stored and revisited after the process. A simplified diagram of the database is shown in Figure 8.4.

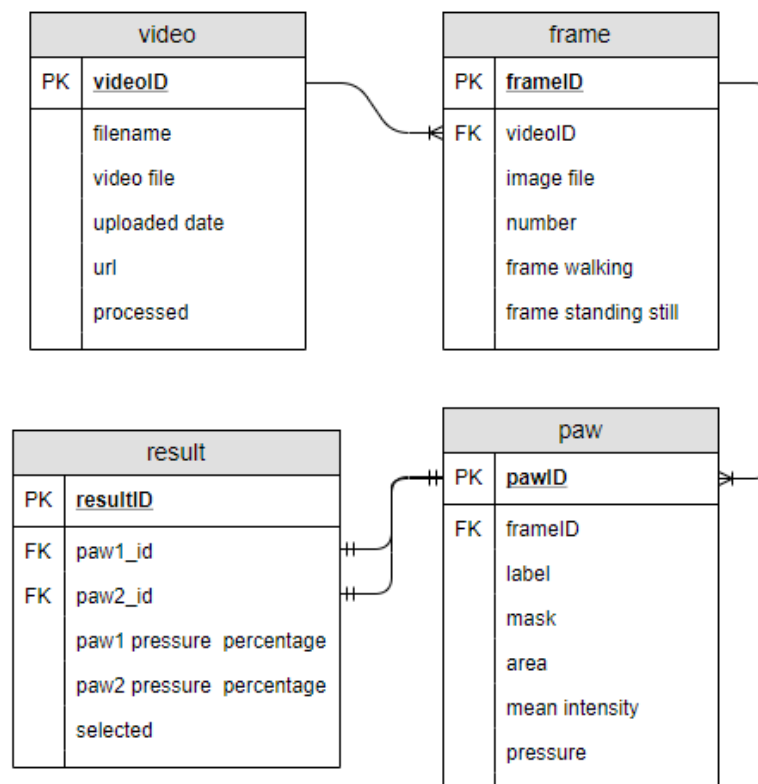


Figure 8.4: Simplified diagram of the database for the rat's gait analysis project.

When the user uploads a video, this is saved into the database and its information such as the filename, the upload date and time and its url are also preserved. The `processed` field is a boolean variable that by default is `False`, turning to `True` only when the user orders the video processing action and the video is entirely processed. When this action is completed, firstly, the frames of interest are saved into the database with information regarding their corresponding video, number and type (`frame walking` and `frame standing still` are two boolean fields that become `True` depending on the type of frame). Secondly, for each frame, the hind paws present on the frame are also updated into the database with information about their corresponding frame, label, mask, area, mean intensity and pressure.

The `result` table allows to save information about the two hind paws of interest, whether they belong to the same frame (standing still) or not (walking frame). It also contains information about the percentage of pressure applied by each one of the paws. The `selected` field is a

boolean which works analogously to the `processed` field from the `Video` table: it is `False` by default, turning `True` when the user selects it for the final set of the results of interest.

8.3 Interface and Functionalities

Gathered the goals and requirements for the web platform and explained the database, one can better understand the actions performed in each step of the process while using the web platform. The use case diagram, presented in Figure 8.5, allows to better understand the interactions between the Web App System, the User and the Database.

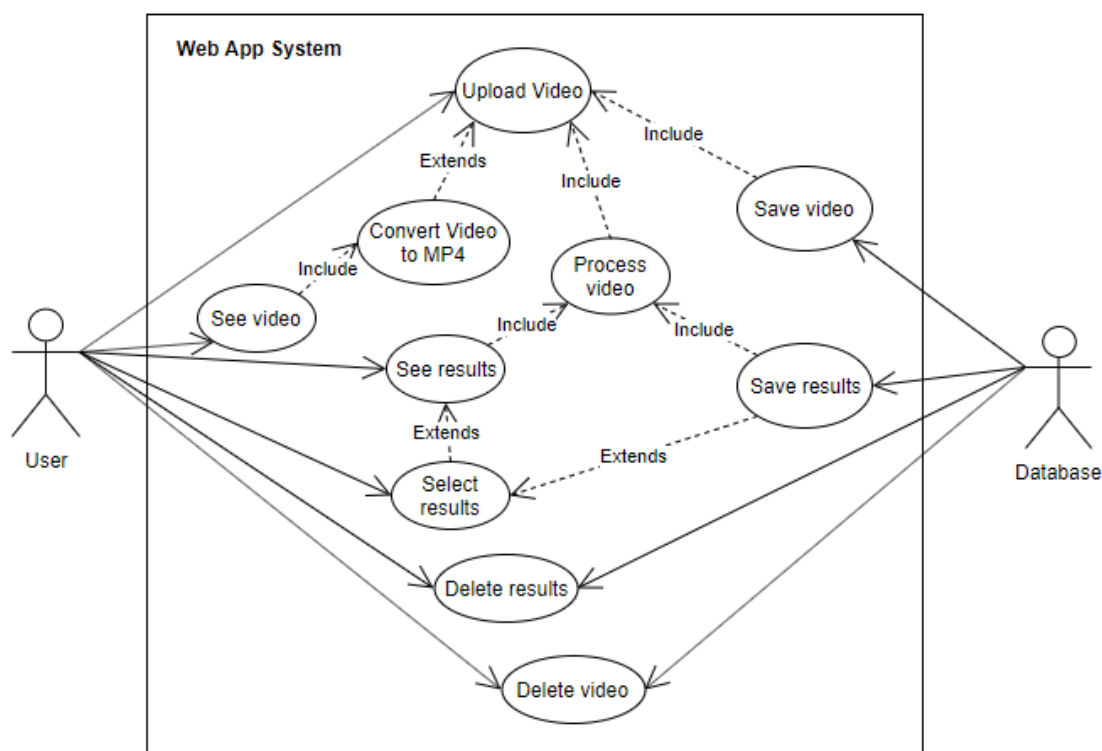


Figure 8.5: Use Case diagram of the Web Application System.

8.3.1 Video Upload

One of the required features of the proposed web application is to upload a video which it will be further processed. To do so, changes in the `settings.py` file had to be performed in order to create a media directory in the root directory of the project, which will be where the uploaded files will be stored.

Once the set up is done, it becomes possible to upload the video file by pressing the `Upload Video` button at the Home Page (Figure 8.6).

After this action, the user is redirected to another URL where he can press the `Escolher arquivo` button and select from his computer one file of any video format. Then, the user can upload it by pressing the `Upload` button (Figure 8.7). This action will save the video into the database and redirect the user to the home page where it will appear the list of all the uploaded files (Figure 8.6).

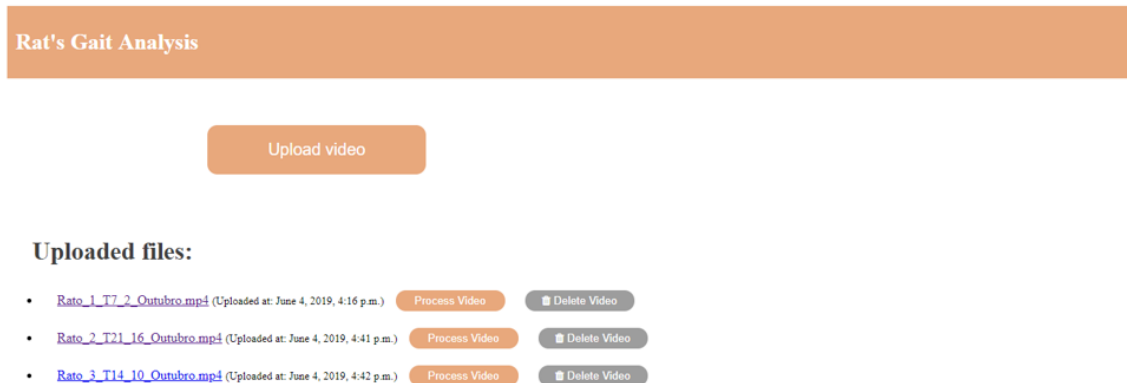


Figure 8.6: Initial home page.



Figure 8.7: Video upload functionality.

8.3.2 Video Conversion

Once the `Upload` button is pressed, if the chosen file is not a MP4 file, `ffmpeg` will run on the back end to convert the uploaded video to a MP4 file. In this way, when the user presses on the file's name, it is possible to display the video on the browser. This action, although not essential, is an example of a complementary feature that may be useful to the user and enriches the web platform.

8.3.3 Video Processing

To process the video the user has to press the `Process Video` button at the main page. This action takes some time since all the framework described in Chapter 6 will run in order to return the results. Since the Django framework is based on Python, there are not problems related with code incompatibility.

8.3.4 See Results

When the video finishes processing, the user is automatically redirected to the results page where, results are displayed in a catalogued way: first, the images with the obtained masks for the paws are displayed followed by the paw's label, area, mean intensity, pressure and its percentage of pressure (Figure 8.8). At the beginning of this page, the video global metrics will appear for all the selected results. These metrics include the mean pressure of both hind paws for the global results and for the walking and the standing still results, separately.

To return to the main page, the user has to press in the page header. On this page, a See Results button will appear ahead of the videos already processed (as well a Delete Results button), as it can be seen in Figure 8.9. Once pressed, this button redirects the user to the page where the results are displayed.

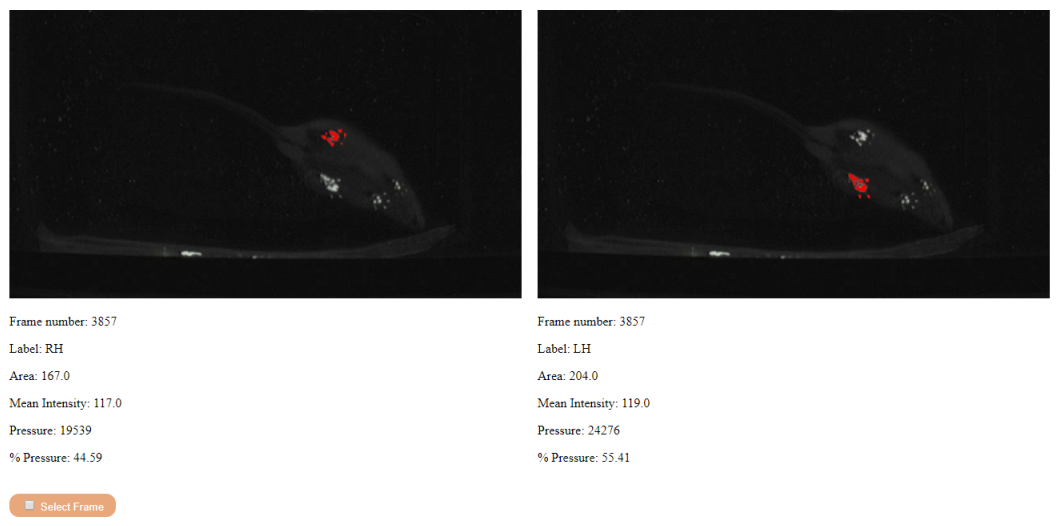


Figure 8.8: Example of a result display.

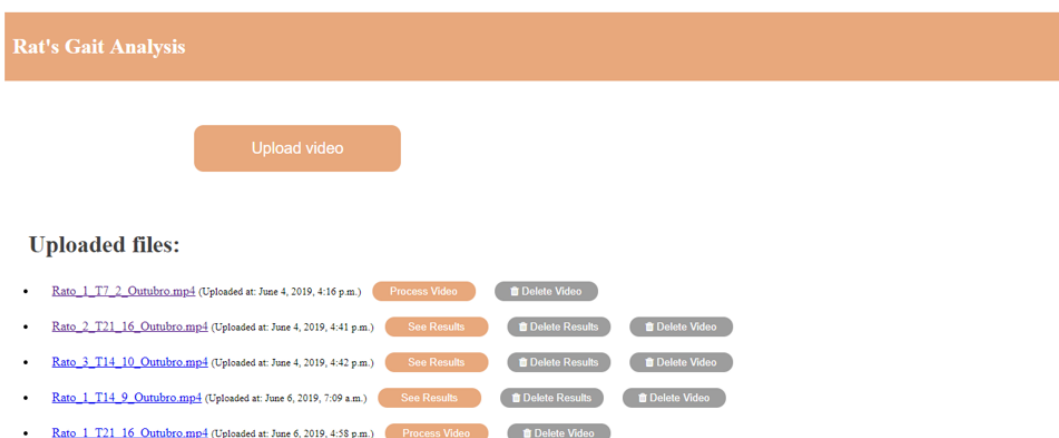


Figure 8.9: Main page with videos already processed.

8.3.5 Select Results

As previously explained, it is given to the user the possibility of selecting, in between the results returned by the algorithm, the final frames of interest. This task allows to make sure that the final metrics correspond to frames with the paws well-segmented and well-classified, giving the user the power to have the final say on the selection of frames, preventing erroneous results. For this reason, at the page where the results are displayed, there is an option for each one of the displayed results to select it or deselect it.

After selecting all the frames of interest, the user must press the `Select Frames` button at the end of the page (Figure 8.10). This button turns `True` the `selected` field of each one of the picked results and it refreshes the display results page, updating the global metrics that appear at the beginning of the page.

The user can update the set of results of interest by selecting and/or deselecting frames according to his interest. To select all the frames returned by the algorithm, the user just needs to press the `Select All frames` button at the end of the page.

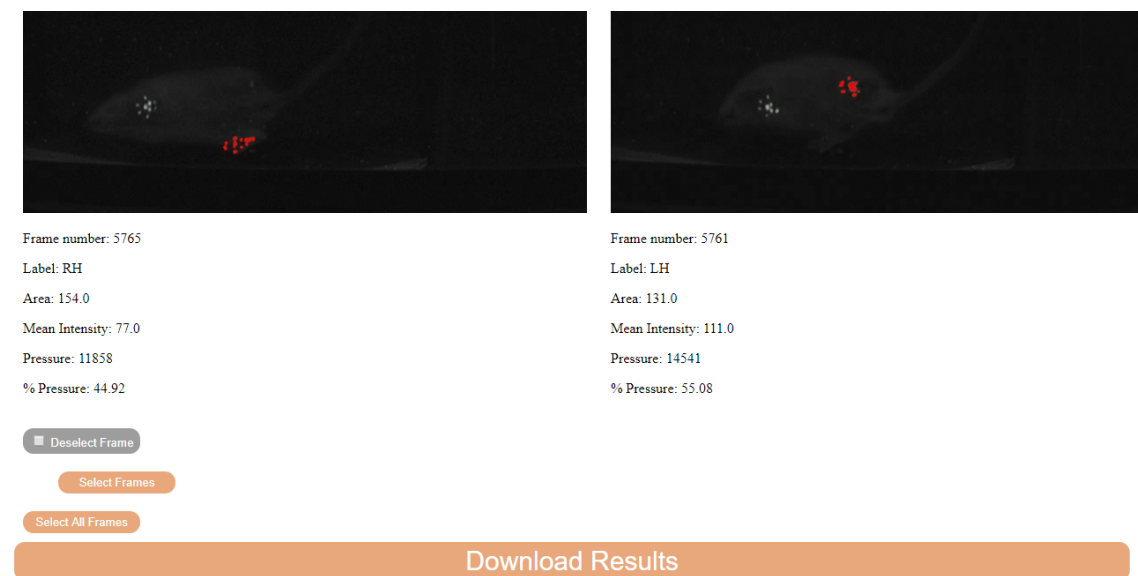


Figure 8.10: End of the display results page.

8.3.6 Download Results

Another functionality of the platform consists in downloading the results to an excel file so that the user can store the results on his computer. To do this, the `Download Results` button that is present at the end of the results page must be pressed (Figure 8.10). In this way, the selected results will be saved into an excel file that has the name of the video and it is organized like as shown in Figure 8.11.

RH - Global	46,3	LH - Global	53,7
RH - Walking	44,2	LH - Walking	55,8
RH - Standing Still	48,4	LH - Standing Still	51,6

Video Name	Type of Frame	Frame Number	Paw Label	Area	Mean Intensity	Pressure	% Pressure
Rato_3_T14_10_Outubro	SS	252	RH	165	80	13200	33,91
			LH	218	118	25724	66,09
Rato_3_T14_10_Outubro	SS	1545	RH	207	106	21942	67,77
			LH	147	71	10437	32,23
Rato_3_T14_10_Outubro	SS	2460	RH	167	92	15364	45,26
			LH	163	114	18582	54,74
Rato_3_T14_10_Outubro	SS	3483	RH	105	89	9345	23,31
			LH	248	124	30752	76,69
Rato_3_T14_10_Outubro	SS	3857	RH	167	117	19539	44,59
			LH	204	119	24276	55,41
Rato_3_T14_10_Outubro	SS	3878	RH	167	114	19038	43,42
			LH	212	117	24804	56,58
Rato_3_T14_10_Outubro	W	1358	RH	155	109	16895	53,85
			LH	127	114	14478	46,15
Rato_3_T14_10_Outubro	W	3833	LH	150	116	17400	54,27
			RH	137	107	14659	45,73




Figure 8.11: Example of a downloaded excel file.

8.3.7 Delete Video and Delete Results

The proposed web tool also allows the deletion of the results and videos from the database, an useful task when experiences are over to free up some space of the app. To do this, the user simply has to press the buttons `Delete Results` and/or `Delete Video` at the main page as desired (see Figure 8.9).

Chapter 9

Conclusions and Future Work

In this dissertation, an automated method for the assessment of the rat's gait is presented, with the aim of not only providing data that is less prone to user variability but also alleviating the work of scientists who resort to the analysis of the gait of animal models to evaluate the effectiveness of drugs used in the treatment of arthritis. The contributions include an annotated dataset publicly available to the scientific community, a published paper in the ninth edition of the IbPRIA conference, an automated framework capable of analyzing the rat's gait and a web platform deploying the proposed framework.

Concerning the adopted methodology to automatically detect and quantify gait changes in the rat, this met the proposed goal being able of detecting and segmenting the animal's body, detecting, segmenting and classifying its paws, and of selecting some frames of interest and extracting some gait metrics from them, making the detection and quantification of the animal gait changes an easier, faster, simpler, more robust and less susceptible to user variability task.

In terms of performance, all the steps of the designed methodology were evaluated individually. For that, 130 frames were manually annotated. Starting with the detection and segmentation of the animal's body, in between all the adopted methodologies, the offline background mean model proved to be one with the best results being capable of detecting the animal in every frame and giving the most similar results with the ground truth. Regarding the paws segmentation task, some interesting results are presented by applying the K-means with a previous pre-processing step. This operation results in a TPR of $85 \% \pm 0,06$ and a F-score of $77 \% \pm 0,09$. Quantitative and visual inspections of the results of paws' segmentation demonstrate a good performance of the algorithm. In what concerns paws classification, in 68% of the frames, all the paws were well classified. With regard to the frame's selection algorithm, it was adopted a semi-automatic task giving the user the power to have the final say on the selection of frames, preventing erroneous results. This algorithm demonstrated to be able of analyzing the rat's gait in 87% of the videos of the dataset, returning 2 to 3 times more frames than the ones currently used by specialists. Concerning the gait metrics extraction task, this proved to return a final value similar to that obtained by the experts with a global mean difference of $3,44 \% \pm 4,25$. Seen this, one can say that the algorithm step that needs further improvement is the paws' classification since it demonstrated to

be the task with less promising results, impacting the following steps of the described pipeline.

Although the overall performance of the tracking-by-detection adopted methodology has proved to be effective, it can potentially be further improved by resorting to the current trend of deep learning. To do this, more annotations should be done in order to increase the dataset, making possible the data to train, test and validate more complex state of the art algorithms. In this way, future work will focus on extending the dataset and improving the algorithm's performance by resorting to more general solutions and deep learning methodologies as an alternative to the adopted framework.

The development of an user friendly web application tool, where it is possible to implement the proposed framework, makes the proposed methodology available as a free service to researchers. Current functionalities of the platform include video uploading and processing, visualization of the results, selection of the frames of interest (giving to the user the final decision) and downloading the results into an excel file. The interface was designed to be simple and intuitive and aiming to make the manual video analysis a simpler and faster process.

Regarding the continuation of this work, since the web platform will be available for the researchers, future work can also include an analysis of the interface and its features in order to meet in more detail the researchers' needs. In the future, new features could be added to the web tool so that it is possible for example to have a personal page for each user, in order to make the platform more private.

Additionally, there is also the possibility of rethinking the gait analysis itself and extracting other metrics that may be useful to researchers.

References

- [1] R. Chambers and M. Schmid, “Making technology-enabled health care work in general practice,” 2018.
- [2] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [3] T. Tolxdorff, T. Deserno, H. Handels, and H.-P. Meinzer, “Advances in medical image computing,” *Methods of information in medicine*, vol. 48, no. 04, pp. 311–313, 2009.
- [4] K. Doi, “Computer-aided diagnosis in medical imaging: historical review, current status and future potential,” *Computerized medical imaging and graphics*, vol. 31, no. 4-5, pp. 198–211, 2007.
- [5] J. K. Willmann, N. Van Bruggen, L. M. Dinkelborg, and S. S. Gambhir, “Molecular imaging in drug development,” *Nature reviews Drug discovery*, vol. 7, no. 7, p. 591, 2008.
- [6] Y.-X. Wang and M. Deng, “Medical imaging in new drug clinical development,” *Journal of thoracic disease*, vol. 2, no. 4, p. 245, 2010.
- [7] J. A. Sturgeon, A. J. Zautra, and A. Arewasikporn, “A multilevel structural equation modeling analysis of vulnerabilities and resilience resources influencing affective adaptation to chronic pain,” *Pain®*, vol. 155, no. 2, pp. 292–298, 2014.
- [8] R. Alonso, “Effect of *sprr1a* overexpression on the nociceptive behaviour of oa animals,” 2017.
- [9] H. Hedrich and G. Bullock, “The laboratory mouse: Handbook of experimental animals,” *London: Elsevier Limited*, 2004.
- [10] S. D. Pathak, L. Ng, B. Wyman, S. Fogarasi, S. Racki, J. C. Oelund, B. Sparks, and V. Chalana, “Quantitative image analysis: software systems in drug development trials,” *Drug discovery today*, vol. 8, no. 10, pp. 451–458, 2003.
- [11] E. H. Lakes and K. D. Allen, “Gait analysis methods for rodent models of arthritic disorders: reviews and recommendations,” *Osteoarthritis and cartilage*, vol. 24, no. 11, pp. 1837–1849, 2016.
- [12] A. F. Adonias, J. S. Cardoso, J. Ferreira-Gomes, F. Neto, and R. Alonso, “Towards automatic rat’s gait analysis under suboptimal illumination conditions,” in *Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2019.
- [13] Y. Zhou, T. Wang, J. L. Hamilton, and D. Chen, “Wnt/ β -catenin signaling in osteoarthritis and in other forms of arthritis,” *Current rheumatology reports*, vol. 19, no. 9, p. 53, 2017.

- [14] OpenStax, *Anatomy and Physiology*. Rice University, 2013.
- [15] H. Gray, *Anatomy of the human body*, vol. 8. Lea & Febiger, 1878.
- [16] A. Foundation, “Arthritis by the numbers,” *Arthritis Foundation*, 2018.
- [17] V. C. Mow, A. Ratcliffe, and A. R. Poole, “Cartilage and diarthrodial joints as paradigms for hierarchical materials and structures,” *Biomaterials*, vol. 13, no. 2, pp. 67–97, 1992.
- [18] D. T. Felson, “The epidemiology of knee osteoarthritis: results from the framingham osteoarthritis study,” in *Seminars in arthritis and rheumatism*, vol. 20, pp. 42–50, Elsevier, 1990.
- [19] J. Ferreira-Gomes, S. Adães, and J. M. Castro-Lopes, “Assessment of movement-evoked pain in osteoarthritis by the knee-bend and catwalk tests: a clinically relevant study,” *The Journal of Pain*, vol. 9, no. 10, pp. 945–954, 2008.
- [20] J. A. Collins, B. O. Diekmann, and R. F. Loeser, “Targeting aging for disease modification in osteoarthritis,” *Current opinion in rheumatology*, vol. 30, no. 1, pp. 101–107, 2018.
- [21] J. Sokolove and C. M. Lepus, “Role of inflammation in the pathogenesis of osteoarthritis: latest findings and interpretations,” *Therapeutic advances in musculoskeletal disease*, vol. 5, no. 2, pp. 77–94, 2013.
- [22] I. B. McInnes and G. Schett, “The pathogenesis of rheumatoid arthritis,” *New England Journal of Medicine*, vol. 365, no. 23, pp. 2205–2219, 2011.
- [23] S. E. Sweeney and G. S. Firestein, “Rheumatoid arthritis: regulation of synovial inflammation,” *The international journal of biochemistry & cell biology*, vol. 36, no. 3, pp. 372–378, 2004.
- [24] J. B. Duarte Pereira, Elisabete Ramos, “Osteoarthritis,” *Acta Med Port*, 2014.
- [25] D. J. Hunter, D. Schofield, and E. Callander, “The individual and socioeconomic impact of osteoarthritis,” *Nature Reviews Rheumatology*, vol. 10, no. 7, p. 437, 2014.
- [26] W. H. Organization, *World report on ageing and health*. World Health Organization, 2015.
- [27] W. H. Organization *et al.*, “Background paper 6.12 osteoarthritis,” *World Health Organization*, http://www.who.int/medicines/areas/priority_medicines/BP6_12Osteo.pdf.
- [28] J. Freeman, “Ra facts: What are the latest statistics on rheumatoid arthritis?,” 2018.
- [29] L. C. Miranda, H. Santos, J. Ferreira, P. Coelho, C. Silva, J. Saraiva-Ribeiro, *et al.*, “Finding rheumatoid arthritis impact on life (frail study): economic burden,” *Acta reumatologica portuguesa*, vol. 37, pp. 134–142, 2012.
- [30] R. A. Benson, I. B. McInnes, P. Garside, and J. M. Brewer, “Model answers: Rational application of murine models in arthritis research,” *European journal of immunology*, vol. 48, no. 1, pp. 32–38, 2018.
- [31] E. L. Kuyinu, G. Narayanan, L. S. Nair, and C. T. Laurencin, “Animal models of osteoarthritis: classification, update, and measurement of outcomes,” *Journal of orthopaedic surgery and research*, vol. 11, no. 1, p. 19, 2016.

- [32] J. Ferreira-Gomes, S. Adães, R. M. Sousa, M. Mendonça, and J. M. Castro-Lopes, “Dose-dependent expression of neuronal injury markers during experimental osteoarthritis induced by monoiodoacetate in the rat,” *Molecular pain*, vol. 8, no. 1, p. 50, 2012.
- [33] S. Adães, M. Mendonça, T. N. Santos, J. M. Castro-Lopes, J. Ferreira-Gomes, and F. L. Neto, “Intra-articular injection of collagenase in the knee of rats as an alternative model to study nociception associated with osteoarthritis,” *Arthritis research & therapy*, vol. 16, no. 1, p. R10, 2014.
- [34] L. G. Ameye and M. F. Young, “Animal models of osteoarthritis: lessons learned while seeking the ‘holy grail’,” *Current opinion in rheumatology*, vol. 18, no. 5, pp. 537–547, 2006.
- [35] A. Bendele, “Animal models of osteoarthritis,” *J Musculoskelet Neuronal Interact*, vol. 1, no. 4, pp. 363–76, 2001.
- [36] B. Y. Jacobs, H. E. Kloefkorn, and K. D. Allen, “Gait analysis methods for rodent models of osteoarthritis,” *Current pain and headache reports*, vol. 18, no. 10, p. 456, 2014.
- [37] J. M. Williams, J. Zurawski, K. Mikecz, and T. T. Glant, “Functional assessment of joint use in experimental inflammatory murine arthritis,” *Journal of orthopaedic research*, vol. 11, no. 2, pp. 172–180, 1993.
- [38] I. Amende, A. Kale, S. McCue, S. Glazier, J. P. Morgan, and T. G. Hampton, “Gait dynamics in mouse models of parkinson’s disease and huntington’s disease,” *Journal of neuroengineering and rehabilitation*, vol. 2, no. 1, p. 20, 2005.
- [39] P. Thoumie, D. Lamotte, S. Cantalloube, M. Faucher, and G. Amarenco, “Motor determinants of gait in 100 ambulatory patients with multiple sclerosis,” *Multiple Sclerosis Journal*, vol. 11, no. 4, pp. 485–491, 2005.
- [40] K. J. Duberstein, S. R. Platt, S. P. Holmes, C. R. Dove, E. W. Howerth, M. Kent, S. L. Stice, W. D. Hill, D. C. Hess, and F. D. West, “Gait analysis in a pre-and post-ischemic stroke biomedical pig model,” *Physiology & behavior*, vol. 125, pp. 8–16, 2014.
- [41] A. Muro-De-La-Herran, B. Garcia-Zapirain, and A. Mendez-Zorrilla, “Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications,” *Sensors*, vol. 14, no. 2, pp. 3362–3394, 2014.
- [42] R. Rushton, H. Steinberg, and C. Tinson, “Effects of a single experience on subsequent reactions to drugs,” *British journal of pharmacology and chemotherapy*, vol. 20, no. 1, pp. 99–105, 1963.
- [43] R. B. Johnston, L. Zachary, A. L. Dellon, W. A. Seiler IV, and D. M. Teplica, “Improved imaging of rat hindfoot prints for walking track analysis,” *Journal of neuroscience methods*, vol. 38, no. 2-3, pp. 111–114, 1991.
- [44] T. Incorporated, “Pressure-sensing walkwayTM @ONLINE.” <https://www.tekscan.com/products-solutions/systems/animal-strideway-system>.
- [45] Bioseb, “Dynamic weight bearing test @ONLINE.” https://www.bioseb.com/bioseb/anglais/default/item_id=1213_cat_id=5_Advanced+Dynamic+Weight+Bearing.php.

- [46] C. W. Dorman, H. E. Krug, S. P. Frizelle, S. Funkenbusch, and M. L. Mahowald, "A comparison of digigait™ and treadscan™ imaging systems: assessment of pain using gait analysis in murine monoarthritis," *Journal of pain research*, vol. 7, p. 25, 2014.
- [47] M. Incorporated, "Digigait™ @ONLINE." <https://mousespecifics.com/digigait/>.
- [48] N. I. Technology, "Catwalk @ONLINE." <http://www.noldus.com/animal-behavior-research/products/catwalk>.
- [49] C. S. Mendes, I. Bartos, Z. Márka, T. Akay, S. Márka, and R. S. Mann, "Quantification of gait parameters in freely walking rodents," *BMC biology*, vol. 13, no. 1, p. 50, 2015.
- [50] C. S. Incorporate, "Treadscan - revolutionizing behavior research," 2017.
- [51] C. S. Incorporate, "Runwayscan - revolutionizing behavior research," 2013.
- [52] C. S. Incorporate, "Gaitscan - revolutionizing behavior research," 2010.
- [53] X. Feng, W. Mei, and D. Hu, "A review of visual tracking with deep learning," *Advances in Intelligent Systems Research*, vol. 133, pp. 231–234, 2016.
- [54] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1419–1426, IEEE, 2011.
- [55] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti, "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE signal processing magazine*, vol. 22, no. 2, pp. 38–51, 2005.
- [56] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Statistic and knowledge-based moving object detection in traffic scenes," in *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, pp. 27–32, IEEE, 2000.
- [57] R. J. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," in *The mind's eye*, pp. 573–605, Elsevier, 2003.
- [58] X. Dong, J. Liang, A. Yang, C. Wang, D. Kong, and F. Lv, "In vivo imaging tracking and immune responses to nanovaccines involving combined antigen nanoparticles with a programmed delivery," *ACS applied materials & interfaces*, 2018.
- [59] J. F. Scalisi, D. Mejia, D. M. Morse, *et al.*, "Vehicle tracking systems and methods," Apr. 3 2018. US Patent 9,933,270.
- [60] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.
- [61] S. Ågren, "Object tracking methods and their areas of application: A meta-analysis: A thorough review and summary of commonly used object tracking methods," 2017.
- [62] M. T. N. Truong and S. Kim, "Parallel implementation of color-based particle filter for object tracking in embedded systems," *Human-centric Computing and Information Sciences*, vol. 7, no. 1, p. 2, 2017.

- [63] S. Murray, "Real-time multiple object tracking-a study on the importance of speed," *arXiv preprint arXiv:1709.03572*, 2017.
- [64] R. B. Hatwar, S. D. Kamble, N. V. Thakur, and S. Kakde, "A review on moving object detection and tracking methods in video," 2018.
- [65] S. H. Shaikh, K. Saeed, and N. Chaki, "Moving object detection approaches, challenges and object tracking," in *Moving Object Detection Using Background Subtraction*, pp. 5–14, Springer, 2014.
- [66] S. K. Patel and A. Mishra, "Moving object tracking techniques: A critical review," *Indian Journal of Computer Science and Engineering*, vol. 4, no. 2, pp. 95–102, 2013.
- [67] F. Porikli and A. Yilmaz, "Object detection and tracking," in *Video Analytics for Business Intelligence*, pp. 3–41, Springer, 2012.
- [68] R. Kashyap, "Fast segmentation methods for medical images," *International Journal of Computer Applications*, vol. 156, pp. 18–23, 12 2016.
- [69] M. A.-M. Salem, A. Atef, A. Salah, and M. Shams, "Recent survey on medical image segmentation," in *Handbook of Research on Machine Learning Innovations and Trends*, pp. 424–464, IGI Global, 2017.
- [70] D. Kaur and Y. Kaur, "Various image segmentation techniques: a review," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, pp. 809–814, 2014.
- [71] A. E. Rad, M. S. Mohd Rahim, A. Rehman, A. Altameem, and T. Saba, "Evaluation of current dental radiographs segmentation approaches in computer-aided applications," *IETE Technical Review*, vol. 30, no. 3, pp. 210–222, 2013.
- [72] A. Singh, C.-H. H. Chu, M. A. Pratt, *et al.*, "Multiresolution superpixels for visual saliency detection," in *CIMSIVP*, pp. 90–97, 2014.
- [73] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [74] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. R. Soc. Lond. B*, vol. 207, no. 1167, pp. 187–217, 1980.
- [75] L. G. Roberts, *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.
- [76] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in*, pp. 271–272, 1968.
- [77] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- [78] S. Angelina, L. P. Suresh, and S. K. Veni, "Image segmentation based on genetic algorithm for region growth and region merging," in *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, pp. 970–974, IEEE, 2012.
- [79] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

- [80] A. K. T. Andu and A. S. Thanamani, “Multidimensional clustering methods of data mining for industrial applications,”
- [81] Z. Cebeci and F. Yildiz, “Comparison of k-means and fuzzy c-means algorithms on different cluster structures,” *AGRÁRINFORMATIKA/JOURNAL OF AGRICULTURAL INFORMATICS*, vol. 6, no. 3, pp. 13–23, 2015.
- [82] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.
- [83] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” 1973.
- [84] S. Ruparelia, “Implementation of watershed based image segmentation algorithm in fpga,” Master’s thesis, 2012.
- [85] J. Zhao, G. Ji, Y. Qiang, X. Han, B. Pei, and Z. Shi, “A new method of detecting pulmonary nodules with pet/ct based on an improved watershed algorithm,” *PloS one*, vol. 10, no. 4, p. e0123694, 2015.
- [86] J. Li, L. Chen, and W. Huang, “Detection of early bruises on peaches (*amygdalus persica* l.) using hyperspectral imaging coupled with improved watershed segmentation algorithm,” *Postharvest Biology and Technology*, vol. 135, pp. 104–113, 2018.
- [87] S. K. Hasan, Y. Sarkar, and M. Ahmad, “Watershed-matching algorithm: A new pathway for brain tumor segmentation,” in *Advances in Electronics, Communication and Computing*, pp. 45–52, Springer, 2018.
- [88] F. K. Patel and M. Singh, “Segmentation of cartilage from knee mri images using the watershed algorithm,” *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 2, pp. 1727–1730, 2018.
- [89] Y.-L. You and M. Kaveh, “Fourth-order partial differential equations for noise removal,” *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1723–1730, 2000.
- [90] J. Weickert, “Efficient image segmentation using partial differential equations and morphology,” *Pattern Recognition*, vol. 34, no. 9, pp. 1813–1824, 2001.
- [91] L. Álvarez, L. Baumela, P. Henríquez, and P. Márquez-Neila, “Morphological snakes,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2197–2202, IEEE, 2010.
- [92] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, “A review on deep learning techniques applied to semantic segmentation,” *arXiv preprint arXiv:1704.06857*, 2017.
- [93] A. Bhandare, M. Bhide, P. Gokhale, and R. Chandavarkar, “Applications of convolutional neural networks,” *International Journal of Computer Science and Information Technologies*, pp. 2206–2215, 2016.
- [94] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

- [95] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.
- [96] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
- [97] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.
- [98] N. Dey, P. Nandi, N. Barman, D. Das, and S. Chakraborty, "A comparative study between moravec and harris corner detection of noisy images using adaptive wavelet thresholding technique," *arXiv preprint arXiv:1209.1558*, 2012.
- [99] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," tech. rep., STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1980.
- [100] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, pp. 10–5244, Citeseer, 1988.
- [101] J. Shi and C. Tomasi, "Good features to track," tech. rep., Cornell University, 1993.
- [102] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European conference on computer vision*, pp. 128–142, Springer, 2002.
- [103] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [104] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [105] B. Heisele, "Visual object recognition with supervised learning," *IEEE Intelligent Systems*, vol. 18, no. 3, pp. 38–42, 2003.
- [106] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [107] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *null*, p. 734, IEEE, 2003.
- [108] M. Hofmann, "Support vector machines—kernels and the kernel trick," *Notes*, vol. 26, 2006.
- [109] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Computer vision, 1998. sixth international conference on*, pp. 555–562, IEEE, 1998.
- [110] E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: an application to face detection," in *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*, pp. 130–136, IEEE, 1997.

- [111] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Comparative study of background subtraction algorithms," *Journal of Electronic Imaging*, vol. 19, no. 3, p. 033003, 2010.
- [112] W. Kim and C. Jung, "Illumination-invariant background subtraction: Comparative review, models, and prospects," *IEEE Access*, vol. 5, pp. 8369–8384, 2017.
- [113] D. S. Alex and A. WAHI, "Bsfd: Background subtraction frame difference algorithm for moving object detection and extraction.," *Journal of Theoretical & Applied Information Technology*, vol. 60, no. 3, 2014.
- [114] J. Heikkilä and O. Silvén, "A real-time system for monitoring of cyclists and pedestrians," in *Visual Surveillance, 1999. Second IEEE Workshop on, (VS'99)*, pp. 74–81, IEEE, 1999.
- [115] N. Singla, "Motion detection based on frame difference method," *International Journal of Information & Computation Technology*, vol. 4, no. 15, pp. 1559–1565, 2014.
- [116] I. Kartika and S. S. Mohamed, "Frame differencing with post-processing techniques for moving object detection in outdoor environment," in *Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on*, pp. 172–176, IEEE, 2011.
- [117] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 780–785, 1997.
- [118] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *cvpr*, p. 2246, IEEE, 1999.
- [119] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *European conference on computer vision*, pp. 751–767, Springer, 2000.
- [120] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–II, Ieee, 2004.
- [121] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 8, pp. 809–830, 2000.
- [122] O. Tuzel, F. Porikli, and P. Meer, "A bayesian approach to background modeling," in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pp. 58–58, IEEE, 2005.
- [123] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.
- [124] A. Doshi and M. Trivedi, "" hybrid cone-cylinder" codebook model for foreground detection with shadow and highlight suppression," in *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*, pp. 19–19, IEEE, 2006.
- [125] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

- [126] F. De La Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 117–142, 2003.
- [127] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery," *IEEE signal processing magazine*, vol. 35, no. 4, pp. 32–55, 2018.
- [128] P. Netrapalli, U. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust pca," in *Advances in Neural Information Processing Systems*, pp. 1107–1115, 2014.
- [129] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust pca via gradient descent," in *Advances in neural information processing systems*, pp. 4152–4160, 2016.
- [130] Y. Cherapanamjeri, K. Gupta, and P. Jain, "Nearly-optimal robust matrix completion," *arXiv preprint arXiv:1606.07315*, 2016.
- [131] P. Narayanamurthy and N. Vaswani, "Nearly optimal robust subspace tracking," in *International Conference on Machine Learning*, pp. 3698–3706, 2018.
- [132] A. Rafique, A. M. Sheri, and M. Jeon, "Background scene modeling for ptz cameras using rbm," in *Control, Automation and Information Sciences (ICCAIS), 2014 International Conference on*, pp. 165–169, IEEE, 2014.
- [133] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *IEEE International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava 23-25 May 2016*, pp. 1–4, IEEE, 2016.
- [134] S. Aslani and H. Mahdavi-Nasab, "Optical flow based moving object detection and tracking for traffic surveillance," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 7, no. 9, pp. 1252–1256, 2013.
- [135] A. A. Shafie, F. Hafiz, M. Ali, *et al.*, "Motion detection techniques using optical flow," *World Academy of Science, Engineering and Technology*, vol. 56, pp. 559–561, 2009.
- [136] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2432–2439, IEEE, 2010.
- [137] T.-C. Huang, C.-K. Chang, C.-H. Liao, and Y.-J. Ho, "Quantification of blood flow in internal cerebral artery by optical flow method on digital subtraction angiography in comparison with time-of-flight magnetic resonance angiography," *PloS one*, vol. 8, no. 1, p. e54678, 2013.
- [138] N. Hata, A. Nabavi, W. M. Wells III, S. K. Warfield, R. Kikinis, P. M. Black, and F. A. Jolesz, "Three-dimensional optical flow method for measurement of volumetric brain deformation from intraoperative mr images," *Journal of Computer Assisted Tomography*, vol. 24, no. 4, pp. 531–538, 2000.
- [139] F. Amat, E. W. Myers, and P. J. Keller, "Fast and robust optical flow for time-lapse microscopy using super-voxels," *Bioinformatics*, vol. 29, no. 3, pp. 373–380, 2012.
- [140] K. Liu, S. S. Lienkamp, A. Shindo, J. B. Wallingford, G. Walz, and O. Ronneberger, "Optical flow guided cell segmentation and tracking in developing tissue," in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, pp. 298–301, IEEE, 2014.

- [141] J. Huang, W. Zou, J. Zhu, and Z. Zhu, "Optical flow based real-time moving object detection in unconstrained scenes," *arXiv preprint arXiv:1807.04890*, 2018.
- [142] R. K. Mahdi Rezaei, "Object detection, classification, and tracking," in *Computer Vision for Driver Assistance*, pp. 51–89, Springer International Publishing, 2017.
- [143] P. Singh, B. Deepak, T. Sethi, and M. D. P. Murthy, "Real-time object detection and tracking using color feature and motion," in *Communications and Signal Processing (ICCSP), 2015 International Conference on*, pp. 1236–1241, IEEE, 2015.
- [144] D. Gabor, "Theory of communication. part 1: The analysis of information," *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.
- [145] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International journal of computer vision*, vol. 43, no. 1, pp. 29–44, 2001.
- [146] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [147] E. Prakasa, "Texture feature extraction by using local binary pattern," *INKOM Journal*, vol. 9, no. 2, pp. 45–48, 2016.
- [148] T. Bouwmans, C. Silva, C. Marghes, M. S. Zitouni, H. Bhaskar, and C. Frelicot, "On the role and the importance of features for background modeling and foreground detection," *Computer Science Review*, vol. 28, pp. 26–91, 2018.
- [149] I. S. Kim, H. S. Choi, K. M. Yi, J. Y. Choi, and S. G. Kong, "Intelligent visual surveillance—a survey," *International Journal of Control, Automation and Systems*, vol. 8, no. 5, pp. 926–939, 2010.
- [150] K. Fujii, "Extended kalman filter," *Refernce Manual*, 2013.
- [151] S. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.
- [152] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–194, International Society for Optics and Photonics, 1997.
- [153] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10143–10162, 1994.
- [154] A. Kamann, J. B. Bielmeier, S. Hasirlioglu, U. T. Schwarz, and T. Brandmeier, "Object tracking based on an extended kalman filter in high dynamic driving situations," in *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, pp. 1–6, IEEE, 2017.
- [155] J. Ko, W. J. Yoon, and Y. S. Kim, "A study on surgical robot image stabilization," *Multimedia Tools and Applications*, vol. 77, no. 8, pp. 9871–9883, 2018.

- [156] X. Wang, T. Li, S. Sun, and J. Corchado, "A survey of recent advances in particle filters and remaining challenges for multitarget tracking," *Sensors*, vol. 17, no. 12, p. 2707, 2017.
- [157] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, "Particle filter-based prognostics: Review, discussion and perspectives," *Mechanical Systems and Signal Processing*, vol. 72, pp. 2–31, 2016.
- [158] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.
- [159] W. Chantara, J.-H. Mun, D.-W. Shin, and Y.-S. Ho, "Object tracking using adaptive template matching," *IEIE Transactions on Smart Processing & Computing*, vol. 4, no. 1, pp. 1–9, 2015.
- [160] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on information theory*, vol. 21, no. 1, pp. 32–40, 1975.
- [161] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
- [162] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [163] N. M. Artner, "A comparison of mean shift tracking methods," in *12th Central European Seminar on computer graphics*, vol. 23, Citeseer, 2008.
- [164] J. G. Allen, R. Y. Xu, and J. S. Jin, "Object tracking using camshift algorithm and multiple quantized feature spaces," in *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pp. 3–7, Australian Computer Society, Inc., 2004.
- [165] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," 1998.
- [166] R. Han, Z. Jing, and G. Xiao, "Probabilistic motion switch tracking method based on mean shift and double model filters," in *International Symposium on Neural Networks*, pp. 705–714, Springer, 2007.
- [167] M.-H. Jeong, B.-J. You, and W.-H. Lee, "Color region tracking against brightness changes," in *Australasian Joint Conference on Artificial Intelligence*, pp. 536–545, Springer, 2006.
- [168] J. Wang and Y. Yagi, "Integrating shape and color features for adaptive real-time object tracking," in *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pp. 1–6, IEEE, 2006.
- [169] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge, "Tracking non-rigid objects in complex scenes," in *1993 (4th) International Conference on Computer Vision*, pp. 93–101, IEEE, 1993.
- [170] K. Sato and J. K. Aggarwal, "Temporal spatio-velocity transform and its application to tracking and interaction," *Computer Vision and Image Understanding*, vol. 96, no. 2, pp. 100–128, 2004.

- [171] C.-b. LI, Y. Bo, and C.-h. LI, "Deep learning based visual tracking: A review," *DEStech Transactions on Computer Science and Engineering*, no. smce, 2017.
- [172] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 76, pp. 323–338, 2018.
- [173] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, vol. 2008, p. 1, 2008.
- [174] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.
- [175] C. E. Erdem, B. Sankur, and A. M. Tekalp, "Performance measures for video object segmentation and tracking," *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 937–951, 2004.
- [176] S. Salti, A. Cavallaro, and L. Di Stefano, "Adaptive appearance modeling for video tracking: Survey and evaluation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4334–4348, 2012.
- [177] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2411–2418, 2013.
- [178] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [179] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, 2009.
- [180] J. P. d. S. Monteiro *et al.*, "Automatic behavior recognition in laboratory animals using kinect," 2012.
- [181] F. De Chaumont, R. D.-S. Coura, P. Serreau, A. Cressant, J. Chabout, S. Granon, and J.-C. Olivo-Marin, "Computerized video analysis of social interactions in mice," *Nature methods*, vol. 9, no. 4, p. 410, 2012.
- [182] F. de Chaumont, E. Ey, N. Torquet, T. Lagache, S. Dallongeville, A. Imbert, T. Legou, A.-M. Le Sourd, P. Faure, T. Bourgeron, *et al.*, "Live mouse tracker: real-time behavioral analysis of groups of mice," *bioRxiv*, p. 345132, 2018.
- [183] L. P. Noldus, A. J. Spink, and R. A. Tegelenbosch, "Ethovision: a versatile video tracking system for automation of behavioral experiments," *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 3, pp. 398–414, 2001.
- [184] T. Leroy, M. Silva, R. D’Hooge, J.-M. Aerts, and D. Berckmans, "Automated gait analysis in the open-field test for laboratory mice," *Behavior research methods*, vol. 41, no. 1, pp. 148–153, 2009.
- [185] M. Zimmermann, "Ethical guidelines for investigations of experimental pain in conscious animals," *Pain*, vol. 16, no. 2, pp. 109–110, 1983.

- [186] D. H. Vrinten and F. F. Hamers, “‘catwalk’ automated quantitative gait analysis as a novel method to assess mechanical allodynia in the rat; a comparison with von frey testing,” *Pain*, vol. 102, no. 1-2, pp. 203–209, 2003.
- [187] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris, “A survey of semantic image and video annotation tools,” in *Knowledge-driven multimedia information extraction and ontology evolution*, pp. 196–239, Springer, 2011.
- [188] K. Wada, “labelme: Image polygonal annotation with python.”
- [189] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “Labelme: a database and web-based tool for image annotation,” *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.
- [190] M. Lorbach, E. I. Kyriakou, R. Poppe, E. A. van Dam, L. P. Noldus, and R. C. Veltkamp, “Learning to recognize rat social behavior: Novel dataset and cross-dataset application,” *Journal of neuroscience methods*, vol. 300, pp. 166–172, 2018.
- [191] X. P. Burgos-Artizzu, P. Dollár, D. Lin, D. J. Anderson, and P. Perona, “Social behavior recognition in continuous video,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1322–1329, IEEE, 2012.
- [192] H. Jhuang, E. Garrote, X. Yu, V. Khilnani, T. Poggio, A. D. Steele, and T. Serre, “Automated home-cage behavioural phenotyping of mice,” *Nature communications*, vol. 1, p. 68, 2010.
- [193] G. H. Salem, J. U. Dennis, J. Krynitsky, M. Garmendia-Cedillos, K. Swaroop, J. D. Malley, S. Pajevic, L. Abuhatzira, M. Bustin, J.-P. Gillet, *et al.*, “Scorhe: A novel and practical approach to video monitoring of laboratory mice housed in vivarium cage racks,” *Behavior research methods*, vol. 47, no. 1, pp. 235–250, 2015.
- [194] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pp. 65–72, IEEE, 2005.
- [195] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” in *Video-based surveillance systems*, pp. 135–144, Springer, 2002.
- [196] A. Holovaty and J. Kaplan-Moss, *The definitive guide to Django: Web development done right*. Apress, 2009.