

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Robust Detection and Tracking of Players and Ball in Padel Matches: A Computer Vision Approach

João Pedro Fontes Vilhena e Mascarenhas



Mestrado em Engenharia Informática e Computação

Supervisor: Luís Filipe Teixeira

Co-supervisor: Jorge Grenha Teixeira

March 28, 2024

Robust Detection and Tracking of Players and Ball in Padel Matches: A Computer Vision Approach

João Pedro Fontes Vilhena e Mascarenhas

Mestrado em Engenharia Informática e Computação

March 28, 2024

Resumo

Com a crescente popularidade do padel entre jogadores amadores, a procura por soluções capazes de seguir (tracking) com precisão os jogadores e a bola durante os jogos tem vindo a aumentar rapidamente. Um tracking preciso é crucial como base para aplicações mais complexas que potenciem a experiência dos jogadores e permitam uma análise aprofundada do seu desempenho. Embora existam sistemas profissionais de captura e análise de movimento, estes são frequentemente inacessíveis em termos de custos e impraticáveis para a comunidade amadora devido à sua complexidade. É desta lacuna de opções viáveis para a comunidade amadora que emerge a necessidade de um sistema robusto e acessível. Esta dissertação visa colmatar essa lacuna, recorrendo a uma abordagem robusta e baseada em soluções de software para a deteção de objetos e tracking de jogadores e bola em vídeos de padel, utilizando uma configuração simples e acessível de uma única câmara.

Através de uma revisão extensa da literatura, foram avaliados modelo do estado da arte para deteção de objetos, seguimento, e re-identificação de pessoas, analisando a sua aplicabilidade ao domínio do padel, no qual vários desafios surgem. Uma contribuição significativa deste trabalho é a criação de um vasto conjunto de dados com vídeos de padel anotados, colmatando a escassez de dados disponíveis para este desporto. No que toca a modelos explorados, o YOLOv8 demonstrou elevada precisão e eficiência na deteção e localização de jogadores e bola. Foram explorados também diversos algoritmos de seguimento, incluindo ByteTrack, BoTSORT e DeepOCSORT, integrando informação de aparência através de modelos de re-identificação como OSNet e ResNet-50.

Através de várias experiências, foi possível validar a eficácia do sistema na deteção de objetos e rastreamento de jogadores em cenários desafiantes, incluindo oclusões e condições diversas do campo. A componente de re-identificação, mesmo que de forma menos expressiva do que o esperado, auxiliou na manutenção de identidades consistentes dos jogadores ao longo dos jogos. No entanto, o seguimento da bola revelou-se mais desafiante, realçando a necessidade de melhorias adicionais nessa componente em particular. No geral, este trabalho proporciona uma base sólida para análises avançadas de padel, permitindo obter dados valiosos sobre o desempenho dos jogadores e enriquecendo a sua experiência de jogo. Além disso, a disponibilização pública do conjunto de dados anotados facilita futuras investigações neste domínio.

Abstract

With the increasing popularity of padel among amateur players, the demand for solutions to accurately track players and the ball during matches is rapidly growing. Accurate tracking is crucial as a foundation for more complex applications that enhance the overall experience and enable in-depth analysis of player performance. While professional-level motion capture and analysis systems exist, they are often cost-prohibitive and impractical for the amateur community. From this lack of feasible options for amateur players emerges the need for an accessible and capable system. The present work addresses this gap by developing a robust, software-based approach for player and ball detection and tracking in padel videos using a simple and accessible single-camera setup.

Through a comprehensive literature review, state-of-the-art models for object detection, multi-object tracking, and person re-identification were thoroughly evaluated for their applicability to the padel domain. A significant contribution of this work is the creation of an extensive dataset of annotated padel videos, addressing the scarcity of available data for this sport. With respect to the selection of models, YOLOv8 demonstrated high accuracy and efficiency in detecting and localizing players and the ball. Additionally, various multi-object tracking algorithms, including ByteTrack, BoTSORT, and DeepOCSORT, were explored, integrating appearance information through re-identification models such as OSNet and ResNet-50.

Extensive experiments validated the system's effectiveness in robustly detecting and tracking players across challenging scenarios, including occlusions and varying court conditions. The re-identification component, although not as expressively as expected, aided in maintaining consistent player identities throughout the matches. However, ball tracking proved more challenging, highlighting the need for further enhancements in future work. Overall, this work provides a solid foundation for further advanced padel analytics, enabling insights into player performance and enriching the overall experience. Furthermore, the public release of the annotated dataset facilitates future research in this domain.

“Our intelligence is what makes us human, and AI is an extension of that quality. Artificial intelligence is extending what we can do with our abilities. In this way, it’s letting us become more human ”

- Yann LeCun

Contents

1	Introduction	1
1.1	Context	1
1.1.1	Padel	1
1.2	Motivation	2
1.3	Objectives	3
1.3.1	Innovation and Commercial Value	3
1.3.2	Research Contribution	4
1.4	Dissertation Structure	6
2	Background Concepts	7
2.1	Introduction	7
2.2	Computer Vision	7
2.3	Traditional CV Techniques	8
2.4	Deep Learning and Neural Networks	9
2.4.1	Depth in DL	9
2.4.2	Advantages and Disadvantages	9
2.4.3	Learning Paradigms	10
2.4.4	DL in Computer Vision	10
2.5	Convolutional Neural Networks	10
2.5.1	Feature Extraction and Feature Maps	11
2.5.2	Pooling and Non-linearity	11
2.5.3	Flattening and Fully Connected Layers	11
2.5.4	Applied Example : Recognizing an image of Tweety	11
2.6	Object Detection	12
2.6.1	Early Object Detection Approaches	12
2.6.2	Transition to DL-Based Models	13
2.6.3	One-Stage vs Two-Stage Detectors	13
2.6.4	Advantages of DL in Object Detection	13
2.7	Multiple Object Tracking	14
2.7.1	Evolution of Deep Learning in MOT	14
2.7.2	Temporal Information and RNN-based Tracking	14
2.7.3	Tracking by Detection vs Temporal-Based Tracking	14
2.7.4	Tracking with Sensors and Wearables	15
2.7.5	Semantic Event Detection	15
2.8	Person Re-Identification	15
2.8.1	Visual Feature Extraction in ReID	16
2.8.2	Deep Learning Enhancements in ReID	16
2.8.3	Omni-Scale Feature Learning for Robust ReID	16

2.9	Motion Capture and Analysis	17
2.9.1	Visual Motion Capture	18
2.9.2	Sensor-Based Motion Capture	18
3	Literature Review	20
3.1	Contemporary Computer Vision Applications in Sports	20
3.1.1	Football	21
3.1.2	Tennis	22
3.1.3	Padel	24
3.1.3.1	Court Detection	24
3.1.3.2	Ball Detection and Tracking	24
3.1.3.3	Player Detection and Tracking	27
3.1.3.4	Automatic Generation of Synthetic Multi-View Dataset	28
3.1.4	Generic Cross-Sport Solutions	29
3.1.5	Challenges and Research directions	30
3.2	Existent Datasets	31
3.2.1	Sports-1M	31
3.2.2	OpenTTGames	31
3.2.3	PadelVic	32
3.2.4	Roboflow	33
3.3	Commercial applications	34
3.3.1	Padel Applications	35
3.4	Motion Capture and Analysis Methods	36
3.4.1	Vicon	37
3.4.2	Hawkeye	37
3.4.3	Single-Camera Motion Tracking	38
3.5	Object Detection	38
3.5.1	YOLO	38
3.5.2	Faster RCNN	41
3.5.3	RetinaNet	42
3.6	Multi-Object-Tracking	43
3.6.1	DeepOCSORT	44
3.6.2	ByteTrack	45
3.6.3	BoTSORT	46
3.7	Re-Identification	48
3.7.1	Omni-Scale Network (OSNet)	49
3.7.2	OSNet Variants	49
3.7.3	ResNet-50	51
3.8	Conclusions	52
4	Methodology	53
4.1	Data Collection	53
4.2	Initial Data Preparation and Annotation	54
4.2.1	Resolution Adjustment	54
4.2.2	Frame Extraction and Standardization	54
4.2.3	Manual Annotation	55
4.2.4	Challenges and Limitations	55
4.3	Automated Data Preparation and Annotation Workflow	55
4.3.1	Semi-Automated Annotation for Object Detection	55

4.4	Dataset Expansion	56
4.5	Multi-Object-Tracking Validation Dataset	56
4.6	Classes Definition	57
4.6.1	Six Classes	57
4.6.2	Three Classes	57
4.6.3	Two Classes	58
4.7	Model Selection and Evaluation Metrics	59
4.7.1	Object Detection	59
4.7.1.1	YOLOv8	59
4.7.1.2	Faster R-CNN	60
4.7.1.3	RetinaNet	60
4.7.1.4	Evaluation Metrics	61
4.7.2	Multi-Object-Tracking and Re-Identification	62
4.7.2.1	Evaluation Metrics	63
4.8	Tools and Technologies	64
4.9	Computational Resources	65
4.9.1	Cloud	65
4.9.2	Local	65
4.9.3	FEUP Server	65
5	Experiments and Results	66
5.1	Object Detection	67
5.1.1	Dataset Development and Refinement	68
5.1.1.1	Initial MOG Datasets(220 and 2000 images)	68
5.1.1.2	Enhanced Dataset (6 different matches)	70
5.1.1.3	Final Dataset (12 different matches)	72
5.1.2	Model Architecture Comparison and Selection	73
5.1.3	Resolution and Model Size Experiments	75
5.1.4	Class Configuration Experiments	76
5.2	Multi-Object-Tracking and Re-Identification	78
6	Conclusions	81
6.1	Future Work	82

List of Figures

1.1	Padel court [2]	2
1.2	Overview of PADEL4PRO platform features	4
2.1	Edge Detection example [15]	8
2.2	Corner Detection example [16]	8
2.3	Simple illustration of a CNN Architecture for Feature Extraction and Image Classification [3]	10
2.4	Multi-Object Detection and Localization on a Street Scene Image [23]	12
3.1	Statistics on Sports Research [19]	21
3.2	Example of a correct court detection [41]	25
3.3	Camshift example detection [43]	26
3.4	Canny edge detection example, ball represented in red [43]	26
3.5	YOLO detection example [43]	26
3.6	Example of a frame with corresponding segmentation masks [38]	32
3.7	Example of a frame with a volley detection [35]	33
3.8	Example of a frame with net detection [36]	34
3.9	Example of a frame with players and ball detection [37]	34
3.10	Scorekeeping [98]	36
3.11	Pose Estimation [98]	36
3.12	Match Recording [98]	36
3.13	Hawkeye camera angle example [107]	37
3.14	YOLO example [24]	39
3.15	YOLO model architecture [8]	40
3.16	Evolution of YOLO models [113]	41
3.17	Faster R-CNN, represented as a single unified network for detection tasks [103]	42
3.18	RetinaNet network's architecture [54]	43
3.19	Deep OC-SORT Visual Representation [44]	45
3.20	BotSORT Visual Schema [63]	47
3.21	OSNet building blocks [82]	50
3.22	ResNet-50 architecture [25]	51
4.1	Mog Dataset Camera Angles	54
4.2	Example of a frame with the players numbered according to their positions	58
4.3	Ball examples	58
4.4	Comparing Yolov8 different versions [113]	59
4.5	Comparing Faster R-CNN different versions [105]	60
4.6	Comparing RetinaNet different versions [105]	61

List of Tables

3.1	Benchmark performed on the 10 first frames of each MOT17 sequence [45]. ¹	44
3.2	Comparison of Different Cross-Domain REID Models [81]. ²	48
5.1	Dataset Versions' Description	67
5.2	All Object Detection Models	68
5.3	Data Augmentation Parameters	68
5.4	Initial models validated on MOG match (higher is better)	69
5.5	Initial models validated on external match (higher is better)	69
5.6	Class balance of dataset MOG220Images	71
5.7	First 3 models validated on MOG match (higher is better)	71
5.8	First 3 models validated on external match (higher is better)	72
5.9	Class distribution of dataset Expanded_12Videos	72
5.10	All models from the current iterations validated on the 5 different videos from Expanded_12Videos dataset (higher is better)	73
5.11	All selected architectures trained and validated on the 5 different videos from Expanded_12Videos dataset	74
5.12	All resolutions and model size combination validated on the 5 videos from Expanded_12Videos dataset (720p)	75
5.13	All resolutions and model size combination validated on the 5 videos from Expanded_12Videos_480p dataset	75
5.14	Comparison of the main models trained and validated on 2 classes and 3 classes	76
5.15	Normalized Confusion Matrix for the 12_videos classes model	77
5.16	Normalized Confusion Matrix for the medium_12_videos_2classes model	77
5.17	Description of MOT-ReID Configurations	78
5.18	Comparison of the different MOT-ReID configurations on the Players	78
5.19	Comparison of the different MOT-ReID configurations on the ActiveBall	79
5.20	Comparison of the different MOT-ReID configurations on the InactiveBall	79

Abbreviations and Symbols

AI	Artificial Intelligence
CV	Computer Vision
YOLO	You Only Look Once
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
DL	Deep Learning
SOTA	State-of-the-Art
FCL	Fully Connected Layer
MOT	Multiple Object Tracking
POV	Point of View
RNN	Recurrent Neural Networks
ReID	Re-Identification
ML	Machine Learning
UGSV	User-generated sports video
LSTM	Long Short-Term Memory
FPN	Feature Pyramid Network
FCN	Fully Convolutional Network
OSNet	Omni-Scale Network

Chapter 1

Introduction

The introduction chapter serves as a compass for this thesis. It outlines why the research matters, identifies the objectives, and provides a roadmap for the reader. The context and motivation offer insight into the rise of Padel and the need for tech innovations. The objectives clarify the dissertation goals. Lastly, the dissertation structure gives an overview of the thesis layout. This chapter essentially sets the stage for the forthcoming in-depth exploration of the subject.

1.1 Context

Padel, a racquet sport that uniquely blends aspects of tennis, squash, and badminton, is presently witnessing an impressive uptick in global interest. It counts over 25 million enthusiasts from over 90 countries among its ranks [4]. This sport's allure stems from its easy-to-learn rules and its accessibility to individuals of varying skill levels. Originating in Mexico in 1969 and rapidly spreading globally, padel has evolved beyond a recreational sport to become a subject of scientific and technological exploration, driven by its accessibility, social appeal, and distinctive gameplay features.

Incorporating Computer Vision technologies into sports represents a significant stride towards enhancing player performance, engagement, and the spectator experience. By leveraging the capabilities of real-time image and video analysis, researchers and developers are pioneering systems for automated player detection, tracking, and game analysis, pushing the boundaries of how technology can augment sports.

1.1.1 Padel

This sport is usually played in doubles on an enclosed court that is approximately a third the size of a tennis court. The court is divided by a net and is surrounded by walls. The walls are used as part of the game, with the ball allowed to hit them similarly to squash. A visual representation is included in the figure 1.1.

In respect to scoring mechanisms, padel follows the conventional model employed in tennis. Points are assigned in an ordinal sequence of 15, 30, 40, culminating in a 'game'. In essence,

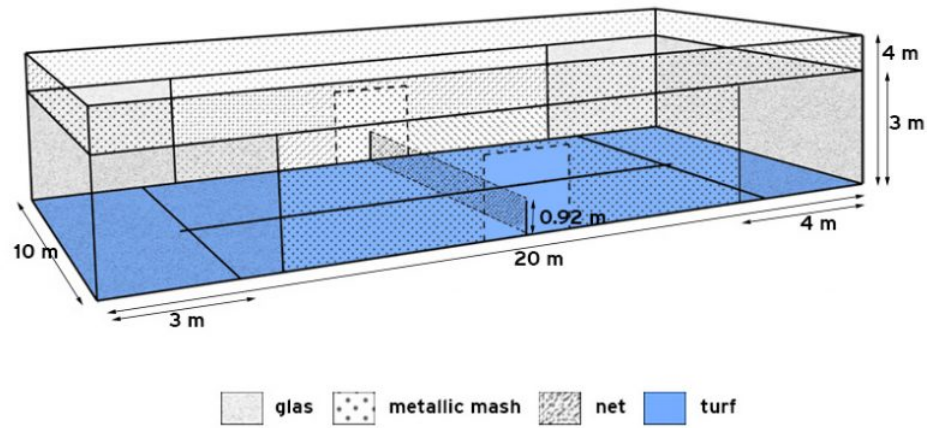


Figure 1.1: Padel court [2]

the duo that first accumulates four points is declared the winner of the game. The match victory is awarded to the team that first completes a set, which consists in the accumulation of six won games.

When it comes to serving, the serving player must bounce the ball behind the service line and hit it below waist height. The ball must then bounce in the opponent's service box before hitting the back wall. If the ball hits the wire fence before bouncing in the service box, it is considered a fault.

If a ball hits the player before it bounces, the player loses the point. Additionally, if the ball bounces and then hits the wire fence before a player has a chance to return it, the player loses the point. When the score is deuce (40-40), the team that wins the next point has the advantage, and if they win the subsequent point, they win the game. However, if the score returns to a deuce, the match continues until one team has a two-point lead [5].

1.2 Motivation

The expansion of Padel is mirrored not only in the increasing number of players but also in the thriving industry that has emerged around it. As of 2021, the estimated market value of Padel stood at USD 190.78 million, with future growth trends showing promise [6]. These indicators underscore the significant economic opportunities the sport presents, emphasizing the imperative for ongoing support and innovation to propel its advancement while ensuring the sustained engagement of players.

The growth in amateur padel has fueled demand for content like match recordings, highlights, player tracking, and performance analysis. The ability to visualize player movements, shot trajectories, and statistics offers opportunities for skill development and understanding the sport. Advancements in computer vision and video analysis provide means to generate this content, enabling the creation of visualizations and analytics tailored to the amateur padel community.

However, manual data generation, such as player and ball positioning, without costly, professional-level motion capture and analysis systems, can be inefficient, unreliable, and often unfeasible, especially in this sport and its particular challenges. Consequently, Computer Vision, as a software-based approach, emerges as a promising solution to address the rising demand for enhanced understanding, performance analysis, and content generation in padel. By automating the extraction of relevant information from video data, Computer Vision techniques can streamline the process, ensuring accuracy, consistency, and scalability. While Computer Vision has found applications in multiple sports-related contexts, its integration into padel, particularly at the amateur level, is relatively uncharted territory. This gap was acknowledged in an article [58] which asserts to be one of the pioneer scientific works to introduce player tracking into padel. This specific article will be examined in greater detail in the chapter 3, which focuses on the Literature Review.

Regarding Padel, the critical information revolves around the players and the ball. Their positional data, coupled with movement analytics, can reveal invaluable information. Object Detection, a subset of Computer Vision, can be employed to extract this positional data through image processing. By processing images to pinpoint the exact locations of players and the ball, it lays the groundwork for tracking mechanisms.

Building on the foundation provided by Object Detection, tracking-by-detection technologies take the analysis a step further by monitoring the movement trajectories of these detected objects over time. This continuous tracking is essential for capturing the dynamic nature of Padel matches, where understanding the flow and patterns of play can offer competitive advantages. Algorithms like YOLO [8], renowned for their speed and accuracy in real-time Object Detection, facilitate the seamless transition to tracking-by-detection algorithms by ensuring that objects of interest are accurately identified from frame to frame. This integration of Object Detection and tracking enables a comprehensive analysis of the game, allowing for the assessment of player positioning, ball movement, and interaction patterns. Such depth of analysis is invaluable not only for coaching and performance improvement but also for enhancing the viewing experience by providing detailed insights into the game's intricacies. By leveraging these technologies, the amateur padel community gains access to analytics and insights only formerly available on a professional level, making the sport more accessible, enjoyable, and enriching for players and fans alike.

1.3 Objectives

This dissertation's main objective is stated in its own title: the design and implementation of a system providing Robust Detection and Tracking of Players and Balls in padel matches. The system's robustness concerns its ability to maintain a proper performance across different scenarios. Nevertheless, on a higher level, its objectives are aligned with two different purposes.

1.3.1 Innovation and Commercial Value

As it was proposed by MOG Technologies, a worldwide provider of digital media technology [7], this dissertation is also framed in a market innovation context and as a value proposition along-

side its scientific purpose. MOG's distinctive business case aims at elevating the padel players' experience and engagement by offering an all-in-one solution, as detailed in figure 1.2. More specifically, the objectives in this domain are :

- A brief market study addressing current platforms proposing the same features and using a single camera setup.
- Developing a working tracking and detection system that can be easily integrated into the Padel4Pro platform (see figure 1.2) and made available to the amateur community, where solutions as such are non-existent due to cost limitations and technical challenges.
- Proposing future work seeking innovation, building upon the robust detection and tracking system, which will serve as a foundational system for more complex features. For instance, extracting highlights from the matches.

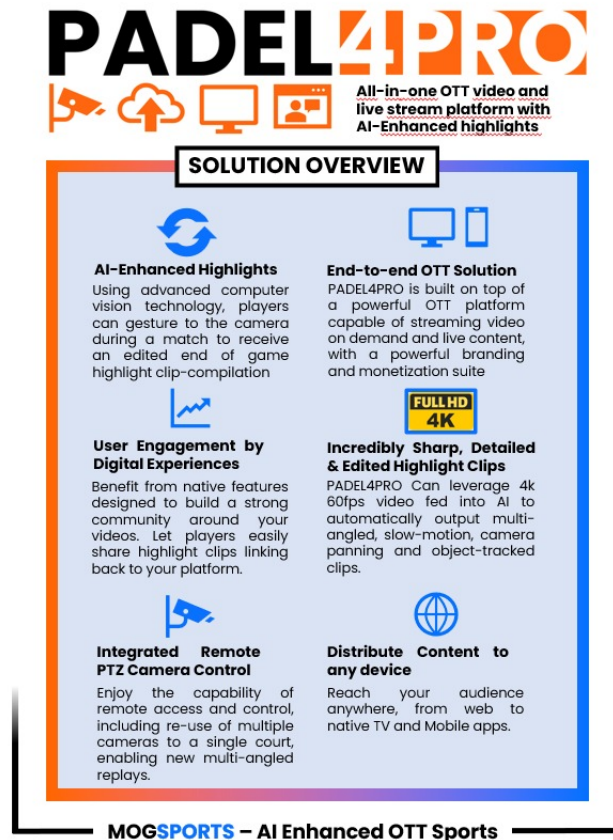


Figure 1.2: Overview of PADEL4PRO platform features

1.3.2 Research Contribution

Through the Literature Review, one of the dissertation's objectives is to provide an extensive and comprehensive overview of the current applications of Computer Vision in sports, with a particular

focus on padel. By conducting that thorough review, the aim is to consolidate and synthesize all the relevant research surrounding this topic. This involves identifying recurring themes, patterns, and methodologies, as well as critically examining the gaps, limitations, and unexplored areas within the field. Ultimately, the objective is to enrich the understanding of this research domain, offering insights and potential conclusions that can guide and inform future work. Moreover, a significant research objective is to experiment with and evaluate various state-of-the-art techniques and approaches for the design and implementation of a robust system for player and ball detection and tracking in padel matches. This involved a rigorous assessment of different object detection models, multi-object tracking algorithms, and person re-identification models, exploring their suitability, performance, and potential adaptations for the specific challenges inherent to the padel environment. The goal is to identify and validate the most effective techniques that can serve as the foundation for a reliable and efficient system. Furthermore, the dissertation aims to contribute to the research community by publicly releasing a valuable and comprehensive dataset of annotated padel videos. A dataset meticulously selected and annotated, ensuring its quality and relevance for computer vision tasks in the padel domain. By making the dataset publicly available, the objective is to address the current scarcity of data and provide a valuable resource that can facilitate and accelerate future research efforts.

1.4 Dissertation Structure

The document is structured as follows:

Introduction as Chapter 1, presents a general outline of the main themes discussed in the document.

Background Concepts as Chapter 2, provides a comprehensive examination of the foundational concepts and technologies that underpin this domain. These include Computer Vision, Traditional CV techniques, Deep Learning and Neural Networks, Convolutional Neural Networks, Object Detection, Multiple Object Tracking, Person Re-Identification, and Motion Capture and Analysis.

Literature Review as Chapter 3, delves into the current state-of-the-art in computer vision applications within sports, with a special focus on padel. It includes a summary of research papers integrating AI in different sports, an assessment of available datasets, and an overview of commercial applications. The chapter also explores specific methodologies for player and ball detection, tracking, and re-identification that are going to be further explored.

Methodology as Chapter 4, presents the proposed approach for implementing a robust detection and tracking system for padel. It covers the tools and technologies used, data collection and preparation, the development and expansion of the dataset, models and algorithms employed, and evaluation metrics.

Experiments and Results as Chapter 5, details the experimental setup, results, and analysis for each component of the system: object detection, multi-object tracking, and re-identification. It also includes an overall system evaluation and discusses the strengths, limitations, and potential improvements.

Conclusion as Chapter 6, summarizes the key findings, research contributions, and sets possible future directions for the work conducted.

Chapter 2

Background Concepts

2.1 Introduction

This chapter consists in a review of the foundational background concepts and technologies that underpin this domain. These include Computer Vision, Traditional CV techniques, Deep Learning and Neural Networks, Convolutional Neural Networks, Object Detection, Multiple Object Tracking, Person Re-Identification, and Motion Capture and Analysis. All these concepts play an important role in contextualizing the application of computer vision in sports, as they form the foundation of many advanced systems used in sports analytics today.

2.2 Computer Vision

Computer Vision (CV) is a field of artificial intelligence that trains computers to interpret and understand the visual world. The roots of computer vision can be traced back to the 1960s when Larry Roberts, often referred to as the "father of computer vision," wrote his PhD thesis at MIT on the possibility of extracting 3D geometric information from 2D views [12].

In the 1970s and 1980s, the focus of computer vision research was on interpreting hand-coded features and geometric models. Researchers manually designed and implemented algorithms to extract specific features from images, such as edges, corners, or texture patterns. These hand-crafted features were then used in geometric models to represent objects or scenes. However, these methods were often brittle and limited in their applicability. The 1990s and early 2000s saw the rise of statistical learning methods, which provided more robustness and flexibility [13].

Through the utilization of digital imagery from cameras and videos, coupled with deep learning models, machines have the capability to recognize and categorize objects accurately, subsequently responding based on their 'visual' understanding [10]. Deep Learning (DL) (read section 2.4), a subset of machine learning, has been a significant driving force in the field of computer vision.

Convolutional Neural Networks (CNNs), further described in section 2.5, are a type of deep learning model that has had an immense influence in the field of CV in recent years.

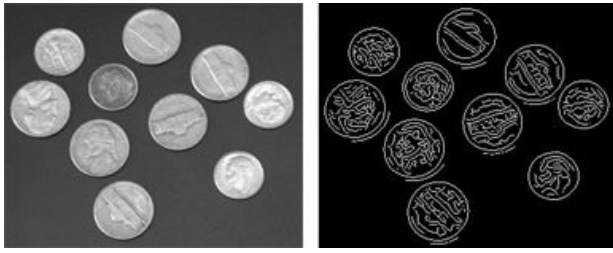


Figure 2.1: Edge Detection example [15]

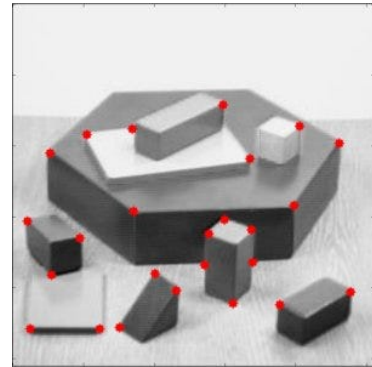


Figure 2.2: Corner Detection example [16]

However, it is important to note that while deep learning has pushed the boundaries of what is possible, traditional computer vision techniques which had been undergoing progressive development in years prior to the rise of DL have not become obsolete. Traditional computer vision techniques such as edge detection, template matching, optical flow, and others have their place in certain applications and laid the groundwork for modern approaches [13].

In the context of sports, computer vision has been used for player and ball detection, tracking and reidentification. These applications leverage traditional computer vision techniques and deep learning models to achieve their goals [14]. The Literature Review, in chapter 3, will provide readers with an extensive and comprehensive examination of the current application of computer vision in sports, offering a detailed discussion of the subject matter, although considering this work's particular scope.

2.3 Traditional CV Techniques

Traditional computer vision techniques, often referred to as classical computer vision, primarily involve methods that extract features from images manually. These techniques have been the foundation of computer vision before the rise of deep learning and continue to be relevant in certain applications [10].

Traditional computer vision techniques include methods such as edge detection, corner detection, blob detection, template matching, optical flow, and histogram of oriented gradients (HOG) [15]. These techniques often involve transforming the image into a form that makes the information contained in the image easier to analyze.

Edge detection (see figure 2.1), for example, is a technique that identifies points in an image where the brightness changes sharply or has discontinuities [15]. The result of this technique is an image emphasizing the edges of detected objects. Similarly, corner detection (see figure 2.2) identifies points in the image where the direction of the edges changes, often corresponding to feature points in the image [16].

Currently, these techniques are not as widely used due to the need of **manual extracting features**, a requirement which can be time-consuming and also requires extensive domain knowledge along with their **limited complexity capabilities** when it comes to handling complex tasks that require understanding, for example, the semantic content of an image. For instance, they are also **sensitive to changes**. Slight differences in lighting, viewpoint, and scale are some of the changes that might highly affect their performance. Accordingly, Deep learning techniques excel in all those previous limitations, surpassing them, thus its ongoing popularity. However, traditional CV techniques, in comparison, **require much fewer quantities of data** since there are no DL models to be trained. Besides, their **interpretability** is also an advantage. The steps and transformations applied are often easy to understand and interpret, in contrast to the often-claimed black boxes when referring to DL models. Usually, they are also less computationally intensive, which makes them suitable for real-time applications.

2.4 Deep Learning and Neural Networks

Deep Learning (DL) is a specific branch of machine learning. It utilizes neural networks with many layers (deep neural networks) and these networks attempt to simulate the behavior of the human brain and how the neurons function, thus its actual name, in order to simulate the process of learning from large amounts of data.

In a **Neural Network**, there are multiple layers of nodes (neurons), where each layer takes input from previous layers (in the case of the first layer, it takes the actual input data), then each node in those layers takes a weighted sum of inputs, adds a bias term, and then feeds the result through a non-linear function, also known as an activation function. The weights and biases are the parameters of the model that are learned during the training process[18]. After performing those computations, each layer passes the output to the subsequent layers. The last layer gives us the final output. The layers between the input and output layers are known as hidden layers.

2.4.1 Depth in DL

The term "deep" in Deep Learning comes from the many layers of neurons in the networks. The "depth" of a model is the number of layers it has. Besides, the more layers, the more complex patterns it can learn to recognize. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help optimize the accuracy of those predictions. Reaching highly complex tasks, DL drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention.

2.4.2 Advantages and Disadvantages

DL has several advantages. It can process large volumes of data, handle multiple inputs and outputs, and model complex, non-linear relationships. It also eliminates the need for manual

feature extraction, as the network learns the features itself.

However, the biggest compromise with deep learning models is that they require a large amount of data and computational resources. Additionally, these models, despite being powerful, are often described as "black boxes", making them difficult to interpret along their decision-making process and can overfit to the training data, meaning they may not be able to generalize well to new, unseen data [18].

2.4.3 Learning Paradigms

DL has two main learning paradigms, the **Supervised** and the **Unsupervised**. In the first one, the data that is given as input to the network as training data has labels. This is, the variable or characteristic that is meant to be classified or predicted, is given. In the latter, it is not. As it will be explained in the chapter 4, only **Supervised Learning** will be utilized in this project despite the difficulties.

2.4.4 DL in Computer Vision

The growth of DL is the main factor for the current improvement of many areas of computer vision, including object detection (see section 2.6) and multiple object tracking (see section 2.7). In particular, Convolutional Neural Networks (CNNs) (see section 2.5), a class of deep learning models, have been particularly effective in processing the spatial and temporal relationships between objects in an image or a sequence of images. These models have been used in applications such as object detection and segmentation, where the goal is to identify objects in an image and delineate their boundaries [18]. Within the scope of this study, DL can be applied for the purpose of tracking players and the ball.

2.5 Convolutional Neural Networks

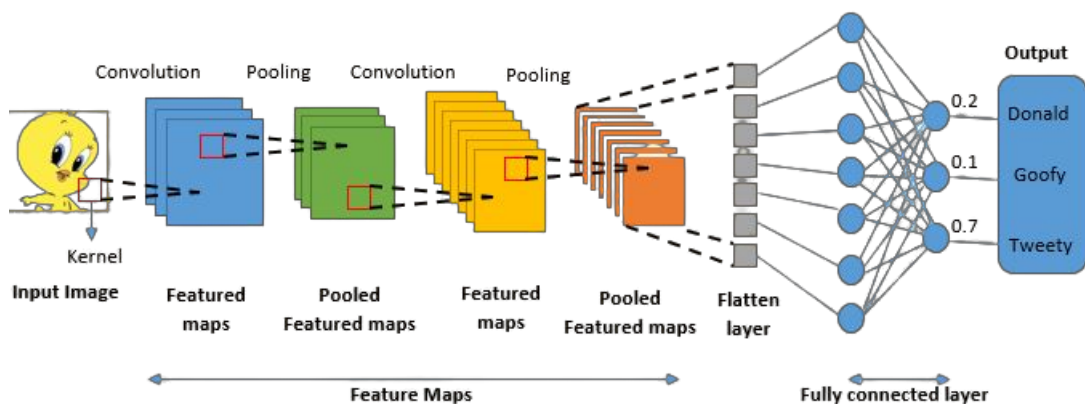


Figure 2.3: Simple illustration of a CNN Architecture for Feature Extraction and Image Classification [3]

After exploring the concept of Deep Learning and Neural Networks, it is important to introduce more in-depth CNNs since they are a specialization of Neural Networks developed specifically for visual data, such as videos and images, which is the present use case. Since they come from Neural Networks, they follow the same general principle of forwarding and backward propagation. CNNs have been particularly effective in processing the spatial relationships between objects in an image or a sequence of images and even temporal relationships depending on their specific architecture.

2.5.1 Feature Extraction and Feature Maps

These mentioned relationships are defined as features and stored in feature maps that undergo successively in operations of pooling and convolution to reach progressively more complex features as seen in Figure 2.3. The Convolutional layers are responsible for applying multiple filters/kernels, which are small matrices of weights, over the pixels of the input image to produce a set of feature maps.

2.5.2 Pooling and Non-linearity

The Pooling layers are responsible for reducing the spatial dimensions of their input to avoid overfitting and reduce unnecessary complexity. Also, it is not displayed in the figure, but after each Convolutional filter and before the Pooling layer there is an activation function such as Rectified Linear Unit (ReLU) that is responsible for non-linearity in the model [20]. Non-linearity is essential in neural networks as it allows them to model complex, real-world relationships in the data that would not be possible with linear functions alone.

2.5.3 Flattening and Fully Connected Layers

Almost in the end, the high-level complex features are flattened, converted to a single-dimensional structure, generally in the form of a 1D vector, and fed into one or more FCLs. Those are responsible to reach even more complex patterns from the high level features.

2.5.4 Applied Example : Recognizing an image of Tweety

For example, in the case of Tweety in figure 2.3, the initial convolutional layers might recognize the sharp edges that define Tweety's outline, the yellow color of Tweety's feathers, and the texture of his feathers. The subsequent pooling layers would downsample these feature maps and keep only the most relevant features. Successively, the layers recognize more complex features such as parts of Tweety's face, its eyes, beak and head. In the end, the FCL is responsible for recognizing those parts and figuring that the image as a whole was Tweety.

In sum, CNNs serve as the basis for more complex models and tasks such as object detection, object segmentation, and multiple object tracking. In the context of this work, these tasks are

crucial for tracking players and the ball in padel matches. For instance, object detection models like YOLO (You Only Look Once), which will be discussed later, utilize CNNs to extract features from the input image and then use these features to detect objects. Similarly, object segmentation and multiple object tracking models also rely on CNNs to extract useful features from the input data.

2.6 Object Detection

Object Detection consists in identifying and localizing objects within an image or a sequence of images such as a video. It is an improvement over image classification or localization because it can identify all the instances of an object class present [21]. For example, in the figure 2.4, an object detection algorithm called YOLO (see section 3.5.1) is able to detect a dog (in yellow), a bicycle (in red) and a truck (in green). It relies on providing a bounding box around each detected object, thus retrieving its location within the image. The coordinates in the output correspond respectively to the ones of each bounding box. The process it carries is twofold: first, it determines where objects are located in a given image (object localization), and second, it identifies to which category each object belongs (object classification).

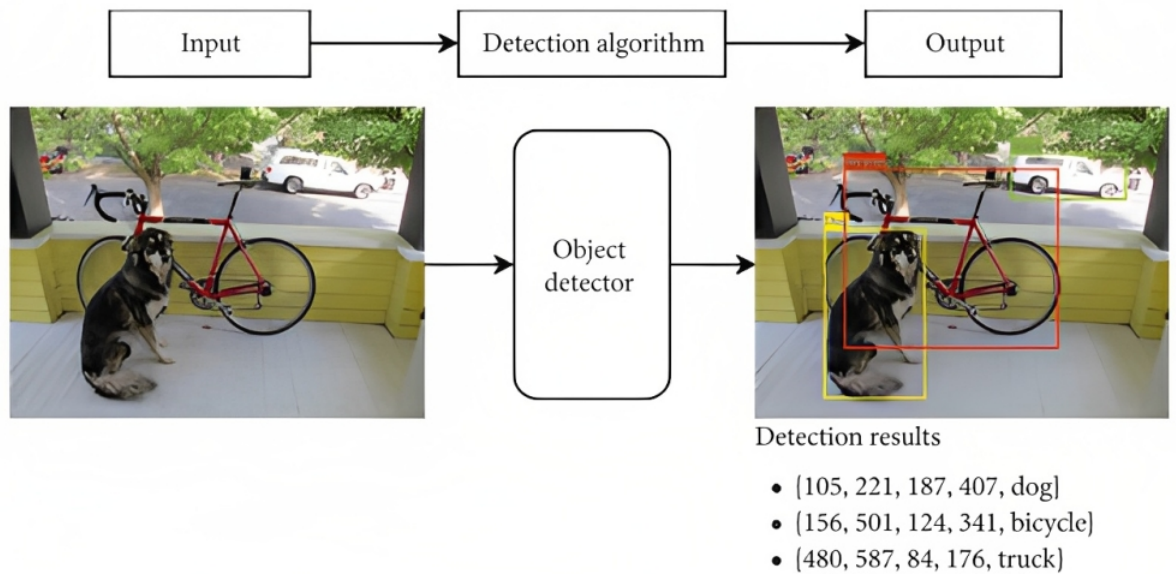


Figure 2.4: Multi-Object Detection and Localization on a Street Scene Image [23]

2.6.1 Early Object Detection Approaches

Traditionally, object detection methodologies followed a structured three-stage pipeline: regions of interest selection, relevant features extraction, and object classification. Initially, a multiscale

sliding window technique was utilized to scan images and detect potential object locations. Despite its straightforward approach, this method suffered from high computational costs and generated excess redundant candidate regions [22].

In feature extraction, visual features that could robustly represent objects were extracted. Classical feature descriptors like Scale-Invariant Feature Transform (SIFT), Histograms of Oriented Gradients (HOG), and Haar-like features, despite achieving certain levels of success, often fell short due to their inability to bridge the semantic gap and the inefficiency in handling the diversity of object appearances and illumination conditions [22].

The classification stage entailed distinguishing between different object categories. This step often employed classifiers such as Support Vector Machines (SVM), AdaBoost, and the Deformable Part-based Model (DPM), which combined object parts with deformation costs to handle complex object shapes [22].

While foundational, these traditional techniques relied heavily on handcrafted features and shallow architectures.

2.6.2 Transition to DL-Based Models

The advent of deep learning, particularly Convolutional Neural Networks (CNNs), has significantly transformed object detection. CNNs, with their deep layered architectures, possess the capability to learn intricate features from data, surpassing traditional shallow models in complexity and expressiveness. This advancement eliminated the need for manual feature engineering, allowing the models to learn informative object representations autonomously [22].

2.6.3 One-Stage vs Two-Stage Detectors

Deep learning-based object detection models can be categorized into two types: Two-stage, also known as region proposal-based frameworks, and One-stage, regression/classification-based frameworks. Region proposal-based methods, such as R-CNN and its successors, operate in two distinct phases. In the first phase, a Region Proposal Network (RPN) generates candidate object bounding boxes, which are then refined in the second phase. Here, the model extracts features using Region of Interest (RoI) pooling from each candidate box and subsequently performs classification and bounding-box regression. On the other hand, regression/classification-based methods, like YOLO and SSD (Single Shot MultiBox Detector), prioritize real-time performance by predicting object classes and bounding boxes directly from the input images without the need for separate region proposals [21].

2.6.4 Advantages of DL in Object Detection

The employment of CNNs in object detection offers numerous advantages over traditional methods. CNNs can learn hierarchical feature representations, ranging from pixel-level to high-level semantic attributes, through multilevel, nonlinear mappings. Their deep architectures exponentially increase the model's expressive capacity. Additionally, they allow for the joint optimization

of related tasks, such as classification and bounding box regression, within a multitask learning framework. This has enabled reframing classic computer vision problems into high-dimensional data transformation challenges, offering innovative solutions from a new perspective [22].

Integrating deep learning into object detection has brought remarkable improvements in performance and efficiency. This progression has established object detection as an essential component in complex visual data analysis and paved the way for new applications across various domains within computer vision. In the context of this work, Object Detection is the main foundation for ball and player detection.

2.7 Multiple Object Tracking

Object Detection sets the stage for Multiple Object Tracking (MOT) in sports, a field where the dynamism of players and the ball is of particular interest. MOT extends object detection capabilities to sequences, maintaining the identity and location of each object across frames, despite occlusions or rapid changes in direction, an area where DL methods have shown significant promise [19] [29]. This can help separating objects that are close to each other or overlapping which can be particularly useful in amateur padel, where the players sometimes move close to each other. By tracking each object's movement over time, MOT should help maintain the separate identities of the players and the ball.

2.7.1 Evolution of Deep Learning in MOT

The transition to DL in MOT is also marked by the adoption of CNNs, which outperform traditional methods, particularly in complex visual sequences common in sports environments. These methods are adept at maintaining accurate object identities, even in instances of visual occlusion [19] [80].

2.7.2 Temporal Information and RNN-based Tracking

Integrating temporal information in MOT plays a crucial role in enhancing tracking systems. Recurrent Neural Networks (RNNs), including variants such as Long Short-Term Memory (LSTM) networks, excel in processing time-series data, allowing for integrating historical information to predict future object states. These networks form the backbone of advanced MOT systems that can effectively track object trajectories over time, a feature particularly useful in sports for predicting player and ball positions after occlusions occur [29] [19].

2.7.3 Tracking by Detection vs Temporal-Based Tracking

While tracking by detection provides real-time capabilities by analyzing detections frame-to-frame, it is prone to inaccuracies during fast movements or when objects become occluded. On the contrary, Temporal-based tracking leverages past data to inform future predictions, thereby significantly reducing errors and improving the robustness of the tracking process [29] [19].

2.7.4 Tracking with Sensors and Wearables

Besides visual MOT systems, sensor-based tracking systems provide additional data layers, such as GPS or inertial measurements, which can be particularly beneficial in occluded or poorly lit scenarios since they can be much more precise and accurate. On the other hand, they may lack the flexibility of camera-based MOT systems and are not always as cost-effective or scalable [19] [80] [65].

2.7.5 Semantic Event Detection

The next step to leverage MOT's capabilities even further, after including temporal information, is semantic event detection. This involves interpreting complex scenes to recognize specific actions or patterns with particular meanings within a sport's context. These interpretations combine the raw tracking data alongside timestamps and the annotations of relevant events on those respective timestamps. Each event carries specific spatial and temporal markers that can be extracted through trajectory analysis. For instance, the positioning of players during a service, the speed and trajectory of the ball during an ace, and the pattern of player movement during rallies provide tangible data points for semantic interpretation. For example, some semantic events that can be extracted from tennis games, analyzing players' trajectory, are Service, Ace, Fault, Net [106]. The complexity of these events can reach even higher levels, including goal scoring in soccer, offsides, and player substitutions [19]. Such complexity opens up possibilities for the strategic breakdown of gameplay, highlighting key moments for broadcast replays and enriching the viewing experience by offering detailed analyses. One crucial aspect that must be emphasized is that the robustness and high accuracy of the tracking process are foundational for effective semantic event detection in sports analytics. It ensures that each movement and interaction is captured with high fidelity, enabling a granular analysis of game dynamics. In tennis, where milliseconds and millimeters can dictate the outcome of exchanges, reliably tracking and analyzing player and ball trajectories is indispensable. In this case, padel, as a sport that introduces even more complex variables than tennis, further underscores the necessity for this level of precision.

In sum, incorporating MOT, DL, temporal information, and the detection of events allows for significant advancements in the field of sports tracking, offering sophisticated systems capable of analyzing complex interactions. The progression of MOT algorithms continues to enhance the accuracy and reliability of tracking systems, underscoring their essential role in sports technology and analytics [19].

2.8 Person Re-Identification

The advancement of Multiple Object Tracking (MOT) in sports analytics, as discussed in Section 2.7, significantly enhances the ability to monitor the dynamic movements of players and objects

across frames. However, MOT's efficacy can be further enhanced when combined with Person Re-identification (ReID) technologies. While tracking algorithms are highly efficient at following objects, such as players or balls, across video frames using bounding boxes and maintaining their trajectories, they face challenges with identity preservation, especially in complex scenarios involving occlusions, rapid movements, or significant appearance changes due to varying camera angles or lighting conditions. ReID delves deeper into visual feature extraction with specialized models to discern between individuals and thus maintain their correct identity throughout the tracking process. This constitutes a key task in dynamic sports environments where players might momentarily exit the scene or when their appearance is significantly altered due to factors such as motion, occlusions, or variations in camera angles.

2.8.1 Visual Feature Extraction in ReID

ReID systems employ deep learning techniques, particularly CNNs (see section 2.5), once again, to extract rich visual features that capture the unique characteristics of individuals. These features include clothing patterns, body shape, and even gait. The effectiveness of CNNs lies in their ability to learn hierarchical feature representations from raw pixels, making them adept at identifying subtle differences and similarities between individuals even under challenging conditions such as varying lighting, occlusions, or changes in perspective. When integrated with tracking algorithms, ReID models provide a robust mechanism for re-identifying individuals after they have been occluded or have left and re-entered the scene. [84].

2.8.2 Deep Learning Enhancements in ReID

Recent surveys highlight the pivotal role of DL advancements in improving ReID technologies[84][83]. Feature Representation Learning (see section 2.8.3), Attention Mechanisms, and Generative Adversarial Networks (GANs) are the three main DL areas that have been instrumental in this regard. **Attention Mechanisms** focus on salient features and suppress irrelevant information by providing larger weights on the most distinctive attributes of individuals. **GANs** allow the augmentation of ReID datasets, generating synthetic images to enhance model training. This addresses the challenge of limited labeled data, enabling models to learn from a wider range of appearances and scenarios, including cross-domain ReID tasks where the model must generalize across different camera networks from the ones in which it was trained.

2.8.3 Omni-Scale Feature Learning for Robust ReID

The concept of Omni-Scale Feature Learning introduces an innovative approach to capturing a wide range of feature scales, enabling the ReID systems to be more resilient against the challenges posed by different player poses and occlusions. Omni-scale features are defined by their combination of variable homogeneous scales (features of a consistent scale across the image) and heterogeneous scales (features that integrate information from multiple scales).

The **Homogeneous Scales** pertain to the analysis of features at a consistent scale throughout an image. This includes **Homogeneous Small-Scale Features**, such as facial details (eye color, facial shape) and the texture of clothing (the knit of a sweater or fabric patterns), which are crucial for identification at closer proximities. These details offer high discriminative power due to their specificity. Additionally, there are **Homogeneous Large-Scale Features**, these encompass broader visual cues like a person's silhouette or overall contour and large color blocks (a red jacket or blue jeans). Those features facilitate identification from a distance where finer details are not perceptible.

Conversely, **Heterogeneous Scales** integrate features across multiple scales to construct a comprehensive feature set. **Heterogeneous Scale Integration** combines detailed and large-scale features, such as facial features with body posture or texture patterns with the silhouette. This integration enables effective recognition by leveraging detailed information up close and more general information at a distance. **Dynamic Feature Fusion** under heterogeneous scales allows ReID systems to adapt to scene dynamics, focusing on small-scale features for individuals close to the camera and large-scale features for those further away. This adaptability is particularly beneficial when a system relies on visible small-scale features while integrating them with discernible large-scale information to maintain identification accuracy.

By applying both homogeneous and heterogeneous scales, omni-scale feature learning ensures robust identification across diverse environmental conditions, embodying a comprehensive approach to the complex variability in real-world scenarios. This is fundamental for ReID, where the ability to recognize individuals based on small, localized details (like facial features or accessories) and larger, global patterns (such as the overall clothing silhouette) can significantly impact identification accuracy[82]. The Omni-Scale Network(OSNet), designed specifically for ReID tasks, encapsulates this approach and will be explored on the chapter 3.

The synergy between MOT and Person Re-identification technologies marks a significant milestone in sports analytics, offering a comprehensive toolset for automating the analysis of individual player dynamics and interactions with great precision.

2.9 Motion Capture and Analysis

Motion capture and analysis technologies provide raw data that allow for the study of insights into player movements, game dynamics, and performance metrics, paving the way for a deeper understanding and enhancement of performance across various sports disciplines. These technologies vary significantly in complexity, cost, and application scope, ranging from professional commercial solutions to more accessible amateur setups.

2.9.1 Visual Motion Capture

These methods can be divided into two main categories, Multi-Camera Systems and Single-Camera Systems.

Multi-camera systems may employ four to more than seven cameras, either static or dynamic and high-speed, to cover the whole playing field. For instance, in certain configurations, two out of four high-speed cameras are positioned on both sides of a goal post to monitor goal events closely. Additionally, a six-camera system might place four cameras along the sides and two at the ends of the court, each pairing with its adjacent end camera to offer a 3D view of the ball. This arrangement ensures that all areas are doubly covered, either by adjacent or opposite cameras, effectively dealing with the occlusion problem by offering clearer views from multiple angles, therefore solving the problem from a hardware standpoint. They are usually employed in professional sports to provide comprehensive coverage and precise motion analysis. In current times, they are crucial for real-time decision-making, aiding in important calls that impact game outcomes. These systems are deployed around the playfield, providing multiple angles for review. For example, in tennis, they help verify line calls and points, while in soccer, they are essential for determining offsides and goal verification. Their real-time data processing and analysis capabilities ensure accurate, unbiased decisions, enhancing the fairness and integrity of the game [107][108] [19].

In contrast, **Single-Camera Systems**, primarily utilize a fixed camera setup to capture events from a specific angle. While more limited in scope, have seen significant advancements thanks to deep learning and computer vision innovations. These systems now offer a practical solution for robust motion tracking, even in less controlled environments. They solve the main problems in detection and tracking from a software standpoint, by focusing on adaptive algorithms that deal with occlusions, variable lighting, and fast-paced action single-camera setups demonstrate a growing generality and accuracy, making them a viable option for applications where multi-camera systems are impractical such as in amateur contexts [58] [59].

2.9.2 Sensor-Based Motion Capture

Sensor-based motion capture utilizes devices like accelerometers, GPS, and heart rate monitors to collect data directly from athletes. This method offers training and injury prevention benefits by providing a diverse set of raw data. For instance, **Inertial Measurement Units (IMUs)** can measure acceleration, which shows how quickly an athlete is speeding up or slowing down; angular rate, which measures the rate of rotation and orientation, which by combining accelerometer, gyroscope, and magnetometer data, determines the orientation of the athlete's body or limbs in space. Additionally, **Flex Sensors** track joint motion by detecting changes in resistance when a force is applied to the sensor, they allow monitoring of the angle and movement of joints, such as the bend of an elbow or knee during a sporting activity. That data can be further analyzed and transformed into real-time feedback on athletes' performance and physiological status [65] [19]. For example, acceleration can inform sprinters about their start performance, angular rate data can

help gymnasts understand their flips and turns and joint angles can help in assessing the technique and ensuring that the athlete is not putting undue stress on particular joints.

While providing valuable data, its application in an amateur context is less viable due to several factors. Primarily, the cost and complexity of deploying wearable sensors for comprehensive data collection can be prohibitive for amateur athletes and teams. Additionally, the expertise required to analyze and interpret the collected data meaningfully may not be readily available in non-professional settings. Analyzing such data requires an understanding of biomechanics, sensor technology, data processing, and statistics. The specific challenges include dealing with the noise in the data, ensuring accurate sensor calibration, and aligning and comparing data from different systems. In practice, the gold-standard references for the task of motion capture are still optical systems such as Vicon [65] [47].

Chapter 3

Literature Review

In the domain of computer vision, deep learning (see section 2.4) has not just enhanced sports analytics but has redefined its potential, pushing boundaries far beyond the limits of traditional CV methods (see section 2.3) and simple machine learning (ML) algorithms. The following review unfolds systematically, beginning with a comprehensive summary of current CV applications within different sports.

Furthermore, particular motion capture and analysis solutions will be explored, providing an overview of their diversity, advantages, and limitations, considering this work’s scope of the amateur sports scene. These methods range from complex optical systems with multi-camera, wearables and sensors to simple single-camera ones.

Additionally, a brief overview of commercial applications is presented in section 3.3, followed by specific padel ones, for the purpose of illustrating real-world capabilities of CV systems within sports.

Finally, building upon the previous foundational concepts presented in chapter 2, special attention will be then drawn to methodologies applied to player and ball detection, tracking, and re-identification domains, respectively, in sections 3.5, 3.6, a 3.7, which fit into this work’s particular scope. Various specific models and architectures will explored. Through this analysis, the review intends to evaluate their effectiveness and how viable their applicability is to the case at hand.

3.1 Contemporary Computer Vision Applications in Sports

This subsection delves into the current applications of CV technologies across various sports disciplines. There is a summary of research papers (see section 3.1) that integrate AI in different sports, aiming to provide a comprehensive overview of the diverse range of applications and their evolving trends. Primarily, the discourse centers on football—identified as the predominant focus within the research realm by a significant margin [19], as highlighted in figure 3.1. Following football, tennis is examined as a general representation of racket sports, and then padel, for which a more

in-depth analysis is conducted for obvious reasons. Additionally, some innovative cross-sport solutions were selected. It is noteworthy that the selection of research papers, with the exception of those pertaining to padel, primarily derives from those discussed in "**A Comprehensive Review of Computer Vision in Sports**" [19]. Furthermore, an assessment of the available datasets in this domain will be conducted.

Commercial applications will be briefly presented since they provide use of these technologies beyond the research domain, therefore beginning to reshape the landscape, bringing AI-powered tools to both the professional and amateur industries, on different scales and with different purposes.

Amidst this technological surge, an essential issue has surfaced: the critical need for advanced player and ball tracking systems, essential for advancing both research and professional and amateur experiences since they form the backbone of most applications. This particular issue and other open issues will be addressed along with DL challenges and future research areas.

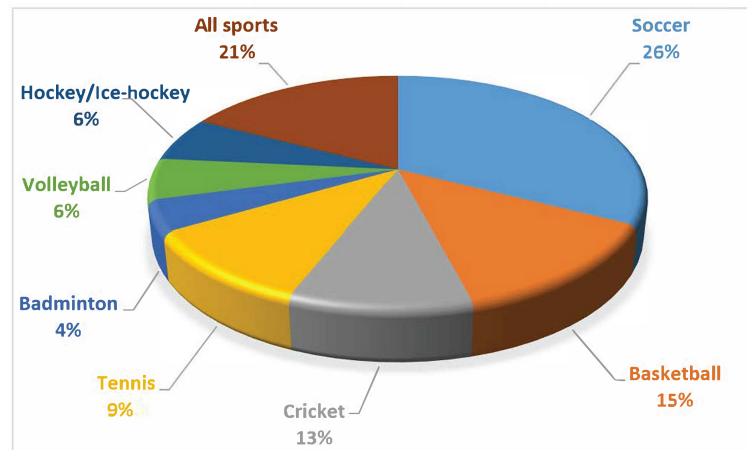


Figure 3.1: Statistics on Sports Research [19]

3.1.1 Football

- **Player and Ball Tracking:** YOLOv3 [9], one of the one-stage versions of YOLO, combined with SORT algorithms, has greatly improved player and ball tracking in football, addressing the challenge of multiple object tracking under occlusions, with notable real-time performance. The main challenge lies in discerning player identities through jerseys for identification amidst rapid and complex movements [66].
- **Tactical Analysis:** Systems like DeepPlayerTrack focus on player, referee, and ball detection, leveraging jersey color recognition to track movements and positioning, albeit with

limitations in occlusion scenarios. The MOTA score obtained, an evaluation metric for MOT, was 96% [67]

- **Event Detection:** Techniques such as variational auto-encoders (VAEs) and convolutional auto-encoders (CAEs) have been applied to detect significant events in football. For reference, VAEs are probabilistic models that learn to encode data into a lower-dimensional latent space and decode it back, often used for tasks like image generation. CAEs are specialized architectures for image processing, leveraging convolutional layers to reconstruct images from encoded representations, commonly applied in tasks such as denoising and feature extraction. These techniques showed decent levels of performance but could be improved through modifications in the DL architectures [?][69].
- **Action Recognition:** On a similar note as Event Detection, several systems using custom-made CNNs, YOLO, are being used for recognizing referee actions and classifying player movements, aiming to provide real-time, accurate analytics for strategic game assessments [70; 71; 72].
- **Game Summarization:** Building upon Event Detection/Extraction and Action Recognition efforts, progress is also being made on automatically generating summaries of sports videos [73].

3.1.2 Tennis

- **Player and Ball Detection:** After comparing several models, YOLOv3 [9] with some minor tweaks have been utilized to detect the players in videos where they appear as small targets, which are particularly difficult conditions. The same model has also been used to detect the ball in several frames and use those detections to predict landing using simple difference equations, which seems like a rudimentary foundation for the tracking-by-detection methodology, which was described in section 2.7.3. The results for high-resolution videos were quite satisfactory, as shown by the mean-average-precision(mAP) evaluation metric [74].
- **Ball Tracking:** A well-known solution for this task is TrackNet. It is fundamentally an object detection model that uses CNNs to process sequential video frames and generate a heatmap indicating the position of objects, specifically designed for tracking the ball in racket sports. The heatmap is based on a probability-like detection that is trained to represent the object's location. The system can work with a single frame input, which would be a conventional CNN, or with multiple frames, which improves the detection by learning the trajectory pattern of the moving object. In essence, while TrackNet does incorporate aspects of tracking by using consecutive frames to refine the detection accuracy, it is not a full tracking system. The consecutive frames help in recognizing the trajectory patterns, which is essential for predicting the position of fast-moving objects like a tennis ball, especially when occlusions occur. The network does not track the object over extended sequences to predict future

positions or maintain object identities over time; rather, it focuses on accurately detecting the object's position within a short sequence of frames. Therefore it is not designed to provide long-term tracking that would, for example, follow the ball throughout an entire match. Instead, it specializes in moment-to-moment detection with high accuracy, which is useful for analyzing specific plays or sequences within a sports event [40]. Nevertheless, it was trained on a large dataset of 20844 frames and obtained F1 measures above 90%. However, it is important to note that the test set included frames from the same videos as the training set, therefore they present exactly same conditions in some cases. Consequently, the results present a bias and might not evaluate correctly the model's ability to generalize.

- **Player Action Recognition:** Using 3D video data, a complex process of feature extraction and optimization, resorting to tailored ML algorithms is conducted. Afterward, temporal DL unsupervised models (refer to 2.4.3), such as Temporal Deep Belief Networks, showed high accuracy, proving their ability to leverage contextual information from preceding and subsequent frames. On a side note, there is potential for improvement in differentiating similar movements [75].
- **Swing Classification:** A combination of SVM, neural networks, and decision trees has led to high classification accuracy, yet the challenge remains in generalizing across different play styles [76]. A possible approach could be focusing mainly on DL techniques, resorting to high quantities of quality annotated data.
- **Semantic events detection:** There are six main events in tennis according to its rules. The service, ace, fault, double fault, baseline rally, and net approach. A proposed method for the detection of those events builds upon a pre-processing phase in which there is a manual filtering of the playfield and court line, followed by a kalman-based [112] tracking-by-detection robust player tracking system [106]. Afterward, the players' trajectories are mathematically refined for a two-dimensional perspective. A decision tree-like logic is then applied, which uses tennis rules to determine the occurrence of an event, although the use of an AI model is not explicitly stated. The results obtained were satisfactory, reaching values of precision and recall over 70% for all different events. Nevertheless, it is relevant to note the fact the dataset used is small, being composed of only 103 clips in which the "double fault" event, in particular, is very rare among the clips.
- **Highlight Extraction:** Similarly to football, applying motion analysis methods, such as tracking and action recognition, as previously presented, paves the way for more complex semantic extractions. After video summarization, extracting highlights becomes a possible task. However, due to its naturally complex nature, it is mostly dominated by solutions based on audio cues or a mixture between some motion analysis and audio [31].

3.1.3 Padel

3.1.3.1 Court Detection

Extracting the court lines in padel is a useful task since it provides a foundation to understand the spatial layout of the court. Once the court is detected, CV algorithms can determine the boundaries of it, the location of the net, and the position of other objects inside. This information might be of great aid for other applications. For example, player tracking algorithms can use the court boundaries to determine whether a player is inbounds or out of bounds. Additionally, automated line-calling algorithms can use the court boundaries to determine whether a ball has landed in or out of bounds.

A novel approach for this task was recently conducted [41]. It explores the ability of detecting the court even when some important elements are not visible which opens up the possibility for more versatile and practical court detection systems. In respect to the methodology, it employs Design Science Research (DSR), which is a research paradigm that focuses on the design and evaluation of artifacts in order to solve real-world problems. In this case, the artifact is a court detection program. The program is developed using a combination of Traditional CV techniques, including background extraction, filtering, canny edge detection, hough lines, and homography. The images are extracted from a single camera angle for ease

The evaluation of the program is based on a dataset of real-world padel sports videos. This is a strength of the study, as it ensures that the findings are grounded in practical applications. The accuracy of the program is measured based on two units: court detection inaccuracy percentage and average pixel distance. The figure 3.2 demonstrates a correct identification of a court.

The empirical results indicate that the developed artifact demonstrates a relatively consistent level of accuracy in court detection across multiple courts whenever a detection was made. However, the frequency of successful detections exhibited some inconsistency. This suggests that the program is capable of accurately detecting courts in a variety of conditions, but there are still some challenges to be addressed in terms of ensuring consistent performance. The study also found that external factors did not significantly influence the accuracy of court detection, yet they posed challenges to the program's overall consistency. This is an important finding, as it suggests that the program has the potential of being robust to external factors, i.e., able to generalize. Some potential improvements could be the integration of ML algorithms with a larger quantity of data [41] and a formal evaluation of the impact from additional external factors such as lighting conditions and camera angle on the accuracy in order to further optimize the artifact when it comes to its consistency issue. Additionally, the proposed solution is not compared to any other solutions for the same task; therefore, there are no relative means of comparison or proper benchmarking.

3.1.3.2 Ball Detection and Tracking

Despite the growing popularity of padel, there is a noticeable absence of literature specifically addressing ball tracking in this sport. This gap in research presents an opportunity for novel contributions to the field.

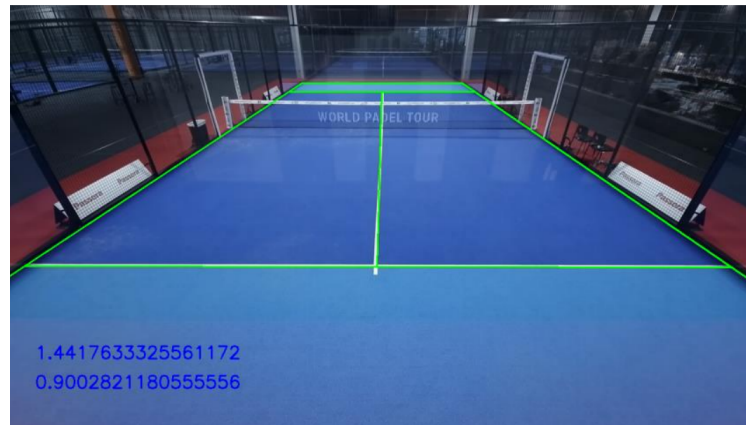


Figure 3.2: Example of a correct court detection [41]

One of the research papers directly addressing ball detection in padel [43] explore three different object detecting techniques for small and fast-moving objects using image processing. In this particular use case, the target of study was the ball from professional padel matches. The techniques proposed include two Traditional CV techniques 2.3, CAMshift and Canny edge detection, and a DL model, YOLO, pre-trained with the COCO dataset. Unfortunately, the version of YOLO used was not specified. Nevertheless, comparing traditional techniques with DL is an interesting and novel approach. On the other hand, the dataset is not properly described but seems very limited, including only two videos, and it is not clarified whether the testing or validation was done from a different video of the training set, which is an important measure to verify generalizability. Furthermore, it is stated that it does not consider occlusions, which is obviously one of the challenges in ball detection during padel matches. When it comes to the results, unfortunately, CAMshift was discarded due to irrelevant output, which can be seen in figure 3.3. Canny edge detection revealed good empirical results, displayed in figure 3.4, but with the crucial downside of them having to be manually observed since the algorithm itself doesn't perform any classification. As a consequence, the comparison with the YOLO model is not totally unbiased. Canny is claimed as a superior option, reaching a five times higher performance in terms of "uptime", which is calculated as the total number of frames in the video where the algorithm successfully detects the ball, divided by the total number of frames in the video. These numbers, with respect to YOLO, on this custom-made evaluation metric, can probably be explained by the low amount of data, particularly in the detection of small objects such as the ball. On the other hand, since the dataset consists of only two videos in similar visual conditions, manually fine-tuning Canny provides expectedly good results. Considering all those factors, it's fair to assume the study lacks an unbiased comparison between those methods. Nevertheless, besides expanding the dataset for a balanced comparison, one possibly interesting addition could be mixing canny edge detection for extracting smaller regions of the image and then using those images as input to YOLO, to test whether the performance increases, compared to their previous individual outcomes.

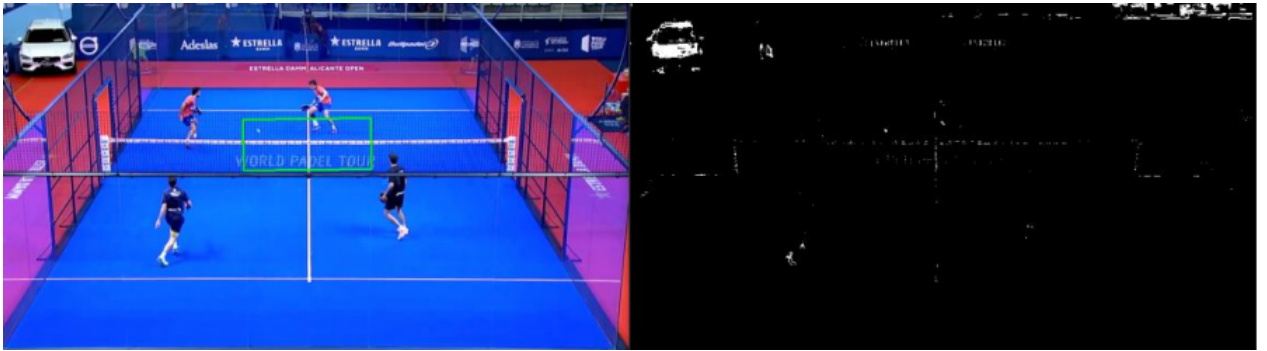


Figure 3.3: Camshift example detection [43]



Figure 3.4: Canny edge detection example, ball represented in red [43]



Figure 3.5: YOLO detection example [43]

When it comes to actual ball tracking, some adaptations to the previously mentioned Track-net [40] in tennis, addressed in section 3.1.2m were made for proper experiments in padel [48],

employing a tracking-by-detection methodology. An ablation study was conducted to optimize the detection performance by varying input variants and parameters, such as the number of input frames and the training schedule. For trajectory fitting, a RANSAC-style method was adopted, which involves fitting parabolic curves to represent the ball's path. This method was applied incrementally, linking seed triplets across frames to build a comprehensive trajectory. The dataset used in this research consists of videos of amateur padel matches captured from an overhead or standard view. The videos were manually annotated to provide ground truth data for ball detection and trajectory fitting, consisting of 10 videos of amateur matches, each lasting approximately 10 minutes. The videos were captured from an overhead view at a resolution of 1280x720 pixels and a frame rate of 25 frames per second. A total of 150,000 frames were manually annotated to provide ground truth data for ball detection and trajectory fitting, which are remarkable numbers. Unfortunately, the dataset is not provided for public access. In respect to the results, the system demonstrates high accuracy in ball detection, achieving a mean average precision (mAP) of 0.95. The trajectory fitting is also effective, capturing the ball's path with a mean error of less than 5 pixels. Nevertheless, as previously stated in section 3.1.2, Tracknet is highly dependent on subsequent frames detection and, therefore, not fit for a scenario where the purpose is tracking the ball throughout a full match, which is this work scope, being better suited for small clips of matches.

3.1.3.3 Player Detection and Tracking

In the context of padel, it is known that player tracking is particularly challenging due to the unique features of the sport. Unlike tennis, padel is almost exclusively played in doubles, which increases the risk of interplayer occlusion in low-angle and moderately high-angle shots. Additionally, the playing field is enclosed by walls, which can cause visibility occlusion and impact game development [58]. On the other hand, since there are only four players to be tracked and a single static camera can be used for all of them, it composes fewer challenges than field sports.

Despite these challenges, recent research has begun to explore the potential of computer vision techniques for player tracking in padel. In 2021, Javadiha, M et al. [58] robustly evaluated and compared the accuracy of state-of-the-art computer vision methods on a large set of images from both amateur videos and publicly available videos from the major international padel circuit. The methods analyzed included object detection, image segmentation, and pose estimation techniques, all based on deep CNNs. For object detection, algorithms such as Faster R-CNN [102] (see section 3.5.2), Cascade R-CNN [46], and MMDetection were used. For image segmentation, Hybrid Task Cascade. For pose estimation, OpenPose [51], Simple Baselines for Human Pose Estimation and Tracking, and HigherHRNet [104]. The results of the study showed that the best results were obtained by Top-Down pose estimation methods, which offered a detection rate of 99.8% and a Root Mean Square Error (RMSE) below 5 and 12 cm for horizontal/vertical court-space coordinates, respectively. Among the top-down pose estimation methods, the combination of Faster R-CNN [102] with HRNet [104] for locating the key points of each detected player provided the best results. This demonstrated the suitability of pose estimation methods based on CNNs for estimating player positions from single-angle padel videos. From an analytical perspective, the

fact that top-down approaches achieved better results than bottom-up approaches was due to the lower number of objects to detect, which is the case of racket sports. It is important to note that the accuracy of these methods was approximately the same as human annotators, demonstrating the effectiveness and viability of these computer vision techniques in player tracking in padel, with a rather simple setup such as a single static camera, without the need for complex sensors or complex set of cameras. On the other hand, a major criticism of this study is that no metrics regarding efficiency or real-time viability are presented.

Another relevant study in this domain is a Master's thesis entitled "**Aprendizagem automática para treinador virtual de padel**" [59], which means "Machine Learning for Virtual padel Coach". Moreira, M [59] proposed a tool that receives a video of a padel session and suggests a better game strategy for the player than the ones presented in the video sequence given as input. The tool is divided into two major components: Vision System and Virtual Coach. The most important component for this study is its Vision System, which detects and tracks each player individually. For the players' detection YOLOv4 was used. It was trained on a dataset of padel images and achieved a high mean Average Precision (mAP) of 99.82%, indicating its effectiveness at detecting players in the video frames. The Object Detection model was trained with transfer learning on a part of a padel dataset and it consisted of 320 manually annotated images. For tracking Moreira, M resorted to DeepSort [62] leveraging a pre-trained CNN that was trained on a dataset related to person re-identification, a theoretical concept address in section 2.8. There was no other training process, only parameter manipulation and experimentation. The best configuration achieved an IDP (Identification Precision) of 94.4%, IDA (Identification Accuracy) of 99.4%, and IDF1 (Identification F1 Score) of 96.8%. Since only one video was used for testing, overfit might be a very considerate risk in this study. Once again, in this study, there are not any mentions regarding efficiency even though the choice on YOLOv4 suggests an intention of speed and real-time performance [59]. Another critic is the fact that there was not an effort to compare to other possible algorithms despite the positive results delivered.

3.1.3.4 Automatic Generation of Synthetic Multi-View Dataset

Due to the large gap of datasets explicitly containing padel content, namely for player tracking purposes, recent efforts have been made in order to study a pipeline for the generation of a complete dataset containing player positions and poses [42].

The methodology respected the following order :

- Motion capture: A professional, suit-based motion capture system was used to capture the pose and position of one of the players during a padel match. The motion capture data was collected at a rate of 120 Hz and was compared to a gold-standard system of motion capture [47], which will be presented on section 3.4.1. It achieved comparable accuracy on both positional and orientation measures, only varying less than 5 millimeters and 2 degrees, respectively.

- 3D model creation: A 3D model of the player was created using the motion capture data. The 3D model was created using Blender, an open-source 3D modeling software package.
- Animation: The 3D model was animated using the motion capture data. The animation was created using MotionBuilder, a professional motion editing software package.
- Video rendering: The animated 3D model was rendered from a variety of camera views to create synthetic videos. The synthetic videos were rendered using Maya, a professional 3D rendering software package.
- Player detection and pose model training: After generating the dataset, its practicality was tested. The method used for detection was the HTC [77], and the pose estimation was implemented with a top-down HRNet [104] approach. This solution was based on the best performing option from the paper [58], presented in the section 3.1.3.3.
- Player tracking training: For this purpose, it was used a non-specified, custom-built algorithm that matches detected persons with players based on the overlap of detections on the video frames, using the IoU(Intersection Over Union) as an overlap metric [42].

The generated dataset is described in more depth on section 3.2.3. Despite the innovative and valuable aspect of this approach, the solution has some margin for improvement, namely in the lack of explicit ball tracking data, in the fact that only one amateur match is included, resulting in a small dataset with low variability and in only including one player on the synthetic 3D generation. The second issue could perhaps be solved by resorting to simulations using a physics-based simulator. That simulator would use the previously collected data to learn the laws of physics that govern padel matches. This data would include the players' position and pose and the friction and other properties of the playing surface. This would, in theory, allow for unlimited data with high variety and controllable conditions. Another alternative could be the use of a GAN for generating more data but the realism obtained from this option would be theoretically inferior.

The lack of ball data is justified by the remaining challenge of robust and accurate ball tracking [42]. In the particular case of padel, this challenge is mainly due to occlusions and the occurrence of wall bounces in-game [42].

3.1.4 Generic Cross-Sport Solutions

- Cross-Sport Video Summarization [33]: Relying on user-generated sports video(UGSV), i.e., shared by individuals rather than professional media outlets or sports organizations, this study deals with unique challenges. Its approach circumvents this issue by focusing on the actions of the players as the primary cue for determining highlights, shifting from relying on editing cues generally presented on professional broadcasts such as replays and narration.

Additionally, its cross-sport applicability is also intriguing as a novel approach, leveraging the universal language of player movements rather than depending on sport-specific structures or rules. The methodology presented employs a two-stream neural network to extract both body joint-based features (3D and 2D) and holistic features from video segments. LSTM networks are then used to model the temporal dependencies among the extracted features, classifying each video segment as either a highlight or not.

When it comes to results, the F-Score metric revealed its superiority relative to other UGSV summarization methods. The evaluation of this system was not only objectively targeted by the F-Score metric but also subjectively, surveying different participants, considering both experienced and inexperienced people with the sport, which is an interesting approach. With this assessment, it was highlighted the challenge of maintaining highlight completeness and continuity of the video, which affected user experience.

- **Classification of sports video [32]:** This study aims to classify sports videos into 15 categories using a DL approach that combines CNNs and RNNs, enhanced by transfer learning with the VGG-16 model, which was pretrained on a common generic dataset called Imagenet. It addresses the challenge of accurately categorizing sports content for easier archiving and retrieval, contributing to broader video scene understanding. With a hand-crafted dataset focused on action-based classification, the method achieved high test accuracies of 94% and 92% for 10 and 15 sports classes, respectively, showcasing its effectiveness.

Although the study's method demonstrates significant effectiveness in classifying sports videos within a limited scope of 15 categories, this range only scratches the surface of the vast array of sports disciplines. A more expansive approach could involve categorizing sports into broader, more encompassing categories such as field sports, combat sports, indoor sports, and outdoor sports and expanding from that. This adjustment would not only extend the applicability of the classification model but also enhance its utility in archiving and retrieving a wider variety of sports content, thereby contributing more substantially to the field of video scene understanding

3.1.5 Challenges and Research directions

The application of CV in sports is a growing field with immense potential, as already overviewed, yet some key challenges remain. Two needs stand out across sports disciplines: robust player tracking and ball tracking systems.

The lack of large-scale, high-quality datasets for many sports presents a significant obstacle, as deep learning models require substantial annotated data for training. For emerging sports like padel, publicly available datasets are scarce, though efforts like PadelVic [42] demonstrate early attempts at addressing this gap. Expanding open-access datasets will be instrumental in advancing CV research and applications in sports. Relevant examples will be displayed in section 3.2.

Generating accurate systems for tracking players and balls is a complex endeavor but foundational for richer analysis and semantics extraction. Factors like rapid motions, occlusions, and

variance across environments make maintaining identities and trajectories challenging. However, there are some relevant complex motion capture systems specialized in that task, some of which are presented in section 3.4. Besides those systems, progress is also being made on robust tracking methods based on simpler software-focused solutions, often combining SOTA detection, tracking, and re-identification techniques. A selection of those techniques will be overviewed in sections 3.6, 3.5, 3.7. Extending these to provide precise tracking across scenarios is an active research direction.

3.2 Existent Datasets

For DL models to achieve high performance, access to large datasets with quality annotations is critical. This requirement is foundational as it allows models to learn and generalize from vast and varied examples, enhancing their accuracy and robustness in real-world applications. CV applications within sports are no exception. Therefore, publicly accessible sports video datasets complete with annotations are needed to alleviate the extensive labor in manual video collection and annotation and provide a common ground for benchmarking between different models [19].

3.2.1 Sports-1M

A comprehensive collection¹ of over one million YouTube video URLs, meticulously annotated with 487 distinct sports labels. These labels were generated automatically using the YouTube Topics API and are organized within a manually curated taxonomy, encompassing a wide range of sports categories such as Aquatic Sports, Team Sports, Winter Sports, Ball Sports, Combat Sports, Sports with Animals, and more [34]. It offers a vast array of video content that can be used to train and benchmark algorithms aimed at understanding and analyzing sports activities through video data.

3.2.2 OpenTTGames

A dataset² which evaluates various computer vision tasks in Table Tennis, including ball detection, semantic segmentation, and spotting fast in-game events. It features full-HD videos recorded at 120 fps, enriched with annotations for frame-specific events and ball coordinates, alongside segmentation masks, as displayed in figure 3.6. The dataset comprises 5 training videos (10 to 25 minutes each) and 7 shorter testing videos, all equipped with detailed annotations [38]. The dataset was manually labeled for in-game events such as ball bounces, net hits, or empty event targets and for ball coordinates and segmentation masks, with the aid of DL annotation models [64].

¹<https://paperswithcode.com/dataset/sports-1m>

²<https://lab.osai.ai/>



Figure 3.6: Example of a frame with corresponding segmentation masks [38]

3.2.3 PadelVic

This dataset³ consists in a collection of multimodal data captured during an amateur padel match and it is the only publicly available dataset for padel that includes both multi-view video streams and accurate motion capture data. It is composed of:

- **Multicamera Videos:** The dataset includes high-quality videos from multiple cameras, capturing different angles of the match. The main camera provides a standard viewpoint of the court, while additional cameras offer alternative perspectives, allowing for a richer analysis of player movements and actions.
- **Motion Capture Data:** One of the players during the match was equipped with professional motion capture (MoCap) system, which recorded detailed body poses at a high frame rate. This data is valuable for tasks such as pose estimation and movement analysis.
- **Synthetic Videos:** Leveraging the motion capture data, synthetic videos of virtual avatars replicating the recorded player's movements were created using Blender, a 3D modeling software. These videos come with pixel-accurate annotations and can be rendered from any desired camera angle, providing ground truth for various computer vision tasks.
- **Annotations:** The dataset includes a rich set of annotations, such as the 3D position of joints, derived data like the orthogonal projection of the feet and hips onto the court plane, and labels that are accurate even for keypoints obscured by clothing. They were explicitly composed by the player position, i.e., the 2D position of each player in each frame of the video, as the projection of their hip onto the court and by the player pose, i.e., the 3D pose of each player in each frame of the video, as the orientation of their body parts.

It can be valuable for player tracking, pose estimation, and action recognition. The availability of both real and synthetic data makes it a considerable tool for testing the robustness of CV algorithms across different scenarios and for training models with accurate ground-truth data.

³<https://github.com/UPC-ViRVIG/PadelVic>

Despite only containing data from a single match, PadelVic dataset⁴, thus, represents a significant contribution to sports analytics, providing a resource that addresses the significant gap in the availability of specialized datasets for padel [42].

3.2.4 Roboflow

It is not a single dataset but a platform⁵ that offers a vast collection of publicly shared datasets for various computer vision projects. It enables any user to find and utilize datasets for training machine learning models, democratizing its access. Three main types of simple padel datasets were found :

- Detection and classification of player actions [35], presented on figure 3.7
- Detection of the net [36], presented on figure 3.8
- Player and ball detection [37], presented on figure 3.9

Unfortunately, neither pose estimation or tracking annotations are available on the platform, the only possible types are bounding boxes and segmentation masks. Nevertheless, it is still a valuable tool and a great aid in democratizing access to CV datasets.



Figure 3.7: Example of a frame with a volley detection [35]

⁴<https://github.com/UPC-ViRVIG/PadelVic>

⁵<https://universe.roboflow.com/>



Figure 3.8: Example of a frame with net detection [36]

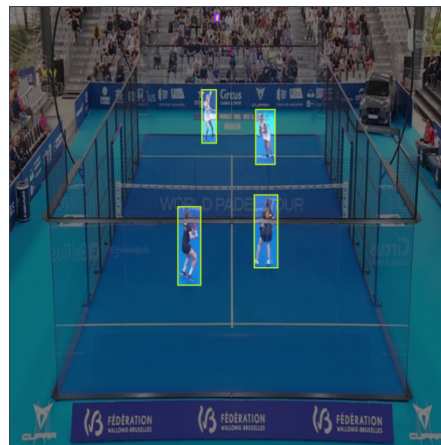


Figure 3.9: Example of a frame with players and ball detection [37]

3.3 Commercial applications

Computer vision is transforming aspects of the sports industry. Algorithms can now analyze sports imagery to automate tasks like video production, personalized coaching, referee assistance, and even fully automated officiating. There are 4 major areas where commercial computer vision is adding value [19].

- Automated highlight reel production, CV techniques are being applied to auto-generate sports highlight reels from live footage. For instance, there was proposed an approach to automatically identify interesting moments such as goals in soccer video and edit together highlight clips to share online [49]. WSC Sports, a company that specializes in using AI and CV technology for sports video analysis and automated highlight generation, uses similar techniques during live games [50]
- Referee decision support systems, for instance, Video Assistant Referee (VAR) leverage CV

to assist referees in reviewing critical decisions during games. Algorithms analyze footage from multiple camera angles to help detect fouls, offsides, and other events. Key tasks, such as player and line detections and 3D reconstruction, give referees more information for decision-making. This technology has been implemented by major organizations like FIFA to improve officiating accuracy.

- Fully automated officiating technologies are being made possible through advanced CV. The Hawk-Eye system [107], which will be further analyzed in section 3.4.2, uses real-time visual tracking and predictive impact analytics to automatically make line calls in tennis, removing human subjectivity. This technology aims to bring more precision and consistency to sports officiating decisions and reduce human error.
- AI Coaching systems, already established systems utilize sensor data, explored in section 2.9.2, and biomechanical CV algorithms to provide coaches with detailed performance analysis related to factors such as acceleration, jumping, and cutting mechanics. This allows for individualized training programs and technique improvements.

3.3.1 Padel Applications

Despite this work not being focused on delivering a mobile application for the amateur players themselves, there was conducted a short market analysis for the purpose of evaluating the current capabilities of the platforms at their disposal, thus exploring existing solutions and innovation attempts for their community. After a comprehensive review, four main types of applications for padel were found:

- **Amateur Leagues Official Apps:** allow players to manage their matches, access their individual statistics, keep track of game schedules, rankings, and contact information of opponents [?] [95]. They might also have a social component.
- **Court Rental Apps:** These simply allow players to find the nearest courts and schedule a rental of a specific one, allowing for a reservation [96].
- **Score Management and Performance Tracking Applications:** These go a step closer to the intended functionalities in this study since their primary purpose is to enhance amateur players' experience. However, there is barely any AI, and the score counting is done manually on a wearable smartwatch. Furthermore, the performance tracking functionality is solely a "Workout" feature in resemblance to many other exercise-focused apps [97]
- **AI assisted Multi-Functional Apps:** These apps provide different functionalities such as scorekeeping seen in figure 3.10, real-time full body motion capture with player detection and pose estimation, seen in figure 3.11 and match video sharing, integrated into a video distribution system and allowing the sharing of matches, seen in figure 3.12. All of those resorting to user-generated sports videos, which is a difficult approach.



Figure 3.10: Scorekeeping [98]



Figure 3.11: Pose Estimation [98]



Figure 3.12: Match Recording [98]

3.4 Motion Capture and Analysis Methods

As already described in section 2.9, motion capture technologies span professional multi-camera systems, wearable sensors, and accessible single-camera computer vision solutions. Each approach comes with trade-offs in applicability, cost, and capability.

Sensor-based systems, while affordable and accessible, are limited in scope and affected by device calibration, noise, and interpretation needs. They offer personal motion insights but lack multi-camera rigor. Therefore, only concrete optical methods will be presented.

Multi-camera systems, such as Vicon and Hawk-Eye, which are further described in sections 3.4.1 and 3.4.2, provide comprehensive motion data through specialized camera networks and reconstruction software. These systems achieve robust results by solving occlusion and accuracy challenges from a primarily hardware perspective, using extensive camera infrastructure. However, this leads to high costs and constrained applicability outside dedicated environments.

Single-camera systems, addressed in section 3.4.3, on the other hand, offer a scalable middle ground by tackling challenges in motion tracking mainly through software techniques. Advances in algorithms such as YOLO enable players and ball detection using accessible video camera hardware. While view-dependent, single-camera methods trade off hardware complexity for soft-

ware adaptability across diverse real-world contexts. This helps overcome cost and environmental barriers to motion capture for amateur athletes and coaches.

3.4.1 Vicon

Vicon utilizes a network of high-speed infrared cameras surrounding the capture space to precisely track reflective markers placed on a subject. Each camera provides a separate view of the markers, with the system combining the 2D data points into highly accurate 3D coordinates through triangulation. For example, a typical setup may involve 10 cameras recording at 300 Hz to track a runner's biomechanics. Fifty reflective markers are carefully placed on anatomical landmarks on the runner. As the subject runs on a treadmill, the cameras track the 3D trajectories of each marker. This produces detailed quantitative motion data on factors like gait mechanics, joint angles, ground reaction forces, and more. While unmatched in precision, Vicon requires a controlled lab setting, technical staff, and significant infrastructure costs.

3.4.2 Hawkeye

Whereas Vicon focuses on scientific analyses, Hawk-Eye demonstrates multi-camera motion tracking tailored for officiating and broadcasts [107]. Deployed in professional tennis tournaments, the Hawk-Eye system uses 10 synchronized high-speed cameras around the court for officiating. Each camera covers a section of the court to provide complete visual coverage. As a serve occurs, the cameras capture the ball trajectory from multiple angles. The system reconstructs a 3D map of the ball path by combining the 2D views. Physics models are used to predict the ball's trajectory over the net. This 3D visualization and prediction enables determinations of whether serves are in or out. Hawk-Eye demonstrates how tailored multi-view setups can provide accurate motion tracking tailored for specific sports environments. This visualization enhances officiating calls and viewer experiences [108] as there are cameras angled directly to the lines that limit the courts, such as in figure 3.13. But, similarly to Vicon, Hawk-Eye depends on specific, expensive, multi-camera installations.

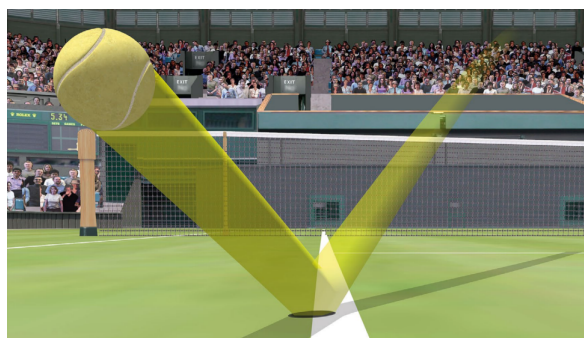


Figure 3.13: Hawkeye camera angle example [107]

3.4.3 Single-Camera Motion Tracking

Recent research has demonstrated the potential of single-camera motion tracking through innovations in computer vision algorithms. For instance, a player position and pose system [58] was developed using only a single, fixed camera angle overlooking a padel court, as previously analyzed in section 3.1.3.3. The single-camera system achieved comparable accuracy to multi-camera setups for tracking players in the padel videos. This exemplifies the capabilities unlocked by advancing software techniques rather than hardware alone. However, factors like occlusion handling remain challenges to expand the applicability to real-world situations, requiring a robust system. In an attempt to fill this gap, potential DL techniques for the design of such a system will be addressed in the following sections.

3.5 Object Detection

Building upon the background concepts of object detection presented in section 2.6, this section will provide an in-depth analysis of specific object detection models and their viability for player and ball tracking in padel. The ability to accurately and efficiently detect these objects in each video frame lays the foundation for robust tracking-by-detection.

The object detection models explored here can be categorized into two main architectural approaches - two-stage models and one-stage models. **Two-stage detectors** such as Faster R-CNN (see section 3.5.2) operate in two phases, first generating region proposals which are then classified. In contrast, **One-stage detectors** like YOLO and RetinaNet, which are further explored in sections 3.5.1 and 3.5.3, predict bounding boxes and class probabilities directly from the full image in a single inference pass. Each approach has trade-offs between accuracy and speed. Although TrackNet [40] was used in several studies [?, [43] for ball detection, its architectural simplicity and the need for subsequent frames turns it into an enviable candidate for videos of full matches. Therefore it won't be analyzed in-depth.

Real-time processing requirements, small object detection, and dealing with occlusions are factors that will inform the assessment of these models' suitability for the task of player and ball detection in padel. Both concrete prominent two-stage and one-stage detectors will be analyzed based on recent research and benchmarks. Their advantages and limitations will be discussed in relation to the needs of a padel robust detection system. This evaluation will provide a recommendation for the optimal object detection approaches to employ in the proposed Methodology, explored in chapter 4.

3.5.1 YOLO

YOLO (You Only Look Once), an object detection algorithm, is considered a revolutionary approach to solving the task of object detection in real-time. It does so by applying a single neural network to the full image, dividing it into regions (see figure 3.14), and predicting bounding boxes

and probabilities for each region. These predictions are then combined to produce the final detection results. Other general methods, such as Faster-RCNN, perform detection by first identifying regions of interest and then classifying those regions using CNNs, which is a longer process and is not as suitable for real-time applications such as live player tracking.

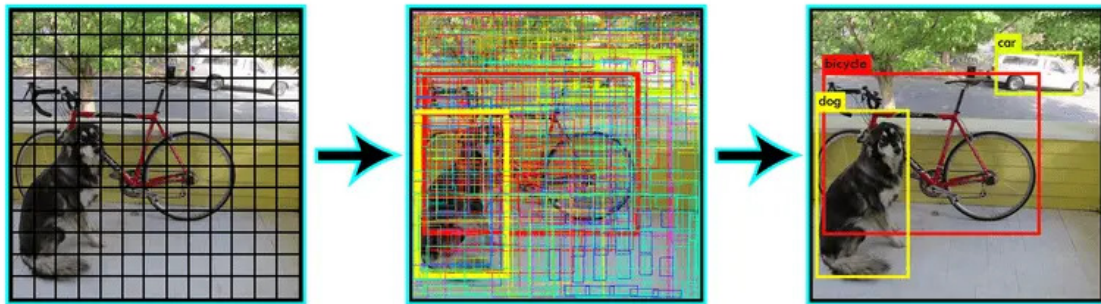


Figure 3.14: YOLO example [24]

On a deeper analysis, YOLO begins by dividing the input image into an $S \times S$ grid, being S a fixed size. This means each cell can predict multiple boxes if it believes that there are multiple objects within it. Then, a single CNN is applied to the entire image, in which its convolutional layers apply filters to the image to create feature maps that abstract the raw pixel data into a form where features, such as edges, textures, and shapes are highlighted. After the feature extraction, bounding box along associated scores prediction occurs, in each grid cell, independently. These scores are, respectively, the confidence score, which reflects the model's certainty that the box contains an object and how accurate it considers the box is, and, where an object is detected, the C class probabilities of it. These bounding boxes are parameterized by the coordinates of their center (x, y) , their width (w) , and height (h) . The (x, y) coordinates are normalized to fall between 0 and 1 within the grid cell, and the width and height are predicted relative to the whole image. The raw output of the network is in the form of a tensor, with an $S \times S \times (B \cdot 5 + C)$ shape, with S rows, S columns, and $B \cdot 5 + C$ channels. However, there is still a post-processing step, where the output undergoes through a Non-Max Suppression because, since each grid cell can predict several bounding boxes, there can be overlapping boxes for a single object. This process filters out all bounding boxes with a confidence score below a user-given threshold or predefined in a configuration and then selects the highest confidence box while suppressing non-maximal boxes (i.e., boxes that overlap significantly with the selected box but have a lower confidence score) [8].

This unified architecture can be trained end-to-end on full images for fast, accurate detection. As seen in figure 3.16, YOLO models have evolved through multiple versions with various improvements [9]:

- YOLOv2 added batch normalization for faster training, trained on higher resolution images, and introduced anchor boxes, which significantly increased its accuracy and speed.
- YOLOv3 improved feature extraction, added a new detection head, and generated predictions at 3 scales to improve detection of small objects.

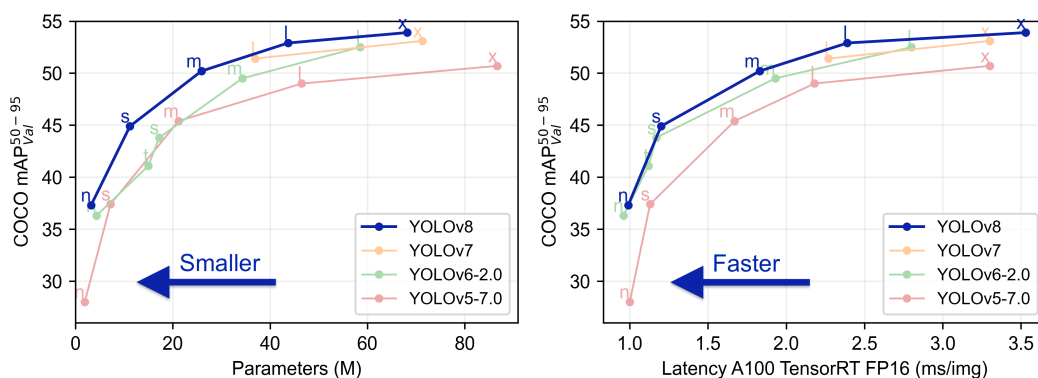


Figure 3.16: Evolution of YOLO models [113]

3.5.2 Faster RCNN

Faster R-CNN is a popular two-stage object detection algorithm introduced in 2015 [103]. It improves on previous region-based CNN methods such as R-CNN [52] and Fast R-CNN [53] by introducing a Region Proposal Network (RPN) as show on figure 3.17, that shares convolutional features with the detection network, thus enabling nearly cost-free region proposals.

The Faster R-CNN architecture consists of two modules. The Region Proposal Network (RPN), which is a fully convolutional network that takes an input image and outputs a set of rectangular object proposals each with an objectness score. The RPN ranks region boxes and proposes the ones most likely to contain objects. The second module is the Fast R-CNN Detector which extracts regions of interest (ROI) pooled features using the proposals from the RPN and performs classification and bounding box regression. The features between the RPN and detector are shared.

During training, RPN and Fast R-CNN are trained alternately to update the shared convolutional layers. At test time, the RPN generates region proposals which are used by Fast R-CNN to make the final detections in a single forward pass.

Some key advantages of Faster R-CNN are its nearly cost-free region proposals by sharing computation with the detection network, the high-quality region proposals from the learned RPN improve detection accuracy and its unified framework for object detection with shared convolutional features.

Overall, Faster R-CNN provides an efficient and accurate approach to object detection by combining semantic feature extraction, region proposal generation, classification and regression in a single unified framework.

In the context of padel video analysis, Faster R-CNN shows promise for reliably detecting players and balls, as demonstrated by its use as a means of comparison on that particular scope [58]. Its region proposal network can hypothesize object locations even amidst occlusions, while the Fast R-CNN module has proven adept at discerning objects from such proposals. As a consequence, Faster R-CNN's accuracy across scales, rotations, and heavy occlusion will aid in robustly identifying players in diverse poses and positions. Theoretically, the model should also

be adequate for near real-time rates aswell. Therefore, being a larger network than YOLO but still efficient, Faster R-CNN is well-positioned as a candidate model for experimentation in padel player and ball detection, as its unified detection framework combining proposals, features, and classification provides advantages for this application.

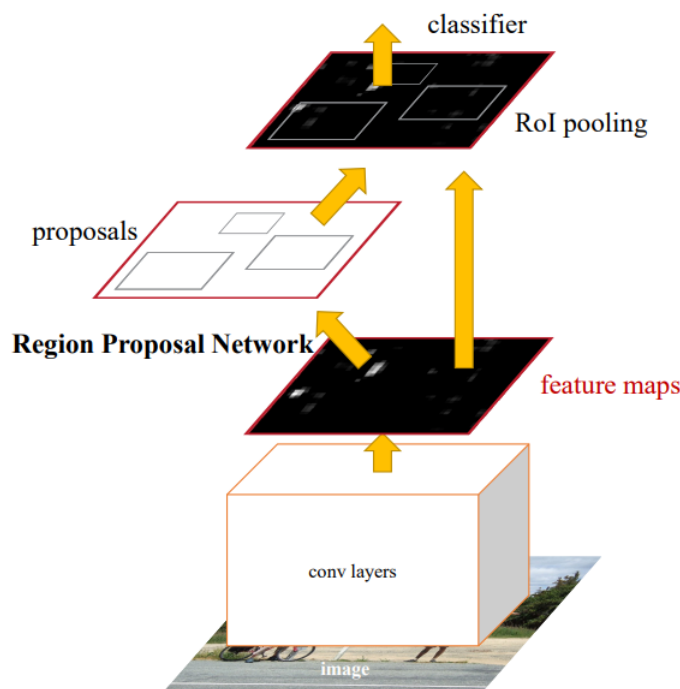


Figure 3.17: Faster R-CNN, represented as a single unified network for detection tasks [103]

3.5.3 RetinaNet

As a one-stage dense object detector (see section 2.6.3), it compares directly to YOLO. However, it introduces the focal loss to handle class imbalance [54], which might reveal itself as a relevant feature, possibly eliminating the accuracy differences between one-stage detectors and state-of-the-art two-stage detectors such as Faster R-CNN 3.5.2 with FPN [55], while being more efficient [54]. Its unified network [54], as displayed in figure 3.18 is composed of:

- **Backbone:** RetinaNet uses a Feature Pyramid Network (FPN) based on a feedforward ResNet [56] architecture. The FPN extracts multi-scale convolutional feature maps efficiently from a single input image.

Classification Subnet: This small Fully Convolutional Layer (FCN) attached to each FPN level predicts the probability of object presence for each anchor at each spatial location. It applies four 3x3 convolutional layers followed by a 3x3 conv output layer.

Box Regression Subnet: A similar small FCN regressing from each anchor to a nearby ground truth object, if one exists. Outputs are class-agnostic.

Anchors: Translation-invariant anchors with areas of 32×32 to 512×512 cover multiple scales and aspect ratios at each spatial location.

Focal Loss: A dynamically scaled cross-entropy loss adding a modulating factor down-weighting easy examples so the model focuses on hard ones during training.

Training: End-to-end alternating optimization trains the classification subnet with focal loss and Box regression subnet with smooth L1 loss.

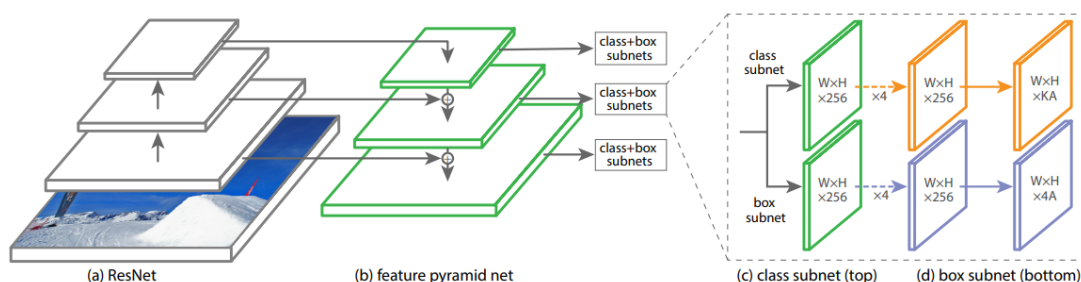


Figure 3.18: RetinaNet network's architecture [54]

In the padel domain, RetinaNet offers several advantages that make it well-suited for detecting players and balls in videos. Its dense sampling of object locations provides robustness to occlusions and rotations of players and balls, which is important given the dynamic nature of padel gameplay. The focal loss used in RetinaNet helps mitigate the background class imbalance inherent in padel videos, where the background tends to dominate frames as well as the player class dominating the ball class. The anchor design can handle the various scales and deformations of bounding boxes around players and balls in different poses and positions, and its dense sampling of object locations makes it robust to occlusions and rotated/deformed objects. Since it is a unified architecture, it optimizes all the model components together rather than separately. Additionally, with an efficient ResNet-50 [56] backbone, which was also used as a comparison method in section ??[58], it can operate at 5fps while maintaining high precision [54]. This speed is theoretically viable for real-time tasks. With these advantages taken together, RetinaNet shows potential as a candidate one-stage model for experimentation in a robust and efficient detection of players and balls system.

3.6 Multi-Object-Tracking

Building upon the foundational concepts of object detection discussed in section 2.7, this section delves into the specific multi-object tracking (MOT) models that will be employed in the subsequent methodology and experiments chapters. MOT is a crucial component in the proposed system, as it enables the continuous tracking of multiple players and the ball throughout the padel

match. The primary focus will be on tracking-by-detection algorithms, which leverage the object detection results to associate detections across frames and maintain object identities over time.

The table 3.1 presents a benchmark comparison of several state-of-the-art MOT models, namely BoTSORT, DeepOCSORT, OCSORT, and ByteTrack, on a subset of the MOT17 dataset ⁶. These models have been selected for their strong performance and potential applicability to the padel tracking task. In the following subsections, they will be analyzed in detail, highlighting their key features, architectures, and comparative strengths. OCSort [110] will be spared from a deeper analysis based on the fact that DeepOCSORT [44] is an improvement over it and that ByteTrack [30] already provides a benchmark for a pure motion-based method. BoTSORT, DeepOCSORT, and ByteTrack are explored in sections 3.6.3, 3.6.1 and 3.6.2.

Tracker	HOTA↑	MOTA↑	IDF1↑
BoTSORT	77.8	78.9	88.9
DeepOCSORT	77.4	78.4	89.0
OCSORT	77.4	78.4	89.0
ByteTrack	75.6	74.6	86.0

Table 3.1: Benchmark performed on the 10 first frames of each MOT17 sequence [45]. ⁷

3.6.1 DeepOCSORT

DeepOCSORT [44] is an extension of the SORT (Simple Online and Realtime Tracking) [110] algorithm that incorporates appearance information through a deep association metric. The model uses a combination of motion and appearance cues to associate detections across frames, building upon the pure motion-based method OC-SORT [110].

The motion model in DeepOCSORT is based on a Kalman filter [112] with a constant velocity assumption, similar to OC-SORT. However, the key innovation lies in the integration of an appearance model that utilizes a deep neural network to extract discriminative features from the detected objects. This allows DeepOCSORT to leverage the visual appearance of objects to improve tracking robustness, especially in scenarios with occlusions or missed detections. For instance, when a tracked object is occluded or not detected in a frame, the deep appearance features (ReID module displayed in figure 3.20) enable DeepOCSORT to still maintain the object’s identity by matching it with previously tracked instances. This is particularly relevant in the context of padel, where players may occlude each other or the ball during fast-paced rallies.

Delving deeper, DeepOCSORT introduces three specific components to enhance the tracking performance:

- Camera Motion Compensation (CMC): This module accounts for camera motion by estimating and correcting the Kalman filter predictions, improving localization accuracy in moving camera scenarios. In this work, the position of the camera is fixed

⁶<https://paperswithcode.com/dataset/mot17>

⁷Available on Boxmot Github : https://github.com/mikel-brostrom/yolo_tracking/

- **Dynamic Appearance (DA):** Instead of using a fixed weight for the appearance model update, DA adjusts the weight based on the detector's confidence. This allows the model to selectively incorporate appearance information only when the detections are reliable.
- **Adaptive Weighting (AW):** AW modulates the contribution of the appearance similarity based on its discriminative power. When the appearance features are highly discriminative for a specific detection-track pair, AW increases the weight of the appearance term in the association cost matrix. This could be particularly beneficial in padel. As players often wear similar clothing or uniforms, the appearance features may not always be highly discriminative.

All these components are illustrated in the figure 3.20.

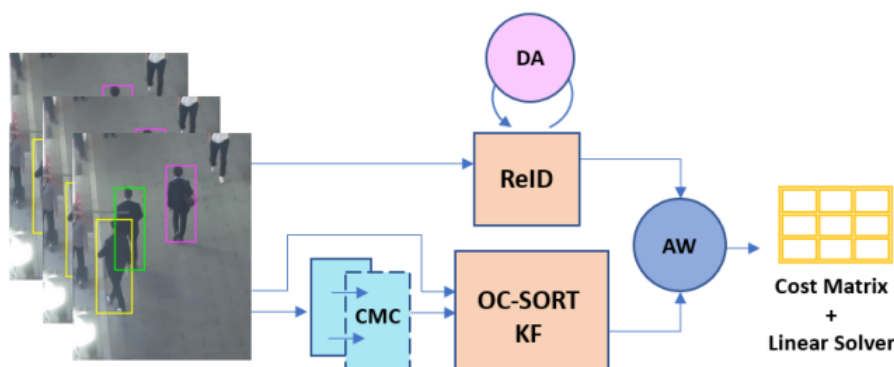


Figure 3.19: Deep OC-SORT Visual Representation [44]

In the benchmark comparison present in table 3.1, DeepOCSORT [44] achieves superior performance compared to OC-SORT [110] across all metrics. It obtains a higher HOTA score of 77.4, indicating better overall tracking accuracy. The improvement in IDF1 from 89.0 to 89.0 suggests enhanced identity preservation, which can be attributed to the integration of the deep appearance model.

The combination of robust tracking, adaptive appearance weighting, camera motion compensation, and strong benchmark performance makes DeepOCSORT a compelling candidate for multi-object tracking in padel. It is potential to handle occlusions, similar player appearances, and moving camera scenarios aligns well with the challenges encountered in padel video analysis.

3.6.2 ByteTrack

ByteTrack [30] is a simple yet effective multi-object tracking algorithm that introduces a novel tracking approach called "tracking by associating every detection box." Unlike traditional tracking-by-detection methods that only consider high-confidence detections, ByteTrack [30] associates every detection box, including low-confidence ones, with the tracked objects.

The algorithm starts by matching high-confidence detections with existing tracks using a combination of motion and appearance similarity. The motion similarity is computed using the Kalman

filter [112] predictions, while the appearance similarity is based on the cosine distance between the appearance features of the detections and the tracks. For the remaining unmatched tracks, it performs a second association step with the low-confidence detections. This allows the algorithm to recover missed or occluded objects that may have been discarded by the object detector due to low confidence scores. However, it is important to note that ByteTrack’s performance relies heavily on the quality of the object detector. If the detector produces a high number of false positive detections, it may struggle to filter out the noise and maintain accurate tracking. Therefore, careful tuning of the object detector and the confidence thresholds is crucial for optimal performance [30].

ByteTrack’s ability to handle low-confidence detections is particularly valuable in scenarios where object detection is challenging, such as in crowded scenes or when objects are partially occluded. By considering every detection box, it can potentially improve the tracking robustness and reduce fragmentation in the object trajectories.

In the context of padel, ByteTrack’s approach could be beneficial for tracking players, especially in situations where players are partially occluded or the detector struggles to produce high-confidence detections consistently. The second association step with low-confidence detections could help maintain player trajectories even during challenging moments in the game. On the other hand, it does not contain any module that extracts appearance features such as DeepOC-Sort and BoTSORT. This is also depicted by the benchmark comparison in table 3.1, where all its evaluation metrics perform worse than the other algorithms, most probably penalized by the identity switches and difficulty maintaining consistent identity labels across all frames. Therefore, it is already clear it may face challenges in maintaining consistent player identities. Additionally, while its ability to handle low-confidence detections could be advantageous for player tracking, it may require additional techniques or modifications to effectively track the fast-moving and frequently occluded padel ball. Nevertheless, as already stated, it is a potential candidate for benchmarking due to its simpler pure motion-based method and being classified as one of the best ones on that scope, along with its efficiency [63].

3.6.3 BoTSORT

BoTSORT [63] is a SOTA MOT algorithm that builds upon the SORT (Simple Online and Real-time Tracking) algorithm and incorporates appearance information through a Re-ID (re-identification) module. The primary goal is to improve the robustness of the tracking process by leveraging both motion and appearance cues, along with camera motion compensation.

One of the key features is its modified Kalman filter [112] state vector, which directly estimates the width and height of the bounding boxes instead of the aspect ratio. This modification leads to more accurate bounding box estimations and improved tracking performance. Additionally, BoTSORT [63] introduces a novel IoU-ReID fusion mechanism that adaptively combines the motion and appearance similarities based on their discriminative power.

All these components are illustrated in the figure 3.20.

The IoU-ReID fusion works by first computing the IoU (Intersection over Union) and appearance similarity matrices between the tracked objects and the detections. The appearance similarity

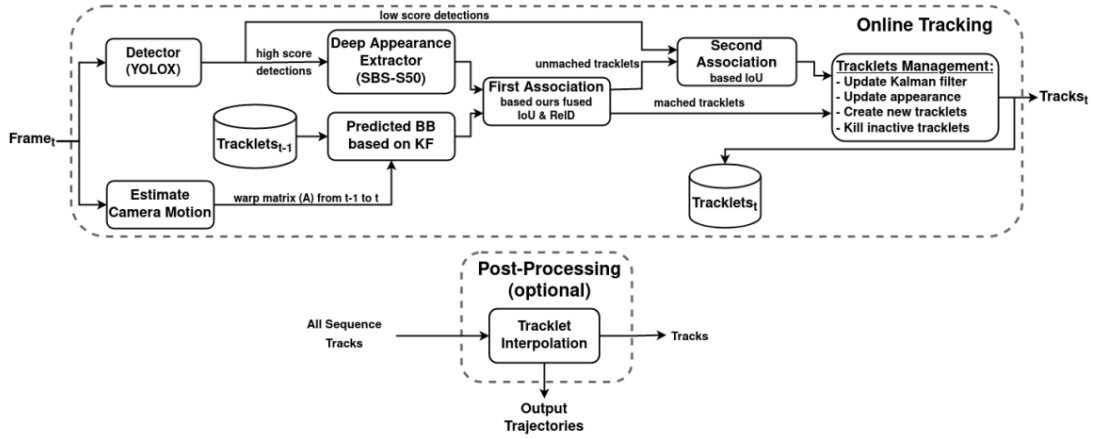


Figure 3.20: BotSORT Visual Schema [63]

is calculated using the cosine distance between the appearance features extracted by a deep neural network. BoTSORT then applies a gating mechanism based on the IoU scores to filter out unlikely associations. Finally, the IoU and appearance similarities are fused using an adaptive weighting scheme that considers the discriminative power of each similarity cue. This adaptive fusion allows BoTSORT to effectively balance the contribution of motion and appearance information based on their reliability in each specific tracking scenario.

Another notable component of BoTSORT is the camera motion compensation module, which aims to mitigate the challenges posed by camera motion in the tracking process. By estimating and correcting for the camera motion, BoTSORT can more accurately predict the positions of the tracked objects, reducing the impact of sudden camera movements on the tracking performance.

In the benchmark comparison present in table 3.1, BoTSORT achieves state-of-the-art performance, outperforming other tracking methods in terms of HOTA, MOTA, and IDF1 scores. The high HOTA score indicates a strong balance between detection accuracy, association accuracy, and localization accuracy, while the MOTA and IDF1 scores demonstrate BoTSORT's effectiveness in both overall tracking performance and identity preservation.

In the context of padel tracking, several theoretical characteristics reinforce BotSORT as a viable candidate for experimentation :

- **Robust tracking:** BoTSORT's fusion of motion and appearance information, along with its adaptive weighting scheme, enables robust tracking even in challenging scenarios with occlusions or appearance changes.
- **Camera motion compensation:** In the event of an angle change in the camera, BoTSORT's camera motion compensation module could be particularly beneficial for maintaining accurate player trajectories.

- Identity preservation: The incorporation of the Re-ID module and the adaptive IoU-ReID fusion mechanism in BoTSORT could help preserve player identities throughout the match, especially in situations where players have similar appearances or undergo occlusions.
- State-of-the-art performance: BoTSORT’s strong performance on benchmark datasets suggests its potential.

However, similar to the other tracking algorithms, BoTSORT [63] may require adaptations and fine-tuning to address the specific challenges of padel, such as the fast-moving ball and the unique court layout. Additionally, the effectiveness of the appearance-based Re-ID module in BoTSORT may depend on the quality and diversity of the appearance features extracted from the padel players. In an attempt to fill this gap, better solutions for this specific task were explored in the following section.

3.7 Re-Identification

While BoTSORT and DeepOCSort presented the possibility of integrating appearance features into the tracking pipeline [44], the choice of the specific ReID model can have a significant impact on the overall performance. Recent advancements have introduced novel architectures and training strategies that push the boundaries of appearance-based object association.

In the following sections, there will be a deeper analysis of ReID models, exploring SOTA approaches that can potentially enhance the appearance-based association in tracking algorithms such as BoTSORT and DeepOCSort. By leveraging ReID techniques, the aim is to further improve the identity preservation and robustness of multi-object tracking in complex scenarios, allowing the tracker to handle multiple players with similar appearances and complex situations where, for example, players may leave and reenter the court or the single camera’s field of view.

Model	Param ($\times 10^6$)	GFLOPs	msmt17 $\rightarrow$ market1501	msmt17 $\rightarrow$ dukemtmc-reid
resnet50	23.5	2.7	46.3 (22.8)	52.3 (32.1)
osnet_x1_0	2.2	0.98	66.6 (37.5)	66.0 (45.3)
osnet_x0_75	1.3	0.57	63.6 (35.5)	65.3 (44.5)
osnet_x0_5	0.6	0.27	64.3 (34.9)	65.2 (43.3)
osnet_x0_25	0.2	0.08	59.9 (31.0)	61.5 (39.6)
osnet_ibn_x1_0	2.2	0.98	66.5 (37.2)	67.4 (45.6)
osnet_aia_x1_0	2.2	0.98	70.1 (43.3)	71.1 (52.7)

Table 3.2: Comparison of Different Cross-Domain REID Models [81] ⁸

The table 3.2 refers to performance metrics for models that have been trained on the MSMT17 dataset, which is the largest one, and then tested on the Market-1501 and DukeMTMC-reID

⁸Kaiyang Zhou Model Zoo https://kaiyangzhou.github.io/deep-person-reid/MODEL_ZOO.html#msmt17-combineall-true-market1501-dukemtmc-reid

datasets, respectively. For reference, Market1501⁹ contains images taken from six different cameras and has a large number of individuals captured in a campus environment. MSMT17¹⁰ is a large-scale multi-scene multi-time person re-ID dataset that contains a comprehensive set of images covering a wide variety of lighting conditions, weather conditions, and seasons. DukeMTMC-reID¹¹ also contains images from multiple cameras, focusing on persons captured in an academic campus. These are three widely used benchmark datasets in the person re-identification (ReID) research community. They provide a standardized evaluation framework for comparing the performance of different ReID algorithms and architectures.

3.7.1 Omni-Scale Network (OSNet)

The Omni-Scale Network (OSNet) [82] is a SOTA architecture specifically designed for person ReID, which builds upon the concept described in section 2.8.3. It addresses the challenge of learning discriminative features at multiple scales to capture both global and local information. This introduces a novel concept called "omni-scale features," which are a combination of multi-scale features that encapsulate a wide range of receptive field sizes.

The key component of OSNet is the omni-scale residual block (OSR), which consists of multiple convolutional streams with different receptive fields. Each stream is responsible for capturing features at a specific scale, determined by a linearly increasing scale factor. The multi-scale features generated by these streams are then dynamically fused using a channel-wise gating mechanism, allowing the network to adaptively combine features from different scales based on the input data.

OSNet also incorporates a lightweight design by utilizing depthwise separable convolutions, which significantly reduces the number of parameters and computational complexity compared to standard convolutional layers. This efficient design makes OSNet well-suited for deployment in resource-constrained scenarios.

Extensive experiments conducted on various person ReID datasets, such as Market1501, CUHK03, and DukeMTMC-reID, have demonstrated the effectiveness of OSNet. It achieves state-of-the-art performance while maintaining a compact model size, making it an attractive choice for real-world applications.

3.7.2 OSNet Variants

To further enhance the performance and/or generalization capability of OSNet, several variants have been proposed.

- OSNet-x0_25: Along with OSNet-x0_5 and OSNet-x0_75 are slimmed-down versions of the standard OSNet, utilizing a width multiplier of 0.25, 0.25, 0.75, respectively, to reduce

⁹<https://paperswithcode.com/dataset/market-1501>

¹⁰<https://paperswithcode.com/dataset/msmt17>

¹¹<https://www.kaggle.com/datasets/whurobin/dukemtmcrid>

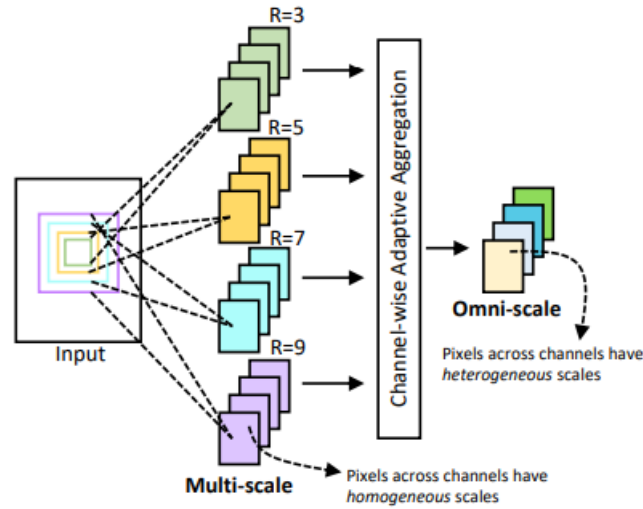


Figure 3.21: OSNet building blocks [82]

the number of channels in each convolutional layer. The compact designs aim to mitigate overfitting and improve efficiency while maintaining a high level of performance.

- OSNet_AIN [81]: OSNet_AIN(Adaptive Instance Normalization) incorporates Instance Normalization (IN) layers into the OSNet architecture. The placement of IN layers is determined through a differentiable architecture search algorithm, ensuring an optimal configuration for domain adaptation. By aligning feature distributions across different domains, OSNet-AIN promotes cross-domain generalization.
- OSNet_IBN [81]: Inspired by the IBN-Net architecture [26], OSNet-IBN (Instance Batch Normalization) introduces IN layers in the shallow layers of OSNet. By aligning the feature distributions of the early stages through batches, OSNet-IBN aims to capture appearance variations which are primarily present in these early stages of the network and enhance its ability to handle cross-domain discrepancies.

These OSNet variants showcase different approaches to improving the generalization and adaptability of the network. In the context of player tracking, they can help maintain the identities of players across different camera views and occlusions by learning the discriminative features at multiple scales and adapting to domain variations throughout the matches.

In respect to results, as seen in the table 3.2, the baseline OSNet_x1_0, with 2.2 million parameters and 0.98 GFLOPs, obtains a mAP of 66.6% on Market1501 and 66.0% on DukeMTMC-reID, which are good numbers taking into account the variety these datasets possess. This demonstrates the effectiveness of the architecture in learning discriminative features for ReID, even in cross-domain scenarios. The reduced variants, which have fewer parameters and GFLOPs compared to the baseline model, also exhibit strong performance. For instance, the OSNet_x0_5, with only 0.6 million parameters and 0.27 GFLOPs, achieves an mAP of 64.3% on Market1501 and 65.2%

on DukeMTMC-reID. This highlights the efficiency of the OSNet, as it can maintain high performance while significantly reducing the model size and computational complexity [81]. The OSNet_ain_x1_0, which employs adaptive instance normalization (AIN) and uses Cosine distance instead of Euclidean distance, achieves the best performance among all the variants. It obtains an mAP of 70.1% on Market1501 and 71.1% on DukeMTMC-reID, outperforming the baseline and the other IBN variant. This might indicate the advantage of AIN in aligning feature distributions between different domains and the benefit of it using cosine distance for similarity measurement.

3.7.3 ResNet-50

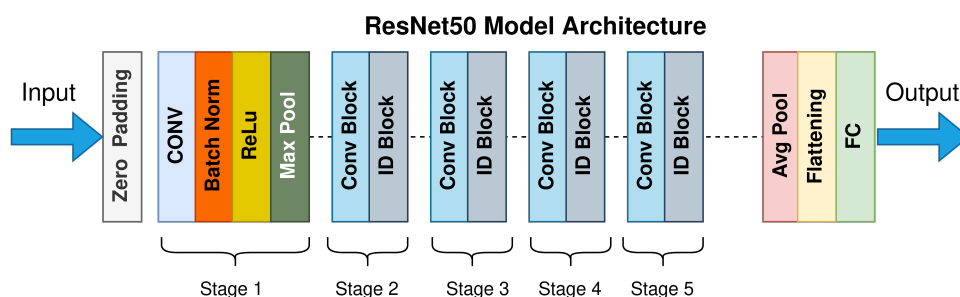


Figure 3.22: ResNet-50 architecture [25]

The ResNet50 [56] is a widely-used deep CNN architecture that has shown remarkable performance in various computer vision tasks, person re-identification (ReID) being no exception. The architecture of ResNet50 is based on the concept of residual learning, which introduces skip connections that allow the network to learn residual functions instead of directly learning the underlying mapping. It consists of 50 layers, including convolutional layers, batch normalization layers, and ReLU activation functions. The network is divided into four main stages, each containing multiple residual blocks. Each residual block consists of three convolutional layers with 1x1, 3x3, and 1x1 filters, respectively. The 1x1 convolutions are used to reduce and then restore the dimensionality, while the 3x3 convolution is the main feature extractor. The skip connections in each residual block allow the gradients to flow directly through the network, mitigating the vanishing gradient problem and enabling the training of deeper networks [56].

One of the key advantages of the ResNet50 architecture for ReID is its ability to learn rich, high-level features that can effectively capture the discriminative information necessary for distinguishing individuals. The deep structure allows it to learn a hierarchical representation of the input images, with the earlier layers capturing low-level features such as edges and textures and the later layers learning more complex and abstract features that are specific to the ReID task. Another benefit is its pre-training capability [56]. It can be pre-trained on large-scale datasets such as ImageNet¹², which contains millions of images across a wide range of object categories. This pre-training allows the network to learn general image features that can be transferred to the ReID task through fine-tuning. Fine-tuning a pre-trained ResNet50 on a smaller ReID dataset can

¹²<https://paperswithcode.com/dataset/imagenet>

significantly improve performance compared to training the network from scratch, as it leverages the knowledge learned from the larger dataset.

When it comes to actual results, as shown in the table 3.2, the original OSNet and its variants can achieve superior performance while being much more lightweight and efficient. Nevertheless, due to its larger size and higher computational complexity compared to the OSNet variants and the fact that it has been extensively studied and optimized, the ResNet50 architecture stands as an interesting candidate for experimentation, as it allows for evaluating the trade-off between performance and efficiency, while still being a viable option for ReID because of its theoretically proven strong feature learning capabilities and pre-training potential, as previously described.

3.8 Conclusions

This comprehensive literature review has shed light on the current state and potential future directions of CV applications in sports, with a special focus on padel. A clear hierarchy was demonstrated in section 3.1 when it comes to the development of these applications, with motion analysis systems, such as player and player and ball tracking, forming the foundational layer upon which more advanced functionalities are built, such as using temporal information, extracting or detecting events, and game summarization. As also revealed by the review presented in section 3.1.3, padel appears to be in the early stages of this hierarchy, emphasizing a research gap that needs to be filled. Another significant challenge in advancing computer vision applications in padel is the scarcity of comprehensive and annotated datasets [42]. Nevertheless, efforts such as PadelVic [42] and others presented in section 3.2 have begun to address this issue by providing a valuable resource for researchers and developers. Nevertheless, this gap also needs to be addressed.

Moreover, the analysis of motion capture and analysis methods (see section 3.4) has highlighted the complementary roles of multi-camera and single-camera systems. While multi-camera setups address tracking and detection challenges primarily from a hardware perspective, single-camera systems tackle these issues through advanced software techniques, offering a more flexible and accessible approach, such as for amateur communities in sports. Regarding software-focused solutions, throughout the review of section 3.1, it has also become apparent DL approaches often outperform traditional ML methods in sports computer vision tasks. The ability of deep learning models to learn hierarchical features and capture complex patterns in visual data has proven invaluable in this domain.

Having thoroughly examined the literature and identified promising candidates for experimentation, including state-of-the-art models for object detection, multi-object tracking, and person re-identification, on the sections 3.5, 3.6 and 3.7, the stage is now set to proceed with the methodology.

Finally, the insights gained from this extensive review will guide the selection and implementation of appropriate techniques to approach a robust system for the identification and tracking of players and the ball in padel, while having reinforced the need for such a system in the current research scenario along with the necessity of datasets for that purpose.

Chapter 4

Methodology

In this chapter, the necessary steps to implement a Robust Detection and Tracking system for padel will be outlined. After conducting a comprehensive review of existing literature in Chapter 3, a variety of techniques and state-of-the-art algorithms that are best suited for the project have been identified. The subsequent stage involves the initial data preparation and annotation, as well as the emergent need to accelerate that process and how it was solved. This chapter will also delve into the logic behind defining the classes. Furthermore, we will discuss the transformation of the dataset, encompassing changes in the videos used due to limitations in the models when it comes to generalization capabilities. Afterward, the selection of appropriate models and algorithms, as well as defining evaluation metrics for each different task, will be addressed. Finally, the tools and computational resources utilized in the implementation will be presented.

4.1 Data Collection

The process of video collection was done by MOG Technologies. The recordings were done by high-quality 3840 x 2160 resolution cameras at 60 frames per second (fps), resulting in a large number of frames and pixels that need to be analyzed but consequently offering greater potential for high-quality tracking.

All the video recordings are of amateur padel matches taking place on the same court, with the same set of players, and under consistent lighting and environmental conditions. This consistency across the dataset might seem beneficial at first but was later seen as a disadvantage. As a consequence, the dataset suffered several modifications and a resulting expansion in diversity, as addressed in the Dataset Expansion, section 4.4

The positions of the cameras here are fixed and static, in 2 different angles, as seen in Figures 4.1a and 4.1b.

More specifically, a total of 12 video recordings were captured, six from a back camera angle and six from a side camera angle, accumulating over 86 minutes of gameplay footage. When broken down into individual frames, this translates to a substantial volume of data for analysis,



Figure 4.1: Mog Dataset Camera Angles

with 2746 frames from the back angle (example in figure 4.1a) and 3003 frames from the side-camera angle (example in figure 4.1b). Unfortunately, since the side camera angle did not provide a symmetric view over both sides of the court and its field-of-view was too narrow, it was deemed as not useful for this use case. Therefore, only the back camera angle was selected for annotation (see figure 4.1a), as further explained in section 4.2.3.

It is also relevant to mention that the high resolution of 4K (3840 x 2160 pixels) and the frame rate of 60 fps ensure that the video captures detailed visual information, including rapid player movements and the fast-paced trajectory of the ball. However, the 4K resolution is too demanding in terms of computational resources and storage requirements for real-time object detection. Therefore, this challenge will be addressed in the next section.

4.2 Initial Data Preparation and Annotation

It is noteworthy that on the first iteration of the data preparation process, all its intrinsic tasks were performed on custom-written scripts. The initial approach was an attempt to achieve independence from typically paid online platforms with limited free plans, having control over the entire data processing pipeline and ensuring the confidentiality of the data. It also provided more flexibility in handling formats and fostered an understanding regarding the practical implementation of such tasks, clarified in the following sections. However, with that approach, critical limitations surged.

4.2.1 Resolution Adjustment

To reduce computational requirements while retaining sufficient detail, the videos, which were initially in 4k, were resized to a lower resolution of 1280x720 pixels (720p). For further means of comparison, a resolution of 854x480 pixels (480p) was also considered in the process of resizing.

4.2.2 Frame Extraction and Standardization

After resizing the videos, their respective frames were extracted. A custom script was developed to extract frames in a manually set interval in an attempt to generate representative sub-datasets from each video. To ensure consistent input to the computer vision models, the extracted and resized

frames underwent a standardization process, which involved normalizing color values, adjusting aspect ratios, and padding the frames to a uniform size.

4.2.3 Manual Annotation

During the first iteration, considering the labor-intensive nature of manual image annotation, a total of only 220 frames were annotated. These frames were extracted from all the videos captured at the back camera angle, adhering to a fixed interval of 10 seconds between each frame. The annotation was carried out using the VGG Image Annotator tool.

4.2.4 Challenges and Limitations

During the initial data preparation phase, several challenges and limitations were encountered. One significant challenge was the time-consuming nature of the manual annotation process. Annotating a large number of frames accurately required significant human effort and attention to detail. Another challenge was the need to debug and update the manual scripts several times, which also required time and effort. To mitigate these challenges, the next section will discuss the transition to an automated data preparation and annotation workflow using another platform in order to accelerate the initial pipeline.

4.3 Automated Data Preparation and Annotation Workflow

Addressing the previously mentioned challenges and limitations, there was a need to streamline the data preparation and annotation process. Therefore, the project transitioned to using Roboflow¹, an automated platform that provided several advantages, including automated frame extraction, preprocessing, and standardization. It replaced all the previously described manual scripts. However, it is important to mention that this brought dependence on a new external platform with inherent risks, such as sudden unavailability, data security, vendor lock-in, and free plan limitations. The existing first dataset, described in section 4.2.3, including the manual annotations, was easily migrated.

4.3.1 Semi-Automated Annotation for Object Detection

Besides automating the Data Preparation Pipeline, Roboflow also allows a semi-automated annotation process. It does so by allowing the import of a pre-trained YOLOv8 model on one previously generated dataset from the workflow. Therefore, it requires at least one totally manually annotated dataset at first. Then, with that initial model, it automatically generated annotations on new images, identifying and labeling objects but allowing for manual correction afterward, ensuring the labels' accuracy and reliability. The improved annotations, once validated, were fed back into the model, creating a cycle that incrementally enhanced the model's predictive accuracy. This tool

¹<https://roboflow.com/>

also allows for some degree of customization by giving control over the confidence thresholds for predictions and the classes to be annotated.

The semi-automated process maximized the efficiency of the annotation process, being empirically estimated as tenfold faster than manual annotation alone.

4.4 Dataset Expansion

Despite labeling over two thousand images and applying a thorough data augmentation process, the resulting model's generalization capabilities were found to be worse than the initial model trained on only two hundred images, showing clear signs of overfit, which are deeper explored in chapter 5. Therefore, the need for a complete reformation of the dataset to improve the model's performance and robustness emerged.

To address this issue, a new dataset was carefully handpicked. Initially, six diverse padel matches were selected, each with varying court conditions, to ensure a representative sample of the expected variations in real-world scenarios. The annotation aid model was then trained on these six videos, and its performance was evaluated. Subsequently, with the assistance of that model, six additional videos were chosen for training, along with five separate videos for validation. It was crucial to ensure that the validation set contained frames from completely different videos than those used in the training set to assess the model's ability to generalize to unseen data.

The primary objective of the video selection process was to incorporate a wide range of conditions, such as different camera angles (always positioned at the back of the court), varying lighting conditions (including both day and night matches), and a mix of indoor and outdoor courts. Moreover, a higher proportion of frames with active balls were purposefully selected to mitigate class imbalance, which could potentially hinder the model's performance.

Another important consideration was the inclusion of videos with both 30fps and 60fps frame rates, as the difference in motion blur between these configurations could impact the model's ability to accurately detect and track objects.

The entire dataset selection process was designed to improve the model's robustness and generalization capabilities. In addition to the careful curation of diverse videos, several data augmentation techniques were applied to further expand the dataset and introduce additional variations, thereby, once again, reducing overfitting and enhancing the model's ability to handle a wide range of real-world scenarios.

All the relevant experiments related to this dataset reformation are presented in Chapter 5.

4.5 Multi-Object-Tracking Validation Dataset

In addition to the semi-automated annotation process using Roboflow, a separate manual annotation task was performed specifically for creating a dataset in order to evaluate the Multiple Object Tracking (MOT) algorithms and Re-Identification (REID) methods. This involved annotating a complete video of a match using the Computer Vision Annotation Tool (CVAT) since Roboflow

did not allow annotating for tracking tasks. The annotation process in CVAT focused on accurately labeling and tracking individual players throughout the video. Each player was assigned a unique identifier, and their bounding boxes were drawn in each frame where they were visible. The video utilized for the study was chosen based on its challenging tracking conditions, characterized by scenarios where players frequently moved in close proximity to one another, reentered the court after exiting for several seconds, and switched sides. These particular conditions were deemed ideal for rigorously testing the tracking algorithm's robustness and accuracy in complex dynamic environments. Regarding the ball classes, identifiers were not applied during the tracking annotations, only the bounding boxes and corresponding classifications.

4.6 Classes Definition

Defining classes for detection and tracking purposes is a crucial step in ensuring that the model can accurately identify and differentiate between different objects or entities within the scene. By properly defining these classes, the model can focus its attention on the specific objects of interest and effectively handle variations in their appearance and context. This ultimately leads to a more robust and adaptable system. In the following subsections, the different class configurations considered in the experiments will be discussed. The results and findings behind each configuration will be thoroughly explored in the chapter 5.

4.6.1 Six Classes

In an initial approach to distinguish between players, six classes were defined, four of which corresponded to the four players on the court. Each player class was assigned to a specific zone of the court, as illustrated in Figure 4.2. Player1 was assigned to the bottom-right corner, Player2 to the bottom-left, Player3 to the upper-right, and Player4 to the upper-left.

The remaining two classes were dedicated to the ball. The first class, InactiveBall, represented balls that were lying on the court for at least 1 second without being used. The second class, ActiveBall, corresponded to the ball currently being used in the frame being annotated or a ball that had been lying still for less than five seconds after an action occurred. This differentiation was introduced to compare the model's performance when detecting a moving ball versus a static one. Additionally, this distinction provides semantic information, as there can only be one instance of an ActiveBall at any given time. Moreover, visual cues such as motion blur (see Figure 4.3c) can be leveraged to identify the ActiveBall. In the context of tracking, this differentiation might also help in optimizing computational resources by excluding the InactiveBalls from being further tracked.

4.6.2 Three Classes

Upon further consideration, it was concluded that a three-class configuration would be more suitable. By associating player identities with specific court divisions, any instances where players

switch sides would result in a loss of their established identities. Therefore, the player identification task was delegated to the tracking phase. In this new configuration, all players were grouped under a single class named *Player*, without any distinction between individual players, while the ball classes were kept the same. This became the standard configuration for all subsequent experiments.

4.6.3 Two Classes

To serve as a basis for comparison, a two-class configuration was also experimented with, which removed the distinction between the *Active* and *Inactive* balls. This configuration allowed for the assessment of the impact of differentiating between ball states on the model's performance, providing more instances of the ball class as a whole.

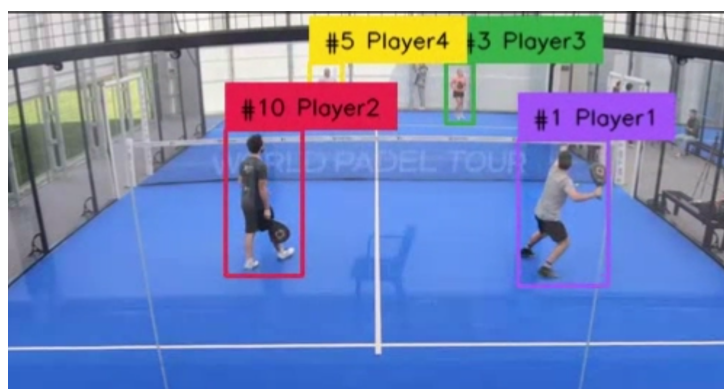
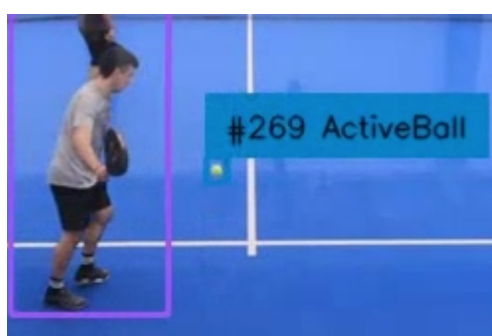
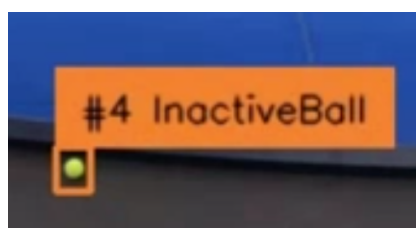


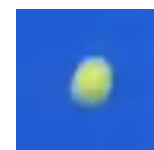
Figure 4.2: Example of a frame with the players numbered according to their positions



(a) ActiveBall Example



(b) InactiveBall Example



(c) Example of a frame with motion blur on an ActiveBall

Figure 4.3: Ball examples

4.7 Model Selection and Evaluation Metrics

4.7.1 Object Detection

Following the extensive review of the literature, detailed in Section 3.5, three distinct models were identified as suitable.

4.7.1.1 YOLOv8

The latest version of the YOLO object detection algorithm was selected as the primary object detection model for this work. As previously stated, it is known for its real-time performance and high accuracy in object detection tasks [9], building upon the success of its predecessors and introducing several improvements, such as a new backbone architecture and advanced data augmentation techniques [113].

Among the YOLOv8 model variants, YOLOv8nano(n) and YOLOv8medium(m) were chosen based on their performance-to-size ratio. YOLOv8n is the smallest and fastest model in the YOLOv8 family, making it suitable for real-time inference on resource-constrained devices. Despite its small size, YOLOv8n maintains a good balance between speed and accuracy. On the other hand, YOLOv8m is a medium-sized model that offers relevant higher accuracy while still being computationally efficient. The choice of these two models allows for a comparison of performance and trade-offs.

The decision to exclude the other versions was based on the observation that the larger models in the YOLOv8 family, such as YOLOv8l and YOLOv8x, did not provide a significant improvement in performance that justified their increased computational requirements. More explicitly, this was depicted by the figure 4.4, in which it is evident that moving from 'm' to 'l' requires a significant increase in parameters for a relatively small gain in mAP. On the other hand, moving from 'n' to 'm' requires fewer additional parameters per unit increase in mAP.

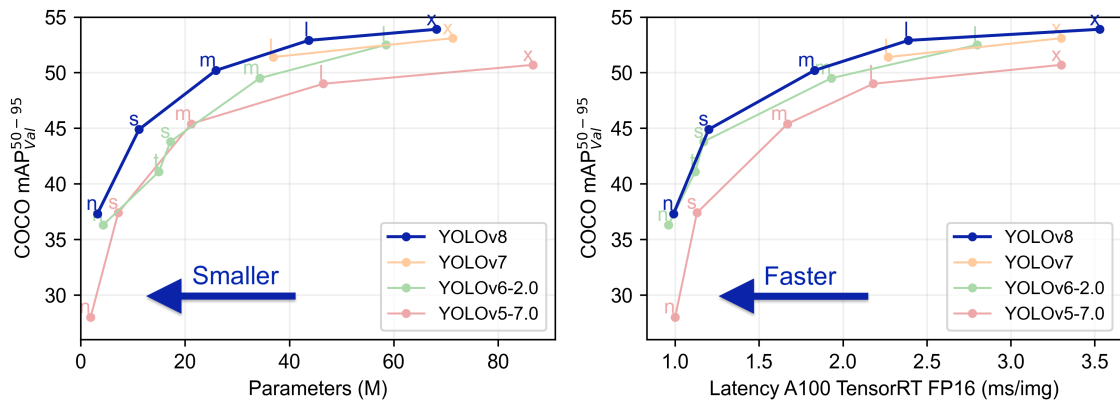


Figure 4.4: Comparing YOLOv8 different versions [113]

4.7.1.2 Faster R-CNN

Faster R-CNN was chosen as a representative two-stage object detection algorithm. Two-stage detectors first generate region proposals and then classify and refine them in a second stage, as already stated in the previous chapter 3. Faster R-CNN has been widely adopted and has demonstrated strong performance in various object detection benchmarks [103]. The inclusion of this model serves as a reliable reference point for comparing the performance of YOLOv8 against a well-established two-stage detector. Faster R-CNN's larger model size and higher computational requirements are offset by its proven accuracy and robustness in complex scenarios. By evaluating it alongside YOLO, insights can be gained into the trade-offs between speed and accuracy.

Regarding the particular version of Faster R-CNN chosen, as shown in figure 4.5, the (x101-FPN,3x) achieves the best Average Precision among all versions while being in the middle with respect to the number of parameters. Therefore, it strikes the best balance between its size and box AP. All these versions are implementations from the Detectron2 [105] library.

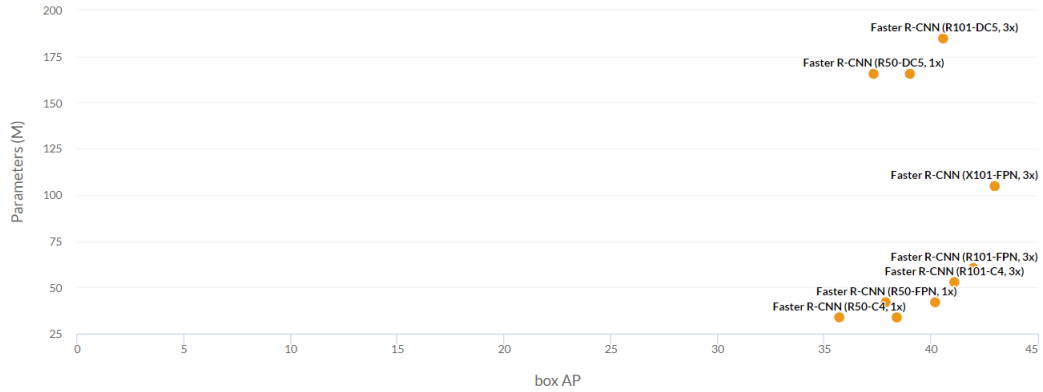


Figure 4.5: Comparing Faster R-CNN different versions [105]

4.7.1.3 RetinaNet

RetinaNet [54] was selected as an additional one-stage object detection algorithm for comparison purposes. It introduced the focal loss function, which helps address the class imbalance problem commonly encountered in object detection tasks. Class imbalance is particularly relevant in this scenario, where the "active ball" class is significantly less frequent compared to the "player" class. Therefore, its ability to handle class imbalances makes it an interesting choice for evaluating its performance in detecting the active ball accurately while maintaining high precision for player detection and assessing the effectiveness of its focal loss function in mitigating the impact of class imbalance on the overall detection performance.

In respect to the chosen version, observing figure 4.6, considering the trade-off between box AP (Average Precision), as the evaluation metric and the number of parameters (as the size variable), the (R101, 3x) seems the better choice. It has a higher box AP compared to the (R50, 1x), suggesting better accuracy. Although it comes with more parameters, which could mean more

computational resources are required, the increase in accuracy is justified. All these versions are implementations from the Detectron2 [105] library.

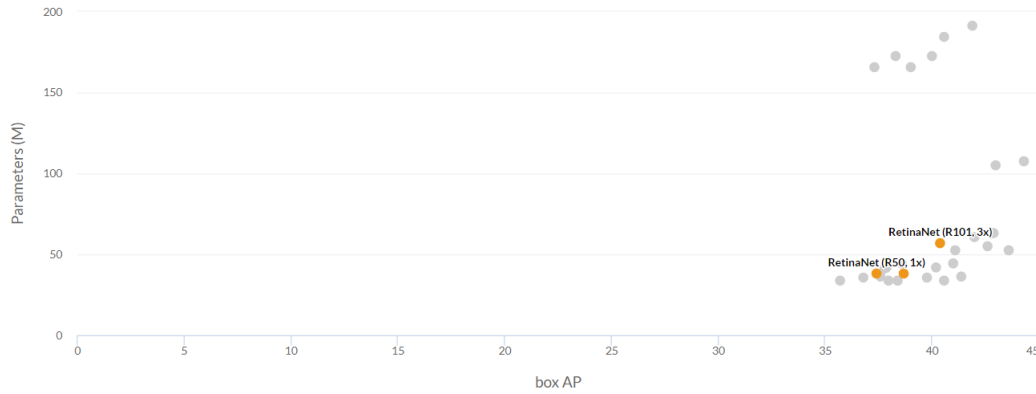


Figure 4.6: Comparing RetinaNet different versions [105]

4.7.1.4 Evaluation Metrics

To comprehensively compare the performance of the YOLOv8, Faster R-CNN, and RetinaNet object detection models, various metrics were utilized. It is important to note that since the YOLO models come from the Ultralytics library [113], while Faster R-CNN and RetinaNet are part of the Detectron2 library [105], the available metrics for each differ.

For the **YOLOv8** models, for all classes at once and separately, four metrics were considered: the mAP@.5 (mean Average Precision with an IoU threshold of 0.5), mAP@.5:.95, Precision and Recall, these will be the main metrics. When found relevant, the confusion matrix was considered, which includes false negatives (FN) and false positives (FP) for each class combination, with the background included for situations where the detected Bounding Box did not have any object inside it. This will provide insights into the model's performance in correctly identifying and misidentifying objects of each class with respect to each other.

To compare **RetinaNet and Faster R-CNN** (Detectron2), the focus will be on mAP@.5 and mAP@.5:.95, considering all classes, as well as each class separately with the AP metric. Additionally, mean Average Precision and Average Recall were reported considering small, medium, and large objects separately.

When comparing **all the models across libraries**, the main metrics used will be mAP@.5 and mAP@.5:.95 considering all classes and the AP@.5:.95 for each class separately. These metrics are commonly available in both libraries and enable the ranking of the models in terms of overall performance.

Delving deeper into the meaning and reasoning of each metric, IoU is a measure of overlap between the predicted bounding box and the ground truth bounding box. It is defined as the area of overlap divided by the area of the union between the predicted and ground truth bounding boxes, providing localization performance. Precision quantifies the accuracy of the True Positive predictions over all the predictions for that class, while recall measures the model's ability to detect

all instances from a respective class. The AP for a single class is computed as the area under the precision-recall curve for that class. The mAP is the overall mean Average Precision, providing a single value that summarizes the model's ability to accurately detect and localize objects. The mAP@.5 is the mean Average Precision at a specific Intersection over Union (IoU) threshold of 0.5. It evaluates the model's performance considering a predicted bounding box as a true positive if its IoU with the ground truth box is greater than or equal to 0.5. The mAP@.5:.95 is the mean Average Precision averaged across a range of IoU thresholds from 0.5 to 0.95, reflecting not only the model's ability to detect objects accurately (high IoU) but also its ability to detect objects with varying degrees of overlap with the ground truth, giving both insights into classification and localization performance, thus being the most important metric in this case.

4.7.2 Multi-Object-Tracking and Re-Identification

For MOT, three prominent algorithms were considered: **ByteTrack** [30], **BoTSORT** [63], and **DeepOCSORT** [44]. The selection of these algorithms was guided by the findings from the literature review, presented in section 3.6 (refer to table 3.1). In sum, the ByteTrack was selected for its performance-to-cost ratio, being one of the most efficient trackers while still being able to compete in total performance numbers. Furthermore, of the three, it is the only tracker that does not incorporate any visual features into its algorithm, emphasizing efficiency and simplicity and representing the pure tracking-by-detection, motion-based object association. On the other hand, BoTSORT and DeepOCSORT were selected for their particularly high performance and the leveraging of visual features for object association, not being solely based on motion cues. Additionally, unlike ByteTrack, they allow the integration of external specialized re-identification models, which further explore the extraction of distinctive, person-specific features. This should improve performance in maintaining the individual identities of the players during matches, which is one of the main objectives.

In order to investigate the impact of re-identification on tracking performance, a set of re-identification models was also selected. On the same note as the MOT algorithms, the selection was guided by the previous findings from the literature review 3, particularly in section 3.7 (refer to table 3.2) including **ResNet-50** [56] and several OSNet variants [?] [81], such as **OSNet x1.0**, **OSNet x0.5**, **OSNet.AIN x1.0**, and **OSNet.IBN x1.0**. In sum, the choice of these models was informed by their strong performance on person re-identification benchmarks, as evidenced in Table 3.2, demonstrating the cross-domain generalization capabilities of these models. In this context, the ability to generalize across different domains is crucial, especially considering the wide range of variations in player appearances. These variations can result from diverse lighting conditions, varying camera angles, and differences in clothing. Given these factors, fine-tuning re-identification models specifically for the padel domain would present significant challenges. The necessity to manually label another large volume of images for such fine-tuning would not be feasible.

After selecting the MOT algorithms and Re-identification models, almost all their possible combinations were considered for a comprehensive comparison and analysis:

- Bytetrack
- BoTSORT without external ReID model; with Resnet-50; osnet_x1_0; osnet_x0_5, osnet_ain_1_0 and osnet_ibn_1_0.
- DeepOCSORT with Resnet-50; osnet_x1_0, and osnet_ibn_1_0

All algorithms and models were implemented using the Boxmot [45] library.

4.7.2.1 Evaluation Metrics

In assessing the performance of multiple object tracking systems, a evaluation involves various metrics, each providing insights into different aspects of tracking accuracy and effectiveness. They can be categorized into three primary groups for a structured overview: overall tracking performance, identity preservation, and additional detailed metrics. In contrast to the Object Detection, the library used for validation was the same across all algorithms and ReID models. Therefore, the following metrics were applied to every combination.

Overall Tracking Performance

- **MOTA (Multiple Object Tracking Accuracy):** Focuses on the tracking accuracy of objects, excluding the aspect of identity preservation. It penalizes false positives, missed detections, and identity switches, reflecting overall tracking performance including the preservation of identities.

Identity Preservation

- **IDF1:** Centers on the accuracy of maintaining correct identity labels over time. It calculates the ratio of correctly identified detections against the average number of ground-truth and computed detections, highlighting the system's effectiveness in consistent identity tracking.
- **IDP (Identification Precision):** Measures the precision in assigning correct identities, defined as the ratio of correctly identified detections to the total number of detections.
- **IDR (Identification Recall):** Assesses the recall in capturing correct identities, calculated as the ratio of correctly identified detections to the total number of ground-truth detections.

Additional Detailed Metrics

- **IDs (ID Switches)** address the errors in tracking in regards to wrongful identity changes and track interruptions.
- **MOTP (Multiple Object Tracking Precision):** Focuses on the precision of detected object positions, emphasizing spatial accuracy, independent of object identities.

It is important to note that the identity-related metrics, such as IDF1, IDP, IDR, and IDs, were only applied to the player classes since they were assigned unique identifiers. Nevertheless, the tracking performance of the ball classes is reflected in the overall tracking performance metrics, such as, Recall, Precision, MOTP, FP, and FN.

4.8 Tools and Technologies

Python² was the primary programming language used in this project due to its extensive ecosystem of deep learning and computer vision libraries. OpenCV³ was used for basic image processing and manipulation tasks. PyTorch⁴ was the main DL framework used, although on a more abstract level, below three other higher-level libraries which will be presented next.

Detectron2⁵[105], Facebook AI Research’s object detection, which provides a simple configuration for models such as RetinaNet and Faster-RCNN. Ultralytics⁶[113] library, which provides a user-friendly interface for training and deploying the different versions of YOLOv8 models. Finally, the Boxmot GitHub repository⁷, which delivers a framework with the implementations of various multi-object tracking algorithms that were explored for the object tracking and re-identification tasks. For the re-identification task, it provided integration with a Model Zoo⁸ which offered a collection of pre-trained models for person re-identification, that can be fine-tuned or used as feature extractors for identifying and tracking objects across frames.

For GPU acceleration in those three higher-level libraries, enabling faster training and inference times, PyTorch CUDA⁹ was utilized.

Regarding working environments, for the initial data preparation process 4.2, Google Colab¹⁰, which is a cloud-based Jupyter notebook environment that provides free access to GPUs and TPUs, was used. In that same initial stage, VGG Image Annotator (VIA)¹¹ was used, a standalone manual annotation software developed by the Visual Geometry Group at the University of Oxford, completely free and open source. It supports various annotation types, such as rectangles, polygons, and points, making it suitable for initial image annotation tasks. VIA has a simple interface and the ability to export annotations in csv,json or COCO format.

For the annotations regarding Multi-Object-Tracking and Re-Identification, CVAT¹² (Computer Vision Annotation Tool) was the chosen platform, consisting in an open-source web-based tool for annotating videos and images. It supports various annotation types, including bounding boxes, polygons, and tracking and has semi-automated options for the tracking task.

²<https://www.python.org/>

³<https://opencv.org/>

⁴<https://pytorch.org/>

⁵<https://github.com/facebookresearch/detectron2>

⁶<https://www.ultralytics.com/>

⁷https://github.com/mikel-brostrom/yolo_tracking/

⁸https://kaiyangzhou.github.io/deep-person-reid/MODEL_ZOO

⁹<https://pytorch.org/docs/stable/cuda.html>

¹⁰<https://colab.research.google.com/>

¹¹<https://www.robots.ox.ac.uk/~vgg/software/via/>

¹²<https://www.cvat.ai/>

4.9 Computational Resources

4.9.1 Cloud

First, Google Colab’s free tier was used for all the described initial data preparation tasks, from the resolution adjustment to the integration of the manual annotations. Colab provides access to GPUs and TPUs, allowing for faster execution of computationally intensive tasks. However, the free tier has limitations on runtime and resource availability. Those limitations range from its internal storage being non-existent and depending on an external Google Drive to the constraints of the runtime environment, which has a maximum session duration, typically around twelve hours. After this time, the session will automatically disconnect, and all unsaved work will be lost. Additionally, there are limitations on how much GPU or TPU usage a user can have within a single session. All these limitations make it suitable only for small-scale projects and short-term projects.

4.9.2 Local

After several difficulties with Google Colab due to the mentioned limitations, for further experiments and model development, a local laptop with an NVIDIA GeForce RTX 3060 Mobile GPU was used. Nevertheless, while attempting to use larger models, the training process for each experiment could reach dozens of hours.

4.9.3 FEUP Server

In order to address the hardware limitations, a server equipped with an NVIDIA GeForce RTX 2080 Ti GPU, 32GB RAM, and an Intel Core i7-9700K CPU was used. The RTX 2080 Ti, being a much more powerful GPU with 11GB of VRAM, enabled the training of larger and more complex models, achieving, at times, a 30-fold reduction in training time over the previous setup. The 32GB of RAM ensures that the system can handle large datasets and multiple processes simultaneously. The i7-9700K CPU, with its 8 cores and 8 threads, provides additional computational power for data processing and model inference.

Chapter 5

Experiments and Results

Given the significance of object detection in the tracking-by-detection task, a significant portion of the experimental efforts was dedicated to refining and optimizing this component. The main focus was to develop a reliable and efficient object detection model that could effectively detect players and the ball in padel match videos, even under challenging conditions such as varying player distances, rapid ball movements, and diverse lighting conditions.

An iterative approach was employed to progressively improve the object detection model. That process was primarily driven by the refinement, reformulation, and expansion of the dataset, previously described in section 4.4, used as its input. In each iteration, the dataset was analyzed and modified based on the insights gained from the previous iteration's experiments. This involved identifying and addressing specific issues, such as lack of generalization capabilities, class imbalances, and lack of diversity in the training samples.

After reaching the final dataset, other aspects of object detection were explored, including different model architectures (YOLO, Faster R-CNN, RetinaNet), class configurations (6 classes, 3 classes, 2 classes), and data augmentation techniques. The impact of factors such as image resolution and model size on the detection accuracy and inference speed was also investigated.

As the best model configuration for object detection was selected, the foundation for tracking-by-detection was set and the experiments then focused on the tracking algorithms and external re-identification models of the system, following a comparison between the different configurations presented in the section 4.7.2 of the methodology.

5.1 Object Detection

The following tables provide an overview of the object detection models and datasets used in this work. Table 5.1 describes the various versions of the dataset, including their composition, annotation process, and purpose. Table 5.2 presents the different object detection models trained and evaluated, along with their architectures, input resolutions, and the dataset on which they were trained. These tables serve as a reference for understanding the experimental setup and the characteristics of the data and models used in the object detection task.

Table 5.1: Dataset Versions' Description

Version	Description
MOG220Images	First dataset iteration, composed of 220 manually annotated frames from the first process of Data Collection. The frames were extracted from all the back camera angle videos, adhering to a fixed interval of 10 seconds between each frame, further described in section 5.1.1.
MOG2000Images	Second dataset iteration, composed of 2000 semi-automated manually annotated frames from the first process of Data Collection (see section 5.1.1). It has the same logic as the previous dataset, but now, it expands its size, resorting to an annotation aid model to accelerate the process. The frames extracted adhered to a fixed interval of 1 second between each other.
Expanded_6Videos	Joining the 220 frames from the first version of MOG220Images, 6 additional videos were selected from external sources. These videos were handpicked to include the most diversity possible, as described in the section 4.4, namely in lighting conditions, camera angles, court appearance, and frame rate. A total of 626 frames were annotated.
Expanded_12Videos	Using an aid annotation model trained on Expanded_6Videos, to further enhance the purpose of the previous dataset and to reach a larger one, 6 more videos were selected and annotated for training, and another 5 for validation, following the same diversity objectives. In total, 1528 frames were annotated for training and 259 for validation, following the proportion of 85-15.
Expanded_12Videos_480p	Resized version in 864x480 of the final dataset for experimentation purposes.
Expanded_12Videos_2classes	Modified version of the final dataset considering only the Player class and joining the ActiveBall and InactiveBall into a Ball class for experimentation purposes. All remaining characteristics were not modified.

Table 5.2: All Object Detection Models

Name	Architecture	Dataset_Version	Epochs	InputResolution
220_images	YOLOv8n	MOG220Images	100	1280x720
2000_images	YOLOv8n	MOG200Images	100	1280x720
6_videos	YOLOv8n	Expanded_6Videos	100	1280x720
12_videos	YOLOv8n	Expanded_12Videos	100	1280x720
fasterrcnn_12_videos	Faster R-CNN x101-FPN,3x	Expanded_12Videos	NA	1280x720
retinanet_12_videos	RetinaNet R101, 3x	Expanded_12Videos	NA	1280x720
medium_12_videos	YOLOv8m	Expanded_12Videos	100	1280x720
12_videos_480p	YOLOv8n	Expanded_12Videos_480p	100	864x480
medium_12_videos_480p	YOLOv8m	Expanded_12Videos_480p	100	864x480
12_videos_2classes	YOLOv8n	Expanded_12Videos	100	1280x720
medium_12_videos_2classes	YOLOv8m	Expanded_12Videos_2classes	100	1280x720

Note: All models were trained with the same hyperparameters for equivalent comparison.

5.1.1 Dataset Development and Refinement

In this section, the focus is on clarifying the main three iterations the dataset with 3 classes went through and their respective experiments. The first iteration was solely based on the videos collected from MOG, described in , while the other two encompassed external videos from completely different matches.

5.1.1.1 Initial MOG Datasets(220 and 2000 images)

At first, as previously described in the section 5.1.1 of the methodology, the dataset was composed of only 220 images from the same padel match. A YOLOv8n model was then trained on those 220 images in order to generate the first annotation aid model, which was then imported inside Roboflow for a process of semi-automated annotation (see section 4.3.1). That process allowed for a further expansion to 2000 annotated images.

Since all the annotated frames at this point corresponded to videos with very similar visual conditions (see table 5.1), there was a need to enhance generalization to unseen data; the first approach was data augmentation. Therefore, both datasets encompassed a data augmentation process (see table 5.3), doubling the sizes of the datasets.

Augmentation Type	Parameter
Outputs per training example	2
Crop	0% Minimum Zoom, 18% Maximum Zoom
Brightness	Between -27% and +27%
Exposure	Between -10% and +10%
Rotation	Between -7° and +7°

Table 5.3: Data Augmentation Parameters

The first augmentation was a variation in zoom: there was no minimum zoom applied, ensuring that the original scale of the players and the ball is preserved in some images, while a maximum zoom of 18% was applied to others, allowing for closer views which could help the model learn

to identify objects at different scales. The second augmentation was regarding the brightness of the images being adjusted within a range of -27% to +27%. This ensures the inclusion of various lighting conditions, which is particularly useful for simulating different times of day, outdoors, or artificial lighting. The exposure of the images is also modified between -10% and +10%. This mimics changes in the overall exposure that the camera might capture during the matches, aiding the model to be resilient to changes in visibility and shadowing that can affect the detection of the objects of interest. Lastly, the images undergo a rotation augmentation, where each image is rotated by a degree between -7° and $+7^\circ$. This introduces a variety of orientations for the objects, which contributes to including players and the ball even when they are tilted due to camera angle or motion.

In order to evaluate the effects of expanding the dataset from 220 to 2000 annotated images, a validation set was created. However, once again, only including frames from the initial MOG collection. The results are shown in table 5.4.

Table 5.4: Initial models validated on MOG match (higher is better)

Model	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95		
			ActiveBall	InactiveBall	Player
220_images	0.448	0.214	0.0576	0.176	0.408
2000_images	0.86	0.574	0.3	0.606	0.817

It was clear that the performance of the model trained on the expanded dataset (2000 images) significantly improved when validated on a set with the same conditions as the training data. The model achieved an mAP@0.5 of 0.86 and an mAP@0.5:0.95 of 0.574, compared to 0.448 and 0.214, respectively, for the model trained on the initial 220 images.

Looking at the class-specific average precision (AP) scores, the model trained on 2000 images demonstrated substantial improvements across all classes. The AP@0.5:0.95 for the ActiveBall class increased from 0.0576 to 0.3, while the InactiveBall class saw an increase from 0.176 to 0.606. Similarly, the Player class AP@0.5:0.95 improved from 0.408 to 0.817. These results indicate that increasing the dataset size with images from the same data collection leads to better performance when evaluated on a validation set with similar conditions.

However, to better assess the generalization capability of the models, they were validated on an external match, as shown in table 5.5.

Table 5.5: Initial models validated on external match (higher is better)

Model	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95		
			ActiveBall	InactiveBall	Player
220_images	0.32	0.167	0.0156	0.32	0.476
2000_images	0.302	0.136	0.0397	0.00107	0.367

Surprisingly, the performance of the model trained on 2000 images deteriorated compared to the model trained on 220 images. The mAP@0.5 decreased from 0.32 to 0.302, and the

mAP@0.5:0.95 dropped from 0.167 to 0.136. Analyzing the class-specific AP scores on the external match, the model trained on 2000 images showed a notable decrease in performance for the InactiveBall class, with the AP@0.5:0.95 falling from 0.32 to 0.00107. The Player class also experienced a decline, with the AP@0.5:0.95 reducing from 0.476 to 0.367. The ActiveBall class, however, was the only one that benefitted from the expansion, with a minor improvement from 0.0156 to 0.0397. The other classes most likely had distinct visual appearances or contexts in the external match compared to the MOG dataset, causing the model to struggle in accurately detecting and localizing them.

These results can be attributed to the different characteristics of the two datasets. While the larger dataset (2000 images) provided more training samples, it was derived from the same match as the smaller dataset (220 images) by extracting frames at a higher frequency (every second compared to every 10 seconds). This higher frame extraction rate likely introduced a higher degree of redundancy and similarity between adjacent frames, leading to a narrower range of variations in terms of player positions, ball movements, and overall scene dynamics. Consequently, the model trained on the larger dataset became overly specialized to the specific characteristics and nuances present in the MOG data collection, limiting its generalization capability to the external match with different visual conditions. On the other hand, the smaller dataset, despite its limited size, captured a broader range of variations by sampling frames at a lower frequency. This diversity, although limited, encouraged the model to learn more generalizable representations of the objects and their visual characteristics rather than memorizing specific patterns or nuances of the MOG dataset. Simply increasing the number of samples without introducing sufficient variations can lead to overfitting and hinder the model's ability to generalize to new and unseen conditions. This highlights the importance of considering not only the size of the dataset but also the diversity and representativeness of the samples during dataset expansion. The next sections follow that approach.

5.1.1.2 Enhanced Dataset (6 different matches)

To address the limitations of the previous datasets and improve the model's generalization capability, an enhanced dataset was created by incorporating frames from 6 different padel matches in addition to the 220 frames from the initial data collection. The selection of these matches and the specific frames within them was guided by several key considerations.

Firstly, one of the primary goals was to annotate more frames featuring active balls. The previous models showed a clear difficulty regarding the ActiveBall class. That was likely due to the dynamic nature of the ball but also due to a class imbalance (see table 5.6), with significantly fewer instances of active balls compared to inactive balls and players. To mitigate this issue, a conscious effort was made to select frames where the ball was in motion and actively being played. This aimed to provide the model with more diverse examples of active ball scenarios, enabling it to better learn and detect this crucial class. By carefully handpicking frames with visible active balls, motion blur was also taken into account.

Table 5.6: Class balance of dataset MOG220Images

Class	Instances
Player	691
InactiveBall	418
ActiveBall	98

Note: ActiveBall class is underrepresented.

Secondly, the enhanced dataset aimed to improve the model’s ability to detect players at various distances. In real-world scenarios, players are often positioned at different depths within the court. Additionally, the different camera’s aperture angles also accentuate those different depths. Therefore, frames showcasing distant players were intentionally selected to expose the model to a wider range of player scales and positions.

Furthermore, the inclusion of videos with individual padel and outdoor settings was a key consideration. The previous dataset consisted of an indoor match with multiple players. By incorporating frames from individual padel sessions and outdoor courts, the enhanced dataset introduced visual variations, such as different court surfaces, lighting conditions, and player arrangements.

Lastly, the enhanced dataset aimed to cover a broader range of environmental conditions. While the previous datasets primarily focused on indoor matches, it is known that padel is also played outdoors in amateur matches. To bridge this gap, efforts were made to include frames from daytime and nighttime outdoor matches, exposing the model to different lighting conditions and shadow patterns. All these considerations aimed to bring diversity in order to enhance the model’s robustness and adaptability.

When it comes to actual results after this expansion, table 5.7 shows the performance of the models on the MOG match. While the 6_videos model achieved lower mAP scores compared to the 2000_images model, it still outperformed the 220_images model. The class-specific AP scores also highlighted improvements, with the ActiveBall class increasing from 0.0576 to 0.153, the InactiveBall class improving from 0.176 to 0.275, and the Player class rising from 0.408 to 0.567.

Table 5.7: First 3 models validated on MOG match (higher is better)

Model	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95		
			ActiveBall	InactiveBall	Player
220_images	0.448	0.214	0.0576	0.176	0.408
2000_images	0.86	0.574	0.3	0.606	0.817
6_videos	0.75	0.332	0.153	0.275	0.567

In the following table 5.8, it is visible the significant improvement in performance on an external match when using the model trained on the enhanced dataset (6_videos). The mAP@0.5 increased from 0.302 (2000_images model) to 0.762 (6_videos model), and the mAP@0.5:0.95 improved from 0.136 to 0.409. The class-specific average precision (AP) scores also showed notable improvements across all classes, with the ActiveBall class experiencing a substantial increase

from 0.0397 to 0.247, the InactiveBall class improving from 0.00107 to 0.238, and the Player class rising from 0.367 to 0.742.

Model	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95		
			ActiveBall	InactiveBall	Player
220_images	0.32	0.167	0.0156	0.32	0.476
2000_images	0.302	0.136	0.0397	0.00107	0.367
6_videos	0.762	0.409	0.247	0.238	0.742

Table 5.8: First 3 models validated on external match (higher is better)

The results from both tables elucidate the effectiveness of the enhanced dataset in improving the model’s generalization ability while still maintaining good performance on the original MOG match. Therefore, it was proven that the careful selection of diverse matches and frames, guided by the considerations of active ball annotation, distant player detection, individual padel, outdoor settings, and daytime and nighttime conditions, contributed to the model’s improved performance.

Nevertheless, it is important to acknowledge that there is still room for further improvement since the Expanded_6Videos dataset only contains a total of 626 frames (see table 5.1) and a validation set was not yet created.

5.1.1.3 Final Dataset (12 different matches)

This final iteration follows the same considerations defined for the Expanded_6Videos dataset, discussed in the previous section 5.1.1.2. Once again, the annotation aid model trained on the previous iteration (Expanded_6Videos dataset) was used to annotate 6 more videos for training and 5 others for validation (see table 5.1). The class imbalance between the ball types was also addressed (see table 5.9). One important aspect was to ensure the validation videos were completely different from the training ones. This separation allowed for a more reliable assessment of the model’s generalization capability. By evaluating the model on unseen videos with distinct visual characteristics, it was possible to gauge how well the model could perform on entirely new padel matches.

Furthermore, data augmentation techniques (see table 5.3) were again employed.

To assess the performance of all models from the current iterations, they were validated on the final 12_videos dataset, which includes the 5 new validation videos. The results are presented in table 5.10.

Table 5.9: Class distribution of dataset Expanded_12Videos

Class	Instances
Player	6161
InactiveBall	1367
ActiveBall	1256

As expected, the final model, trained on the Expanded_12Videos dataset, achieved the highest performance across all metrics. It obtained a mAP@0.5 of 0.8 and a mAP@0.5:0.95 of 0.438,

Table 5.10: All models from the current iterations validated on the 5 different videos from Expanded_12Videos dataset (higher is better)

Model	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95		
			ActiveBall	InactiveBall	Player
220_images	0.457	0.227	0.0549	0.189	0.437
2000_images	0.292	0.13	0.0386	0.00912	0.261
6_videos	0.546	0.289	0.176	0.0905	0.602
12_videos	0.8	0.438	0.276	0.379	0.659

outperforming all previous models. The class-specific average precision scores also showed significant improvements, with the ActiveBall class reaching 0.276, the InactiveBall class achieving 0.379, and the Player class obtaining 0.659.

Comparing the 12_videos model to the 6_videos model, there is a notable increase in performance, which reinforced the considerations for the expansion defined in the previous iteration. The mAP@0.5 improved from 0.546 to 0.8, and the mAP@0.5:0.95 increased from 0.289 to 0.438. The ActiveBall class AP increased from 0.176 to 0.276, the InactiveBall class AP improved from 0.0905 to 0.379, and the Player class AP rose from 0.602 to 0.659.

The 220_images and 2000_images models, which were trained on the initial MOG dataset, showed worse performance compared to the models trained on the expanded datasets. The 220_images model achieved an mAP@0.5 of 0.457 and an mAP@0.5:0.95 of 0.227, while the 2000_images model obtained an mAP@0.5 of 0.292 and an mAP@0.5:0.95 of 0.13. This highlights the importance of dataset diversity and the inclusion of challenging scenarios in improving the model's generalization capability.

In sum, the results demonstrate that the iterative process of expanding the dataset, guided by the considerations of balancing active ball annotation against the remaining classes, varying player depths, individual padel settings, outdoor settings, court visual appearance and varied lighting conditions, has led to significant improvements in the model's performance and generalization ability. The 12_videos model, trained on the most diverse and comprehensive dataset, has shown the best results when validated on the final Expanded_12Videos dataset, which includes handpicked new and unseen validation videos.

5.1.2 Model Architecture Comparison and Selection

After the iterative process of creating a comprehensive dataset, the next step was to experiment with different model architectures to determine the most suitable one for the robust detection task at hand. As discussed in section 4.7.1 of the methodology chapter, three architectures were selected for evaluation: YOLOv8, Faster R-CNN, and RetinaNet.

The experiments were conducted using the final dataset, which underwent the iterative expansion process described in the previous sections, enabling a fair comparison between the different architectures. The results are shown in the following table 5.11.

Table 5.11: All selected architectures trained and validated on the 5 different videos from Expanded_12Videos dataset

Model	InferenceTime	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95		
				ActiveBall	InactiveBall	Player
12_videos	2.6ms	0.8	0.438	0.276	0.379	0.659
fasterrcnn_12_videos	98ms	0.784	0.394	0.237	0.294	0.65
retinanet_12_videos	54ms	0.625	0.327	0.13	0.21	0.65
medium_12_videos	9.4ms	0.792	0.438	0.242	0.405	0.669

Analyzing the performance of each model in depth, considering their specific characteristics, **YOLOv8n**(12_videos), being the lightest model, achieves an impressive inference time of just 2.6ms per image. Despite its small size and efficiency, it also demonstrates strong performance with a mAP@0.5 of 0.8 and a mAP@0.5:0.95 of 0.438. Looking at the class-specific average precision (AP) scores, compared with the other models, it performs well across all classes, achieving an AP of 0.276 for the ActiveBall class, 0.379 for the InactiveBall class, and 0.659 for the Player class.

Faster R-CNN(fasterrcnn_12_videos), being the largest model among the compared architectures, has an inference time of 94ms per image, which is 36x of the YOLO's. It achieved an AP of 0.237 for the ActiveBall class, 0.294 for the InactiveBall class, and 0.65 for the Player class. While Faster R-CNN's performance is slightly lower than YOLOv8n in terms of mAP and class-specific AP scores, its larger size, and higher computational requirements may contribute to its ability to capture more complex features and handle challenging scenarios.

RetinaNet(retinanet_12_videos), with its key characteristic being a focal loss function designed to address the class imbalance, shows an inference time of 54ms per image. Additionally, its performance is substantially lower, with a mAP@0.5 of 0.625 and a mAP@0.5:0.95 of 0.327. Unexpectedly, it achieved an AP of 0.13 for the ActiveBall class, which is the lowest among all models. This suggests that despite the focal loss function, which was the main reason for its choosing, RetinaNet struggles to accurately detect and localize the ActiveBall, which was an imbalanced class in the dataset, even after the enhancement. For the InactiveBall and Player classes, AP scores of 0.21 and 0.65 were obtained, respectively.

YOLOv8m(medium_12_videos) is the medium-sized variant of YOLOv8 and theoretically aims to provide a better balance between inference speed and performance, as discussed in the methodology. However, the results show otherwise; it achieved a mAP@0.5 of 0.792 and a mAP@0.5:0.95 of 0.438, which are slightly lower than YOLOv8n. Looking at the class-specific AP scores, it barely performed slightly better than the smallest variant for the InactiveBall and Player classes, with AP scores of 0.405 and 0.669, respectively, and for the ActiveBall, it even performed worse. Additionally, with an inference time of 9.4ms per image, although significantly faster than Faster R-CNN and RetinaNet, it is still five times slower than YOLOv8n.

A general remark to be considered is the fact that the performance of localization and detection of the players is considerably high across all models, proving the effectiveness of the final dataset in providing robustness for their detection. On the other hand, the ActiveBall continues to be the

most challenging part of the object detection task, justified by both its dynamic characteristics and the inherent difficulty of balancing its annotated instances. Many times, even the human eye fails to detect and localize it in individual frames.

In sum, considering the maximum performance and lowest inference times, YOLOv8n emerged as the definitive choice for the object detection task, being the chosen model for the foundation module of the tracking-by-detection. Nevertheless, some other relevant experiments regarding other configurations, such as resolution varying along the model size and class configurations, will be presented in the following sections.

5.1.3 Resolution and Model Size Experiments

To further investigate the impact of input resolution and model size on the performance of the padel tracking system, additional experiments were conducted using the YOLOv8 nano and medium models. The models were trained and validated on two different resolutions: 720p and 480p. The results are presented in tables 5.14 and 5.13

Table 5.12: All resolutions and model size combination validated on the 5 videos from Expanded_12Videos dataset (720p)

Model	InferenceTime	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95		
				ActiveBall	InactiveBall	Player
12_videos	2.6ms	0.8	0.438	0.276	0.379	0.659
medium_12_videos	9.3ms	0.792	0.438	0.242	0.405	0.669
12_videos_480p	3.6ms	0.596	0.281	0.134	0.143	0.564
medium_12_videos_480p	16ms	0.661	0.31	0.129	0.247	0.545

Table 5.13: All resolutions and model size combination validated on the 5 videos from Expanded_12Videos_480p dataset

Model	InferenceTime	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95		
				ActiveBall	InactiveBall	Player
12_videos	2.2	0.715	0.364	0.173	0.263	0.655
medium_12_videos	9.6	0.675	0.362	0.134	0.277	0.673
12_videos_480p	2.1	0.688	0.354	0.155	0.264	0.642
medium_12_videos_480p	9.8	0.701	0.367	0.171	0.281	0.648

The models trained on higher resolution data (720p) consistently outperform their counterparts trained on lower resolution data (480p) when validated on the higher resolution dataset. Validating on the lower resolution, there is almost an equivalence in performance. This suggests that, as expected, training on higher-resolution data provides more detailed information and improves the model's ability to generalize across different resolutions.

The performance gap between the 720p and 480p models is more pronounced when validated on the higher resolution dataset (720p) compared to the lower resolution dataset (480p). This indicates that while training on higher resolution data is beneficial, the impact of resolution mismatch between training and validation data is less severe when the input resolution is lower.

Interestingly, the inference time remained largely unaffected by changes in the input resolution. The main differences in inference time were observed between the YOLOv8 nano and medium models, with the nano models being faster, as also seen in previous sections. This suggests that the choice of input resolution can be made based on performance requirements without impacting the inference speed.

In conclusion, training on higher resolution data proves highly advantageous, as it enables the model to capture more detailed information and generalize better across different resolutions. Since the performance in inference time remains unaffected, real-time applicability remains possible, despite any increase in resolution. The only downside that should be considered is the consequential increase in training times. With respect to model sizes, the differences in performance are minimal. Despite not being shown on the tables, the training times for the different model sizes changed in direct proportions.

5.1.4 Class Configuration Experiments

As discussed in the section 4.6 of the methodology, three class configurations were considered. However, the 6 class configuration was discarded for the reasons described in section 4.6.1 of the methodology. Therefore, only the 2-class configuration remains to be explored. The main purpose of this configuration is to investigate the influence of separating or combining the "ActiveBall" and "InactiveBall" classes on the model's performance, i.e., if joining them into a single class would result in any improvement. Theoretically, this configuration should help mitigate the class imbalance, increasing the training data for the ball itself while reducing class confusion between the "ActiveBall" and "InactiveBall". As such, the model can focus on learning the general characteristics of the ball, such as shape, size, and color, without the need to distinguish between its active and inactive states. Table 5.14 presents a comparison of the usual metrics of the medium and nano YOLOv8 models trained and validated on 2 classes (combined "Ball" class and "Player" class) and the best-performing model with the 3 classes.

Table 5.14: Comparison of the main models trained and validated on 2 classes and 3 classes

Model	InferenceTime	mAP@0.5	mAP@0.5:0.95	AP@0.5:0.95			
				ActiveBall	InactiveBall	Player	Ball
12_videos	2.6ms	0.8	0.438	0.276	0.379	0.659	NA
12_videos_2classes	2.2ms	0.783	0.461	NA	NA	0.633	0.289
medium_12_videos_2classes	9.7ms	0.842	0.499	NA	NA	0.676	0.322

The results show that the best model with the combined "Ball" class (2 classes) achieved slightly higher mAP@0.5 and mAP@0.5:0.95 scores than the model with 3 classes, which was probably related to the better Player class performance, which is irrelevant in this case. Unfortunately, with respect to the "Ball" class, the improvement over the previous "ActiveBall" was barely existent, which refutes all previously mentioned theoretical advantages. However, to better understand the effects on ball classification, it is essential to analyze the confusion matrices. Table 5.15

presents the confusion matrix for the 12_videos model (3 classes), while Table 5.16 shows the confusion matrix for the best 2 classes model, the medium_12_videos_2classes model (2 classes).

Table 5.15: Normalized Confusion Matrix for the 12_videos classes model

	ActiveBall	InactiveBall	Player	Background
ActiveBall	0.59	0.06	0	0.57
InactiveBall	0.04	0.83	0	0.08
Player	0	0	0.97	0.35
Background	0.36	0.11	0.03	NA

Note: The Columns refer to the True labels and the Rows to the predicted.

Table 5.16: Normalized Confusion Matrix for the medium_12_videos_2classes model

	Ball	Player	Background
Ball	0.66	0	0.55
Player	0	0.97	0.45
Background	0.33	0.03	NA

Note: The Columns refer to the True labels and the Rows to the predicted.

From the confusion matrix of the 12_videos model, it is evident that the main misclassifications occur between the "ActiveBall" and "Background" classes (false positives and false negatives). The model correctly classifies 59% of the "ActiveBall" instances, while 57% are misclassified as "Background". Similarly, for the "InactiveBall" class, 83% of the instances are correctly classified, while 8% are misclassified as "Background". The misclassification between "ActiveBall" and "InactiveBall" is relatively low (6% and 4%, respectively).

In the confusion matrix of the medium_12_videos_2classes model, the combined "Ball" class achieves a correct classification rate of 66%, with 55% of the misclassifications being attributed to the "Background" class.

In conclusion, these class configuration experiments demonstrated that the main challenge for the model is distinguishing between the ball (active or inactive) and the background rather than differentiating between the "ActiveBall" and "InactiveBall" classes. Combining the ball classes into a single "Ball" class does not significantly improve the model's performance in terms of reducing misclassifications.

5.2 Multi-Object-Tracking and Re-Identification

As a first note, given the importance of accurate object detection for tracking performance in the tracking-by-detection paradigm, the most significant portion of the experiments conducted in this study focused on optimizing the object detection model. The first priority was to develop the robust detector which will be applied to each frame of the video to locate and classify the objects of interest during the tracking. Therefore, regarding the MOT and ReID aspects, the focus was on testing how several combinations of tracking algorithms (see section 3.6) performed with and without external specialized ReID models (see table 3.2 and section 3.7) on a padel video from the initial data collection, using the previous 12_videos model as their object detector. The various configurations tested are described in table 5.17. It is important to note that the hyperparameters were kept as consistent as possible across all trackers to ensure a fair comparison.

Table 5.17: Description of MOT-ReID Configurations

Configuration Name	Description
bytetrack	ByteTrack algorithm in its native form
botsort_non_reid	BotSORT algorithm without ReID model
botsort_osnet_ain_x1_0	BotSORT algorithm and OSNet AINx1.0 ReID model
botsort_osnet_ibn_x1_0	BotSORT algorithm and OSNet IBNx1.0 ReID model
botsort_osnet_x0_5	BotSORT algorithm and OSNetx0.5 ReID model
botsort_osnet_x1_0	BotSORT algorithm and OSNetx1.0 ReID model
botsort_resnet50	BotSORT algorithm and ResNet-50 ReID model
deepsort_osnet_ibn_x1_0	DeepOCSORT algorithm and OSNet IBNx1.0 ReID model
deepsort_osnet_x1_0	DeepOCSORT algorithm and OSNetx1.0 ReID model
deepsort_resnet50	DeepOCSORT algorithm and ResNet-50 ReID model

Table 5.18: Comparison of the different MOT-ReID configurations on the Players

Configuration	MOTA	IDF1	Recall	Precision	IDs	MOTP	FP	FN
bytetrack	95.9%	54.6%	98.3%	97.7%	44	0.736	1715	1266
botsort_non_reid	96.0%	53.2%	98.3%	97.7%	45	0.740	1724	1253
botsort_osnet_ain_x1_0	96.1%	56.3%	98.2%	98%	39	0.742	1519	1348
botsort_osnet_ibn_x1_0	96.1%	58.4%	98.2%	98%	42	0.742	1528	1348
botsort_osnet_x0_5	96.1%	58.6%	98.2%	97.9%	48	0.742	1539	1350
botsort_osnet_x1_0	96.1%	59.5%	98.2%	98%	41	0.742	1527	1349
botsort_resnet50	96.1%	60.8%	98%	98%	44	0.742	1530	1349
deepsort_osnet_ibn_x1_0	96.3%	58.5%	97.8%	98.6%	52	0.741	1066	1624
deepsort_osnet_x1_0	96.3%	58.5%	97.8%	98.6%	52	0.741	1066	1624
deepsort_resnet50	96.3%	58.5%	97.8%	98.6%	52	0.741	1066	1624

Table 5.18 presents a comprehensive comparison of various MOT-ReID configurations with respect to the players. These include MOTA (Multiple Object Tracking Accuracy), IDF1 (ID F1 Score), Recall, Precision, IDs (Number of ID Switches), MOTP (Multiple Object Tracking Precision), FP (False Positives), and FN (False Negatives). Unfortunately, it seems the DeepOCSORT

algorithm did not correctly employ the ReID models. Therefore, its results with the different models will be discarded and simply considered as the native algorithm. Since the embeddings hyperparameter was enabled, it still used visual descriptors. Comparing the BotSORT configurations with and without ReID models, it is important to note that the differences in IDF1 scores are relatively small, ranging from 53.2% to 60.8%, as well as the identity switches, which go from 39 to 48. This suggests that, although the ReID models are working and improving identity preservation, opposing DeepOCSORT, their effects are not as expressive as expected. For instance, the Bytetrack configuration, which does not incorporate any direct visual information or external ReID models, achieves a MOTA score of 95.9% , an IDF1 score of 54.6%, and only 44 id switches. These results indicate that Bytetrack, despite its simplicity and lack of appearance information, is still effective in accurately tracking players.

There is not a single best-performing configuration, as they all seem to trade off on different metrics, and their differences are not that significant; therefore, the choice depends on whether minimizing identity switches is more critical or maximizing the overall identity accuracy.

Regarding the pure tracking performance without considering identity maintenance, the Recall, Precision, MOTP, FP, and FN metrics are essentially the same across all configurations within each tracking algorithm, revealing solid numbers, with recall and precision almost reaching 100% and a MOTP higher than 0.730. The consistency in this regard is coherent with the fact that all configurations utilize the same underlying object detector 12_videos model (see table 5.11) as their foundation. The object detector’s performance largely determines the overall tracking accuracy, while the ReID models primarily contribute to identity preservation.

Table 5.19: Comparison of the different MOT-ReID configurations on the ActiveBall

Configuration	Recall	Precision	MOTP	FP	FN
botsort_non_reid	10.8%	27.7%	0.371	1721	5445
botsort_osnet_x1_0	10.8%	27.7%	0.371	1721	5445
bytetrack	10.5%	28%	0.367	1632	5466
deepsort_osnet_x1_0	11.6%	35.1%	0.372	1303	5400

Table 5.20: Comparison of the different MOT-ReID configurations on the InactiveBall

Configuration	Recall	Precision	MOTP	FP	FN
botsort_non_reid	13.8%	31.0%	0.235	8549	23924
bytetrack	13.9%	31.1%	0.236	8542	23912
botsort_osnet_x1_0	14.8%	33.3%	0.242	8234	23651
deepsort_osnet_x1_0	15.7%	38.2%	0.294	7046	23405

On the other hand, moving on to tables 5.19 and 5.20, ball tracking results are quite unsatisfactory. For instance, the recall and precision scores are significantly lower compared to player tracking. For the ActiveBall class, the highest Precision score is only 35.1% (deepsort_osnet_x1_0), while for the InactiveBall class, it is 38.2% (deepsort_osnet_x1_0). Once again, the MOTP scores are relatively consistent across all configurations, which is coherent. On top of that, they are also

expectedly low for both ball classes (ranging from 0.235 to 0.372) compared to the player class (around 0.74), reiterating the challenges in accurately detecting and localizing the ball. Additionally, the high number of False Positives and False Negatives suggests that the tracking algorithms struggle with accurately detecting and maintaining the ball's trajectory when it is stationary or not actively being played. As previously seen in the object detection results, the struggle comes in distinguishing the ball from the background. It's also worth noting that the choice of the ReID model has no impact on the ball tracking performance due to the absence of identity labels. Furthermore, the disparity of FP and FN between the ActiveBall and InactiveBall is justified by the high number of InactiveBall instances in the particular video that was used for validation.

Nevertheless, DeepOCSort achieved better results than the other tracking algorithms. In theory, this might be due to its adaptive weighting scheme dynamically adjusting the importance of motion and appearance cues based on their reliability, relying more on motion cues when the appearance information is less discriminative or reliable[44], which is often the case with the ball due to its small size and blurred appearance when moving rapidly. On the other hand, BotSORT is a more person-focused algorithm as it was initially designed for multi-pedestrian tracking [63], therefore not as adequate for small and fast-moving objects with few visual characteristics.

In conclusion, regarding the players, ReID models, when not fine-tuned for the padel domain, do not seem much of a relevant addition for the player re-identification task, despite their cross-domain pretraining. Nevertheless, their results revealed robustness, indicating the system's viability for motion player tracking and even reasonable re-identification capabilities, despite the difficult scenarios in which they were tested. The ball tracking results, on the other hand, underline the need for additional and possibly alternative methods in order to achieve a more robust tracking system, specifically for its particular challenges. A starting point could be expanding the dataset even further, only for the ball classes.

Chapter 6

Conclusions

In this dissertation, the application of computer vision and deep learning techniques to automatically detect, track, and re-identify players and the ball in padel sports videos has been explored. The primary goal was to develop a robust and efficient system that, with a simple, accessible single-camera setup, could provide a valuable foundation for further complex systems, formerly only available in the professional domain with costly setups. Further complex features can be built on top of this robust motion analysis system, allowing for insights and analytics for players, coaches, and sports enthusiasts, enhancing their experience, which was the main motivation.

The exploration began with a comprehensive literature review, examining the state-of-the-art methods and approaches in computer vision for sports analysis, reaching the realization that at the base of those complex systems always stands a robust motion analysis system, providing accurate player and ball tracking. This review highlighted that, despite its growing popularity, padel lacks the advanced motion analysis systems prevalent in more widely studied sports such as tennis or soccer while presenting a challenging scenario for their design.

To address these challenges, a multi-stage pipeline was proposed that combines object detection, multi-object tracking, and person re-identification. The best solution leveraged the power of deep learning architectures, specifically the YOLOv8 model, for accurate and efficient detection of players and the ball in video frames. The detected objects were then tracked across multiple frames using several SOTA algorithms along person ReID, leveraging their ability to incorporate appearance and motion information to maintain consistent identities.

Extensive experiments were conducted to evaluate the performance of the proposed system on a diverse set of padel sports videos. The results demonstrated the effectiveness of the approach in accurately detecting and tracking players, even in challenging scenarios with fast motion, occlusions, and varying camera and court conditions. The re-identification module, although not very expressively, showed good results in maintaining consistent player identities throughout the videos, which can be further leveraged with the fine-tuning of the models on the padel domain. Regarding the ball tracking, the results indicated the need for further experimentation with alternative methods.

Throughout the dissertation, the key contributions of the work were highlighted, including

the exploration of current literature in CV applied to sports, the development of a comprehensive pipeline for padel sports analysis which addresses observed gaps in the research domain, the adaptation of state-of-the-art deep learning techniques to the specific challenges of padel videos, and the creation of a valuable resource for future research in the form of a quality, annotated padel dataset.

Looking ahead, there is excitement about the potential impact of this work and the opportunities for further advancements. By continuing to refine and expand upon the techniques and approaches presented in this dissertation, new possibilities for sports analysis, player development, and fan engagement can be unlocked. It is hoped that these contributions will inspire and guide future researchers and practitioners in this field, driving innovation.

6.1 Future Work

Finally, while significant progress has been made, building upon the defined techniques and availability of the annotated dataset, there are several avenues for future research and development that can further enhance the capabilities and performance of the proposed system. Consequently, annotating videos to validate the ball tracking across more broad scenarios is also relevant.

The first important step to build upon the work done is to improve the final dataset for the Object Detection. In particular, it is important to include more quantity and quality of ball instances in an attempt to achieve better robustness and consequent tracking continuity for the ball.

One other promising direction, in case the ball tracking is not improving even with a much larger dataset, is to shift from the tracking-by-detection approach, incorporating temporal information and trajectory prediction and analysis into the tracking process. By considering the progression of events over time, the system can better understand and predict the movement of the ball, leading to more accurate and robust tracking results. This can be achieved by employing models that take time into account, such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks, which are capable of capturing and leveraging the temporal dependencies within the video data. For instance, by feeding these models with information from multiple subsequent frames, the system can better predict and track the ball's movement, even in challenging scenarios such as fast motion or occlusions.

Another area for improvement is the refinement of the re-identification (REID) module. Fine-tuning the REID model on a larger and more diverse dataset specific to padel can enhance its ability to distinguish between players and maintain consistent identities throughout the video. Additionally, incorporating semantic information, such as player positions or actions, can provide valuable context for the re-identification process and improve its accuracy.

To gain a deeper understanding of the game dynamics and player performance, future work can also focus on extracting and analyzing player trajectories. By tracking the movements of players over time, valuable insights can be obtained regarding their positioning, speed, and overall strategy. Furthermore, pose estimation techniques can be employed to capture the detailed body

movements and actions of players, enabling a more comprehensive analysis of their technique and performance.

Event detection is another crucial aspect that can be addressed in future work. By identifying and classifying key events, such as shots, rallies, and referee actions, the system can provide a more structured and meaningful analysis of the game. This can be achieved by detecting and recognizing specific gestures, analyzing player and ball trajectories during critical moments, and classifying different types of shots performed by players. Additionally, incorporating sound analysis can further enhance the event detection capabilities by analyzing the audio signals present in the video; the system can detect and recognize specific sounds associated with events, such as the distinctive sounds of a ball being hit, a referee's whistle or the crowd cheering. This multimodal approach, combining visual and audio cues, can improve the accuracy and robustness of event detection, particularly in situations where visual information alone may be ambiguous or occluded.

In summary, future work in this domain should focus on improving ball tracking, either by expanding the dataset or incorporating temporal information, refining the REID module, extracting semantic information, analyzing player trajectories and poses, and detecting key events. By addressing all these aspects, a full-fledged system providing comprehensive, accurate, and insightful analysis of padel sports videos can be delivered and even introduced into the real world, benefiting players, coaches, and fans alike.

Bibliography

- [1] Zillur Rahman, Amit Ami, and Muhammad Ullah. A real-time wrong-way vehicle detection based on yolo and centroid tracking, 10 2022.
- [2] UsaPlaysPadel. Official padel court measurements. <https://www.usaplayspadel.com/what-is-padel/>.
- [3] Saily Shah. Convolutional neural network: An overview, 2022. <https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/>.
- [4] Teemo. Surprising stats about the growth of padel around the world. <https://www.padel.fyi/blog/padel-statistics>.
- [5] World Padel Tour. Padel rules, 2023. <https://worldpadeltour.com/en/rules>.
- [6] Business Research Insights (BRI). Padel sports market report - 2023 | forecast, 2030. <https://www.businessresearchinsights.com/market-reports/padel-market-102910>.
- [7] Mog technologies official website. <https://www.mog-technologies.com/>.
- [8] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.
- [9] Muhammad Hussain. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, 11(7), 2023.
- [10] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.
- [11] Niall O' Mahony, Stewart Campbell, Andre Carvalho, Shashank Harapanahalli, Gustavo Vargas Hernandez, Lucie Krpalkova, David Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. *arXiv preprint arXiv:1910.13796*, 2019.
- [12] Lawrence G Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963.
- [13] David A Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

- [14] Thomas B Moeslund, Graham Thomas, and Adrian Hilton. *Introduction to the Use of Computer Vision in Sports*, pages 1–14. Springer, 2014.
- [15] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing*. Pearson Education, 2006.
- [16] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O’Reilly Media, Inc., 2008.
- [17] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, volume 1, pages 886–893. IEEE, 2005.
- [18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.
- [19] Banoth Thulasya Naik, Mohammad Farukh Hashmi, and Neeraj Dhanraj Bokde. A comprehensive review of computer vision in sports: Open issues, future trends and research directions. *Applied Sciences*, 12(9):4429, April 2022.
- [20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [21] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE Access*, PP:1–1, 09 2019.
- [22] Zhong-Qiu Zhao, Peng Zheng, Shou tao Xu, and Xindong Wu. Object detection with deep learning: A review, 2019.
- [23] ResearchGate. Learning feature fusion in deep learning-based object detector, 2023. [Online; accessed 20-June-2023].
- [24] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
- [25] Gorlapraveen123. Resnet50.png. <https://commons.wikimedia.org/wiki/File:ResNet50.png>. Accessed: 2023-11-20.
- [26] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net, 2020.
- [27] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Better than yolov4, scaled-yolov4, and efficientdet. *arXiv preprint arXiv:2207.02696*, 2022.
- [28] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, April 2021.

- [29] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, March 2020.
- [30] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box, 2022.
- [31] Guangyu Zhu, Changsheng Xu, Liyuan Xing, Wen Gao, and Hongxun Yao. Human behavior analysis for highlight ranking in broadcast racket sports video. *Multimedia, IEEE Transactions on*, 9:1167 – 1182, 11 2007.
- [32] Ashraf Russo, Laksono Kurniangugoro, and Kang-Hyun Jo. Classification of sports videos with combination of deep learning models and transfer learning. pages 1–5, 02 2019.
- [33] Antonio Tejero-de Pablos, Yuta Nakashima, Tomokazu Sato, Naokazu Yokoya, Marko Linna, and Esa Rahtu. Summarization of user-generated sports video by using deep action recognition features. *IEEE Transactions on Multimedia*, 20(8):2000–2011, 2018.
- [34] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.
- [35] Jaime Sendra. Pp_actions dataset. https://universe.roboflow.com/jaime-sendra/pp_actions, Jan. 2024. Visited on 2024-03-01.
- [36] Eduardo de la Paz. Net detection dataset. <https://universe.roboflow.com/eduardo-de-la-paz-l5r2d/net-detection>. visited on 2024-03-01.
- [37] Giuseppe Paolini. padel players detection on yolov4 tiny dataset. <https://universe.roboflow.com/giuseppe-paolini-xj6ip/padel-players-detection-on-yolov4-tiny>. visited on 2024-03-01.
- [38] Roman Voeikov, Nikolay Falaleev, and Ruslan Baikulov. Ttnet: Real-time temporal and spatial video analysis of table tennis. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.
- [39] Tayeba Qazi, Prerana Mukherjee, Siddharth Srivastava, Brejesh Lall, and Nathi Ram Chauhan. Automated ball tracking in tennis videos. In *2015 Third International Conference on Image Information Processing (ICIIP)*, pages 236–240, 2015.
- [40] Yu-Chuan Huang, I-No Liao, Ching-Hsuan Chen, Tsì-Uí Ík, and Wen-Chih Peng. Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications, 2019.
- [41] David Wennerblom and Andrey Arronet. Padel court detection system, 2023.

- [42] Mohammadreza Javadiha, Carlos Andujar, Michele Calvanese, Enrique Lacasa, Jordi Moyés, Jose Luis Ponton, Antonio Susín, and Jiabo Wang. Padelvic: Multicamera videos and motion capture data of an amateur padel matchpadelvic: Videos multicámara y datos de captura de movimiento de un partido de pádel amateur. *Padel Scientific Journal*, 2:89–106, 01 2024.
- [43] Simon Gustafsson and Andreas Persson. Detecting small and fast objects using image processing techniques: A project study within sport analysis, 2021.
- [44] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *CoRR*, abs/1703.07402, 2017.
- [45] Mikel Broström. BoxMOT: pluggable SOTA tracking modules for object detection, segmentation and pose estimation models.
- [46] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1483–1498, 2021.
- [47] Anirudh Chakravarthy, Honnesh Rohmetra, Divyam Goel, Harshan Baskar, Prateek Garg, and B. Rout. Complete scene parsing for autonomous navigation in unstructured environments. pages 41–45, 02 2020.
- [48] Michele Calvanese. Ball tracking for padel videos. Master’s thesis, Universitat Politècnica de Catalunya, 2023.
- [49] Alan Hanjalic. Adaptive extraction of highlights from a sport video based on excitement modeling. *Multimedia, IEEE Transactions on*, 7:1114 – 1122, 01 2006.
- [50] Wsc sports highlight detection. <https://www.sportcal.com/features/wsc-sports-how-ai-has-led-the-highlights-revolution/>. Accessed: 2023-09-30.
- [51] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Real-time multi-person 2d pose estimation using part affinity fields, 2019.
- [52] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.
- [53] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.
- [54] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

- [55] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.
- [56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [57] Siyuan Li, Li Sun, and Qingli Li. Clip-reid: Exploiting vision-language model for image re-identification without concrete text labels, 2023.
- [58] Mohammadreza Javadiha, Carlos Andujar, Enrique Lacasa, Angel Ric, and Antonio Susin. Estimating player positions from padel high-angle videos: Accuracy comparison of recent computer vision methods. *Sensors*, 21(10), 2021.
- [59] Miguel Teodoro Moreira. Aprendizagem automática para treinador virtual de padel. Master’s thesis, Universidade NOVA de Lisboa, February 2022.
- [60] Mohammadreza Javadiha, Carlos Andujar, and Enrique Lacasa. A query language for exploratory analysis of video-based tracking data in padel matches. *Sensors*, 23(1), 2023.
- [61] Christian Janiesch, Patrick Zschech, and Kai Heinrich. Machine learning and deep learning. *CoRR*, abs/2104.05314, 2021.
- [62] Kristina Host, Marina Ivasic-Kos, and Miran Pobar. Tracking handball players with the deepsort algorithm. In *ICPRAM*, pages 593–599, 2020.
- [63] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking, 2022.
- [64] Papers with Code. Openttgames dataset. <https://paperswithcode.com/dataset/openttgames>. Accessed: 2023-11-04.
- [65] Yewande Adesida, Enrica Papi, and Alison H. McGregor. Exploring the role of wearable technology in sport kinematics and kinetics: A systematic review. *Sensors*, 19(7), 2019.
- [66] Banoth Thulasya Naik and Md. Farukh Hashmi. YOLOv3-SORT: detection and tracking player/ball in soccer sport. *Journal of Electronic Imaging*, 32(1):011003, 2022.
- [67] Banoth Thulasya Naik, Mohammad Farukh Hashmi, Zong Woo Geem, and Neeraj Dhanraj Bokde. Deepplayer-track: Player and referee tracking with jersey color recognition in soccer. *IEEE Access*, 10:32494–32509, 2022.
- [68] Hyeonah Cho, Hyunyoung Ryu, and Minseok Song. Pass2vec: Analyzing soccer players’ passing style using deep learning. *International Journal of Sports Science & Coaching*, 17(2):355–365, 2022.
- [69] Ali Karimi, Ramin Toosi, and Mohammad Ali Akhaee. Soccer event detection using deep learning. *CoRR*, abs/2102.04331, 2021.

- [70] Yaparla Ganesh, Sri Teja, Sai Munnangi, and Rama Murthy Garimella. *A Novel Framework for Fine Grained Action Recognition in Soccer*, pages 137–150. 05 2019.
- [71] Genki Suzuki, Sho Takahashi, Takahiro Ogawa, and Miki Haseyama. Team tactics estimation in soccer videos based on a deep extreme learning machine and characteristics of the tactics. *IEEE Access*, PP:1–1, 10 2019.
- [72] Yoonhyung Kim, Chanhoo Jung, and Changick Kim. Motion recognition of assistant referees in soccer games via selective color contrast revelation. EasyChair Preprint no. 2604, EasyChair, 2020.
- [73] Hossam Zawbaa, Nashwa El-Bendary, Aboul Ella Hassanien, and Tai-Hoon Kim. Event detection based approach for soccer video summarization using machine learning. *International Journal of Multimedia and Ubiquitous Engineering (IJMUE)*, 7, 01 2012.
- [74] Guodong Wu. Monitoring system of key technical features of male tennis players based on internet of things security technology. *Wireless Communications and Mobile Computing*, 2021:1–6, 05 2021.
- [75] Bai Ning and Liu Na. Deep spatial/temporal-level feature engineering for tennis-based action recognition. *Future Generation Computer Systems*, 125:188–193, 2021.
- [76] Kevin Ma. A real time artificial intelligent system for tennis swing classification. In *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 000021–000026, 2021.
- [77] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation, 2019.
- [78] Jiaxin Cai, Junlin Hu, Xin Tang, Tzu-Yi Hung, and Yap-Peng Tan. Deep historical long short-term memory network for action recognition. *Neurocomputing*, 407:428–438, 2020.
- [79] Silvia Vinyes Mora and William J. Knottenbelt. Deep learning for domain-specific action recognition in tennis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 170–178, 2017.
- [80] M. Nehru J. Attzs, Trupti Mahendrakar, Mackenzie J. Meni, Ryan T. White, and Isaac Silver. Comparison of tracking-by-detection algorithms for real-time satellite component tracking. In *Proceedings of the Small Satellite Conference*, Logan, UT, August 2023. Utah State University. Accessed: [Date of Access].
- [81] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning generalisable omni-scale representations for person re-identification, 2021.

- [82] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification, 2019.
- [83] Apurva Bedagkar-Gala and Shishir K. Shah. A survey of approaches and trends in person re-identification. *Image and Vision Computing*, 32(4):270–286, 2014.
- [84] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. Deep learning for person re-identification: A survey and outlook, 2021.
- [85] Nvidia v100: The most powerful data center gpu. <https://www.nvidia.com/en-us/data-center/v100/>. accessed 20-June-2023.
- [86] Tom Polk, Dominik Jäckle, Johannes Häußler, and Jing Yang. Courttime: Generating actionable insights into tennis matches using visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):397–406, 2020.
- [87] Russ Tedrake. *Robotic Manipulation*. 2022.
- [88] Yuanbo Wang, Unaiz Ahsan, Hanyan Li, and Matthew Hagen. A comprehensive review of modern object segmentation approaches. *Foundations and Trends® in Computer Graphics and Vision*, 13(2-3):111–283, 2022.
- [89] Li Tan, Xu Dong, Yuxi Ma, and Chongchong Yu. A multiple object tracking algorithm based on yolo detection. In *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5, 2018.
- [90] Shinfeng D. Lin, Ting-Hsun Chang, and Wensheng Chen. Multiple object tracking using yolo-based detector. *Journal of Imaging Science and Technology*, 2021.
- [91] Davy Neven. Pose estimation example. https://uploads-ssl.webflow.com/63c6be5d69abf87798adedb7/64789132b267233f4969f3e5_Screenshot%20from%202023-06-01%2014-37-55.png, 2023. [Online; accessed 20-June-2023].
- [92] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.
- [93] Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss, 2022.
- [94] PadelBox. Padelbox mobile app. <https://www.padelbox.pt/apppadelbox>, 2023. Accessed: 2023-06-21.
- [95] BeerSocialPadel. Beersocialpadel mobile app. <https://beersocialpadel.pt/app>, 2023. Accessed: 2023-06-21.

- [96] AirCourts. Aircourts - reserva campos. <https://apps.apple.com/pt/app/aircourts-reserva-campos/id985659535>, 2023. Accessed: 2023-06-29.
- [97] Padel Watch. Padel watch mobile app. <https://apps.apple.com/es/app/padel-watch/id6443518532?l=en>, 2023. Accessed: 2023-06-20.
- [98] PadelCourt. Padelcourt - a computer vision based scoring detection system for padel matches. <https://padelcourt.app/>, 2023. Accessed: 2023-06-23.
- [99] Radha Penumala, M. Sivagami, and Suresh Srinivasan. Automated goal score detection in football match using key moments. *Procedia Computer Science*, 165:492–501, 2019. 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019.
- [100] Xu-Bo Fu, Shao-Long Yue, and De-Yun Pan. Camera-based basketball scoring detection using convolutional neural network. *International Journal of Automation and Computing*, 18(IJAC-2020-05-119):266, 2021.
- [101] Hyoung-Gook Kim, Steffen Roerber, Amjad Samour, and Thomas Sikora. Detection of goal events in soccer videos. In Rainer W. Lienhart, Noboru Babaguchi, and Edward Y. Chang, editors, *Storage and Retrieval Methods and Applications for Multimedia 2005*, volume 5682, pages 317 – 325. International Society for Optics and Photonics, SPIE, 2005.
- [102] Kaiqiang Wei and Xu Zhao. Multiple-branches faster rcnn for human parts detection and pose estimation. In Chu-Song Chen, Jiwen Lu, and Kai-Kuang Ma, editors, *Computer Vision – ACCV 2016 Workshops*, pages 453–462, Cham, 2017. Springer International Publishing.
- [103] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.
- [104] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *CoRR*, abs/1908.07919, 2019.
- [105] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [106] Chi-Kao Chang, Min-Yuan Fang, Chung-Ming Kuo, and Nai-Chung Yang. Event detection for broadcast tennis videos based on trajectory analysis. In *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pages 1800–1803, 2012.
- [107] N. Owens, C. Harris, and C. Stennett. Hawk-eye tennis system. In *2003 International Conference on Visual Information Engineering VIE 2003*, pages 182–185, 2003.

- [108] George Mather. Perceptual uncertainty and line-call challenges in professional tennis. *Proceedings. Biological sciences / The Royal Society*, 275:1645–51, 04 2008.
- [109] Michał Kopania, Jarosław Nowisz, and Artur Przelaskowski. Automatic shuttlecock fall detection system in or out of a court in badminton gamesmdash;challenges, problems, and solutions from a practical point of view. *Sensors*, 22(21), 2022.
- [110] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking, 2023.
- [111] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, 2019.
- [112] Michael Vaiman, VR Energy, and Marianna Vaiman. Use of pmu-based framework to improve power system resilience and reliability. In *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pages 1–6, 2018.
- [113] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolo. <https://github.com/ultralytics/ultralytics>, 2023.