NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

TOMÁS MAURÍCIO CAMPAS MARTINS

Bachelor of Computer Science and Engineering

ENHANCING RACKET SPORTS VIDEO ANALYSIS THROUGH OBJECT DETECTION AND OBJECT TRACKING

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
September, 2024

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

ENHANCING RACKET SPORTS VIDEO ANALYSIS THROUGH OBJECT DETECTION AND OBJECT TRACKING

TOMÁS MAURÍCIO CAMPAS MARTINS

Bachelor of Computer Science and Engineering

Adviser: Nuno Manuel Robalo Correia
Full Professor, NOVA School of Science and Technology

Examination Committee

Chair: Carla Maria Gonçalves Ferreira
Full Professor, NOVA School of Science and Technology

Rapporteur: Nuno Cruz Garcia
Assistant Professor, Faculty of Sciences of the University of Lisbon

Member: Nuno Manuel Robalo Correia
Full Professor, NOVA School of Science and Technology

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
September, 2024

Enhancing Racket Sports Video Analysis through Object Detection and Object Tracking

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.5) [35].

Acknowledgements

First and foremost, I would like to thank the NOVA School of Science and Technology, who helped me grow on both a personal and academic level. I would also like to thank all my professors, especially Professor Nuno Correia, for the support and guidance throughout the development of this thesis, as well as Rui Rodrigues and João Diogo, who provided valuable feedback during the project.

Furthermore, my appreciation goes to all my friends and colleagues, in particular the ones who provided suggestions and helped me during this journey. Additionally, I am also thankful to all the participants of the user tests, who dedicated some of their time to test the system and contributed to its improvement.

Finally, I thank my family, especially my parents, for their continuous availability, encouragement and support.

”

“Dedication makes dreams come true.”

— Kobe Bryant

Abstract

With the technological advancements in recent years, such as cameras and videos with higher resolution, more computational resources, portable devices, and the growth of Artificial Intelligence (AI), our everyday life has become increasingly integrated with technology. Thus, one of the possible applications for these technologies is sports, which could benefit from such developments in many distinct use cases, such as enhancing player analysis, training, and performance.

Video, in particular, represents a transversal element used across a wide range of sports, enabling athletes, coaches, and spectators to visualize content. Therefore, the main objective of this thesis is to enhance sports analysis through intelligent video techniques.

This work focuses on padel as a use case. This racket sport has rapidly increased in popularity worldwide and thereby has potential in several implementation routes. Even though this project could target other sports, padel presents several favorable traits, such as well-defined sports rules, consistent court sizes, and controlled occlusion scenarios, making it a compelling and suitable developmental option.

A web application that automatically detects and analyzes the main elements in racket sports videos, such as the net, ball, and players was developed. By using object detection and tracking models, the system identifies these elements in each video frame, providing a more detailed view of the game. The system provides a customizable experience, enabling users to focus on specific game elements and providing novel functionalities related to court coverage, such as players' heatmaps and trajectories.

Two series of user studies were conducted to evaluate the usefulness and usability of the system: a preliminary user test where the prototype was assessed, and a final user test where participants evaluated the fully developed system. Overall, the feedback from users was positive in both phases.

Keywords: Object Detection, Object Tracking, Computer Vision, Machine Learning, Sports Analysis.

Resumo

Com os avanços tecnológicos nos últimos anos, como câmeras e vídeos com maior resolução, mais recursos computacionais, dispositivos portáteis e o crescimento da Inteligência Artificial (IA), a nossa vida quotidiana tornou-se cada vez mais integrada com tecnologia. Deste modo, uma das possíveis aplicações dessas tecnologias é o desporto, que pode beneficiar desses desenvolvimentos em muitos casos distintos, como na melhoria da análise, treino e performance de jogadores.

O vídeo, em particular, representa um elemento transversal utilizado em vários desportos, permitindo que atletas, treinadores e espetadores visualizem conteúdo. Assim, o principal objetivo desta tese é melhorar a análise desportiva através do uso de técnicas de vídeo inteligentes.

Este trabalho tem o padel como caso de estudo. Este desporto tem ganho rapidamente popularidade mundial, apresentando potencial em várias rotas de implementação. Embora este projeto pudesse focar-se em outros desportos, o padel tem várias características favoráveis, como regras bem definidas, tamanhos de campo consistentes e cenários de oclusão controlados, tornando-o uma opção de desenvolvimento adequada.

Foi desenvolvida uma aplicação web que deteta e analisa automaticamente os principais elementos em vídeos de desportos de raquetes, como a rede, a bola e os jogadores. Através do uso de modelos de deteção e rastreamento de objetos, o sistema identifica estes elementos em cada frame de vídeo, oferecendo uma visão mais detalhada do jogo. O sistema proporciona uma experiência personalizável, permitindo aos utilizadores concentrar-se em elementos de jogo específicos e oferecendo funcionalidades inovadoras relacionadas com a cobertura do campo, como mapas de calor e trajetórias dos jogadores.

Realizaram-se duas séries de estudos com utilizadores para avaliar a utilidade e a usabilidade do sistema: um teste preliminar em que o protótipo foi avaliado, e um teste final em que os participantes avaliaram o sistema totalmente desenvolvido. Em geral, as opiniões dos utilizadores foram positivas em ambas as fases.

Palavras-chave: Deteção de Objetos, Rastreamento de Objetos, Visão Computacional, Aprendizagem Automática, Análise Desportiva.

Contents

List of Figures

List of Tables

Listings

Glossary

Convolutional Neural Network Convolutional Neural Networks are deep learning networks designed to recognize two-dimensional (2D) shapes in images, allowing fast information extraction from an extensive data set. [53] *(p. 8)*

Embedded Systems A mix between computer hardware (usually microcontrollers) and a limited amount of software, part of a broader system/product. Cellphones, heart monitors, traffic lights, and credit/debit card readers are examples of embedded systems. [55] *(p. 9)*

Real-time Applications Applications where the users expect more than to have correct logical results, they also expect results computed in the shortest amount of time possible. Video chatting, message exchange, and online gaming are examples of real-time applications. These are examples of systems where the minimum delay in the response upsets the user. [55] *(pp. 9, 11)*

URL Uniform Resource Locator (URL) is a unique identifier to a web resource. *(p. 23)*

Acronyms

AI Artificial Intelligence *(pp. iv, 8)*
AR Augmented Reality *(p. 19)*

CMC Camera Motion Compensation *(p. 14)*
CRUD Create, Read, Update, and Delete *(p. 34)*
CV Computer Vision *(pp. 1, 2, 4, 6–8, 15, 16, 18, 19, 21, 22, 28, 29, 35, 46, 64, 65)*

Faster R-CNN Faster Region-based Convolutional Neural Network *(pp. 8, 9, 11)*
FPS Frames Per Second *(pp. 10, 14)*

GenOD Generic Object Detection *(pp. 17, 18)*

HCI Human-Computer Interaction *(p. 1)*
HOTA Higher Order Tracking Accuracy *(p. 15)*

mAP Mean Average Precision *(pp. 10, 16, 41)*
MOTA Multiple Object Tracking Accuracy *(pp. 14, 15)*
MRAS Microsoft Research Annotation System *(pp. 24, 25)*
MS COCO Microsoft Common Objects in Context *(pp. 11–13, 36)*

RPN Region Proposal Network *(p. 8)*

SSD Single Shot MultiBox Detector *(pp. 8–11, 36)*
SUN Scene UNderstanding *(pp. 12, 13)*
SUS System Usability Scale *(pp. 55, 56, 58, 59)*

UEQ User Experience Questionnaire *(pp. 58, 62)*
UI User Interface *(pp. 24, 29, 33, 63)*

YOLO You Only Look Once *(pp. 8, 10, 11, 16)*

1 Introduction

This first chapter introduces the context of the project in which this thesis is integrated, the motivation behind it, a proposed solution, the main contributions, and the document's structure.

1.1 Context

This thesis is related to previous developments of a web-based video annotation tool (Motion Notes [49]) that uses multimodal components and contributes to the Human-Computer Interaction (HCI) field of study.

The project's main objective was to enhance sports analysis by using Computer Vision (CV) technologies in sports videos, improving insights for athletes and coaches. This led to the development of a web application that utilizes object detection and tracking to identify the main elements in racket sports videos, such as the players, the net, or the ball. After enhancing the object detection and tracking features, other derived features were computed to further improve the overall video analysis.

In addition, this document presents a narrative showcasing the usefulness of these technologies and how they can be integrated into sports training/analysis and various other industries.

1.2 Motivation and Problem Definition

In recent years, technology has evolved tremendously, and its use in sports is more and more recurrent. There are several examples of technology uses in sports, not only to help the referees' decisions (like the Video Assistant Referee (VAR) [36] in football or the Hawk-Eye [33] in tennis) but also to monitor player's performances (e.g., GPS vests [50] in football are used to track the player's total distance, speed, heat maps).

Adding to the technological evolution, the current competitiveness in professional sports shows that it is pivotal to identify player's strengths and especially weaknesses to improve their game. Besides professional athletes, younger players who aspire to become

professionals and even people who occasionally play friendly matches would also benefit from such technologies.

Sensors could be an option to consider due to being a precise and reliable technique. In racket sports, for example, embedded devices in rackets and balls provide valuable data on attributes such as spin, speed, and impact points [63], and they can be extended to body-worn sensors [60]. Despite that, such technologies (e.g., motion capture sensor systems) are usually not used by the average user, due to their accessibility (higher cost or lack of access to specialized equipment). Another downside is that these systems may be too invasive or uncomfortable for the athlete, displaced from their original position due to body movement [59], and even banned from official matches of some sports[1].

Alternatively, the use of video has become increasingly common in sports and CV techniques, such as object detection and object tracking, have become much more reliable, rapidly evolved over the last few years, and play a major role in the sports industry [58]. These techniques have greater accessibility and can, in some cases, be a cheaper and easier mechanism to evaluate players' performances and improve their self-awareness. This is attributable to the widespread use of digital devices (such as smartphones, cameras, and computers), which have become integral parts of modern daily life, and the significant expansion of sports video archives.

Such mechanisms could be used with a wide range of players, from amateurs to top-level professional athletes, since nowadays it is effortless to record videos of training sessions (e.g., at an amateur level, even a smartphone could be used). The widespread televised transmission of major sports tournaments (and sometimes even smaller ones like under-18 or college tournaments) also facilitates the use of these techniques, making it possible to analyze professional games. In addition to improving player performance and technique [48], these technologies can also be used to evaluate players' risk of injury [4].

1.3 Solution

This thesis contributes to the development of an object detection and tracking web application to improve racket sports coaching and analysis. A prototype was initially developed, composed of four main features: automatically identify key elements of the game (e.g., players), selective highlight of game elements (e.g., increase contrast with surroundings), search for specific plays (such as serves), and show game statistics. Figure 1.1 shows a prototype of this application, specifically the bounding boxes feature.

The development of this application encountered many challenges. Placing the bounding boxes correctly and making a smooth transition between frames was the first big challenge, since racket sports elements, like the players, the racket, and especially the ball, can travel at a considerable speed and become blurry in a specific video frame [37]. Another

[1] https://www.si.com/media/2017/02/02/nba-data-analytics-new-cba-wearable-device (visited on January 2024).

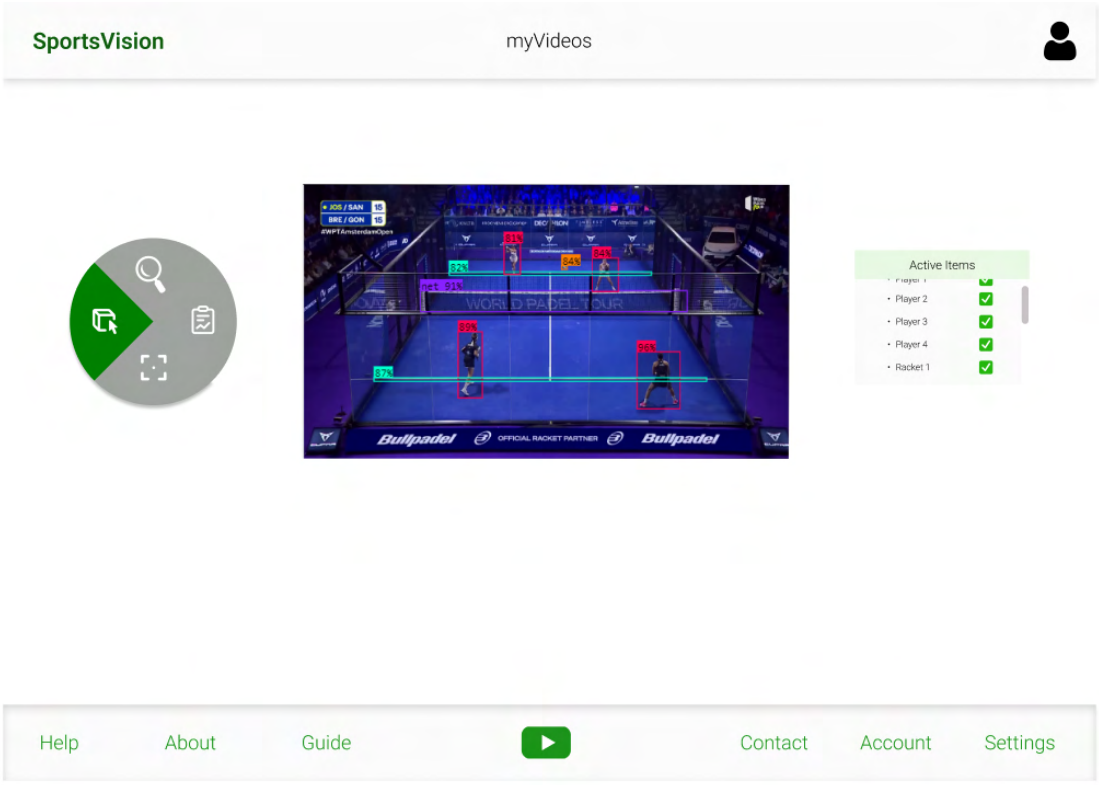


Figure 1.1: Initial prototype of the application.

challenge that proved to be particularly problematic was the occlusion of game elements during the video, which made detection and tracking more difficult. New techniques were developed to address this issue, leading to enhanced performance and improving the overall accuracy of the system. These and other challenges will be addressed further in this document.

Even though this work could be applied across many industries, the developed solution focuses on racket sports, more specifically padel. According to the Global Padel Report 2024[2], by the end of 2023, there were approximately 43,000 padel courts worldwide, with over 5,000 built in that year alone, averaging 16 courts per day. The report anticipates a growth rate of 17% per year until 2026, reaching almost 70,000 courts globally. This report is just one of many indicators of padel's tremendous growth over recent years, highlighting its potential for many uncharted implementation routes.

Padel's advantageous characteristics, such as well-defined sports rules, consistent court sizes, and controlled occlusion scenarios also contributed to validating it as a suitable use case. Unlike other sports such as basketball or football, where players from both teams can be anywhere on the pitch, in padel matches, each team is exclusively on their side of the court. As a result, the occurring occlusions are controlled (e.g., limited player movement intersection), enabling fine-grained collision resolution.

Due to the limited amount of padel datasets and the absence of padel objects in general

[2] https://products.playtomic.io/global-padel-report/ (visited on September 2024).

object detection datasets, it was pertinent to develop a custom dataset to achieve better detection and tracking results. Therefore, a novel padel dataset was created with over 1500 manually annotated images to optimize results. Despite the substantial challenges resulting from this phase, such as accurately labeling game elements and selecting representative images, visible improvements were achieved as the dataset grew. Even though this process was very time-consuming, it visibly improved the results' accuracy.

1.4 Contributions

As a result of the work executed during this thesis, the main contributions are the following:

- **Creation of a custom dataset**: Selection and labeling of images to create a custom dataset that is adequate to the racket sports context. Additionally, this custom dataset was used to train an object detection/tracking model. This was an iterative process, with several phases, showing improvements with each iteration.
- **Object detection and tracking on video content**: The CV model trained with the custom dataset was used to implement object detection and tracking features, enhancing game analysis and serving as foundational functionalities for the system.
- **Development of additional features**: Other features were implemented that derived and used the results from the previous features to provide other valuable data about each player's performances (e.g., players' heatmap and trajectory).
- **System Evaluation and Publication**: Evaluation of the system through the use of usability tests on practitioners, athletes and coaches. The system's evaluation consisted of two phases: a preliminary evaluation after the prototype was created, and a final evaluation, following the system's complete development. Additionally, there has been a contribution to the development of a scientific paper.

1.5 Document Structure

This document's structure is divided into the five following chapters:

- **Introduction**: The first chapter details the context and the motivation behind the development of this thesis. It also presents a brief description of the proposed solution, the main challenges and the main contributions that can be expected.
- **Related Work**: The second chapter discusses relevant theoretical concepts in developing this system's features (e.g., object detection). After introducing these concepts, an overview of some object detection and object tracking algorithms is presented. This chapter also presents a detailed analysis of the related work in this area and some similar systems.

- **Design and Implementation**: The third chapter explains the proposed solution, which technologies were used, and how the system evolved during its development. It presents all features of the system, the alternatives that were considered, and what was ultimately implemented.
- **Evaluation and Results**: The fourth chapter describes the two phases of the system's evaluation, analyzing its results and the overall feedback from participants.
- **Conclusions and Future Work**: The last chapter discusses the conclusions drawn from the development and evaluation of the system, as well as some improvements and further work that should be performed in the future.

2

Related Work

The previous chapter presented an overview of the context and motivation behind this thesis. This chapter is divided into seven sections: Computer Vision Concepts, Object Detection Algorithms, Datasets, Object Tracking Algorithms, Similar Systems, Video Annotation, and Summary.

The first section presents some CV principles that will be important to get a clearer insight into this thesis' topic. The second section addresses a few object detection algorithms, comparing their performance and purposes. The third section explores the significance of datasets and how they can influence the performance of object detection models. The fourth section compares the performance of some object tracking algorithms. The fifth section details applications that use similar technologies and then focuses on sports-driven applications. The sixth section presents concepts related to annotation and various systems that take advantage of such techniques. The final section provides a summary of this chapter.

2.1 Computer Vision Concepts

Before discussing some algorithms and their practical applications, it is relevant to get a better understanding of CV, its main concepts, and the differences between them. This analysis is inspired by the Computer Vision: Algorithms and Applications book by Richard Szeliski [56] and by the Ultralytics YOLOv8 GitHub page[1], a well-known object detection algorithm.

The most basic notions are **classification** followed by **localization**. In classification, a frame is classified based on the objects of the image (e.g., if an image has a car in it, it will be categorized as a car). Alternatively, localization locates objects in the frame, typically drawing a box around it. **Object detection** combines both classification and localization to achieve better results.

Instance segmentation takes an additional step by not only detecting objects, but also identifying individual objects in a frame, drawing a contour around each object. Unlike

[1] https://github.com/ultralytics/ultralytics (visited on January 2024).

the previous concepts, instance segmentation distinguishes objects: even if two different objects belong to the same class, they are considered unique instances.

Object tracking classifies and locates a specific object (or multiple ones) in a sequence of frames. Similarly to instance segmentation, different objects that belong to the same class are considered distinct instances. However, object tracking can be much more challenging due to many outside factors that could pollute this computation, such as object intersection or the rapid appearance of an object. Even if a video is clear and everyone can determine the object's location, the resulting video frames can have less quality and become blurry, making it harder to locate the object in individual frames.

Pose estimation consists of estimating the human pose in a frame based on a set of 2D point projections. Its main goal is to identify specific keypoints, such as a person's body parts/joints.

Figure 2.1 shows examples of the CV concepts that were previously described.

Figure 2.1: Examples of computer vision concepts[1].

2.2 Object Detection Algorithms

Having established a better understanding of some CV principles, it is now relevant to approach some object detection algorithms. Since it is unfeasible to discuss all existing algorithms, the analysis will focus on 3 well-known algorithms: Faster Region-based Convolutional Neural Network (Faster R-CNN), Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO). This choice was based on the comparison made by S. Srivastava et al. [54] between deep learning image detection algorithms.

Note that technology, especially AI, is evolving rapidly, meaning that some results may become obsolete or outdated over time.

2.2.1 Faster Region-based Convolutional Neural Network (Faster R-CNN)

Since Faster R-CNN originates from earlier algorithms, it is relevant to offer information about the previous algorithms. Hence, before Faster R-CNN was developed, R. Girshick et al. proposed R-CNN [20], that incorporated region proposals for object segmentation with high-capacity Convolutional Neural Network (CNN) for object detection. Although R-CNN had excellent accuracy, training was composed of a multi-stage pipeline, resulting in slow object detection (expensive in time and space). Figure 2.2 displays how R-CNN computes.

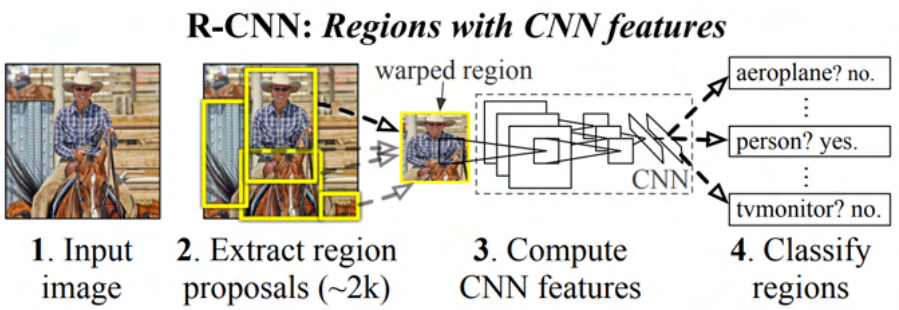


Figure 2.2: R-CNN object detection overview [20].

Later, R. Girshick proposed the Fast R-CNN [19] algorithm, which uses deep convolutional networks to classify objects. Fast R-CNN improves its predecessor by using the image directly to generate a convolutional feature map instead of region proposals. This new detection method grants higher detection accuracy and several improvements in training and testing speed. The architecture of Fast R-CNN can be seen in Figure 2.3.

Faster R-CNN, proposed by S. Ren et al. [47], was the successor of Fast R-CNN. It is described as a Region Proposal Network (RPN) that uses full-image convolutional features for the detection network, allowing a low, nearly free, region proposal cost. The authors of Faster R-CNN describe an RPN as "a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position" [47].

Faster R-CNN is composed of 2 modules. The first module uses a fully convolutional network to propose regions, and the second module uses the proposed regions as input

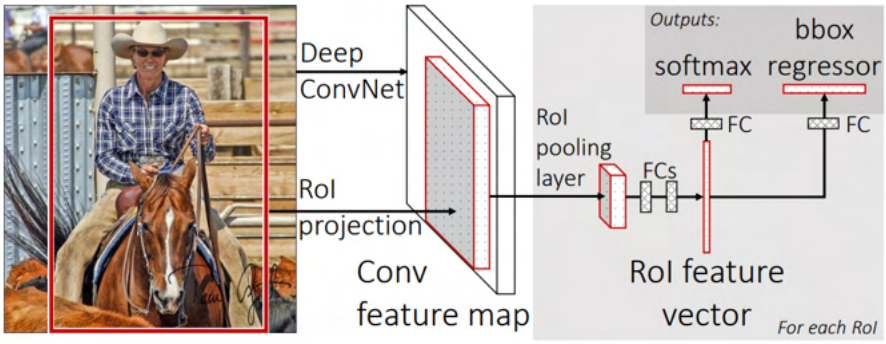


Figure 2.3: Fast R-CNN object architecture [19].

for the Fast R-CNN detector. Figure 2.4 displays how Faster R-CNN functions.

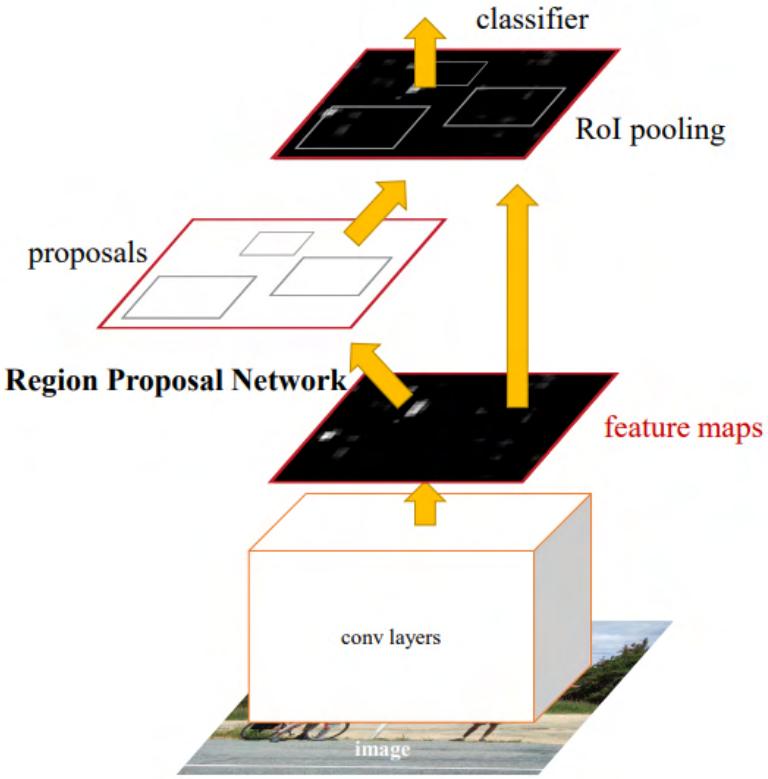


Figure 2.4: Faster R-CNN's structure [47].

2.2.2 Single Shot MultiBox Detector (SSD)

W. Liu et al. developed SSD [34], an object detection algorithm that uses a single deep neural network. According to their studies, most object detection approaches were based on Fast R-CNN, and even though these techniques had high precision, they were just too computationally expensive to execute in embedded systems and too time-consuming for real-time applications. There were several attempts to develop faster algorithms, but due

to a trade-off between performance and accuracy, all those solutions would come at a cost of an expectable and notably lower detection accuracy.

SSD was one of the first and most accurate deep network-based object detection algorithms that did not resample pixels or features for bounding boxes (and was as precise as approaches that did). This improvement was the main reason for the increase in the algorithm's speed.

To prove their point, the authors compared SSD with another single-shot object detector algorithm: YOLO (addressed in Section 2.2.3). As shown in Figure 2.5, SSD achieved 74.3 Mean Average Precision (mAP) and 59 Frames Per Second (FPS) with a 300 x 300 input image, while YOLO only managed to get 63.4 mAP and 45 FPS with a 448 x 448 input. So, at the time of this study, it was clear that SSD was the more accurate and faster algorithm.

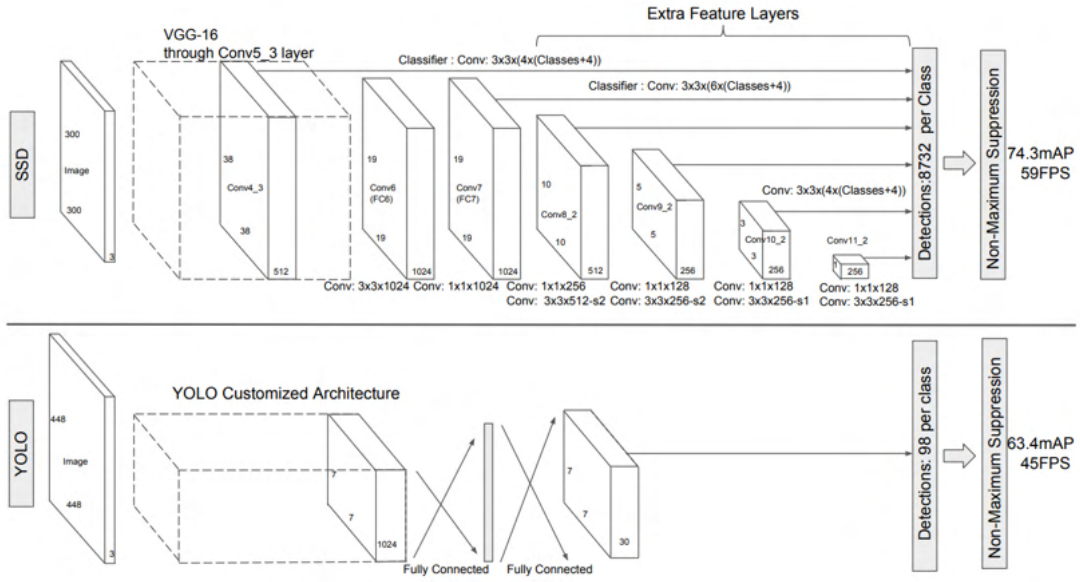


Figure 2.5: Comparison between SSD and YOLO [34].

2.2.3 You Only Look Once (YOLO)

YOLO was first presented by J. Redmon et al. in 2015 [46], just a few months before SSD. At its release, the main highlight was speed, achieved by using a single CNN to predict bounding boxes and classify images in a single evaluation. Figure 2.6 illustrates how YOLO functions:

1. Resizes the original image to 448 x 448 resolution.

2. Runs the single CNN.

3. Produces detections displaying confidence intervals that are above an intended threshold.

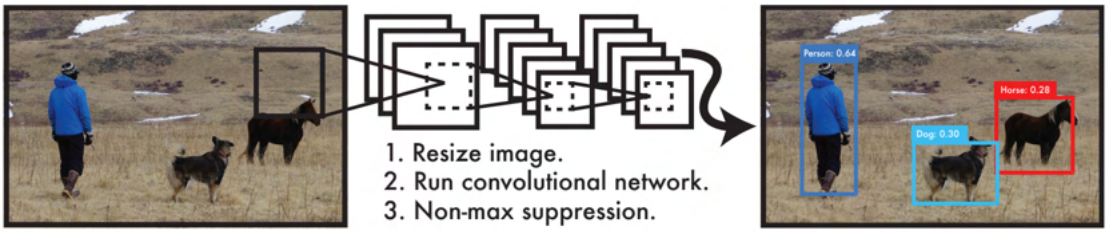


Figure 2.6: YOLO detection overview [46].

Despite having high speed, when compared with other algorithms such as Fast R-CNN, YOLO had a lower detection accuracy and struggled the most when trying to locate small objects. While Fast R-CNN located 71.6% of the objects correctly, YOLO only had 65.5% accuracy. Nevertheless, YOLO had far fewer background errors (4.75% versus 13.6%). As said in Section 2.2.2, YOLO had not only lower accuracy but also lower speed than SSD.

In its subsequent versions, YOLO managed to always be one step ahead of its competitors. When YOLOv2 [44] was released, the authors made several benchmarks comparing YOLO with Fast R-CNN, SSD, and other algorithms. In general, YOLO kept up with all of the other state-of-the-art detectors, with similar accuracy but much faster results (in some cases, from 2 up to 10 times faster).

A study in 2021 was made by S. Srivastava et al. [54], comparing YOLOv3 [45] with SSD and Faster R-CNN. Faster R-CNN had higher accuracy but it was a lot slower than the others, being best suited for small datasets and applications that do not need real-time responses. SSD was considered a middle ground between Faster R-CNN and YOLO, having a good balance between both accuracy and speed. YOLOv3 would be the best option for real-time applications and was considered the best overall detector.

YOLO is currently in its eighth version, YOLOv8, developed by Ultralytics and it is an open-source project available on the Ultralytics GitHub page[1]. YOLOv8 takes advantage of the success of its predecessors, adding new features and improving overall performance and efficiency. YOLOv8 has several new functionalities (including all concepts described in Section 2.1), allowing a more versatile use in different applications and domains.

YOLOv8 offers several pre-trained models for detection, segmentation and posing that are designed to identify specific classes of objects. These models are trained using the Microsoft Common Objects in Context (MS COCO) [32] dataset. Thus, it is relevant to introduce some concepts about datasets, to provide a better understanding of them.

2.3 Datasets

As stated by T. Gebru et al. [18], data is really important in machine learning. Machine learning models, such as the ones addressed in Section 2.2, are trained using datasets. The datasets' characteristics influence the model's behavior: a model is improbable to perform well in a context that is different from its training/evaluation datasets. Detection errors or

biased datasets may cause severe repercussions in areas like criminal justice, hiring, and finance, or produce results that are socially or racially discriminatory.

When benchmarks are made to compare different models, it is important to use the same dataset to produce results with the same starting point. Therefore, to get a clearer insight into the various types of datasets, the enumeration of some dataset examples and their differences will now be presented.

ImageNet [15] has several object categories, many of which are divided into several subcategories. It has over 3.2 million images and its main goal was to be a better dataset than the ones already existing, providing a larger, more diverse, and more accurate dataset.

Scene UNderstanding (SUN) [62] focuses on categorizing scenes, such as indoor (e.g., police office, bowling, and sports stadium), urban (e.g., cathedral, campus, and skyscraper), and nature (e.g., archipelago, mountain, and river). Usually, datasets have hundreds of different categories, but the largest available scene dataset only had 15 categories. SUN improves the lack of diversity in scene datasets by having 899 categories and over 100,000 images.

PASCAL[2] **Visual Object Classes (PASCAL VOC)**[3] is a dataset composed of annotated summer photographs obtained from the flickr[4] website and its main purpose is to detect objects in natural images. From 2005 to 2012, the PASCAL VOC Challenge [16] was an annual benchmark that used the PASCAL VOC dataset to perform standard evaluation procedures. It was considered *the* benchmark for object detection.

MS COCO [32] was developed with the intent of advancing the state-of-the-art in object recognition by using it in the context of scene understanding. To do this, its developers collected pictures of complex scenes from everyday life, including common objects within their natural environment.

The MS COCO annotation pipeline (see Figure 2.7) is composed of 3 main phases: (a) Label all the existing categories in the image, (b) locate and mark all instances of those categories, and (c) complete instance segmentation.

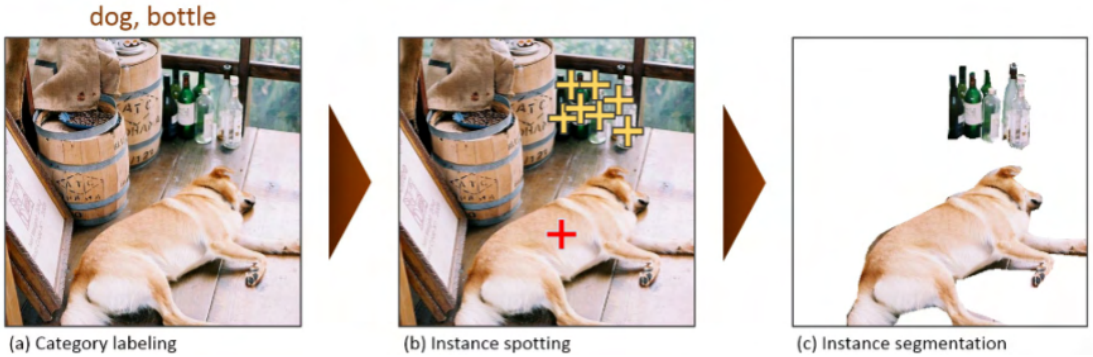


Figure 2.7: MS COCO annotation pipeline [32].

[2]PASCAL means pattern analysis, statistical modeling and computational learning and is an EU Network of Excellence funded by the IST Programme of the European Union.

[3]http://host.robots.ox.ac.uk/pascal/VOC/ (visited on January 2024).

[4]https://www.flickr.com/ (visited on January 2024).

When compared with the other datasets discussed in this section, MS COCO had, on average, more categories and instances per image than ImageNet and PASCAL VOC. MS COCO also has 90% of its images with more than one category, unlike ImageNet and PASCAL VOC which have around 60% of images with only a single category. The SUN is composed mostly of contextual information due to being scene-based. Both MS COCO and SUN have smaller objects on average, which are usually more difficult to identify.

All these characteristics show that MS COCO is an overall complete dataset, with more complex and harder-to-identify images. Harder images may not always help the recognition and may pollute the model if the model is not rich enough. This dataset allows the investigation of these issues.

There are also larger-scale datasets, like OpenImages [29] developed by Google, LVIS [23] developed by Facebook, and VisualGenome [28], that have a wider diversity of categories, with up to thousands of distinct categories.

Additionally, sports datasets such as SoccerNet[5] could be used for more specific and accurate use in sports-driven applications.

As previously stated, there are many distinct datasets available and it is important to know the differences between them. This will not only help in the choice of the dataset that best fits our purpose but it will also contribute to a better mitigation of errors.

2.4 Object Tracking Algorithms

Since the developed solution takes advantage not only of object detection algorithms but also object tracking ones, it is pertinent to discuss some of these algorithms in more detail. This examination will cover two distinct algorithms: ByteTrack [65] and BoT-SORT [1]. The reason for choosing these two algorithms is that they are the options available in the tracking feature of YOLOv8[6], with BoT-SORT being the default option.

2.4.1 ByteTrack

ByteTrack is a multi-object tracking algorithm proposed by Y. Zhang et al. in 2022 [65]. Before the development of ByteTrack, most trackers determined identities by relating detection boxes with scores higher than a certain threshold. While this approach was common and effective, it resulted in the discarding of objects with low detection scores, such as occluded objects, leading to significant true object misses and fragmented trajectories.

To solve this issue, the developers of ByteTrack introduced a tracking-by-association method where nearly every box is associated, not just the high-score boxes. To handle detection boxes with low scores, the authors use their similarities with tracklets to distinguish true objects and eliminate background detections. Figure 2.8 displays an example of the use of ByteTrack. In Figure 2.8 (c), the dashed boxes represent predicted boxes using

[5] https://www.soccer-net.org/ (visited on January 2024).
[6] https://docs.ultralytics.com/modes/track/ (visited on July 2024).

the Kalman filter [31]. This filter is widely used for tracking purposes. It utilizes various data observed over time, containing noise and inaccuracies, to estimate the coordinates of each bounding box in the subsequent frames. To obtain the bounding boxes, ByteTrack uses YOLOX [17].

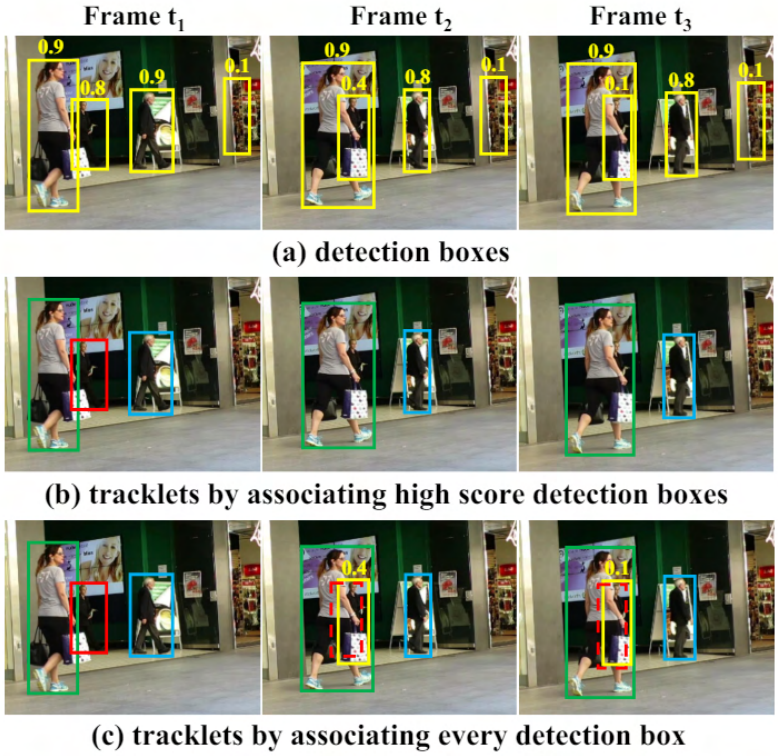


Figure 2.8: ByteTrack example [65].

The authors compared ByteTrack with nine other trackers using the test set of MOT17. ByteTrack's evaluation metrics include 80.3 Multiple Object Tracking Accuracy (MOTA) and 77.3 IDF1, with a frame rate of 30 FPS, surpassing the performance of all previous trackers.

2.4.2 BoT-SORT

BoT-SORT [1], developed by N. Aharon et al., is a multi-object tracking algorithm that provides enhancements to the ByteTrack algorithm.

To improve ByteTrack, BoT-SORT utilizes a more accurate Kalman filter state vector. Figure 2.9 illustrates the differences between the filters used by both algorithms. While the Kalman filter in ByteTrack (dashed blue) intersects the objects' legs (in red), the proposed Kalman filter (green) more accurately fits the object width.

In addition to this improvement, BoT-SORT takes advantage of Camera Motion Compensation (CMC) to prevent inaccurate results in dynamic camera scenarios. In such situations, the location of the bounding boxes in the image plane can vary drastically. Therefore, CMC is employed to mitigate increasing ID switches and false positives.

Figure 2.9: Kalman filter proposed by BoT-SORT [1].

The authors compared BoT-SORT with five other state-of-the-art trackers, including ByteTrack, on the MOT17 and MOT20 test sets. BoT-SORT outperforms all the other trackers across all evaluated metrics: IDF1, MOTA, and Higher Order Tracking Accuracy (HOTA).

2.5 Similar Systems

Now that the basic CV concepts and object detection algorithms have been discussed, it is pertinent to discuss the practical applications of these technologies. To display the diversity of use of such techniques, this section will present applications across several industries that use similar technologies, followed by a more in-depth review of sports-driven applications.

2.5.1 Applications Across Different Industries

Several industries, such as healthcare [8], security [41], agriculture [12], automotive [52], and gaming [38], take advantage of the benefits CV techniques provide.

For instance, these technologies could be used to automatically recognize license plates. The detection and recognition of license plates is an important task in traffic surveillance, parking management, vehicle recognition, and tracking for security purposes.

V. Jain et al. [27] addressed Automatic License Plate Recognition (ALPR) by using a Deep CNN methodology in real-time traffic videos. This can be quite challenging since license plates vary from country to country and sometimes even within the same country (e.g., India). License plates can vary in number of lines, font, shape, size, and color. Other factors that may interfere are the poor resolution of traffic cameras, complex backgrounds, text signs/boards, and light reflection.

Their methodology was composed of 3 main parts: (i) generation of license plate candidates by converting the image to grayscale and then applying edge filters such

as mean filters, dilation and erosion, (ii) filtering false positives with binary license plate/non-license plate CNN, and (iii) recognition of characters with 37 class CNN.

The automotive industry may also take advantage of object detection. T. Karasawa et al. [57] proposed the use of multispectral images for object detection in traffic. Multispectral images are not only composed of RGB images but also near, middle, and far-infrared images since some objects cannot be recognized in RGB images, only in infrared images.

As stated in Section 2.3, a dataset that fits our goals can have a critical role in the results. If the dataset is inadequate for our purpose, the results are not as accurate. Consequently, since there was not a multispectral image dataset available, a new dataset was generated by T. Karasawa et al.

The object detection was done with YOLO and the detection pipeline is represented in Figure 2.10. By using multispectral images, the mAP of their detection was 13% higher than that of RGB-only detection.

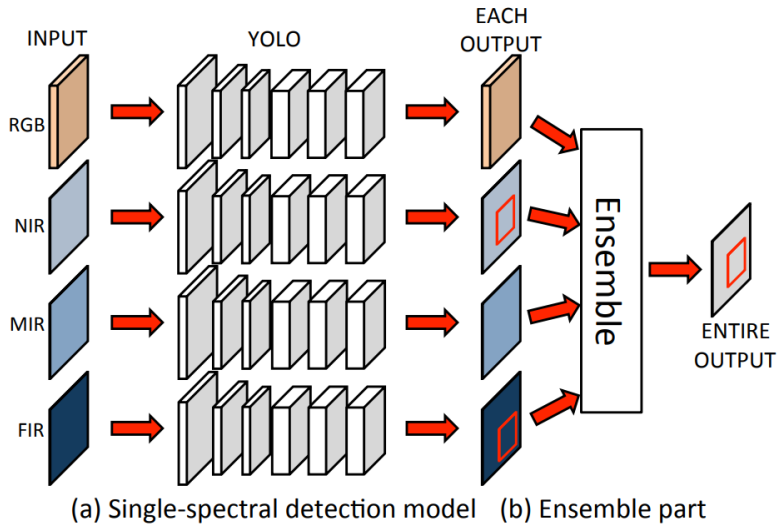


Figure 2.10: The pipeline of multispectral detection [57].

Some of the major automotive companies are already using object detection. In particular, Tesla[7] uses deep neural networks to perform image segmentation, object detection, and depth estimation to make autonomous vehicles, bi-pedal robots, and other similar technologies. Besides sensors and other mechanisms, Tesla Autopilot[8] uses the input from several cameras to perform a per-camera network identification of objects, road features, and other vehicles on the road.

While the automotive industry has seen considerable improvements in the usage of CV algorithms for many different purposes, it is relevant to display other areas that also benefit from these technologies.

Plastic waste is a big problem in today's society because it can alter habitats, accelerate climate change processes, and impact food production, ultimately affecting the lives of

[7] https://www.tesla.com/en_eu/AI (visited on January 2024).
[8] https://www.tesla.com/pt_pt/support/autopilot (visited on January 2024).

billions of people. Recycling is a major countermeasure to this problem and can also benefit from object detection, both for home and industrial use.

An object recognition method was proposed by I. W. R. Ardana et al. [2], where the authors used YOLOv3 to detect and classify real-time plastic waste. Six classes of plastic waste were proposed, namely plastic bags, plastic bottles, crushed bottles, cups, cartoons, and straws. To achieve better results, more than 1800 images were collected and then labeled (averaging around 300 images per class). The last steps were to train the model using YOLO and test it using a threshold confidence set to 0.2.

Plastic bottles and cartoons had a high confidence value for single object detection, around 85% and 75%, respectively. The straw achieved 65% and the other classes had confidence between 30% and 40%. However, in multi-object detection, only the bottle achieved a high confidence value, above 85%, the cartoon dropped to around 30% and the others maintained low values. This means that while the bottle training dataset is ready to be used, more images of the other classes should be added to the dataset, preferably using different angles and positions of the ones already present in the dataset.

Object detection is also used by the biggest search engines. In particular, Microsoft Bing[9] developed Generic Object Detection (GenOD) [9], a large-scale object detection system with over 900 categories for visual search queries in near real-time. When the user uploads an image, Bing automatically identifies visual concepts, shows similar images, and searches for product information.

Figure 2.11 depicts an example of the use of GenOD. On the left side of the figure, the desktop view shows how users can click on hotspots of an image to search for similar products. On the right, the Bing Mobile Camera uses the input of a smartphone camera to detect objects in real-time.

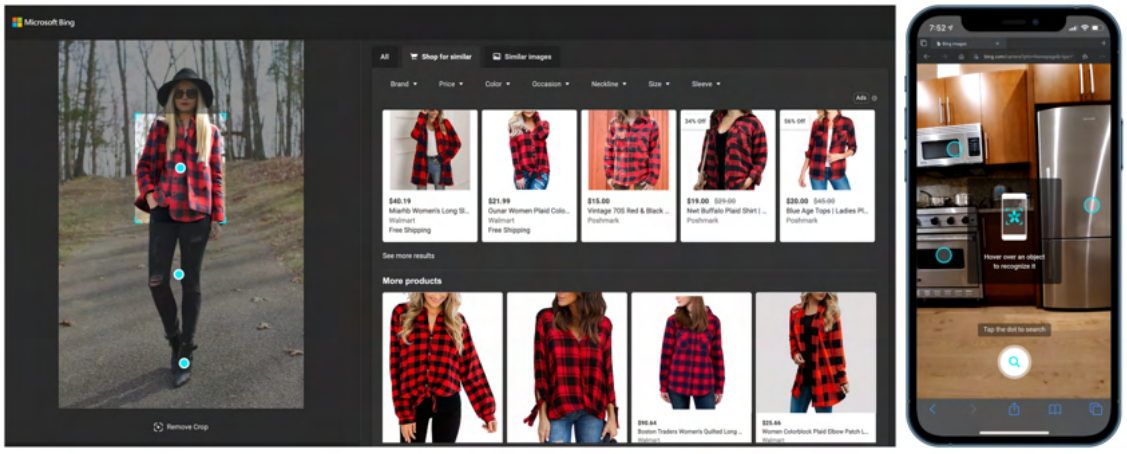


Figure 2.11: GenOD search example [9].

The data collection has 3 main stages: (i) discover all the categories present in the image, (ii) mark all the instances discovered in the previous stage, and (iii) draw bounding

[9] https://www.bing.com/ (visited on January 2024).

boxes for each category.

When developing GenOD, Microsoft trained the base model with a large amount of data to have a default detector, easy to maintain and with less frequent updates. Several datasets with different characteristics were used to train this model, varying in the number of images, categories, and density of categories per image. This design was improved by using disjoint detectors on the shared backbone, granting agile updates while not disturbing existing dependencies.

Similarly, some of the biggest retailers are already using CV techniques to enhance the customers' shopping experience. Amazon developed the Just Walk Out technology [61] that allows customers to shop as they would normally do but, instead of doing the usual checkout, the customer can simply walk out of the store and be automatically charged.

When customers remove items from shelves, these items are automatically added to their virtual cart. If the customer decides not to buy an item and puts that item back, whether in the same location or elsewhere, the item is removed from the cart. This is achievable by combining CV techniques, sensors, and several cameras positioned throughout the store. The combination of such technologies enables the automation of tasks such as checkout, item tracking, customer identification, inventory management, and fraud prevention.

2.5.2 Sports-driven Applications

The previous examples indicated that CV already has a considerable impact in many different areas. However, since this thesis primarily focuses on sports, it is relevant to thoroughly analyze what kind of similar systems are being developed in the sports industry.

SoccerNet[5] is a large-scale dataset for football video understanding, composed of 550 complete broadcast games and 12 single-camera games from the major European leagues. Initially, it only focused on three important actions: goals, cards, and substitutions. Nowadays, it includes tasks like action spotting, camera calibration, player re-identification, and tracking.

SoccerNet does yearly annual video understanding challenges where multiple teams compete internationally. In 2023, SoccerNet did the third edition of these challenges [11], divided into seven tasks (three additional tasks when compared with the previous edition):

1. **Action Spotting**: Retrieve timestamps associated with actions of interest in football (e.g., a corner kick is defined by the moment where the player kicks the ball).

2. **Ball Action Spotting (new)**: Retrieving timestamps related to changes in the state of the ball, namely passing or driving.

3. **Dense Video Captioning (new)**: Spot and describe broadcast events with natural language, associating timestamps to those events.

4. **Camera Calibration**: The objective is to detect all the lines in a football field, as well as the posts and crossbar. This is an important task that could be used to bring Augmented Reality (AR) graphics into any live game.

5. **Re-identification**: Identify a specific player or referee, using multiple cameras, across different game moments.

6. **Multiple Object Tracking**: Track key elements in a football game, such as the players, the referees, and the ball, across multiple frames.

7. **Jersey Number Recognition (new)**: Recognize the jersey numbers of the players. This may be a challenging task due to the poor resolution and visibility of the jersey numbers in the majority of the broadcast.

These are just some examples of CV tasks that could be incorporated into football video analysis, and most of these tasks could also be integrated into other sports.

Similarly to SoccerNet, DeepSportradar is a yearly competition that started in 2022 and proposes tasks for the improvement of CV techniques related to sports. The first edition [64] introduced two basketball datasets and four tasks related to basketball: ball localization, camera calibration, instance segmentation, and player re-identification. In the second edition [26], two new datasets were introduced, one for basketball and one for cricket, and three new tasks were presented: basketball player instance segmentation, basketball player re-identification and cricket bowl release detection.

Furthermore, several things could be improved in mainstream sports broadcasts. For instance, most television sports broadcasts do not provide on-demand data to meet users' individual needs. Although several apps do this job, they often end up distracting the user from the game itself.

iBall[10] [10], represented in Figure 2.12, tries to tackle this issue by presenting a live basketball video-watching system that uses embedded visualization to improve the game understanding of fans. It is composed of four main features that dynamically update based on the players' positions.

The first feature highlights the players with the ball, showing their names above them. Ball receivers for the next pass are also highlighted and open players are highlighted with a green spotlight. Star players also have small icons close to their names to indicate if they are a good shooter and/or a good defender. To detect the players, the authors fine-tuned the COCO [7] pre-trained YOLOX [17] model with an NBA player dataset.

The second feature draws a ring around the offensive player and uses color and size to represent the expected points of the shooter, based on the player's shooting records (if the ring is large and dark, the expected points are higher).

The third feature presents a defensive shield on the player who is guarding the offensive player with the ball, by using the thickness of the shield to measure the player's defensive ability and the length represents the distance to the offensive player.

[10] https://www.youtube.com/watch?v=BjdByJ5BgxI (visited on January 2024).

Figure 2.12: iBall[10].

The last feature draws lines connecting the player with the ball and the defenders guarding the player.

iBall also uses eye-tracking to perform gaze interactions. If the user keeps looking at a player, the player will start to glow and eventually, his name and in-game statistics will be shown. To avoid too much clutter on the screen, a gaze filter removes the highlights around players outside of the predefined filter radius.

Similar innovative technologies are already being adopted by the major sports leagues, which are constantly trying to improve their broadcasts. For instance, the NBA League Pass[11] provides live in-game players' statistics and other game scores simultaneously during the broadcasts of their games.

In addition, the Premier League, one of the most notable football leagues in the world, has partnered up with Genius Sports and it is now offering the 'Premier League Data Zone'[12][13], which offers enriched graphic overlays for live broadcasts and is already being used by major television broadcasters. Data Zone (see Figure 2.13) provides various features, such as player names, pitch maps, passing accuracy, shot speeds, and distance traveled.

Basketball, football, and cricket are some of the most-watched sports in the world[14]. Therefore, many technologies are being developed to improve player analysis, becoming increasingly more descriptive, like the previous examples. Despite that, smaller or less

[11] https://www.nba.com/watch/league-pass-stream (visited on January 2024).

[12] https://geniussports.com/newsroom/premier-league-productions-partners-with-genius-to-deliver-ground-breaking-premier-league-data-zone/ (visited on January 2024).

[13] https://www.youtube.com/watch?v=sh3RdZBKNfE (visited on January 2024).

[14] https://www.pledgesports.org/2017/03/top-10-most-watched-sports/ (last visited on June 2024).

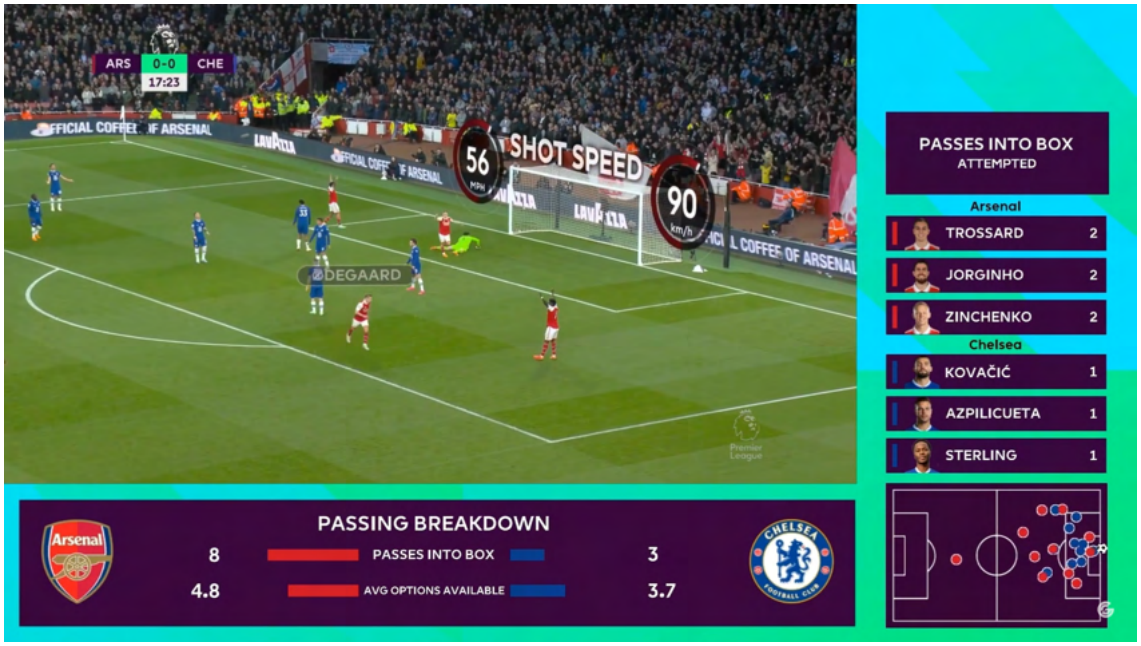


Figure 2.13: Premier League Data Zone[13].

popular sports do not have as much research around them and do not take as much advantage of such technologies. Nevertheless, they too could benefit from such CV techniques.

In particular, Brazilian jiu-jitsu is a grappling martial art with many complex positions that may lead to high levels of occlusion during the matches and create challenges to most CV techniques. Despite these challenges, such techniques could benefit athletes and coaches in training and opponent analysis. Moreover, these techniques could also improve referees' accuracy in match scoring and the overall viewers' experience. V. Hudovernik et al. [25] proposed a method that uses object detection, tracking, and pose estimation to overcome these challenges and improve the use of technology in jiu-jitsu.

Their proposed approach consists of three main stages:

1. **Person Detection and Pose Estimation**: At first, the system locates the athletes on the frame and then performs the keypoint detection to estimate the poses of both athletes.

2. **Combat position classification**: Based on the pose estimation, a combat position is assigned to each frame (e.g., takedown, open guard, and standing).

3. **Automatic Scoring**: Combat position classification enables the automatic scoring of the match.

Besides these three main features, a dataset was also produced. The dataset was composed of six sparring matches, each one focusing on a particular position, shot from three different angles, resulting in a total of 1248 annotated frames.

Given that the primary focus of this thesis is racket sports, it is also important to inquire about the applicability of such CV techniques in racket sports.

For example, EventAnchor [14] is a video analysis framework for racket sports that combines CV techniques and interactive annotation. Based on this framework, a system was developed for table tennis analysis.

The framework has three levels: (i) object level, where objects such as the ball, the player, and the court are recognized through CV techniques, (ii) event level, where event detection (e.g., stroke and ball bounce) is made based on the data from the previous level, (iii) context level, summarizes the information from the event level and can include information such as the type of stroke.

In order to better comprehend which events are relevant in racket sports (e.g., tennis, table tennis, and badminton) two studies were made.

The first study consisted of interviews with domain experts to identify relevant events in analysis and common challenges in data acquisition. These interviews were enlightening in understanding the main commonalities and differences between the analysis of these racket sports. For instance, the ball and player position were relevant in all three sports, while the ball speed was only crucial in tennis and badminton. The domain users considered that the video annotation tools would improve opponent analysis and player preparation for future matches.

In the second study, a survey was conducted to determine the most interesting events for general sports fans with backgrounds and interests that may differ. In the case of tennis, for instance, the three main events were ball position, ball speed, and serve effect. Only a few inquired had experience with video annotation tools. However, in general, they considered that scenarios such as finding important events or a specific rally could benefit from such tools.

With the introduction of a system that uses annotation, it is relevant to address some video annotation concepts and provide pertinent examples, since the solution developed during this thesis also incorporates annotation features.

2.6 Video Annotation

Annotation is a short explanation or note added to an image or a video, for instance. Annotations have always been part of most individuals' lives and date back to ancient times, having been used in areas such as literature and academics.

From a young age, we are taught the importance of taking notes in school to stay current with the contents of classes and to facilitate the later recall of those subjects. As adults, note-taking is also useful at work or while reading a book, for example. Nowadays, note-taking can be performed in notebooks and applications such as Microsoft OneNote[15],

[15] https://www.onenote.com/ (visited on January 2024).

Apple Notes[16] or Notion[17].

Another example where annotations offer benefits is in PDF files, allowing users to add comments and highlights to their documents. These annotations can be useful in many contexts, such as academic research, collaborative projects, or professional documentation.

Additionally, annotations can be applied in videos to better understand the visual content. Nevertheless, before getting into further and more complex examples, it is essential to describe some annotation types, their differences, and how they could improve video content visualization.

2.6.1 Annotation Types

There are several annotation types, each with some commonalities but also different characteristics that can be used to enhance video content visualization. It is pertinent to comprehend the variations between them to maximize the benefits of each annotation type:

- **Text**: This is a fundamental form of annotation and it is usually the most common one. It is intuitive to most individuals due to its extensive use in many areas and can be performed, for instance, through the use of a keyboard.
- **Speech**: Speech annotations require an input audio that is decoded into words displayed in the video. This is a similar technology to the ones used in virtual assistants such as Alexa [43] or Siri [13] and could be used to write annotations at designated moments (e.g., "Write (...) at a specified timestamp").
- **Ink Strokes**: The appearance and growth of mobile devices (e.g., smartphones and tablets) have facilitated writing on interfaces using touch. This is a comfortable and easy way to take notes in many distinct scenarios, such as in annotation systems and day-to-day life. Before the appearance of such technologies, this could also be performed using a standard mouse, which was not as practical.
- **Marks**: Mark annotations use images and symbols. These visual elements provide a fast understanding of the transmitted message since complex ideas can often be conveyed in a single image.
- **Hyperlink**: These annotations could be considered an interactive form of text annotations since the user clicks a given Uniform Resource Locator (URL) and is then redirected to another web page.
- **3D**: This type of annotation allows the user to place 3D model representations of virtual objects on top of a video frame, granting control over those objects. The user is able to move, resize, and rotate the model around the scene. The manipulation of

[16] https://www.icloud.com/notes (visited on January 2024).
[17] https://www.notion.so/ (visited on January 2024).

those models in real-time could improve the overall experience in presentations or collaborative situations.

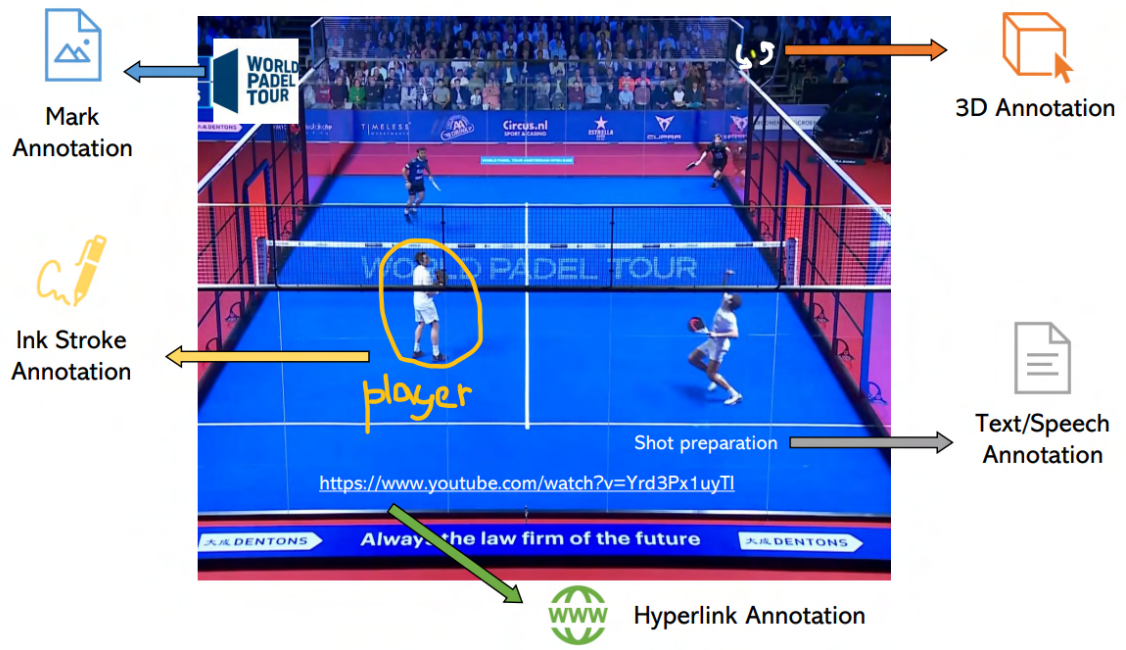


Figure 2.14: Various annotation types combined in one video.

It is often useful to combine some or even all of these annotation types in one single environment. Figure 2.14 displays an example of a racket sports video where distinct annotation types cohabit in the same video annotation tool.

2.6.2 Annotation Systems

Having discussed various types of annotations, it is now important to examine their practical applications and analyze examples of systems that use such technologies. Annotations can be applied in many different domains, for instance, education.

Microsoft Research Annotation System (MRAS) [22] is a conceptual multimedia annotation tool that was originally developed to help students and professors. Microsoft's goal was to develop a general-purpose User Interface (UI) to fit in a scenario where a student would watch his classes from home, using a video of the lecture, the associated slides, and notes referring to each slide. The student could read, answer, and ask questions about a specific topic. These questions would be associated with a timestamp and would pop up on the screen during that timestamp. The student could also participate in smaller discussions, also linked to the lecture, in a public or anonymous manner.

Despite being an easy-to-understand tool, Microsoft made several iterations of tests during the development of MRAS and concluded that this tool would not be adopted as they would have expected unless major changes were made for each deployment. This

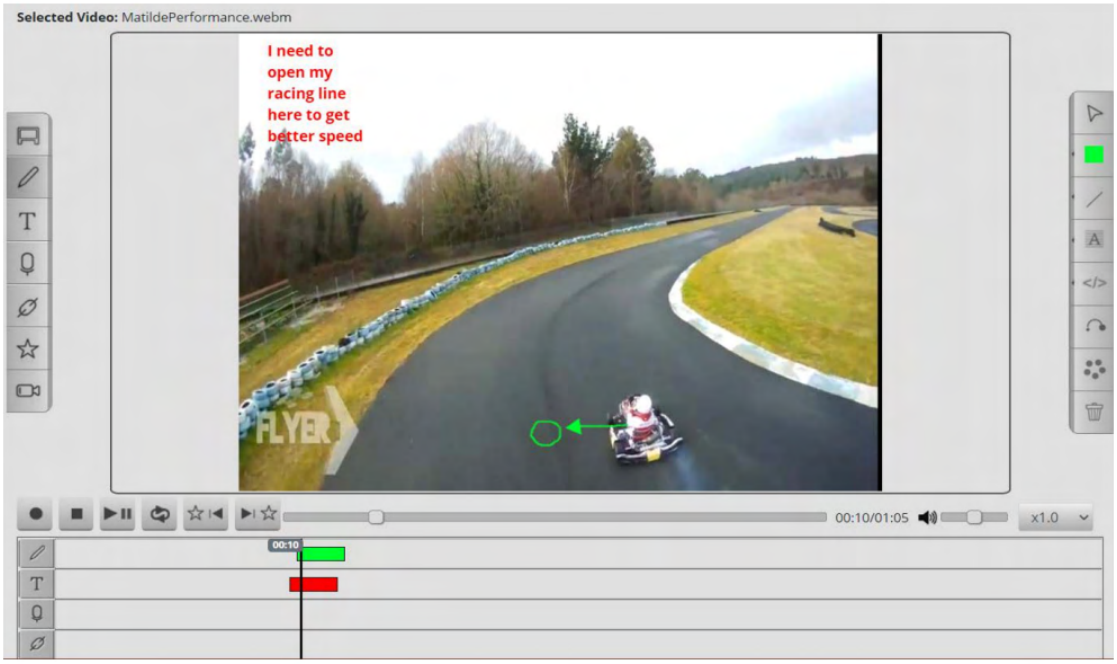


Figure 2.15: Example of an annotated video frame using MotionNotes. It displays both text and ink stroke annotations [49].

was due to the increased demand for new functionalities and overall system improvement. Microsoft believed that MRAS would have worked better 10 years prior when users might have been more prone to adapt to the original functionalities.

In the end, Microsoft concluded that the development of a framework and generic annotation platform would be best suited. Task-specific users could refine the application for their specific purpose with simple programming skills.

Even though this project was developed roughly 20 years ago, its ideas remain current and would be useful, especially during the COVID-19 pandemic, where almost every class, from primary school to college, was lectured online [24] [39].

MotionNotes [49] is a more recent example of a video annotation tool with the intent of helping teachers and students. To present a video during a class, the lecturers have to be close to their devices/desks to pause or rewind the video if any questions arise or if they want to comment about the video. However, teachers usually walk around the class to communicate better with their students. Doing so while presenting a video could cause unnatural pauses during their presentations. If they had to return to their devices to stop or rewind the video, that would potentially decrease the audience's focus.

MotionNotes (see Figure 2.15) tries to tackle this issue by offering a multimodal web video annotation tool that combines manual (e.g., touch, mouse, and keyboard) and speech annotation to help users complement their video visualization. The user can perform speech commands such as pausing the video and adding text annotations. While it was initially developed in the context of dance annotation analysis, this tool proved to be useful in other contexts, including education.

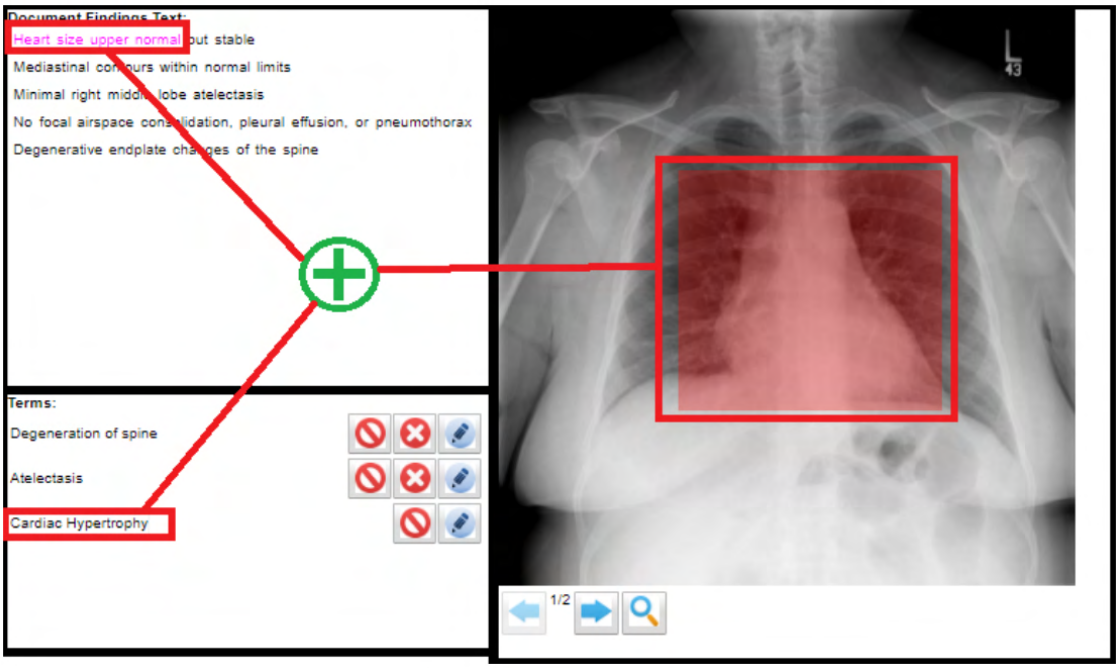


Figure 2.16: Example of a medical annotation tool [6].

Before proceeding, it is relevant to address the importance of user tests in scientific research for validating work. Despite not having previously highlighted their significance, these tests are a regular practice in the scientific field and can help researchers understand if their methodologies and conclusions are robust. Therefore, it is pertinent to introduce one example of such tests.

In the development of MotionNotes, two user studies were conducted. The first study had participants from two universities and focused on comparing user performance and satisfaction using the MotionNotes tool with and without speech recognition. Most users preferred to use both manual and speech annotation, even though their overall performance did not change much between both situations, and all of the inquired users would recommend the system to their friends and colleagues.

The second study tested how MotionNotes behaved during class, collecting feedback from the audience and the lecturer. With the help of a BlueTooth microphone, the teacher was able to watch the videos with the audience, control the video playback, and annotate it with his voice. Similarly to the first study, the overall feedback was positive, reinforcing that such a tool would be helpful in an academic context.

Besides education, the healthcare industry could also benefit from annotation systems. There are several approaches to be explored, such as handwritten annotations from medical professionals [21] and automatic annotation of radiograph images [42]. F. Buendía et al. [6] proposed an instructional annotation tool to help practitioners within the radiology domain (see Figure 2.16).

Regarding the annotation process, several approaches could be implemented. The first

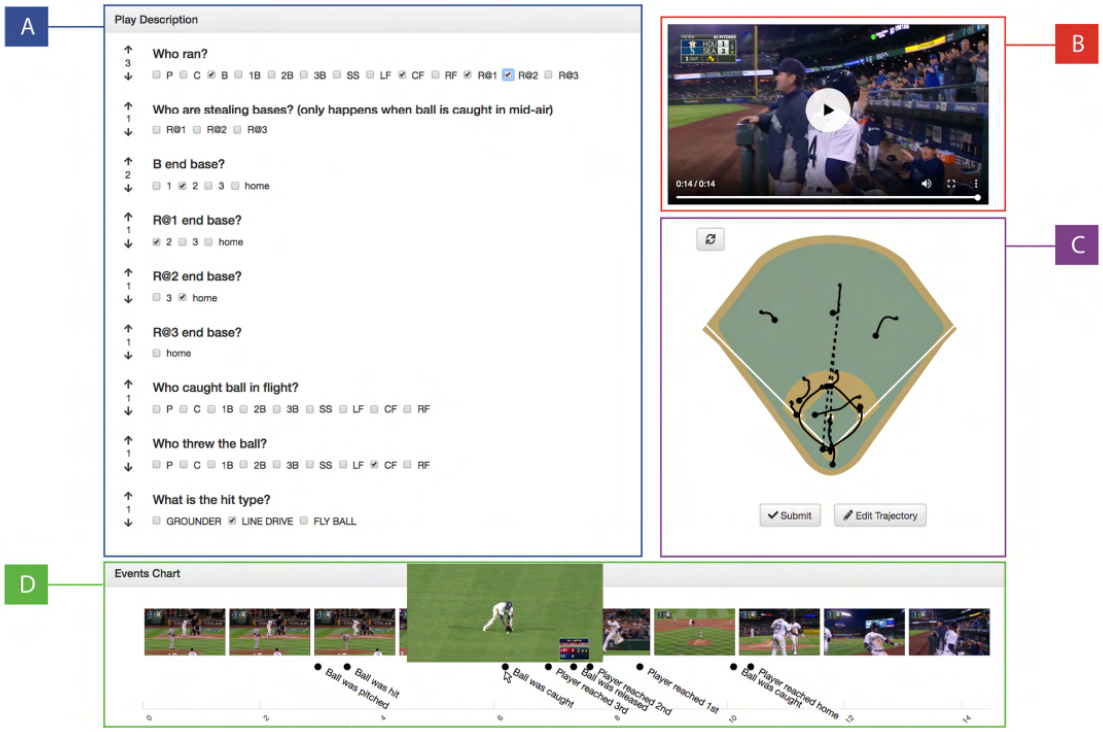


Figure 2.17: HistoryTracker [40].

one is the manual annotation of images by experts. Even though this approach guarantees high-quality annotations, it is also very time-consuming.

The second approach is automatic annotation. This type of annotation relieves the experts from the time-consuming task of manually annotating the images. Still, automatic annotations can frequently produce errors, especially in tasks with a high degree of specificity, namely radiography annotation.

The last approach is semi-automatic annotation, which combines manual and automatic annotation. At first, automatic annotation is performed and then reviewed by experts who can correct or complement the produced annotations.

In addition to the previous presented industries, sports analysis can also be enhanced with the use of such annotation tools. HistoryTracker [40] enables users to produce tracking data for baseball plays.

The importance of analytics in sports has increased tremendously in recent years. With the appearance of precise tracking technologies such as smartwatches, heart rate monitors, specialized sensors, and high-definition cameras, sports teams now have much more data available for analytical purposes.

However, implementing and maintaining such systems produces difficulties since they are expensive, may be affected by hard-to-control factors, and cannot be applied to older games where those technologies were not present. Manual annotation is a valid alternative to tackle those issues and produce reliable sports data, but it may burden the annotators due to its time-consuming aspect.

To address those challenges, HistoryTracker (see Figure 2.17) offers a tool that reduces the burden of manual annotation from scratch.

After a video is selected (see Figure 2.17 B), a summary of the play is collected by a series of easy-to-answer questions (see Figure 2.17 A), and those answers are used to compute an initial set of recommended annotations based on previously tracked similar plays (see Figure 2.17 C). If there is a need to refine the initial annotations, it is possible to edit them by performing manual annotations. Users can also drag and drop events to a specific timestamp of the video at which these events happened (Figure 2.17 D).

The conducted user studies displayed good feedback from the users, who, in general, were satisfied with the initial annotations of plays and considered it less time-consuming than manual annotation, since many of the common movements were already filled by HistoryTracker.

2.7 Summary

In summary, this chapter presented some concepts related to CV, followed by some CV algorithms and datasets, with an emphasis on their main differences. Following that, various systems were discussed to illustrate the usefulness of CV technologies in different areas. Initially, examples from various industries were showcased, followed by a more in-depth analysis of applications in sports, which is the main focus of this thesis.

Based on the systems and concepts discussed, this thesis incorporates CV techniques (e.g., object detection and object tracking algorithms) in sports, focusing on enhancing the analysis of padel games.

Lastly, the text covered some annotation concepts and included examples, since the developed work also incorporates annotation features. When combined with the CV technologies, these features improve the overall user experience and facilitate the analysis.

3

Design and Implementation

The previous chapter covered CV concepts and algorithms, applications that take advantage of CV, and annotation-related systems. This chapter details the decisions made during the development process, as well as the techniques that were implemented.

The chapter starts by detailing some aspects of the design of the web application, followed by a description of the prototype that was initially developed and the rationale behind the development of the web application and its various functionalities. It ends with an analysis of the limitations identified during implementation.

3.1 Design

The system was designed as a single-page application where users can watch their padel videos and analyze the players' performances. To improve player analysis and overall user experience, users benefit from the addition of CV techniques, such as object detection and tracking, as well as video annotation. These improvements transform the task of simply watching videos into a more engaging, practical, and helpful experience.

The system was designed to provide a straightforward and intuitive mechanism for analyzing padel matches. To navigate through the several functionalities, users only need to perform very few clicks. This design intends to provide a simple, effective, and appealing UI that reduces the time spent searching for each functionality and enables users to focus on their primary goal of analyzing padel videos (see Figure 3.1).

Before starting the development of the system, a preliminary study was conducted to obtain initial feedback about the system's concepts and their potential value in sports analysis. Essentially, a prototype was created without the need for any coding, and a complete description of this process is further detailed in Section 3.2.1. The initial design of the system's UI was developed and the operation of four key functionalities was outlined, alongside some other basic features:

- Allow users to upload padel videos into different categories.
- Allow users to watch their videos without any additional features being enabled.

Figure 3.1: Final version of the system's User Interface.

- Allow users to enhance the video visualization with object detection features.
- Allow users to highlight/focus on specific players using object tracking techniques.
- Allow users to analyze automatically generated statistical reports.
- Allow users to search for specific game events and their timestamps.

Following this, an initial user study was conducted to evaluate the prototype's features (see Section 4.1). After reviewing these findings and refining certain aspects, it was determined that the system would offer the following features:

- Allow users to upload padel videos into different categories.
- Allow users to watch their videos without any additional features being enabled.
- Allow users to enhance the video visualization with object detection features.
- Allow users to highlight/focus on specific players using object tracking techniques.
- Allow users to edit the player names to facilitate the navigation between players.
- Allow users to better understand each player's movements and tendencies by providing heatmap and trajectory trace features.

3.2 Implementation

While the previous section provided a broad overview of the system's features, an in-depth description of the development process will now be presented. Several technologies and different approaches were evaluated to achieve the desired development of the system. Therefore, the following sections will present the entire process behind the system's implementation and the various alternatives considered, starting by discussing the initial prototype and how it evolved into the final version of the system.

3.2.1 Initial Prototype

During the initial stages of this research, an interactive web application prototype was developed using Figma[1]. Its primary objective was to serve as a preliminary version of the system, enabling early feedback before the implementation, in order to identify potential issues and improvements. This enabled a direct assessment of how effectively current theoretical advancements in sports analytics can be applied to enhance the understanding of game dynamics and player performance.

The prototype was centered around an intuitive user experience, starting with login and account management. Upon logging in, users were welcomed into a private working area, ensuring a personalized and secure environment for their analytical activities.

After successful login, users were directed to their main working area. This area was designed to provide immediate access to essential features such as account details and video management.

The video management area (see Figure 3.2) enables users to upload new videos and organize their existing video library via a window popup interface. This functionality was designed for simplicity, enabling the efficient management of a large number of videos, thereby supporting extensive, long-term analytical endeavors.

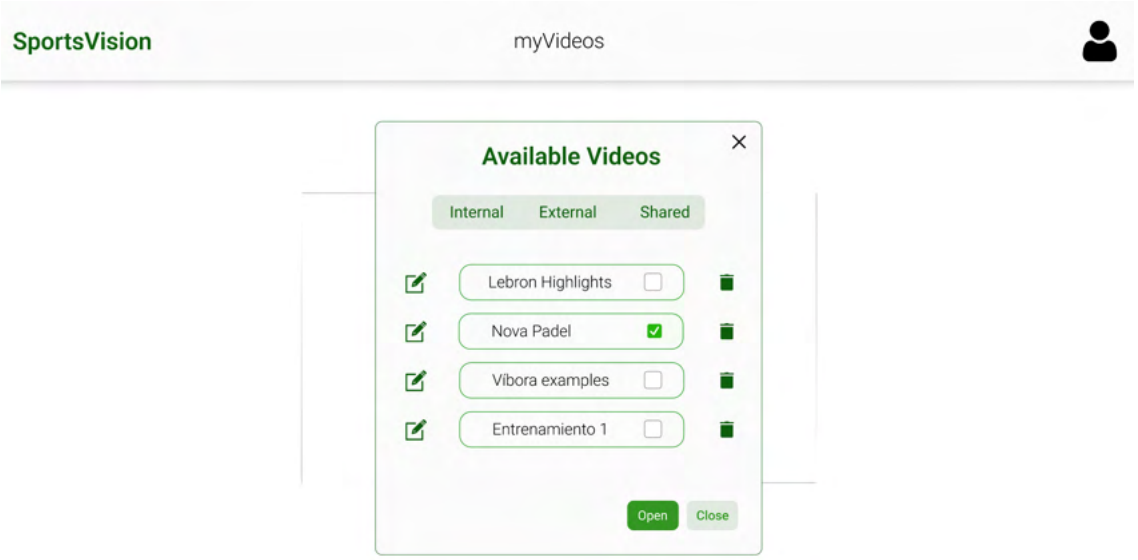


Figure 3.2: Videos' management area.

The most important part of the user interface was the video player, strategically positioned at the center of the screen. It was here where users could play their selected videos, engaging directly with the object recognition features and analysis tools the prototype presented.

The integration of automatic object detection and tracking in this video content analysis tool opened multiple opportunities for in-depth analysis in racket sports. The prototype

[1] https://www.figma.com/ (visited on January 2024).

Figure 3.3: Itemized search.

Figure 3.4: Automated statistical reporting.

had four main functionalities: itemized search, automated statistical reporting, individual player focus, and enhanced game visualization. To choose one of these functionalities, users would select one of the four options from a pie menu on the left side of the video player.

The prototype introduced a refined search capability, allowing users to query specific game events such as winners, unforced errors, smashes, and successful volleys. This feature would significantly reduce the time spent on manual video analysis, providing a quick and efficient way to access moments of interest in a match. When a search was conducted, the results were displayed in a lateral menu on the right side of the screen (see Figure 3.3).

Another feature was the automatic generation of statistical reports. Users could specify the aspects of the game they wished to analyze, such as player performance metrics or game event frequencies. The tool would then process this input to generate a report, presenting insights about the chosen parameters. This functionality would aid in objectively assessing players' performances and game strategies, providing data for coaches and analysts to work with (see Figure 3.4).

The prototype also included a feature for tracking specific players or items within the video. Users could highlight a particular player and follow their movements throughout

Figure 3.5: Individual player highlight.

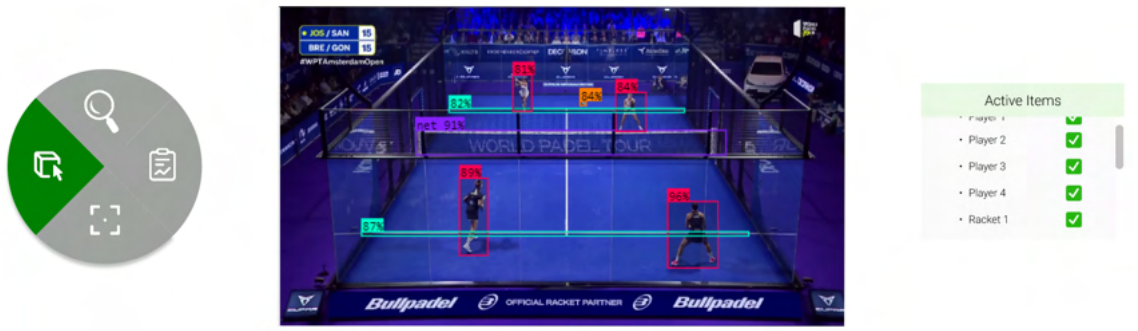


Figure 3.6: Enhanced game visualization.

the game. This tracking would enable a detailed analysis of an individual player's movements and a focused perspective on their contribution to the game (see Figure 3.5).

In addition to player tracking, the tool could automatically highlight all objects of interest in the video, such as the ball, the net, and the players. By visually distinguishing these elements, users could watch the game with an augmented layer of information. This enhanced visualization aided in understanding the spatial dynamics of the game and the interactions between different elements, further enriching the analysis (see Figure 3.6). When developing the prototype and its functionalities, the idea was that this would be the first functionality to be implemented and it would serve as a basis for all the other functionalities.

To better evaluate the prototype's usefulness and pinpoint potential improvements, a series of user tests were conducted, which will be detailed further in Section 4.1 of Chapter 4. As a result, this prototype motivated the development of a scientific paper, which was submitted to a conference dedicated to research in this area.

3.2.2 System Overview

After the prototype was completely developed and tested, the focus shifted to the development of the application. In terms of visual design, there were minimal modifications in the UI when compared with the prototype. The main page remained largely unchanged and the most significant change was the transition from a pie menu to a column menu,

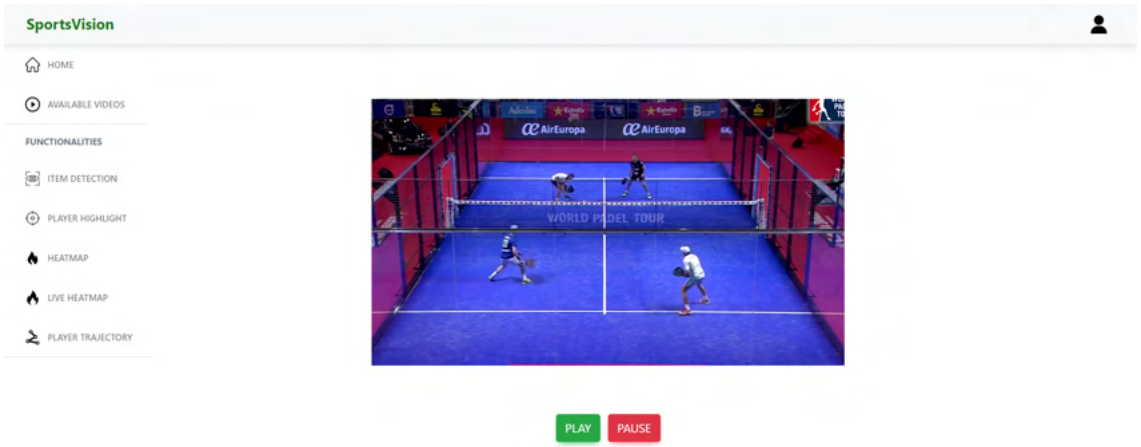


Figure 3.7: System's interface displaying a selected video.

intending to provide a more natural way to select the desired functionality. Figure 3.7 illustrates the system when a video was already selected.

Regarding the technical implementation, the system consists of a single-page web application with a client-server architecture. The client side is built with HTML5, CSS3, and JavaScript (ES6), while the server components are developed using the Node.js framework. For the client to communicate with the server, the implementation follows a simple approach: the client sends a request to the server via an intermediate layer. This layer builds the request and forwards it to the node front, where the request is then processed and a response is sent back to the client.

The Node.js layer accommodates several endpoints that enable communication with the client (Table 3.1). Specifically, there are some POST, GET, and DELETE operations that represent the Create, Read, Update, and Delete (CRUD) features for managing the available videos. Moreover, it contains two GET endpoints that handle the object detection and tracking functionalities. Besides these, there are two other endpoints: a POST endpoint for when the user decides to change a player's name (e.g., from PLAYER 1 to John), and a GET endpoint for loading these saved player names onto the page.

Table 3.1: API endpoints available for clients.

Type	Endpoint	Description
GET	/getVideo/{filename}	Gets the selected video
GET	/getVideoList	Lists all available videos
DELETE	/deleteVideo/{folder}/{filename}	Deletes the specified video
POST	/renameVideo/{folder}/{oldFilename}/{newFilename}	Renames the specified video file
POST	/uploadVideo/{folder}	Uploads a new video to the specified folder
GET	/getDetection/{videoName}	Gets detection data for the video

GET	/getTracking/{videoName}	Gets tracking data for the video
POST	/savePlayerName/{videoName}	Saves the name of a player in the specified video
GET	/getPlayerNames/{videoName}	Retrieves the names of the players in the specified video

To accommodate the object detection and tracking features, the system's architecture incorporates two additional layers: a Python-based server and a JSON file-based database. To perform these CV features, YOLO was considered the best available option (the rationale for this choice is further detailed in Section 3.2.3). Since YOLO was developed in Python and designed for ease of use across various Python environments, using a Python server was considered the most effective and straightforward option.

Unlike the Node.js server, this Python server only contains two endpoints: `GET /computeDetection` for handling object detection and `GET /computeTracking` for managing object tracking. These endpoints are responsible for processing requests that are forwarded from the Node.js layer, as the object detection and tracking computations are executed in the Python layer rather than in Node.js.

The JSON file-based database stores the results from the computations of object detection and tracking. JSON's simplicity, human-readability and compatibility with the existing JSON format in client-server interactions facilitate effective data management. Figure 3.8 illustrates the overall architecture of the system.

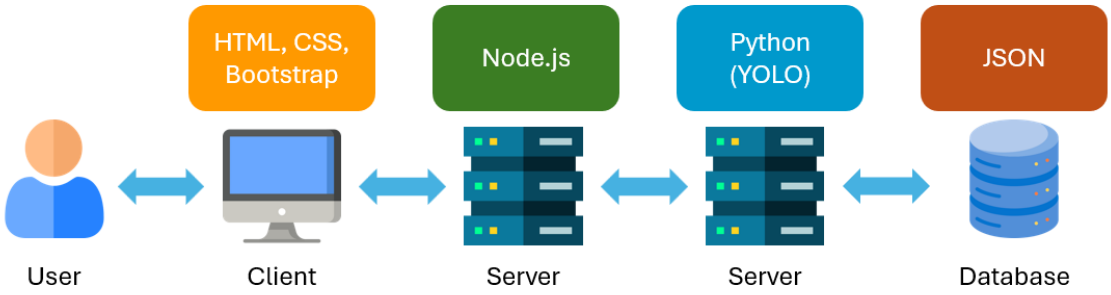


Figure 3.8: General overview of the system's architecture.

3.2.3 Object Detection

Having discussed the system overview, it is now relevant to address the main topics of this work: object detection and tracking. The following sections will discuss the decision-making process behind the chosen algorithms, their implementation, and the features that take advantage of these CV concepts.

During the early stages of development, following the creation and testing of the prototype and the setup of a basic web page to facilitate user video uploads, the next step involved integrating object detection into the application.

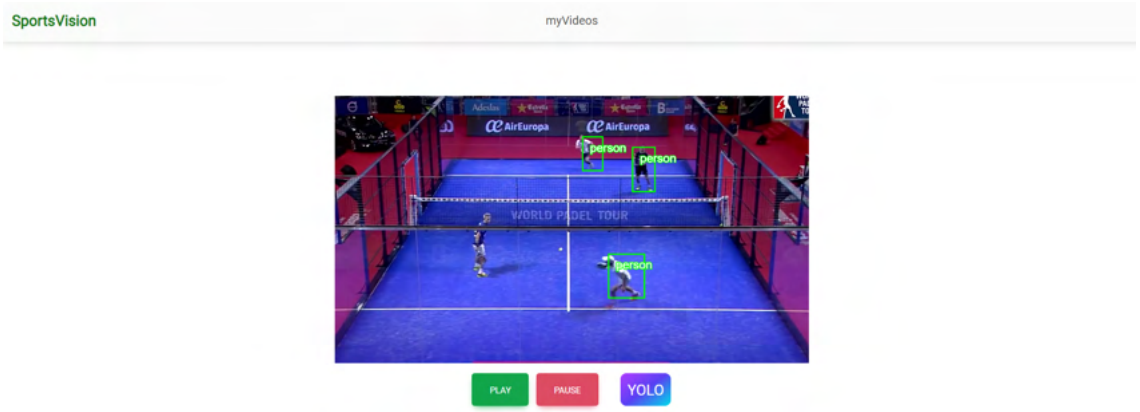


Figure 3.9: Initial implementation of the object detection using COCO-SSD.

The first approach consisted of using the ml5.js object detection feature with the COCO-SSD model[2]. This model uses the SSD algorithm to detect objects defined in the MS COCO dataset. Within this library, the results are structured such that, for each frame, information about each detected object is returned, such as top-left coordinates, width and height, confidence, and the object's label.

To draw the bounding boxes on the screen, the p5.js library[3] was used. This library provides an easy and effective approach to drawing shapes and text on the screen, making it well-suited for our purpose. The initial step consisted of placing a p5 canvas on top of the video player, allowing the drawing operations directly onto the video feed. Following that, it was necessary to adjust the coordinates returned by the detector to match the dimensions of the canvas, as these dimensions may vary considerably from those in the original video. The final step involved drawing the bounding boxes and their labels for each detected object.

Since the detection is performed on video instead of a static image, it is necessary to update the bounding boxes for each frame of the video. To achieve this, the `requestAnimationFrame()` function was used to repeat this process every time a new frame is displayed on the screen. When the video was paused or ended, this loop was canceled with `cancelAnimationFrame()`. The `requestAnimationFrame()` function ensured smooth and accurate drawing of the bounding boxes. If, for instance, a regular while loop was used, the drawing could potentially occur much faster than the video, resulting in misalignment with the current frame of the video.

Even though this algorithm was not chosen as the final algorithm for performing object detection, it was important to visualize and evaluate the initial stage of this feature, including its accuracy and computation time. Figure 3.9 shows the early stages of the main page, where COCO-SSD was being used for object detection.

After the initial experiments with object detection, the next step involved integrating

[2] https://github.com/ml5js/ml5-library/blob/main/docs/reference/object-detector.md (visited on July 2024).

[3] https://p5js.org/ (last visited on July 2024).

and evaluating an alternative algorithm, YOLOv8[4], known for its high speed, while also maintaining accuracy equivalent to other state-of-the-art solutions. Based on the analysis presented in Chapter 2, YOLO was considered the best state-of-the-art option for our purpose due to its high speed, accuracy, versatility, and popularity across different fields, as many of the studied related works used YOLO as their preferred algorithm. Furthermore, YOLOv8 includes a tracking feature, simplifying its integration into the application without requiring much additional coding.

As previously mentioned, the YOLO computations are made in a Python server. This layer uses the Ultralytics library to compute the object detection results as shown in the code snippet in Listing 3.1 below:

Listing 3.1: Loading a pre-trained YOLO model and running predictions

```
from ultralytics import YOLO

model = YOLO("yolov8n.pt")  # load a pre-trained model
results = model.predict(source=<filePath>, conf=<conf_lvl>,
    save=False)
```

The `results` variable stores the results computed by YOLO in the `model.predict()` function, where `source` is the path of the image or video where the detection will be performed, `conf` is the minimum confidence threshold of the bounding boxes (which aims to help reduce false positives), and `save` enables saving of the annotated images or videos to a file. Saving the annotated images is disabled since it is preferred to freely annotate the images in the JavaScript client, without being restricted to the YOLO format.

After computing the results, these are converted to JSON format before being sent as a response to the Node.js layer. The JSON data follows the structure show in Listing 3.2:

Listing 3.2: JSON structure used for storing bounding box data.

```
{
    "frames": [
        {
            "frame": <frame_number>,
            "boxes": [
                {
                    "x": <x_coordinate>,
                    "y": <y_coordinate>,
                    "width": <box_width>,
                    "height": <box_height>,
                    "confidence": <confidence_score>,
                    "class": "<class_label>"
                }
```

[4] https://github.com/ultralytics/ultralytics (visited on July 2024).

```
                ]
            }
        ]
    }
```

To accommodate YOLO in the JavaScript client layer, some changes needed to be implemented when compared to the COCO-SSD implementation. Unlike COCO-SSD, where the x and y coordinates represent the top-left point of the bounding box, in YOLO, the x and y values represent the center of the bounding box. Therefore, there were some changes to the process of adjusting the coordinates to match the dimensions of the canvas.

Additionally, `requestAnimationFrame()` was no longer a valid option since the frame rates used in YOLO and the browser were different. When using this function, the drawings of the bounding boxes would be performed much faster than the video, causing the video and the drawings to become out of sync. To address this, an estimation of the current video frame is computed based on the number of total frames returned by YOLO, and the drawing of new bounding boxes is only performed when a new frame of the video is displayed on the screen. The method for calculating the estimate is as follows:

$$\text{currentFrame} = \left\lfloor \frac{\text{currentTime}}{\text{duration}} \times \text{totalVideoFrames} \right\rfloor$$

where currentTime is the current playback position in seconds, duration is the total length of the video in seconds, and totalVideoFrames is the total number of frames in the video. The floor function ensures that the computed frame index is an integer value.

After performing object detection in several videos, a problem became evident: the computation time of YOLO. Even though the algorithm is extremely fast, since the drawing of the bounding boxes is only performed when all frames are computed and the Python server returns the data to the client, the user could be waiting for a while during these computations. To address this issue, the object detection data is stored in a JSON file after the first time a video is computed. Subsequently, there is no need to recompute the detection again, as solely the JSON file is sent to the client, resulting in significantly improved response times.

3.2.4 Custom Dataset and Model Training

While using the pre-trained models from COCO-SSD and YOLO provided beneficial insights about initial developments and their computation times, both models presented similar problems regarding detecting the game elements of a padel game. As shown in Figures 3.9 and 3.10, both models were able to identify the players as 'person' but struggled to identify the other game elements (e.g., net, ball, rackets, and serve lines).

Although the rackets and the ball were detected in some frames, the detections were abrupt and inconsistent throughout the video, making the detection unreliable and restricting the possibility of deriving other functionalities from the object detection.

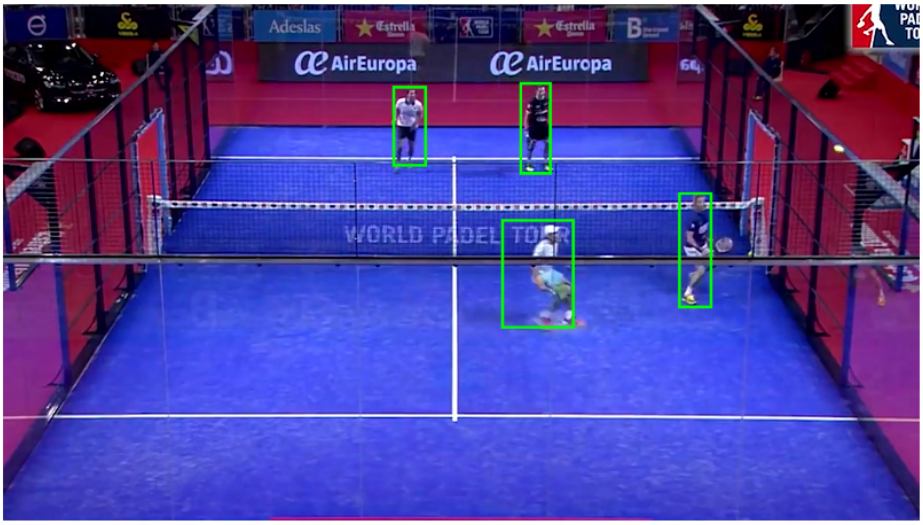


Figure 3.10: Object detection using one of YOLO's pre-trained models (YOLOv8n).

Therefore, it became evident that a new custom dataset and further model training were necessary to fulfill the needs of this system and accurately detect the desired game elements.

To create the dataset, the Roboflow software was used. Roboflow[5] is a state-of-the-art platform designed to facilitate the creation, management, and deployment of custom datasets and computer vision models. It provides various options for data annotation, dataset augmentation, and model training, allowing for a robust and accurate process of building a new machine-learning model.

The developed dataset is composed of images gathered from various professional men's and women's padel games from the World Padel Tour and Premier Padel competitions. This choice was based on the fact that, in televised padel games, the camera placement is consistent across different stadiums. Regardless of the match location, the primary camera is usually placed in a central, elevated position behind one of the courts, providing a clear perspective of the game and capturing all game elements. For that reason, only images from the primary camera were used and other frames from different cameras, such as close-up shots and replays, were discarded.

This dataset comprises 1545 manually annotated images, added over several iterations, with each image labeled to identify one of five classes: ball, net, player, racket, and serve line (see Figure 3.11 for an example of the manual annotation of images). These images are divided into three sets: 70% are allocated to the training set, which is used to train the model; 20% are designated for the validation set, which is used to fine-tune and validate the model; and 10% are assigned to the test set, which is used to evaluate the model's performance.

To decrease the training time and increase performance, all images are preprocessed and suffer two transformations. The first transformation, auto-orientation, ensures that

[5] https://roboflow.com/ (visited on August 2024).

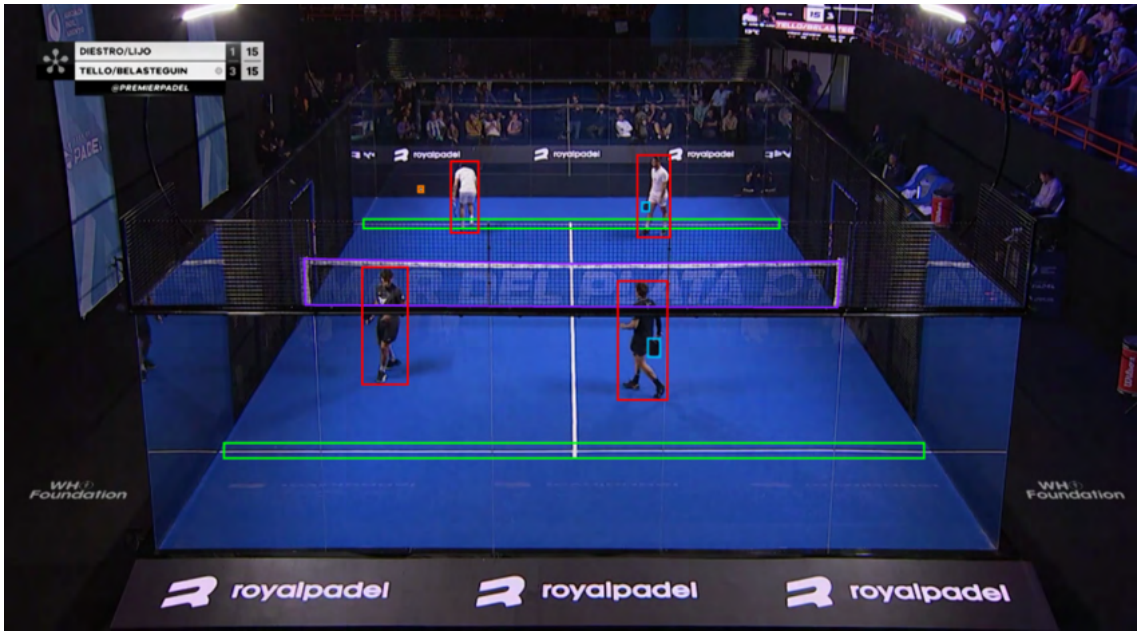


Figure 3.11: Example of manual annotation of images in custom dataset.

each image is correctly oriented regardless of how it was originally captured. The second transformation, resizing, makes sure that all images have the same dimensions, allowing the model to learn from a consistent input size.

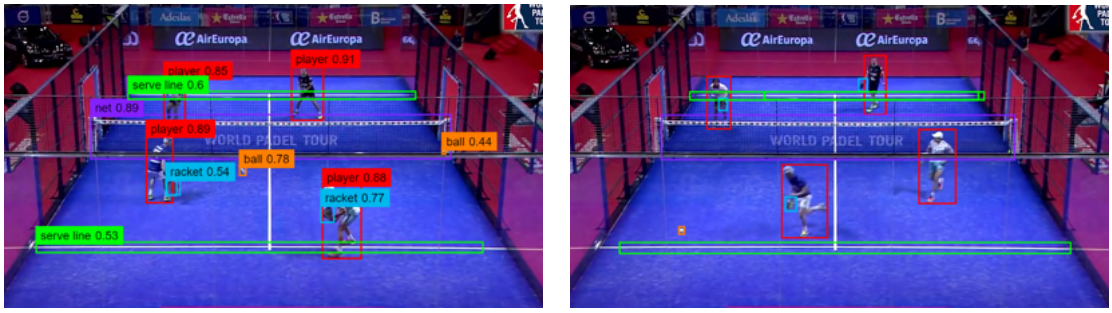
Moreover, to increase the number of images and improve the model's accuracy, new training images were created by generating augmented versions of each image in the training set. These augmentations include horizontal flips, and adjustments to hue, saturation, and brightness. Other augmentations, such as rotation, blur, and noise, were tested but discarded in the final version of the dataset because they negatively impacted the accuracy of the model. With the addition of the augmented images, the dataset increased to over 3700 images.

After each iteration of the annotation process, the next step was to train YOLO's model with a new version of the custom dataset. The model was trained and validated using an interactive notebook provided by Ultralytics on their GitHub page, hosted in Google Colab[6]. This notebook provides a comprehensive guide on training, validating and deploying newly trained YOLO models on custom datasets, facilitating the learning process and implementation of these advanced YOLOv8 features.

Lastly, the best weights of the newly trained model were downloaded and added to the source code, replacing the previously used pre-trained model YOLOv8n. Every time the source code was updated with a new model version, its accuracy was evaluated to determine if additional image annotation was necessary.

The process of manual annotation and model training consisted of 11 versions. New images were added after each iteration and various augmentations were tested to identify

[6] https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb (visited on August 2024).



(a) With labels. (b) Without labels.

Figure 3.12: Comparison of object detection results with and without displayed labels.

the most effective ones for enhancing the model.

This phase was extremely time-consuming, occupying a significant portion of the overall development time. This is due to the arduous task of manually annotating several hundreds of images for each model iteration, which was both a laborious and lengthy process.

Moreover, the model's training on the custom dataset was also slow, with each run taking several hours to complete. Initially, the model was trained for 100 epochs, but after some dataset updates, the number of epochs was adjusted to 80. This change slightly reduced the training time and did not have a significant impact on the model's overall accuracy since the accuracy gains beyond 80 epochs were minimal.

According to data retrieved from the Roboflow page, the final version of the model achieved 89.9% mAP, 90.3% precision, which measures how often the model's predictions are correct, and 87.9% recall, which represents what percentage of relevant labels were successfully identified. This was a significant improvement compared to the first trained version of the model, which only had 200 annotated images and achieved 74.4% mAP, 79.7% precision, and 73.2% recall.

Beyond the numerical analysis of the various metrics used to evaluate the performance of the custom model, it is important to mention how it performs in visual terms. The detection of the players was the one that visually performed the best, being highly consistent throughout all video frames. The detection of the net and serve lines also presented good results, though it was slightly harder to achieve since players constantly moved across the court and partially occluded parts of the net and serve lines.

The rackets were occluded by the players in many frames, increasing the difficulty of its detection. Despite that, while missing some frames, the detection of the rackets was also generally effective.

Since the ball is the smallest game element and travels extremely fast, it was expected that this would be the hardest object to detect. Even when manually annotating the custom dataset, some frames presented challenges in marking the ball's position, as it was sometimes just a blur with very little clarity. Nevertheless, while sometimes inconsistent, the ball detection provided a good visual experience, remaining fairly accurate in most

frames.

An overview of the final version of the object detection feature is shown in Figure 3.12. Figure 3.12a illustrates the usual representation of object detection, which features a rectangle drawn around the detected object, a label to identify the object's class, and the confidence level of the detection. Despite being highly detailed and informative regarding the elements of a padel game, this version of the object detection feature can be somewhat overwhelming to the users, who end up with extensive information on their screens.

Therefore, there is a button to remove both the labels and the confidence level of the objects, providing a cleaner version of this feature (see Figure 3.12b). Consequently, users can choose between a more detailed version, which includes all labels and confidence levels, or a simpler version with these elements disabled to accommodate different user needs and preferences.

3.2.5 Object Tracking

Upon completing the development of the custom dataset, model training, and object detection feature, the focus shifted to working on the object tracking functionality. As previously mentioned, YOLOv8 introduced other features besides object detection, including object tracking, so its integration into the system was fairly simple.

YOLOv8 supports two different tracking algorithms: ByteTrack and BoT-SORT. As described in Section 2.4, BoT-SORT enhances the ByteTrack algorithm, which is why it was considered the best option to perform object tracking. This is also the default algorithm used in YOLOv8. Similar to object detection, object tracking is performed on the Python server for the entire video, and the resulting data is sent to the client via an HTTP response.

Instead of using the `predict()` function, the `track()` function was used. This YOLOv8 function has the same parameters as the object detection one and the value of those parameters remained untouched. However, another parameter was added regarding the classes included in the tracking process.

By default, all classes in a model are included in YOLO's `track()` function but, since the players are the game's main focus, using only the players in the tracking function was considered a better alternative. Apart from that, tracking game elements such as the net and serve lines would not be justified since these are static elements throughout the video. One could make a case for tracking both the rackets and the ball. However, it was considered more prudent to start with only the players.

Furthermore, detecting only the players was expected to make the algorithm's runtime faster, as it involves tracking just one class instead of five. Nevertheless, after evaluating the runtime in each case, the time difference between tracking one or five classes was not substantial.

A small performance test was conducted for three videos, each with different time durations (e.g., 15 seconds with 463 frames, 16 seconds with 501 frames, and 1 minute and 16 seconds with 1840 frames) and the tracking was performed five times on each video.

				Runs (in seconds)					
Video	Duration (in seconds)	Frames	Tracking Type	1	2	3	4	5	Average
1	15	463	players-only	36.96	35.55	36.61	40.98	39.05	37.83
			all classes	42	37.49	34.62	35.54	40.53	38.03

				Runs (in seconds)					
Video	Duration (in seconds)	Frames	Tracking Type	1	2	3	4	5	Average
2	16	501	players-only	43.37	37.84	36.28	43.46	46.09	41.4
			all classes	39.5	35.52	44	39.25	43.54	40.36

				Runs (in seconds)					
Video	Duration (in seconds)	Frames	Tracking Type	1	2	3	4	5	Average
3	76	1840	players-only	101.18	89.28	100.54	89.29	91.03	94.26
			all classes	101.84	92.17	102.14	92.11	91.76	96

Figure 3.13: Benchmark results for object tracking across different videos.

Those videos are relatively small since they display individual padel rallies, which are the videos intended for use in video analysis within this system. Figure 3.13 summarizes the benchmark results for object tracking.

Although the first and third videos differ by approximately 1 minute, the results were similar. In both cases, performing the object tracking with all classes was slightly slower than tracking only the players. For the first video, the difference was less than a second, while for the second video, it was approximately two seconds. Nonetheless, the second video, which has a duration similar to the first, showed a one-second improvement in tracking all classes compared to tracking only the players.

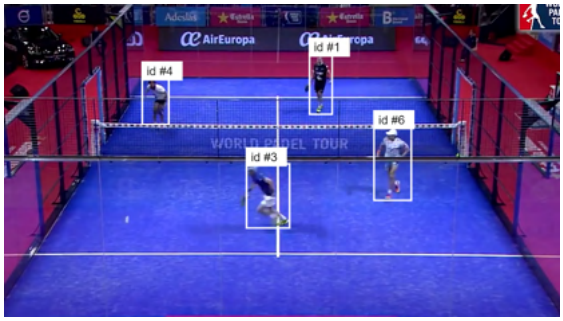
These results show that, despite the initial assumptions, the execution time of YOLO's object tracking was not influenced by performing it in one or five classes, since the differences between those executions were marginal. There is no guarantee that this issue would not arise with extremely long videos (e.g., lasting several hours) or if the difference in the number of classes was significant, in the hundreds or thousands. However, in this test case, the execution time remained relatively constant, and the use of several videos during the implementation phase also supports these findings.

After performing the object tracking in several videos, a problem became evident: when the tracker fails to track a player for some frames (e.g., due to player intersections or a player leaving the court), a new ID is assigned when the player is located again (see Figure 3.14). This is because the YOLO version used does not support re-identification. Re-identification allows the algorithm to recognize and track objects even if they are missing in several consecutive frames, instead of allocating new IDs each time an object is temporarily lost.

Consequently, there were two implementation approaches considered. The first one involved using a different version of BoT-SORT outside the Ultralytics Python library, where the re-identification option was available. The second approach was to continue using the Ultralytics library while developing an alternative method for re-identifying

(a) Before failing to track the player with ID 2.

(b) Failed tracking the player with ID 2.



(c) Assignment of a new ID.

Figure 3.14: Overview of player tracking failure and ID reassignment.

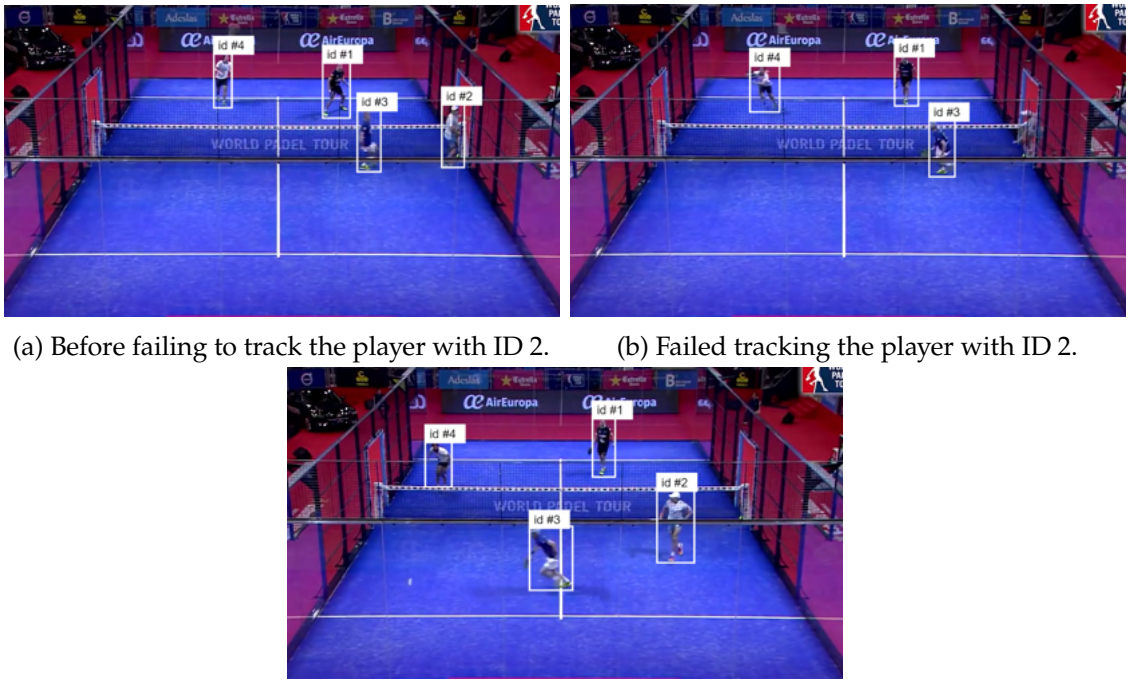
missing objects.

Despite the increased complexity, the second option was selected because it provided a more rigorous challenge and offered better insights into how re-identification processes are performed. This approach also allowed for the continued use of the Ultralytics library, which was already integrated into the source code, thereby keeping the previously written code unchanged and simple.

The first metric developed to perform the re-identification was relatively simple. Since a padel game is composed of four players, if there is a video frame with exactly one ID that exceeds 4, it is adjusted to the missing ID within the range of 1 to 4. This metric performed effectively across all tested videos and was included in the final version of the re-identification. In the final version, even if other metrics are being used throughout the re-identification process, this metric is always utilized when there is exactly one missing ID.

The second metric used was the distance of the players across frames. This process involved saving the last known coordinates from each player for every frame. If more than one ID exceeded 4, the last known coordinates of the missing IDs were used to compute the distance between them and the players with IDs higher than 4. Subsequently, each missing ID was then assigned to the player whose distance to the last known coordinates of that ID was the smallest.

While this approach was straightforward and initially seemed promising, it did not perform as well as anticipated. Despite achieving good results most of the time, this method would fail when two players overlapped during the video, causing their IDs to



(a) Before failing to track the player with ID 2.

(b) Failed tracking the player with ID 2.

(c) ID 2 remains unchanged.

Figure 3.15: Overview of player tracking and ID continuity using the proposed re-identification methods.

be swapped. This issue highlighted the need to replace this method with a more effective one to improve the accuracy in such scenarios.

Following some research, the Kalman filter emerged as a promising alternative. As highlighted in Section 2.4, the Kalman filter is a method commonly used in tracking algorithms. This filter relies on data observed over time, containing noise and inaccuracies, to estimate the coordinates of bounding boxes in subsequent frames. BoT-SORT already takes advantage of the Kalman filter, but it is exclusively used for tracking and not re-identification. However, it was considered that the Kalman filter could also be useful for re-identification, leading to the exploration of this solution.

After the initialization of several parameters (e.g., the initial state for position and velocity, state transition matrix, measurement function, covariance matrix, measurement noise, and process noise), the Kalman filter is composed of two iterative stages: prediction and update. The `predict()` function estimates the coordinates of a bounding box in the next frame based on its previous coordinates and how they evolve over time. The `update()` function adjusts the estimated coordinates by using the observed value during the current video frame. The updated value is then used in the next iteration of the `predict()` function.

Four different Kalman filters are used, one for each player, with their coordinates updated in every frame. When a frame has more than one ID that exceeds 4, the predicted values of the missing IDs are used to compute the distance between them and the players with IDs higher than 4, rather than relying on the last known coordinates.

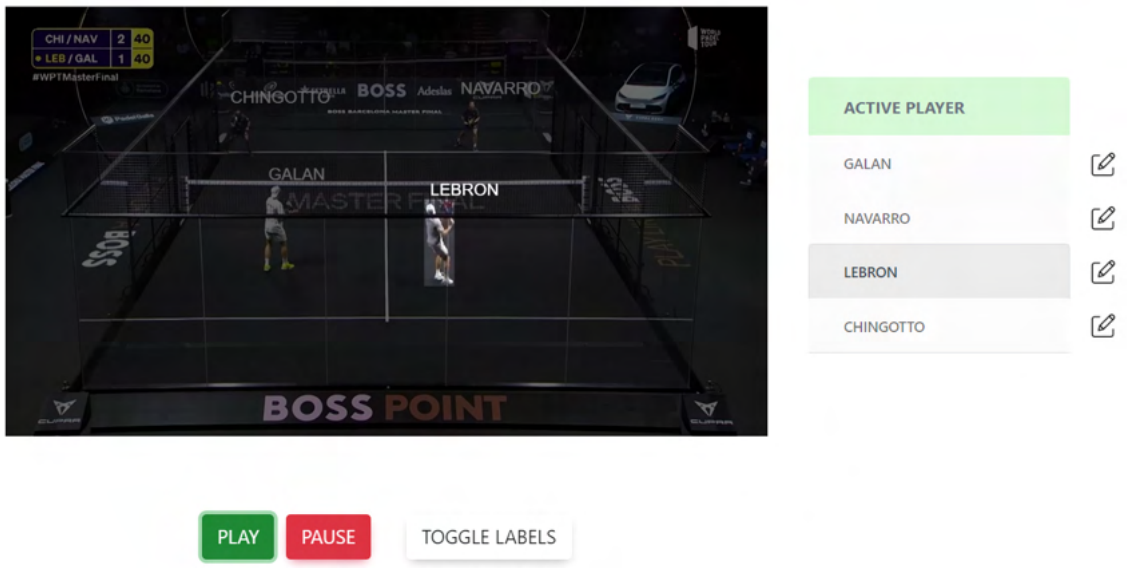


Figure 3.16: Overview of the player highlight functionality.

To further improve the accuracy of this method, which was still facing some problems during player occlusion, the predicted values are only used when a new ID exceeding 4 appears for the first time. The predicted values are used to identify the correct player ID within the range of 1 to 4, and this value is then saved for future frames. For instance, if ID 10 appears for the first time, the predicted values are used to associate it with one of the missing ones (e.g., ID 3). Once this association is established, ID 10 will be replaced by ID 3 in subsequent frames, removing the need to recompute the distance each time.

While there are more complex methods to perform re-identification, such as facial recognition, clothing, and body shape analysis, the proposed method, alongside the improved accuracy of the custom-trained CV model, achieved impressive results across all tested videos and successfully addressed the previously mentioned issues. Figure 3.15 shows the same example previously shown in Figure 3.14, now functioning correctly with the proposed re-identification methods.

Although player tracking is different from object detection and, as described throughout this last chapter, presented several challenges to ensure its implementation was as accurate as possible, visually it may not differ much from object detection and could become less appealing to users (e.g., athletes, coaches, and analysts). Therefore, the next step was to improve this functionality by making it more interesting to the users and more oriented towards player analysis.

Consequently, after receiving the data from the Python server, the client adjusts the coordinates of the bounding boxes to match the dimensions of the canvas, similar to the process used in the object detection functionality. Nevertheless, instead of displaying the class name, confidence level, and the corresponding ID of the player, a different approach is adopted: the selected player (by default the player with ID 1) is highlighted by drawing a contour around that player and darkening the rest of the video.

This approach allows users to focus their analysis on a specific player, allowing for a better understanding of the player's playstyle and the errors made during a particular rally.

Besides highlighting the selected player, the name of each player is annotated above them to provide an easier way of changing between players. Similar to the object detection feature, there is a button to disable the annotations above the players to provide a simpler and cleaner version of this functionality. The user can change the highlighted player by choosing one of the four options on a small menu on the right side of the video player.

Initially, the name of each player is assigned by default, where the prefix "PLAYER" is added to the ID assigned by the tracking algorithm. To change the name of a player, the user can select the edit icon next to the player's name and enter the desired one.

The custom names are saved in a JSON file with the following structure: `{<id>: <custom_name>}`, where each entry maps a player ID to its corresponding name. Each time the video is opened, the custom names are loaded and displayed instead of the default ones, preventing users from re-entering names every time they access the same video. The player highlight functionality is illustrated in Figure 3.16.

3.2.6 Additional Functionalities

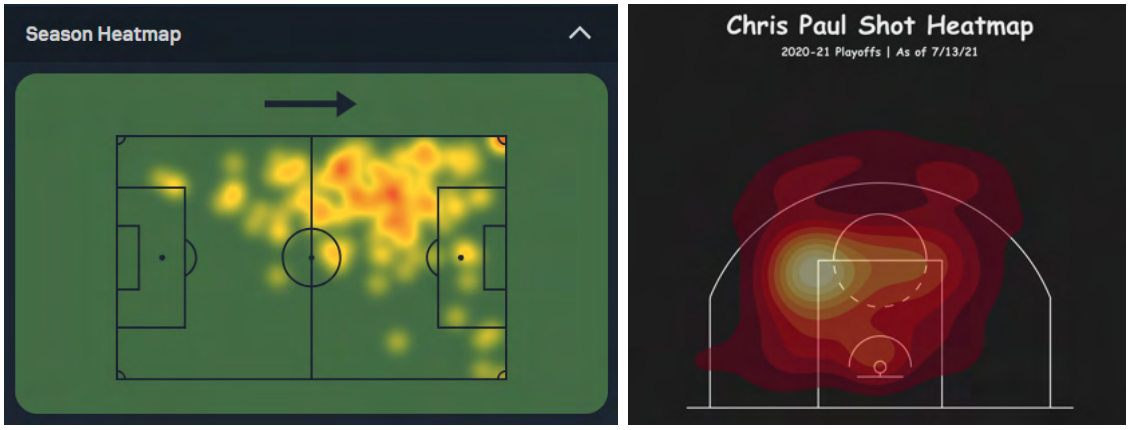
After completing the implementation of object detection and tracking, as well as developing the custom dataset and training the YOLO model with it, which consumed a significant portion of the development time for this thesis, the focus shifted to adding some additional functionalities to make the system more robust and helpful for its users.

There were various options for implementation: develop the other functionalities present in the prototype, create new functionalities, or even both. The prototype's main objective was to engage users from the early stages of development to evaluate the applicability of such a system and explore some implementation routes within the context of padel. However, since the two additional functionalities described in Section 3.2.1 (itemized search and automated statistical reporting) would involve the development of extremely complex metrics to extract game events and statistical reports solely from the results of object detection and tracking, an alternative approach was pursued, focusing on implementing more straightforward features that were also considered useful for game analysis, including some suggested by users in the preliminary tests.

Three new functionalities were developed, featuring two variations of the player's heatmap and the visualization of their trajectories. These new functionalities serve as a foundation for the system. After being tested by users and their usefulness properly evaluated, the other prototype functionalities may be added in future work.

3.2.6.1 Heatmap (Static and Dynamic)

The first additional functionality that was developed was a player's heatmap with two variations: static and dynamic. Heatmaps are a visual representation of data that uses



(a) Football: Player movement heatmap[8]. (b) Basketball: Player shot heatmap[9].

Figure 3.17: Examples of heatmap usage across different sports.

color to indicate different levels of intensity for a measured metric, making it easier to understand and evaluate.

Various industries take advantage of different types of heatmaps[7] (e.g., analyze crime density per area, tracking user behavior on websites, or visualizing traffic congestion). However, in sports, these are usually used to represent a player's movement during a match of a particular sport. They can reveal movement patterns, identify potential positioning mistakes, and improve the overall understanding of a player's playstyle and tendencies. While several color palettes can be used, usually areas where players spend most of their time appear in red while progressively less visited areas are represented by yellow and green.

This kind of visual representation is highly used in sports like football[8] (see Figure 3.17a) or basketball[9] (see Figure 3.17b) and it can also be effectively applied in racket sports to analyze player movement and court coverage, as well as shot distribution. The heatmap developed during this thesis focuses on better understanding player movement and court coverage, with the possibility of adding shot distribution in future work.

To represent the heatmap, the heatmap.js library[10] was used. This library provides a simple and efficient solution to visualize heatmaps on web applications. Major international organizations, such as the international governing body for football (FIFA)[11], take advantage of this library, showcasing its credibility and widespread use.

In the context of the system developed for this thesis, when the users select the heatmap functionality, a request is sent to the Python server to retrieve the tracking data. If there is no available data for the selected video, users wait on a loading screen, similar to the object detection and player highlight features, while tracking is being computed. Once

[7] https://www.patrick-wied.at/static/heatmapjs/showcases.html (visited on August 2024).
[8] https://www.sofascore.com/player/pedro-goncalves/895764 (visited on August 2024).
[9] https://github.com/DomSamangy/R_Tutorials (visited on August 2024).
[10] https://www.patrick-wied.at/static/heatmapjs/ (visited on August 2024).
[11] https://web.archive.org/web/20140712005218/http://www.fifa.com/worldcup/players/player=201200/statistics.html (visited on August 2024).

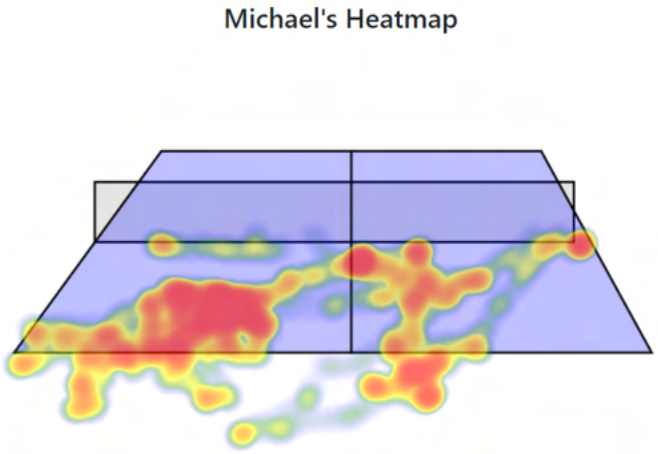


Figure 3.18: Example of a static heatmap generated by the developed system.

the computation is finished, or if it was already previously performed, the Python server sends the data back to the client.

A similar process is performed to obtain the object detection data, which is then used to draw the court where the heatmap will be overlaid. Using object detection data to draw the court enables a more precise heatmap display. In contrast, utilizing a court with pre-defined dimensions could result in an inaccurate heatmap representation caused by different camera angles and positions across different videos.

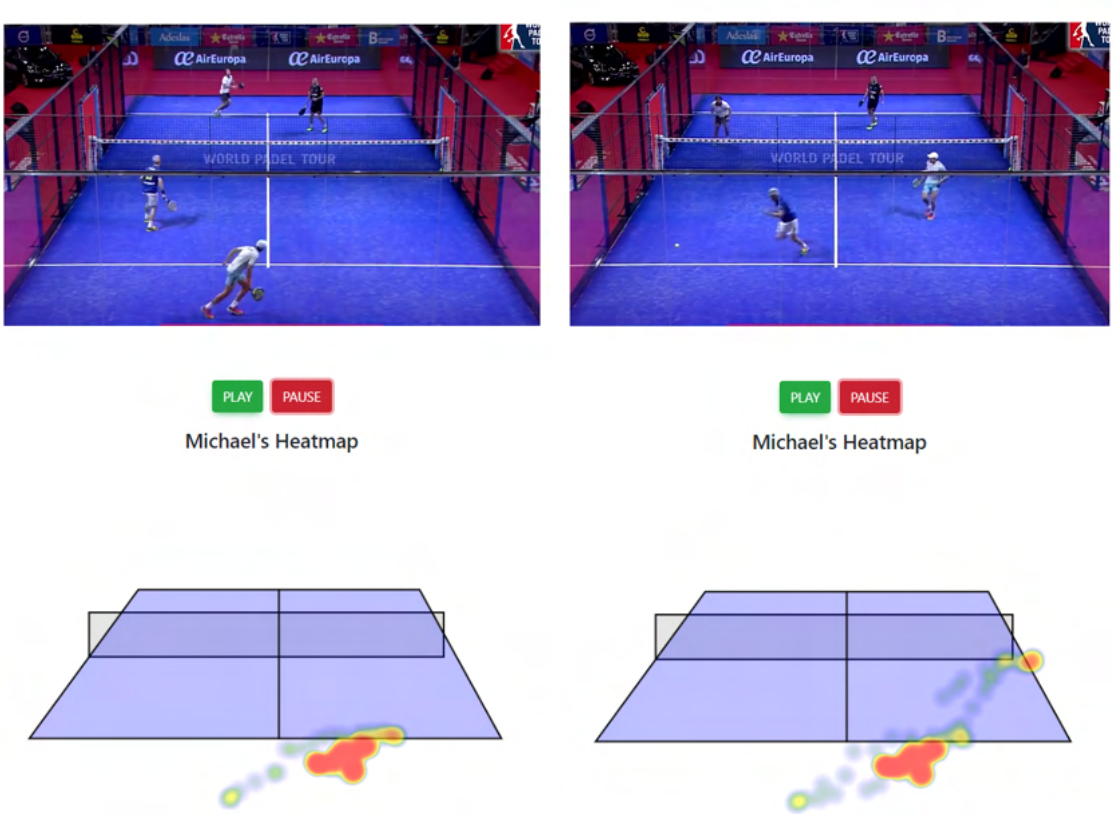
Since some frames can experience inaccurate detections, before drawing the court, the data is first processed to select the first frame where exactly two serve lines are detected and reasonably far from each other (preventing cases where the same serve line is detected twice). The data from this frame is then used to draw the court lines and net.

After initializing the heatmap with parameters such as the color palette and radius, the player coordinates throughout the video are provided to the heatmap. Besides adjusting the coordinates to match the canvas dimensions, these coordinates are also refined to represent the players' feet rather than their center, as this is the standard approach to represent data in sports heatmaps.

The static version of the heatmap (see Figure 3.18) receives the coordinates for the entire video in a single instance while the dynamic variation (see Figure 3.19) receives the coordinates progressively as the video frames advance, ensuring that the heatmap's drawing is synchronized with the player's movement while the video is playing. To ensure that the dynamic heatmap drawing is synchronized with the video, the method previously described in Section 3.2.3 is used to estimate the current frame.

These two different variations of the heatmap feature accommodate distinct user needs. If users decide to take a more direct approach and view the heatmap for the whole video instantly, they can opt for the static version. However, if they prefer a more interactive experience and want to understand how the heatmap evolves throughout the video, they can alternatively choose the dynamic version.

The heatmap is displayed below the video player (as illustrated in Figure 3.19) in both



(a) Frame 1: Heatmap at timestamp T1.

(b) Frame 2: Heatmap at timestamp T2.

Figure 3.19: Dynamic heatmap example across different frames.

the static and dynamic versions of this feature. Similar to the player highlight feature illustrated in Figure 3.16, there is a menu next to the video player that enables users to choose the player whose heatmap they want to view and edit the players' names.

3.2.6.2 Player Trajectory

The last developed functionality is called player trajectory and it shares strong similarities with the dynamic heatmap feature. Both are dynamic representations, but this new functionality is slightly different since it tracks the player's path across the video. Instead of highlighting the areas where the player spent more time throughout the video, we simply draw a green line representing the player's movement (see Figure 3.20).

The whole workflow of the player trajectory is similar to the dynamic heatmap. Two requests are sent to the Python server to obtain the object detection and tracking data for the selected video. Then, a suitable video frame is selected to retrieve the data to represent the court. Lastly, the players' coordinates are adjusted to match the canvas dimensions and further refined to represent the players' feet rather than the center of their bodies.

Apart from providing valuable insights about the player's movement throughout the video, this feature also introduces a different kind of annotation into the application. Previously, the application used only text annotations, but now, a different type of annotation

(a) Frame 1: Player trajectory at timestamp T1. (b) Frame 2: Player trajectory at timestamp T2.

Figure 3.20: Player trajectory example across different frames.

is employed to represent the player's trail, similar to the ink stroke annotation described in Section 2.6.

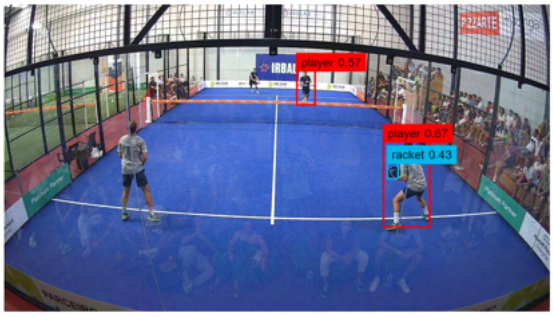
3.2.7 Limitations

As previously mentioned in Section 3.2.4, the custom dataset is only composed of images from the World Padel Tour and Premier Padel competitions. This is because, in televised professional games, the main camera is usually positioned in the same elevated and centered position behind one of the courts, despite the game's location. This standard camera setup captures all game elements and provides a clear view of the whole court, facilitating the object tracking and detection tasks. Additionally, this approach allows for the acquisition of stable detection data, as all frames are recorded in the same environment and there are no outside factors (e.g., camera position or angle) affecting the model's accuracy.

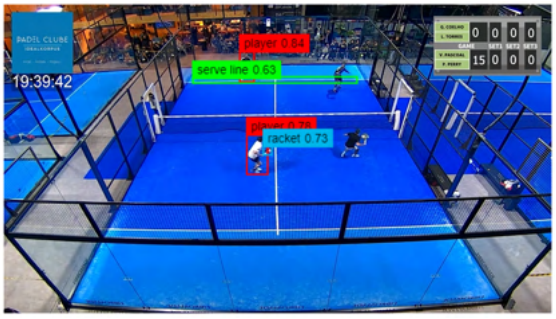
Nonetheless, the model's accuracy was also tested in videos with non-standard camera setups but the results were not optimal (see Figure 3.21). Figure 3.21a displays an example where the video was recorded from behind one of the courts at eye level, rather than an elevated position.

Even if no specific functionalities are used, it is challenging for the viewers to locate certain game elements, especially when they are farther from the camera. For instance, in the provided video frame, it is hard to determine the position of the serving line

(a) Eye level camera setup.



(b) Slightly elevated camera setup.



(c) Substantially elevated camera setup.

Figure 3.21: Object detection in videos with non-standard camera setups, highlighting challenges not seen in World Padel Tour and Premier Padel examples.

farther from the camera, which makes tasks such as object detection, heatmap, and player trajectory analysis extremely difficult.

Moreover, since the camera setup is much different from the one used in professional games, the model struggles to detect even more visible elements, such as the players or the net, detecting them only occasionally.

Figures 3.21b and 3.21c show frames from two other videos where the game elements are somewhat more visible. Despite that, the model still struggled to identify most game elements since these are different camera angles from the ones used in the custom dataset and model training. Even though some elements are detected (e.g., players, rackets, and serve lines), those detections are inconsistent and do not offer the best user experience.

Therefore, in order to experience the best results, only videos from professional games with standard camera setups, such as those from the World Padel Tour and Premier Padel competitions, should be used in the web application.

Another limitation is that only individual rallies, or at best individual games, should be used in the application and not full matches. Since the re-identification approach that was implemented only uses the distance and the predicted coordinates from the Kalman Filter, it is not possible to re-identify players when they change the side of the court.

While it might be possible to re-identify players if they remain within the camera's range and are being continuously tracked, televised broadcasts broadcasts typically feature multiple cameras and various replays, making it very challenging to re-identify players throughout an entire match.

Additionally, when the players switch sides of the court, the camera captures them from the opposite angle (e.g., if it was previously showing their backs, it will now show their fronts). As no advanced re-identification methods, such as facial recognition, clothing, and body shape analysis, are being used, it is unfeasible to account for videos where players change court sides. Still, while such technologies would enhance identification, they could also introduce overhead to the system and negatively impact the user experience. Therefore, it is preferable to only use videos with individual rallies to ensure results with higher accuracy.

4

Evaluation and Results

This chapter presents the user tests utilized to evaluate the system and the results that were obtained.

Due to the iterative nature of the system's design and implementation, the testing phase consisted of two iterations. The first one presented a prototype version of the system, where some functionalities were presented but not yet implemented. The second phase offered the fully implemented version of the system, where the functionalities were not entirely the same as the prototype ones — some were added, and others were removed. Each iteration helped to evaluate the system and identify potential system improvements.

In both the preliminary and final user tests, participants received an interview guide with the various testing steps and had to sign a consent form. The documents used in the final user tests are available in appendices D and E.

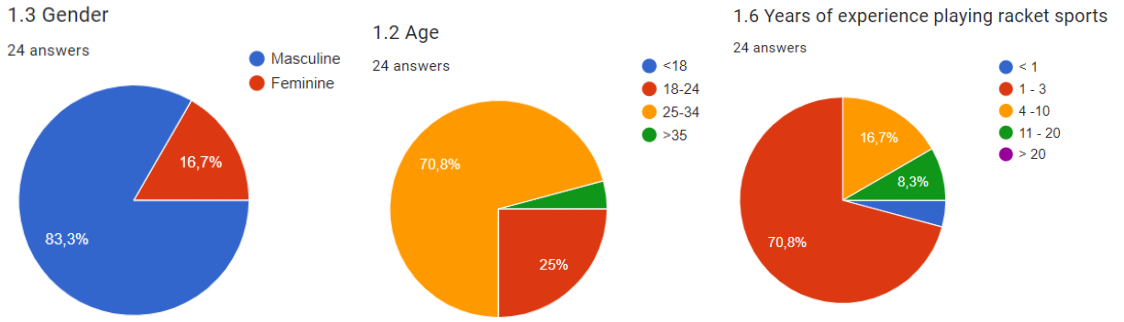
4.1 Preliminary User Tests

The preliminary user tests were conducted in a early stage of the system, where users were presented with the system's prototype, composed of four functionalities. This approach was adopted because it involved the users from the early stages of development and allowed for testing of the system's potential and applicability from the start.

Participants were asked to test all available functionalities and to suggest additional features they believed would enhance the utility of this technology. They were expected to test the system for approximately 20 minutes.

4.1.1 Participants and Evaluation Method

For this study, 24 participants with experience in racket sports were selected, comprising 20 males and 4 females. Most participants were in the 24 and 34 age range, accounting for 75% of the group. Their occupations varied: teachers, engineers, students, and sports coaches. Regarding expertise in racket sports, the participants reported high levels of experience: 4% had over 20 years, 8% between 10 and 20 years, 17% between 5 and 10 years, and 71% between 1 and 5 years (see Figure 4.1).



(a) Gender.

(b) Age.

(c) Years of experience playing racket sports.

Figure 4.1: Preliminary user tests: Demographic distribution of participants.

Each testing session began with a demonstration of the tool and basic training. Essential features such as logging in, managing videos, and video playback were initially explained. Subsequently, the features based on object detection and tracking were introduced, along with guidance on their usage. Participants were then encouraged to use the system freely and provide their comments on each functionality. The sessions were concluded with a questionnaire designed to gather feedback on the prototype and suggestions for improvement.

The questionnaire included a demographic section and professional background, followed by the System Usability Scale (SUS) standard questionnaire. Then, several questions related to object detection and tracking integration were presented, mainly consisting of 5-point Likert scales, ranging from 1 - "strongly disagree" to 5 - "strongly agree". The last question was open-ended to collect suggestions.

4.1.2 Results and Discussion

The results of the questionnaires are presented in appendix A.

The SUS questionnaire [5] evaluates how respondents view a system's usability through ten questions, where each question is composed by a five point Likert scale, ranging from 1 - "totally disagree" to 5 - "totally agree".

To calculate the system usability scores, the following formula was used:

$$((Q1 - 1) + (5 - Q2) + (Q3 - 1) + (5 - Q4) + ... + (Q9 - 1) + (5 - Q10)) * 2.5$$

where Q_i represents the score given for the i-th question on the SUS questionnaire. The SUS questionnaire displayed the following results for the developed prototype, detailed the Table 4.1 below:

Table 4.1: Preliminary User Tests SUS scores.

Mean	Median	Standard Deviation
88.1	90	6.77

On its own, a score does not indicate whether it represents good or poor performance. As a result, several researches have been conducted to analyze SUS scores. A. Bangor et al. [3] proposed a grading scale similar to the university grading system in which SUS scores under 60 were assigned an "F", those from 60 to 69 a "D", from 70 to 79 a "C", from 80 to 89 as a "B", and those 90 and above as an "A". J. R. Lewis et al. [30] analyzed over 200 usability studies to develop a curved grading scale, with the SUS score of 68 at the center of the range for an average grade (C).

Based on the research, the obtained results indicate that the developed prototype had high usability, nearly achieving the "A" grade in the A. Bangor et al. scale. Although the SUS questionnaire is typically conducted in fully developed systems, using it on the prototype provided an initial sense of the system's usability.

All respondents recognized the value of reviewing games and training sessions through video, indicating the popularity of this medium in sports. When asked for their preferred device for video analysis, most participants selected a regular computer, with over 60% favoring this option. Tablets were the second choice, preferred by 25% of participants, while smartphones received the remaining votes.

The feedback was predominantly positive in response to questions about how video analysis could improve game understanding (with an average rating of 4.5) and the likelihood of integrating this tool into their training (averaging 3.7). This confirms the favorable reception of using video in racket sports. However, when asked if they had previously performed video analysis using other methods or tools, only 16.7% of participants responded affirmatively. These findings suggest that video analysis tools are not widely used by practitioners, at least among the respondents in this study.

To determine if there was a difference in how users rated the several details they focused on while reviewing padel videos, the One-Way ANOVA test was conducted. For this question, the p-value was significant ($p < 0.05$), meaning that the null hypothesis — which stated that all details were rated equally — should be rejected and at least one of the proposed details was rated significantly differently from the others.

The statistical analysis of participants' responses revealed a clear preference for concentrating on strategic aspects of gameplay. This emphasizes the importance players and coaches place on understanding and refining game plans, positioning, and decision-making. Alongside strategic elements, there was also notable interest in technique details. This suggests that participants value the opportunity to closely examine and improve individual techniques, such as stroke precision, footwork, and body positioning. These two details scored significantly higher than physical conditions, which was the lowest-rated detail with an average score of 2.8.

Question	1	2	3	4	5	Avg.	Std. Dev.	ANOVA (SF)
1. How much video analysis could enhance your understanding of a game?	0	0	0	12	12	4.5	0.5	
2. How likely are you to integrate this video analysis into your training?	0	2	7	12	3	3.7	0.8	
3. On which kind of details do you focus most while reviewing the video?								
3.1. Technique Details	0	0	4	7	13	4.4	0.8	p = 5,9E-11; p<0.05
3.2. Strategy Details	0	0	2	9	13	4.5	0.7	
3.3. Game Events	0	3	5	10	6	3.8	1	
3.4. Game Status	1	6	7	8	2	3.2	1	
3.5. Physical Conditions	1	9	9	4	1	2.8	1	
4 How effective/useful is each feature below in providing insights about games?								
4.1. Itemized Search	0	0	0	7	17	4.7	0.5	p = 1.0E-7; p<0.05
4.2 Automated Statistical Reporting	1	0	0	6	17	4.6	0.9	
4.3. Individual Player Focus	0	4	2	8	10	4.0	1.1	
4.4. Itemized Search (Object Detection)	2	5	6	8	3	3.2	1.2	
5. How likely will you recommend this tool to other players and coaches?	0	0	1	12	11	4.4	0.6	

Figure 4.2: Preliminary user tests: Questionnaire section 3 statistics.

To evaluate the significancy of the responses regarding the four features based on object detection and tracking, the One-Way ANOVA was used once more. The p-value was also significant ($p < 0.05$), suggesting that the null hypothesis — which stated that all features were rated equally — should be rejected and at least one of the features was rated significantly differently from the others.

The automatic search was the feature that received the highest average rate from users. This preference highlights the importance of the ease and efficiency of finding specific moments or events in the game footage. The feature for automatic statistics generation also garnered significant interest. It appeals to users by instantly compiling statistical data on various gameplay elements. Nevertheless, all features received positive feedback, with player highlight averaging a 4.0 rating and object detection averaging 3.2 (see Figure 4.2).

Overall, the feedback was positive, with some participant comments expressing satisfaction. One participant said, "All the features are well-suited for game analysis, making them relevant and applicable to both amateurs and professionals". Another participant said, “Regarding the statistics, it enables the creation and comparison of data between games, thereby facilitating the assessment of progress".

Another comment highlighted, "I think the player highlight feature would be the most useful regarding the player analysis aspect of the game". On the other hand, six different respondents suggested a heatmap feature, which was ultimately implemented in the developed system.

One participant added, "Video analysis is important in coaching, especially for beginners, as it visually demonstrates techniques, helping them understand concepts faster".

Despite the encouraging results, such as the SUS score and the overall feedback of the respondents, these results needed to be interpreted carefully, as the prototype primarily demonstrated conceptual ideas rather than being a fully functional product. Therefore,

a second and final iteration of user tests was conducted after the system was further developed.

4.2 Final User Tests

In the second and final user study, the final version of the system was evaluated. This version was composed of all functionalities described throughout the Section 3.2.

Similarly to the preliminary user tests, respondents tested all functionalities, commenting on their value and pertinence, and suggested additional improvements and features that could enhance the system. On average, the tests consisted of sessions of 15 to 30 minutes per participant.

4.2.1 Participants and Evaluation Method

Unlike the first testing phase, this round of tests included participants without racket sports experience to gain a wider perspective on the system. In total, 30 respondents evaluated the system, comprising 22 males and 8 females, where 50% of the participants had experience in racket sports and the other half did not.

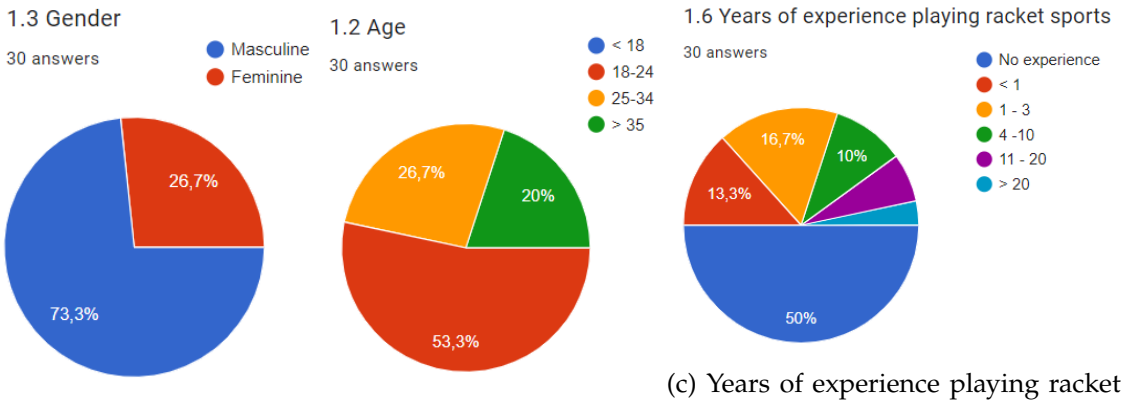
Approximately half of the users were in the 18 and 24 range, while the remaining participants were almost evenly split between the 25-34 age group (27%) and those over 35 (20%). They had various occupations: professors, students, engineers, architects, security managers, and staff from a padel facility.

The respondents with experience in racket sports had different levels of expertise: four had less than a year, five had 1 to 3 years, three had 4 to 10 years, two had 11 to 20 years, and one had over 20 years of experience (see Figure 4.3). Among them, the most common sports were padel (53%) and tennis (46.7%), followed by table tennis (26.7%) and badminton (20%). Additionally, one respondent had experience in squash.

The second round of testing followed a similar process to the first. It began with a basic introduction about the system and its purpose. Participants then tested each feature, providing feedback and offering suggestions for improvement.

The questionnaire was almost identical to the one used in the preliminary user tests. However, besides the SUS questionnaire and some additional questions, the User Experience Questionnaire (UEQ) [51] was also used. The UEQ consists of 26 seven-point scale items, each represented by two opposite terms (e.g., attractive and unattractive). The items are scored from -3 to +3, where -3 corresponds to the most negative option, 0 indicates a neutral response, and +3 represents the most positive answer.

The UEQ divides these items into six scales: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty. Perspicuity, efficiency, and dependability are then categorized as pragmatic quality aspects due to their goal-oriented nature, while stimulation and novelty are considered hedonic quality aspects since they are not goal-oriented. On the other hand, attractiveness is considered an independent category focused



(a) Gender. (b) Age. (c) Years of experience playing racket sports.

Figure 4.3: Final user tests: Demographic distribution of participants.

only on the system's appeal.

4.2.2 Results and Discussion

The results of the questionnaires are presented in appendices B and C. The final version of the system obtained the following results for the SUS questionnaire, detailed in Table 4.2 below:

Table 4.2: Final User Tests SUS scores.

Category	Mean	Median	Standard Deviation
With Racket Sports Experience	87.7	87.5	8.68
Without Racket Sports Experience	93.2	92.5	5.71
All respondents	90.4	91.25	7.43

The system obtained, on average, an 87.7 SUS score among the respondents with racket sports experience, achieving a quite similar value to the one obtained in the preliminary user tests, which only involved participants with racket sports experience. However, in the final user tests, participants without racket sports experience were also included to gain a different perspective on the system. Among this demographic, the system achieved even better results, averaging a SUS score of 93.2. When evaluating all respondents collectively, regardless of their racket sports experience, the system averaged a 90.4 SUS score, an "A" grade in the A. Bangor et al. scale, highlighting the user satisfaction with the system's usability.

When asked about the usefulness of reviewing games and training sessions through video, users responded affirmatively, averaging a rating of 4.6, following the trend observed in the initial tests. Unlike the preliminary tests, when asked for their preferred device for video analysis, users were able to select more than one option, though selecting just one was also allowed. Computers were once again the most voted option, with 80% of votes,

followed by smartphones and tablets, both at 40%, while the television collected 36.7% of the votes.

Similarly to the first phase of tests, the feedback to questions about how video analysis could improve game understanding and the likelihood of integrating this tool into training was generally positive, respectively averaging ratings of 4.3 and 3.8. Nevertheless, video analysis tools appear to be infrequently used by practitioners, with only one participant responding affirmatively when asked whether they previously performed video analysis using other methods, which is even lower than the number obtained in the first round of tests.

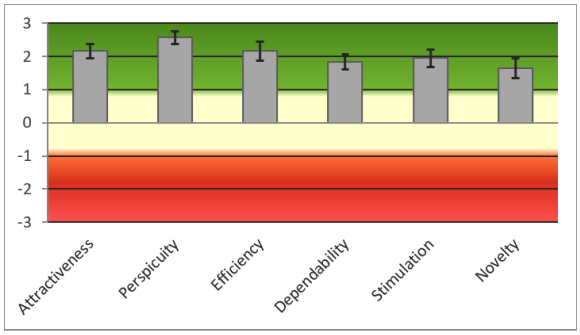
Consistent with the preliminary test, a One-Way ANOVA was conducted to evaluate the significance of the responses regarding the details that users prioritize when reviewing games on video. For this question, the p-value was significant ($p < 0.05$) once more, meaning that at least one of the features was rated significantly differently from the others.

The strategic (scoring 4.3) and technical details (scoring 4.5) remained as the two most important areas of focus. Additionally, the spatial analysis aspect, introduced in this round of testing through features like the heatmap and player trajectory, also achieved a high score (averaging 4.2). This emphasizes the importance of understanding the players' movement, positioning, and tendencies throughout the entire court.

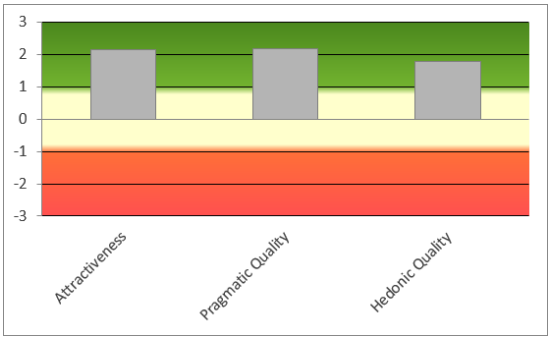
Question	1	2	3	4	5	Avg.	Std. Dev.	ANOVA (SF)
1. In general, how useful is reviewing a game through video?	0	1	1	7	21	4.6	0.7	
2. How much video analysis could enhance your understanding of a game?	0	1	5	8	16	4.3	0.9	
3. How likely are you to integrate this video analysis into your training?	1	3	6	10	10	3.8	1.1	
4. On which kind of details do you focus most while reviewing the video?								
4.1. Technique Details	0	0	1	14	15	4.5	0.6	p = 2.7E-07; p<0.05
4.2. Strategy Details	0	1	4	11	14	4.3	0.8	
4.3. Game Events	1	3	14	9	3	3.3	0.9	
4.4. Game Status	1	6	8	11	4	3.4	1.1	
4.5. Physical Conditions	2	4	7	10	7	3.5	1.2	
4.6. Spatial Analysis	0	2	5	8	15	4.2	1	
5.How effective/useful is each feature below in providing insights about games?								
5.1. Item Detection	3	0	4	5	18	4.2	1.3	p = 0.84; p>0.05
5.2 Player Highlight	0	1	4	8	17	4.4	0.9	
5.3. Static Heatmap	1	2	2	6	19	4.3	1.1	
5.4. Dynamic Heatmap	0	1	3	8	18	4.4	0.8	
5.5. Player Trajectory	1	0	3	7	19	4.4	0.9	
6. How likely will you recommend this tool to other players and coaches?	0	0	2	10	18	4.5	0.6	
7. Could a similar system be effective for other sports besides racket sports?	0	0	1	9	20	4.6	0.6	

Figure 4.4: Final user tests: Questionnaire section 3 statistics.

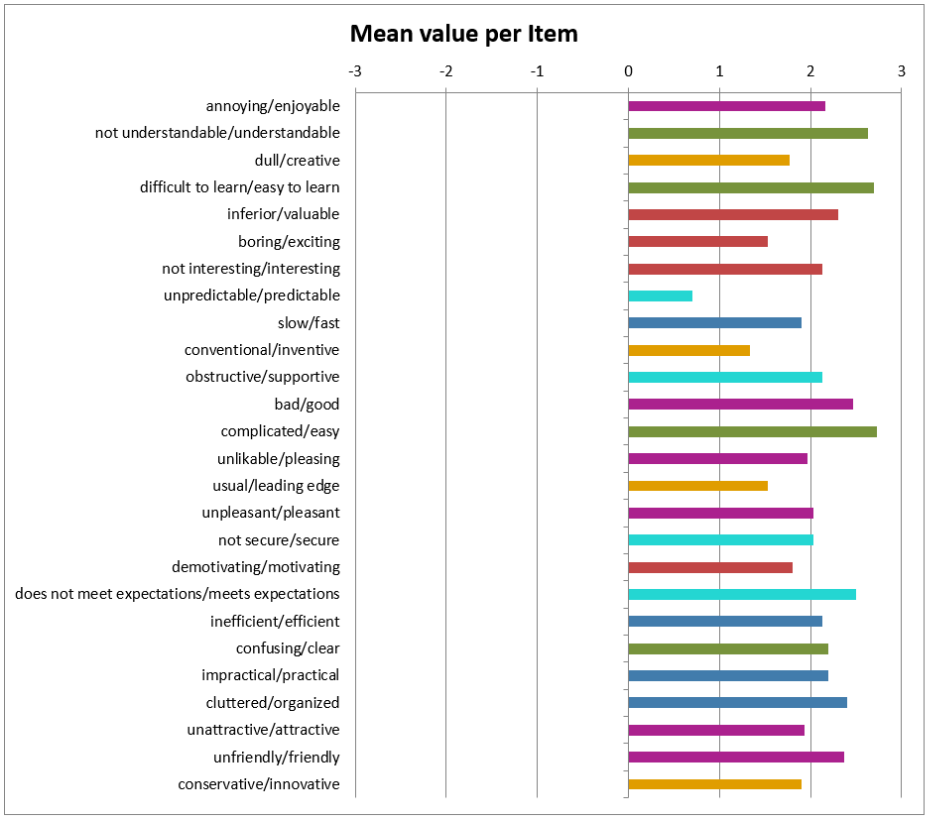
Once more, the One-Way ANOVA was used to assess the significance of the features' ratings. Unlike the other tests, the p-value was not significant ($p > 0.05$), which means the null hypothesis was not rejected and all features were rated similarly.



(a) User Experience Questionnaire scales.



(b) Pragmatic and Hedonic quality.



(c) Average score per item.

Figure 4.5: Final user tests: User Experience Questionnaire results.

All system functionalities received positive feedback from the respondents, each averaging a score above 4.0. Despite having highly similar results across all features, the functionalities with the highest average ratings were the player highlight, dynamic heatmap, and player trajectory, each scoring 4.4 among users. The statistics for this whole section of questions are summarized in Figure 4.4.

The UEQ also presented strong results, which are illustrated in Figure 4.5. Figure 4.5a shows the scores of the six different scales in the UEQ. Values between -0.8 and 0.8 represent neutral evaluations, while values above 0.8 represent positive evaluations, and values below -0.8 represent negative evaluations. Thus, it is evident that all the evaluated aspects received a positive assessment, with everything falling within the green zone of the chart. Perspicuity received the highest score, averaging 2.57, followed by efficiency and attractiveness, both averaging 2.16.

Figure 4.5b shows the pragmatic and hedonic qualities, which were also consistently positive. The pragmatic quality scored the highest, averaging 2.19, and the hedonic quality also achieved favorable results, averaging 1.79. The items with the highest results were both from the perspicuity scale: "easy to learn/difficult to learn" and "complicated/easy", each averaging 2.7.

When analyzing the mean value of each item (see Figure 4.5c), the item "unpredictable/predictable" stands out by being the only item with a neutral score, averaging 0.7. In this item, several respondents asked for clarification about which side represented the positive option, leading many to choose a neutral response when uncertain. This likely accounts for why this item received the lowest rating among the evaluated aspects.

To better understand the quality of a product, UEQ also offers a benchmark containing data from 21175 persons across 468 studies concerning different products (business software, web pages, web shops, social networks). The benchmark classifies products into five categories: excellent (top 10% of results), good (10-25% of results), above average (25-50% of results), below average (50-75% of results), and bad (worst 25% of results). When compared to the benchmark, the system achieved excellent classification across all evaluated scales, with each averaging a score within the top 10% of results (see Figure 4.6).

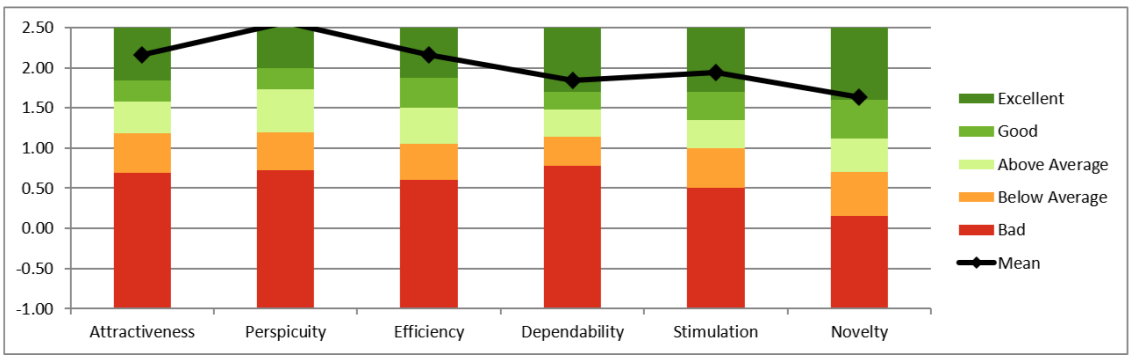


Figure 4.6: Benchmark graph for the User Experience Questionnaire.

Similarly to the preliminary user tests, the overall feedback from the respondents was very positive, and various comments were added to those expressed in the prototype testing phase. One participant said, "All features are well integrated, enabling users to conduct thorough analyses and extract key insights from the videos." Another user said, "I really enjoy the minimalistic UI of the page. It is simple, yet very intuitive."

One of the respondents mentioned, "Even though the object detection feature may not be essential for game analysis, it is still impressive to see it in action and understand how the system works." Another comment highlighted, "The heatmap and player highlight features impressed me the most. Despite their differences, both provide great insights about player performance and overall game analysis." Additionally, one comment noted, "The system offers robust game analysis and could be adapted for other sports, serving them effectively, with great potential for further development."

5

Conclusions and Future Work

This last chapter presents the conclusions from the work developed throughout this thesis and its evaluation, while also describing some ideas and routes of implementation to improve the solution in the future.

5.1 Conclusions

This thesis culminated in the creation of a web application that uses CV techniques such as object detection and tracking to enhance sports analysis. This work focuses on padel, a relatively recent racket sport that has increased its popularity tremendously in recent years, having many unexplored implementation routes.

The developed system allows users to upload custom padel videos. While it is possible to watch the videos by themselves, other tools and features are provided to improve the comprehension of player movement, technique, tendencies, and overall game analysis. Besides the object detection and tracking features, three additional features were developed mainly to improve the comprehension of each player's movement during their games. These features include two variations of the players' heatmap — a static and a dynamic variation — and the drawing of each player's trajectory during the game.

Many challenges were encountered during the development of the system. The lack of padel datasets caused a new dataset to be created, composed of several iterations. Despite its time-consuming aspect and the challenges it presented, such as selecting suitable and representative images and the correct labeling of those images, this step was indispensable and tremendously improved the accuracy and reliability of the CV features as the dataset grew. Another major challenge encountered during the system's development was the assignment of new IDs in the object tracking feature, where YOLO would increase the ID of players if they could not be identified for some consecutive frames. To mitigate this issue a re-identification technique was developed. Despite some limitations, it worked properly for the desired purpose.

The design and development of the system followed an iterative approach, with two main phases. After each iteration, a series of user tests were conducted to evaluate the

system's features, their usefulness, and overall user satisfaction. In the first iteration, participants tested the initial prototype of the system, while in the second iteration, the final version of the system was evaluated.

The user tests feedback was extremely positive, showing that users were pleased with the developed system. The majority of participants were pleasantly surprised about the CV features and how they could be integrated into sports analysis. Despite differing in some features, both the prototype and the final version of the system received favorable comments throughout all the provided tools to improve the game analysis. These comments are of significant relevance, as most of the participants had experience with racket sports, either as players or coaches, and suggest that the overall goal of the system was successfully achieved.

In conclusion, despite leaving some room for improvement in the existing system, the thesis successfully met all of its goals and new future challenges can now be explored.

5.2 Future Work

Even though the system was completely developed, some functionalities could still be further improved. Moreover, in the future, other features and technologies can be developed to improve the game analysis aspect of the system.

One of the aspects that should continuously be improved is the custom dataset and the model training, with the addition of more representative images, to further increase the model's accuracy and make the detection and tracking features even more precise and reliable. With the continuous addition of more images, the detection of troublesome game elements should be increasingly easier, and their detections should become less intermittent.

Moreover, to overcome the system's limitation of only being able to produce accurate results for individual rallies, new and more complex methods of re-identification should be used. These methods would potentially allow for the upload of longer videos — ideally full padel matches — where, for instance, scenarios with the players changing the side of the court would be handled correctly.

Additionally, after improving the accuracy of the model and allowing for the use of complete padel match videos, the two features that were discarded from the prototype — itemized searches and automated statistical reporting — could also be implemented. These functionalities already received positive feedback from users in the preliminary tests and they would provide an even more detailed and comprehensive analysis of the matches. In the second series of user tests, many users also suggested the implementation of automatic statistical reporting, highlighting the demand for such a feature and its potential to significantly improve the user experience.

The participants of the user tests provided other valuable suggestions for future developments. For instance, some participants suggested that the object detection functionality should be more customizable, allowing users to disable the visualization of some game

elements (e.g., only show the detections of the ball and the players) and customize the colors of each class. One participant even mentioned that they were color blind and suggested that such a feature would prevent scenarios where the color of the bounding boxes could be confused with the video.

Another suggestion that was mentioned by various users was the addition of more video player controls. Currently, the users can only play and pause the videos, but other options such as fullscreen mode and a slider to adjust the video timestamp were also suggested by the participants.

Some users also proposed new features regarding the ball, such as drawing the ball's trajectory throughout the video (similar to the player trajectory feature already present in the system) or including the ball in the player highlight functionality.

Finally, some users suggested that the heatmap could be displayed on top of the video rather than below it to provide a more direct visualization method. To support this change and accommodate different user needs, the system could introduce a button to toggle between views and a slider to adjust the heatmap's opacity.

Bibliography

[1] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky. *BoT-SORT: Robust Associations Multi-Pedestrian Tracking*. 2022. arXiv: 2206.14651 [cs.CV]. URL: https://arxiv.org/abs/2206.14651 (cit. on pp. 13–15).

[2] I. W. R. Ardana, I. B. I. Purnama, and I. M. S. Yasa. "Application of Object Recognition for Plastic Waste Detection and Classification Using YOLOv3". In: IEEE, 2020-10, pp. 652–656. ISBN: 978-1-7281-9567-4. DOI: 10.1109/iCAST51016.2020.9557735 (cit. on p. 17).

[3] A. Bangor, P. T. Kortum, and J. T. Miller. "An Empirical Evaluation of the System Usability Scale". In: *International Journal of Human-Computer Interaction* 24 (6 2008-07), pp. 574–594. ISSN: 1044-7318. DOI: 10.1080/10447310802205776 (cit. on p. 56).

[4] N. Blanchard et al. ""Keep Me In, Coach!": A Computer Vision Perspective on Assessing ACL Injury Risk in Female Athletes". In: IEEE, 2019-01, pp. 1366–1374. ISBN: 978-1-7281-1975-5. DOI: 10.1109/WACV.2019.00150 (cit. on p. 2).

[5] J. Brooke. "SUS: A quick and dirty usability scale". In: *Usability Eval. Ind.* 189 (1995-11) (cit. on p. 55).

[6] F. Buendía, J. Gayoso-Cabada, and J.-L. Sierra. "An Annotation Approach for Radiology Reports Linking Clinical Text and Medical Images with Instructional Purposes". In: ACM, 2020-10, pp. 510–517. ISBN: 9781450388504. DOI: 10.1145/3434780.3436651 (cit. on p. 26).

[7] H. Caesar, J. Uijlings, and V. Ferrari. *COCO-Stuff: Thing and Stuff Classes in Context*. 2018. arXiv: 1612.03716 [cs.CV] (cit. on p. 19).

[8] Charleen et al. "Impact of Computer Vision With Deep Learning Approach in Medical Imaging Diagnosis". In: IEEE, 2021-10, pp. 37–41. ISBN: 978-1-6654-4002-8. DOI: 10.1109/ICCSAI53272.2021.9609708 (cit. on p. 15).

[9] S. X. Chen et al. "Web-Scale Generic Object Detection at Microsoft Bing". In: ACM, 2021-08, pp. 2674–2682. ISBN: 9781450383325. DOI: 10.1145/3447548.3467122 (cit. on p. 17).

[10] Z. Chen et al. "iBall: Augmenting Basketball Videos with Gaze-moderated Embedded Visualizations". In: ACM, 2023-04, pp. 1–18. ISBN: 9781450394215. DOI: 10.1145/3544548.3581266 (cit. on p. 19).

[11] A. Cioppa et al. "SoccerNet 2023 Challenges Results". In: (2023-09) (cit. on p. 18).

[12] R. Concepcion et al. "Towards the Integration of Computer Vision and Applied Artificial Intelligence in Postharvest Storage Systems: Non-invasive Harvested Crop Monitoring". In: IEEE, 2021-11, pp. 1–6. ISBN: 978-1-6654-0167-8. DOI: 10.1109/HNICEM54116.2021.9731973 (cit. on p. 15).

[13] *Deep Learning for Siri's Voice: On-device Deep Mixture Density Networks for Hybrid Unit Selection Synthesis*. en-US. URL: https://machinelearning.apple.com/research/siri-voices (visited on 2024-01-18) (cit. on p. 23).

[14] D. Deng et al. "EventAnchor: Reducing Human Interactions in Event Annotation of Racket Sports Videos". In: ACM, 2021-05, pp. 1–13. ISBN: 9781450380966. DOI: 10.1145/3411764.3445431 (cit. on p. 22).

[15] J. Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on p. 12).

[16] M. Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88 (2 2010-06), pp. 303–338. ISSN: 0920-5691. DOI: 10.1007/s11263-009-0275-4 (cit. on p. 12).

[17] Z. Ge et al. *YOLOX: Exceeding YOLO Series in 2021*. 2021. arXiv: 2107.08430 [cs.CV]. URL: https://arxiv.org/abs/2107.08430 (cit. on pp. 14, 19).

[18] T. Gebru et al. "Datasheets for datasets". In: *Communications of the ACM* 64 (12 2021-12), pp. 86–92. ISSN: 0001-0782. DOI: 10.1145/3458723 (cit. on p. 11).

[19] R. Girshick. "Fast R-CNN". In: (2015-04) (cit. on pp. 8, 9).

[20] R. B. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CoRR* abs/1311.2524 (2013). arXiv: 1311.2524. URL: http://arxiv.org/abs/1311.2524 (cit. on p. 8).

[21] G. T. Gobbel et al. "Assisted annotation of medical free text using RapTAT". In: *Journal of the American Medical Informatics Association* 21 (5 2014-09), pp. 833–841. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2013-002255 (cit. on p. 26).

[22] J. Grudin and D. Bargeron. "Multimedia Annotation: An Unsuccessful Tool Becomes a Successful Framework". In: *Communication and Collaboration Support Systems* (2005-01), pp. 62–76. URL: https://www.microsoft.com/en-us/research/publication/multimedia-annotation-unsuccessful-tool-becomes-successful-framework/ (cit. on p. 24).

[23] A. Gupta, P. Dollár, and R. Girshick. "LVIS: A Dataset for Large Vocabulary Instance Segmentation". In: (2019-08) (cit. on p. 13).

[24] E. Hiraki, M. Ishihara, and K. Umetani. "Tiny Approaches to the Interactive Online Lectures Under the COVID-19 Pandemic". In: IEEE, 2021-10, pp. 1–4. ISBN: 978-1-6654-3554-3. DOI: 10.1109/IECON48115.2021.9589676 (cit. on p. 25).

[25] V. Hudovernik and D. Skocaj. "Video-Based Detection of Combat Positions and Automatic Scoring in Jiu-jitsu". In: ACM, 2022-10, pp. 55–63. ISBN: 9781450394888. DOI: 10.1145/3552437.3555707 (cit. on p. 21).

[26] M. Istasse et al. "DeepSportradar-v2: A Multi-Sport Computer Vision Dataset for Sport Understandings". In: ACM, 2023-10, pp. 23–29. ISBN: 9798400702693. DOI: 10.1145/3606038.3616160 (cit. on p. 19).

[27] V. Jain et al. "Deep automatic license plate recognition system". In: ACM, 2016-12, pp. 1–8. ISBN: 9781450347532. DOI: 10.1145/3009977.3010052 (cit. on p. 15).

[28] R. Krishna et al. "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations". In: (2016-02) (cit. on p. 13).

[29] A. Kuznetsova et al. "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale". In: (2018-11). DOI: 10.1007/s11263-020-01316-z (cit. on p. 13).

[30] J. R. Lewis and J. Sauro. "Item benchmarks for the system usability scale". In: *J. Usability Studies* 13.3 (2018-05), pp. 158–167 (cit. on p. 56).

[31] Q. Li et al. "Kalman Filter and Its Application". In: *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. 2015, pp. 74–77. DOI: 10.1109/ICINIS.2015.35 (cit. on p. 14).

[32] T.-Y. Lin et al. "Microsoft COCO: Common Objects in Context". In: 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48 (cit. on pp. 11, 12).

[33] C. Liu et al. "Application of Hawk-Eye Technology to Sports Events". In: IEEE, 2022-06, pp. 1–5. ISBN: 978-1-6654-7025-4. DOI: 10.1109/TCS56119.2022.9918811 (cit. on p. 1).

[34] W. Liu et al. "SSD: Single Shot MultiBox Detector". In: (2015-12). DOI: 10.1007/978-3-319-46448-0_2 (cit. on pp. 9, 10).

[35] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[36] I. Lucic, S. Babic, and D. Vuckov. "Perception of Using VAR Technology in Football After Completion of Training and Education and Experiences of Croatian Video Assistant Referees (VARs) and Assistant VARs (AVARs)". In: IEEE, 2020-09, pp. 905–911. ISBN: 978-953-233-099-1. DOI: 10.23919/MIPRO48935.2020.9245111 (cit. on p. 1).

[37] A. Maksai, X. Wang, and P. Fua. "What Players do with the Ball: A Physically Constrained Interaction Modeling". In: (2015-11) (cit. on p. 2).

[38] A. Nhu et al. "A Comprehensive Defense Approach Targeting The Computer Vision Based Cheating Tools in FPS Video Games". In: IEEE, 2023-11, pp. 168–177. ISBN: 979-8-3503-0293-6. DOI: 10.1109/IPCCC59175.2023.10253881 (cit. on p. 15).

[39] S. M. D. Oca, M. Villada-Balbuena, and C. Camacho-Zuniga. "Professors' Concerns after the Shift from Face-to-face to Online Teaching amid COVID-19 Contingency: An Educational Data Mining analysis". In: IEEE, 2021-12, pp. 1–5. ISBN: 978-1-6654-2763-0. DOI: 10.1109/IEEECONF53024.2021.9733778 (cit. on p. 25).

[40] J. P. Ono et al. "HistoryTracker: Minimizing Human Interactions in Baseball Game Annotation". In: ACM, 2019-05, pp. 1–12. ISBN: 9781450359702. DOI: 10.1145/3290605.3300293 (cit. on p. 27).

[41] N. A. Othman and I. Aydin. "A new IoT combined body detection of people by using computer vision for security application". In: IEEE, 2017-09, pp. 108–112. ISBN: 978-1-5090-5001-7. DOI: 10.1109/CICN.2017.8319366 (cit. on p. 15).

[42] O. Pelka, F. Nensa, and C. M. Friedrich. "Annotation of enhanced radiographs for medical image retrieval with deep convolutional neural networks". In: *PLOS ONE* 13 (11 2018-11), e0206229. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0206229 (cit. on p. 26).

[43] R. Prasad. "Alexa Everywhere". In: ACM, 2019-01, pp. 3–3. ISBN: 9781450359405. DOI: 10.1145/3289600.3291377 (cit. on p. 23).

[44] J. Redmon and A. Farhadi. "YOLO9000: Better, Faster, Stronger". In: (2016-12) (cit. on p. 11).

[45] J. Redmon and A. Farhadi. "YOLOv3: An Incremental Improvement". In: (2018-04) (cit. on p. 11).

[46] J. Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: (2015-06) (cit. on pp. 10, 11).

[47] S. Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: (2015-06) (cit. on pp. 8, 9).

[48] P. Rethinam et al. "Olympic Weightlifters' Performance Assessment Module Using Computer Vision". In: IEEE, 2023-09, pp. 8–12. ISBN: 979-8-3503-1605-6. DOI: 10.1109/STAR58331.2023.10302649 (cit. on p. 2).

[49] R. Rodrigues, R. N. Madeira, and N. Correia. "Studying Natural User Interfaces for Smart Video Annotation towards Ubiquitous Environments". In: ACM, 2021-05, pp. 158–168. ISBN: 9781450386432. DOI: 10.1145/3490632.3490672 (cit. on pp. 1, 25).

[50] A. Rossi et al. "GPS Data Reflect Players' Internal Load in Soccer". In: IEEE, 2017-11, pp. 890–893. ISBN: 978-1-5386-3800-2. DOI: 10.1109/ICDMW.2017.122 (cit. on p. 1).

[51] M. Schrepp. *User Experience Questionnaire Handbook*. 2015-09. DOI: 10.13140/RG.2.1.2815.0245 (cit. on p. 58).

[52] H. Song. "The Application of Computer Vision in Responding to the Emergencies of Autonomous Driving". In: IEEE, 2020-07, pp. 1–5. ISBN: 978-1-7281-9481-3. DOI: 10.1109/CVIDL51233.2020.00008 (cit. on p. 15).

[53] J. Song et al. "A survey of remote sensing image classification based on CNNs". In: *Big Earth Data* 3 (3 2019-07), pp. 232–254. ISSN: 2096-4471. DOI: 10.1080/20964471.2019.1657720 (cit. on p. xiv).

[54] S. Srivastava et al. "Comparative analysis of deep learning image detection algorithms". In: *Journal of Big Data* 8 (1 2021-12), p. 66. ISSN: 2196-1115. DOI: 10.1186/s40537-021-00434-w (cit. on pp. 8, 11).

[55] J. A. Stankovic. "Real-time and embedded systems". In: *ACM Computing Surveys* 28 (1 1996-03), pp. 205–208. ISSN: 0360-0300. DOI: 10.1145/234313.234400 (cit. on p. xiv).

[56] R. Szeliski. *Computer Vision: Algorithms and Applications*. 2010. URL: http://szeliski.org/Book/. (cit. on p. 6).

[57] K. Takumi et al. "Multispectral Object Detection for Autonomous Vehicles". In: ACM, 2017-10, pp. 35–43. ISBN: 9781450354165. DOI: 10.1145/3126686.3126727 (cit. on p. 16).

[58] G. Thomas et al. "Computer vision for sports: Current applications and research topics". In: *Computer Vision and Image Understanding* 159 (2017-06), pp. 3–18. ISSN: 10773142. DOI: 10.1016/j.cviu.2017.04.011 (cit. on p. 2).

[59] N. Verdel et al. "Reliability and Validity of the CORE Sensor to Assess Core Body Temperature during Cycling Exercise". In: *Sensors* 21 (17 2021-09), p. 5932. ISSN: 1424-8220. DOI: 10.3390/s21175932 (cit. on p. 2).

[60] J. Wang et al. "Tac-Trainer: A Visual Analytics System for IoT-based Racket Sports Training". In: *IEEE Transactions on Visualization and Computer Graphics* (2022), pp. 1–11. ISSN: 1077-2626. DOI: 10.1109/TVCG.2022.3209352 (cit. on p. 2).

[61] K. Wankhede, B. Wukkadada, and V. Nadar. "Just Walk-Out Technology and its Challenges: A Case of Amazon Go". In: IEEE, 2018-07, pp. 254–257. ISBN: 978-1-5386-2456-2. DOI: 10.1109/ICIRCA.2018.8597403 (cit. on p. 18).

[62] J. Xiao et al. "SUN database: Large-scale scene recognition from abbey to zoo". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 3485–3492. DOI: 10.1109/CVPR.2010.5539970 (cit. on p. 12).

[63] T. Yamashita and T. Kobayashi. "Smart ping pong racket by ultrathin piezoelectric strain sensor array". In: IEEE, 2018-05, pp. 1–3. ISBN: 978-1-5386-6199-4. DOI: 10.1109/DTIP.2018.8394237 (cit. on p. 2).

[64] G. V. Zandycke et al. "DeepSportradar-v1: Computer Vision Dataset for Sports Understanding with High Quality Annotations". In: ACM, 2022-10, pp. 1–8. ISBN: 9781450394888. DOI: 10.1145/3552437.3555699 (cit. on p. 19).

[65] Y. Zhang et al. *ByteTrack: Multi-Object Tracking by Associating Every Detection Box*. 2022. arXiv: 2110.06864 [cs.CV]. URL: https://arxiv.org/abs/2110.06864 (cit. on pp. 13, 14).

A

Questionnaire Results of Preliminary User Tests

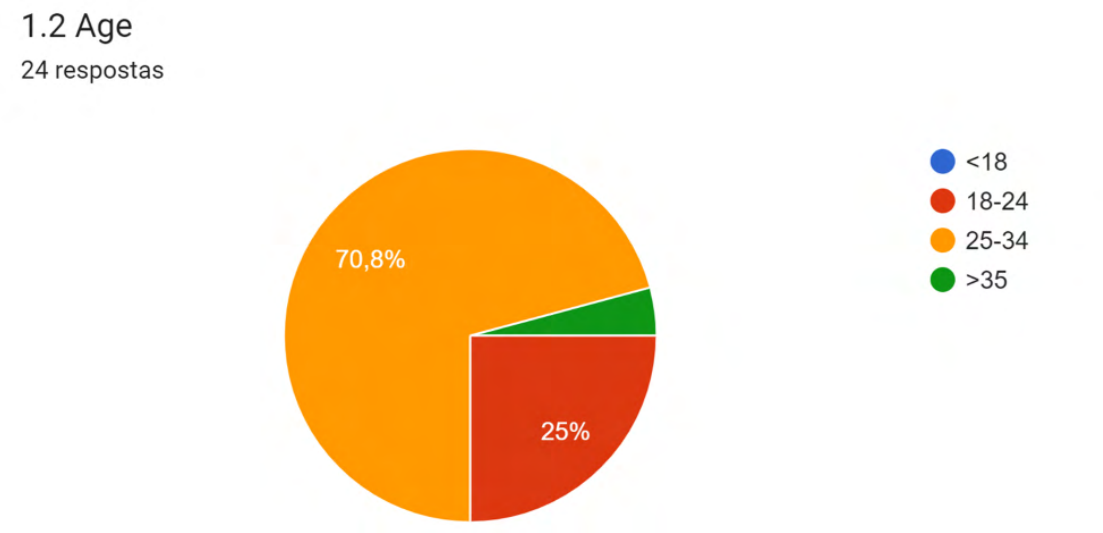


Figure A.1: Age.

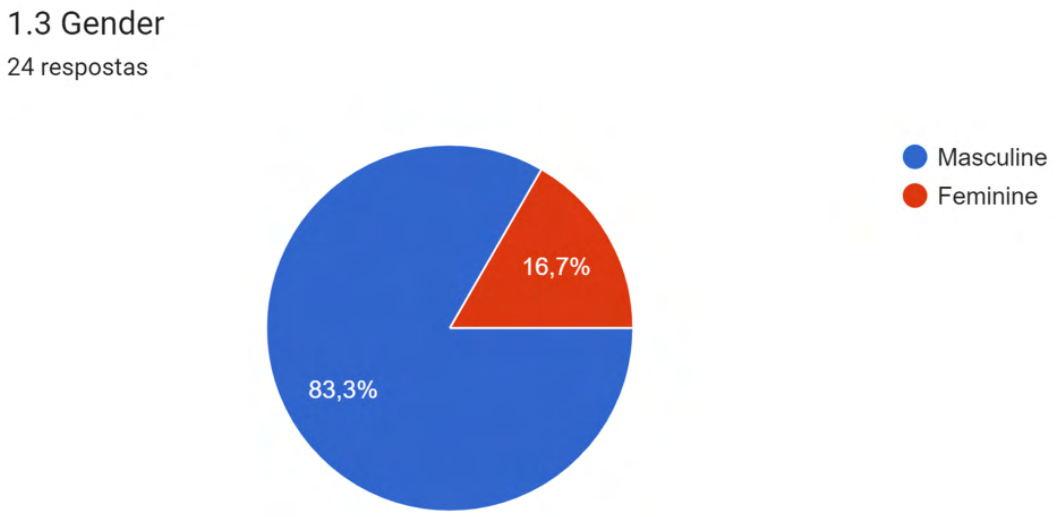


Figure A.2: Gender.

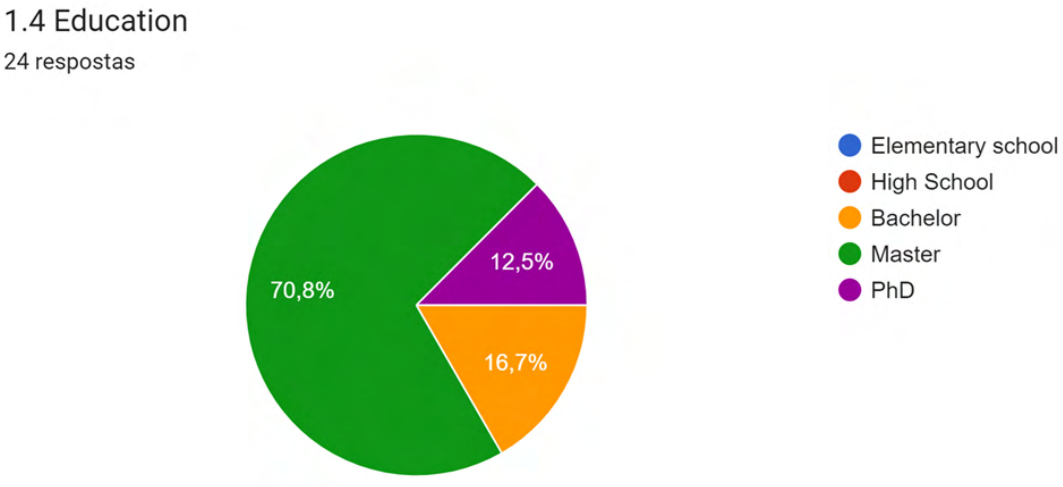


Figure A.3: Education.

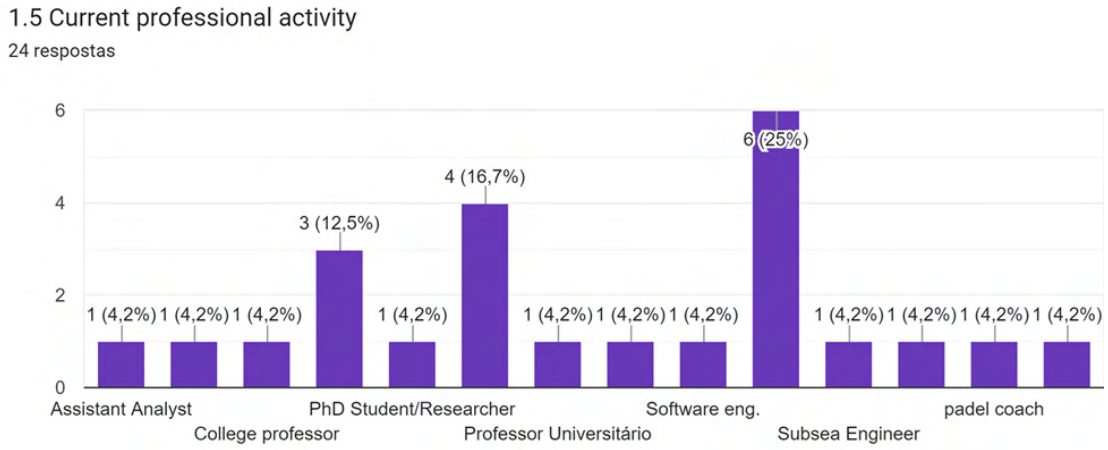


Figure A.4: Current professional activity.

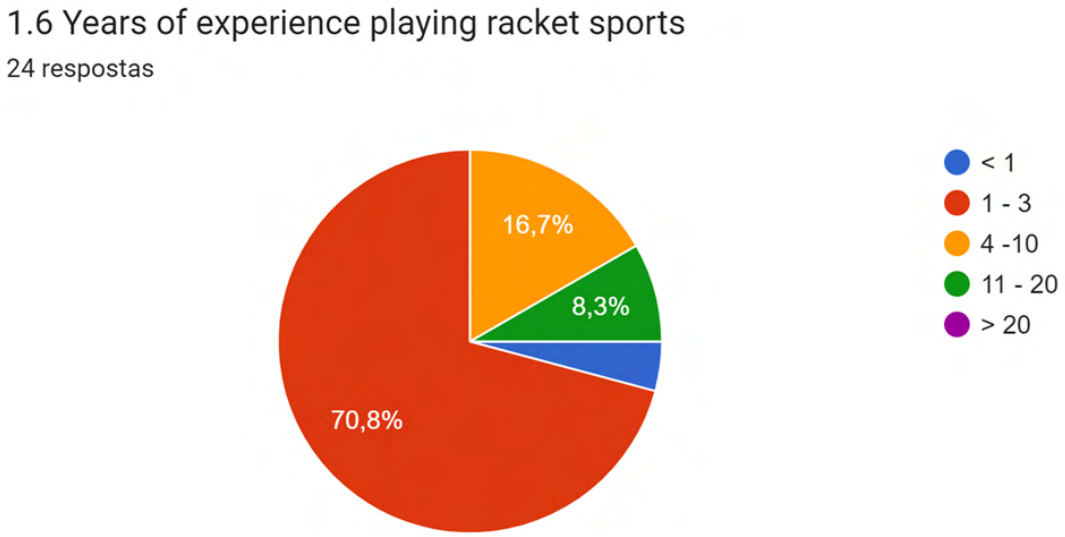


Figure A.5: Years of experience playing racket sports.

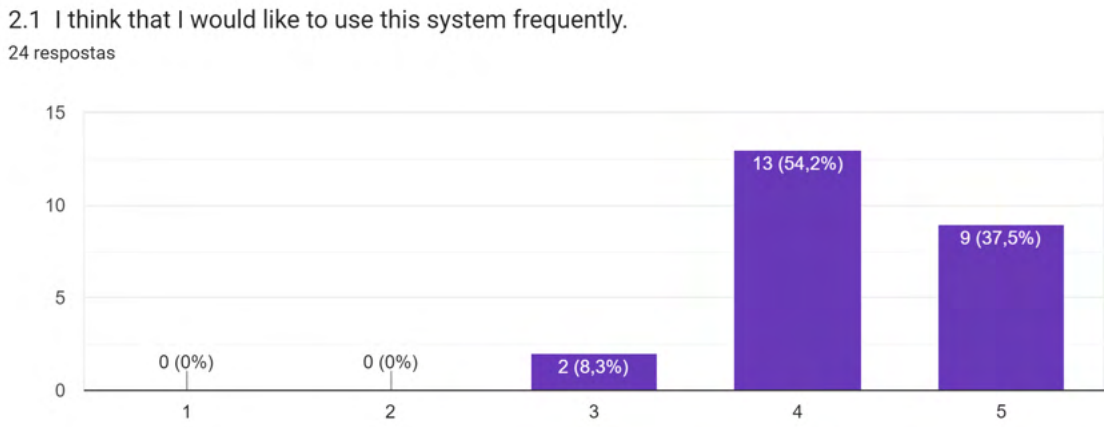


Figure A.6: System Usability Scale: I think that I would like to use this system frequently.

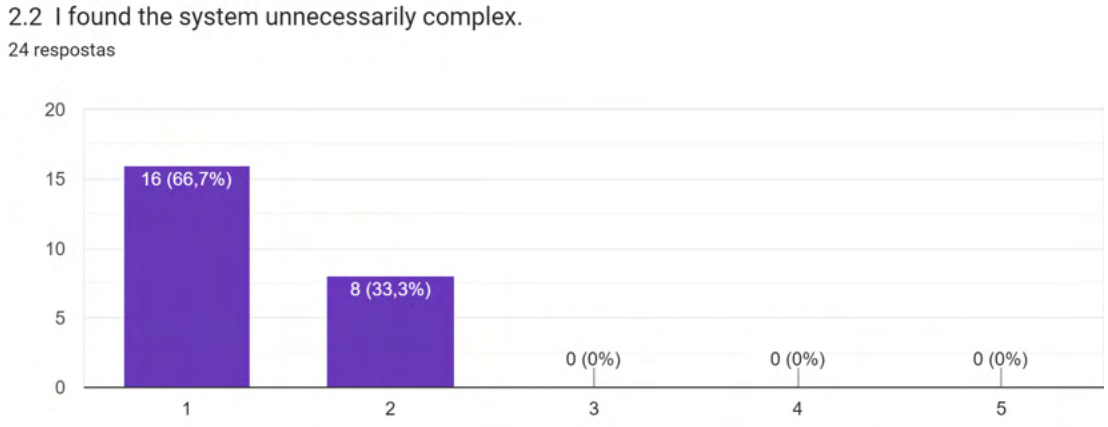


Figure A.7: System Usability Scale: I found the system unnecessarily complex.

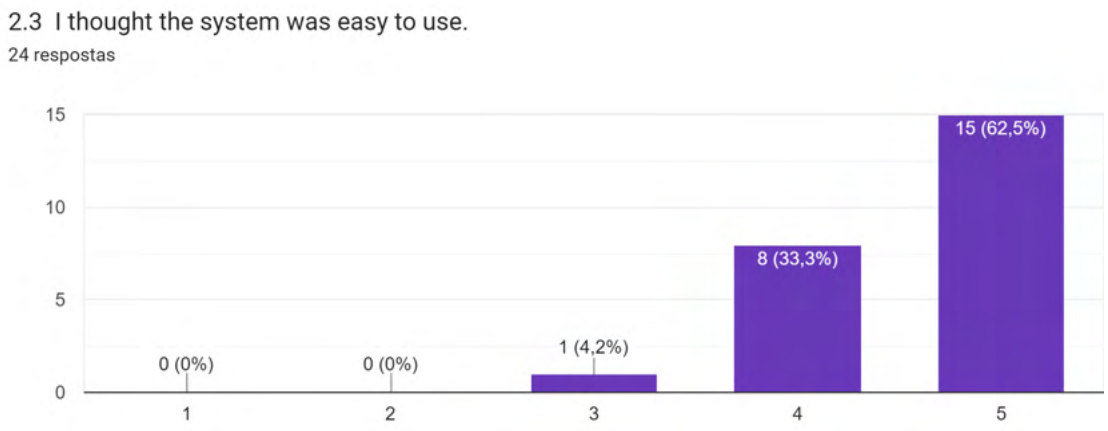


Figure A.8: System Usability Scale: I thought the system was easy to use.

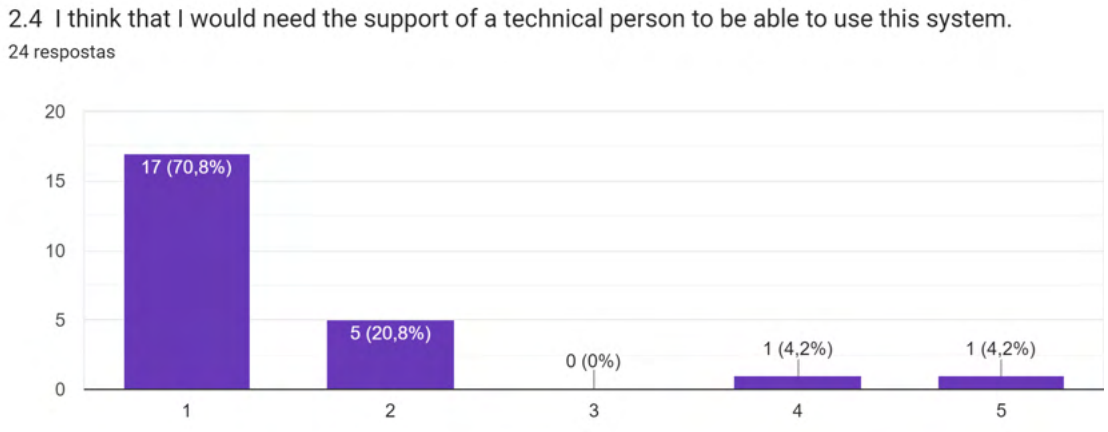


Figure A.9: System Usability Scale: I think that I would need the support of a technical person to be able to use this system.

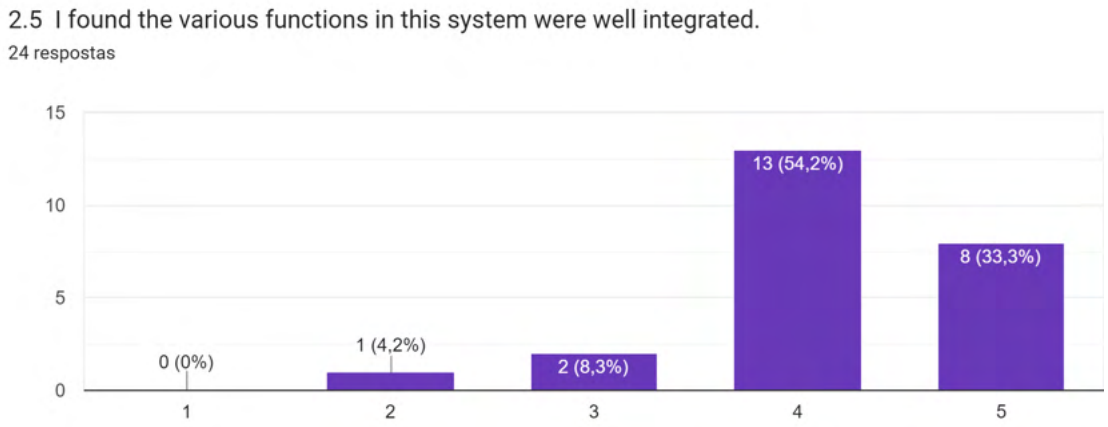


Figure A.10: System Usability Scale: I found the various functions in this system were well integrated.

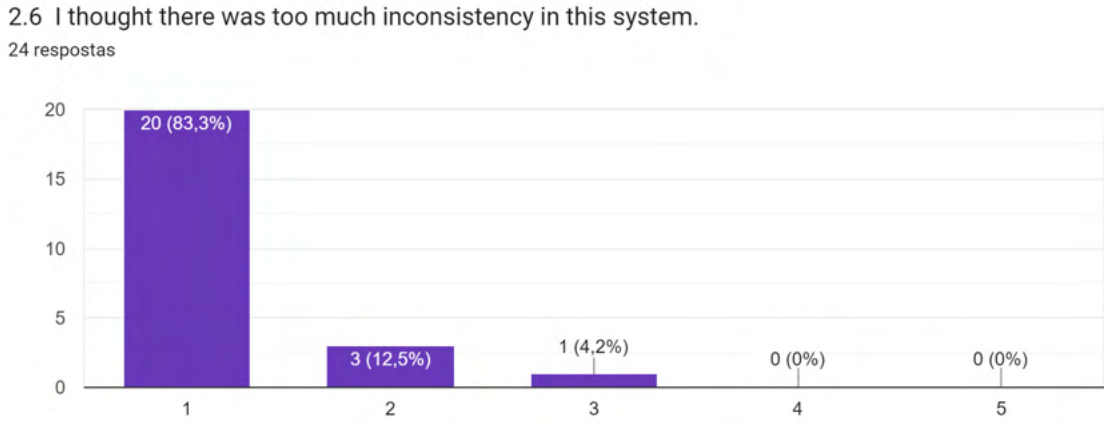


Figure A.11: System Usability Scale: I thought there was too much inconsistency in this system.

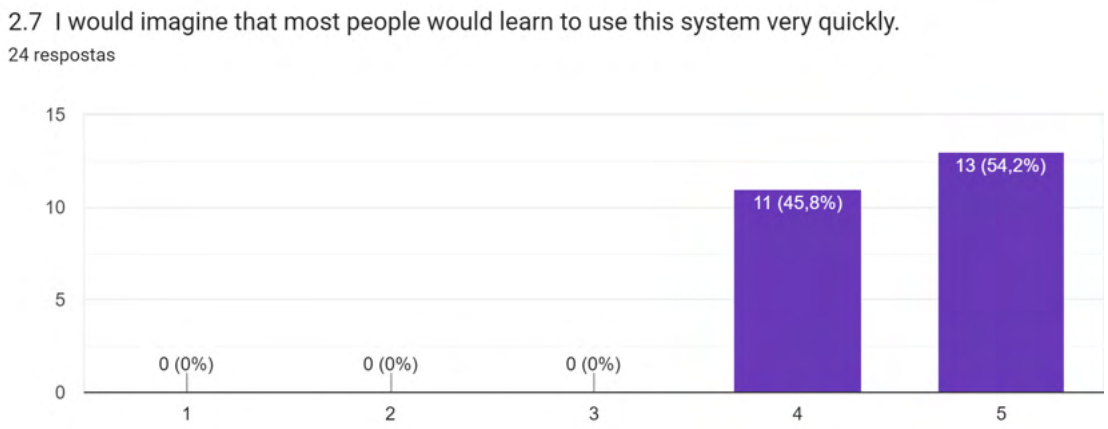


Figure A.12: System Usability Scale: I would imagine that most people would learn to use this system very quickly.

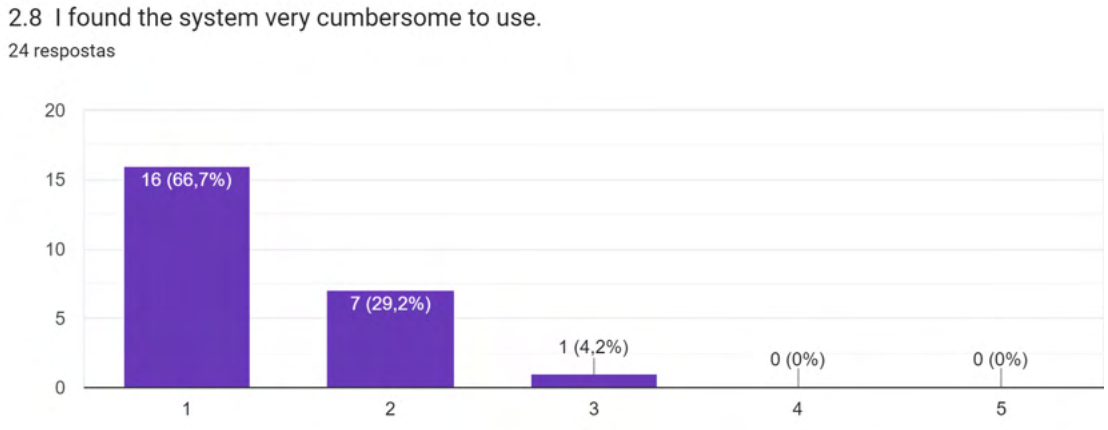


Figure A.13: System Usability Scale: I found the system very cumbersome to use.

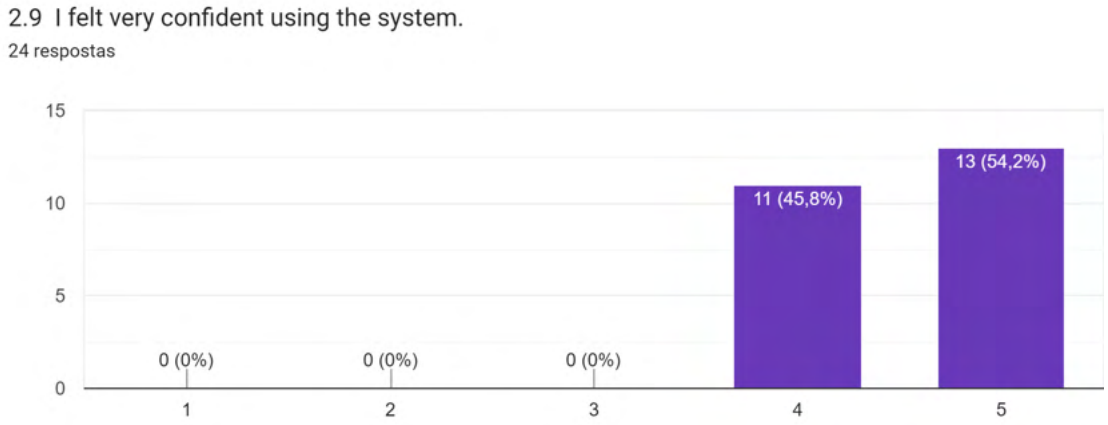


Figure A.14: System Usability Scale: I felt very confident using the system.

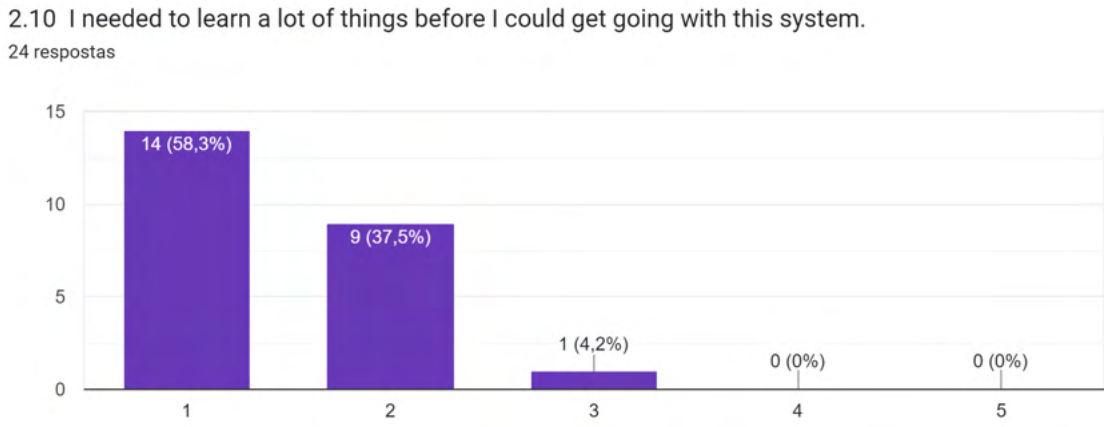


Figure A.15: System Usability Scale: I needed to learn a lot of things before I could get going with this system.

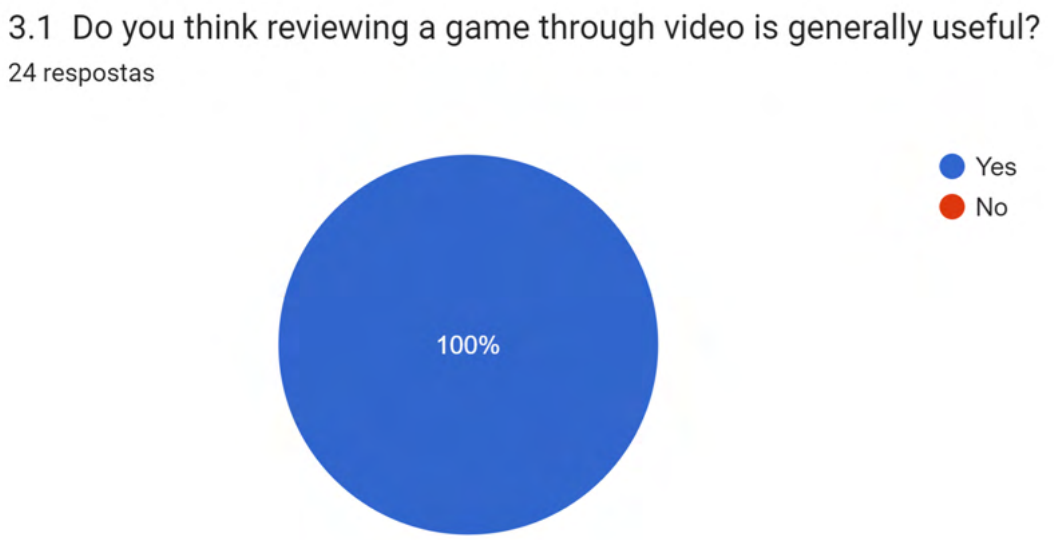


Figure A.16: Do you think reviewing a game through video is generally useful?

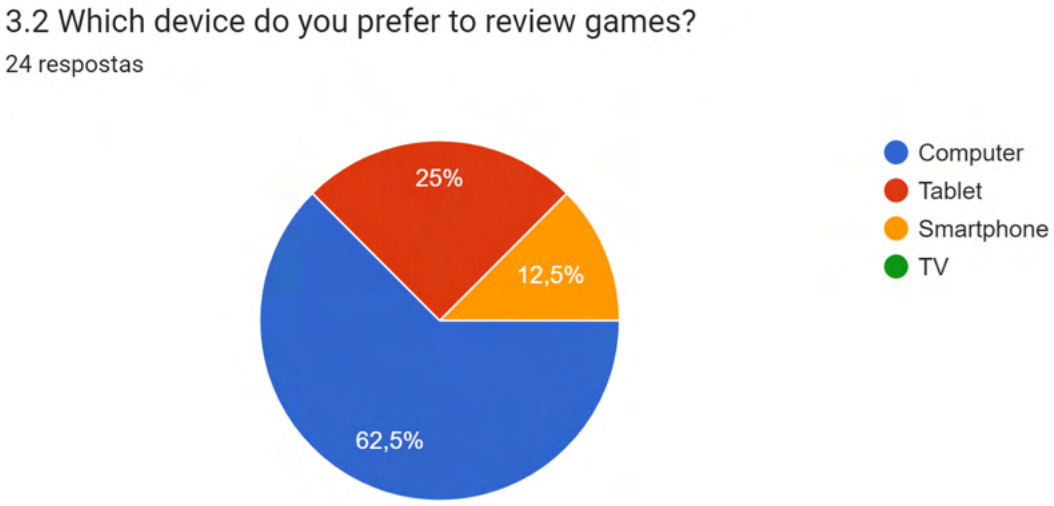


Figure A.17: Which device do you prefer to review games?

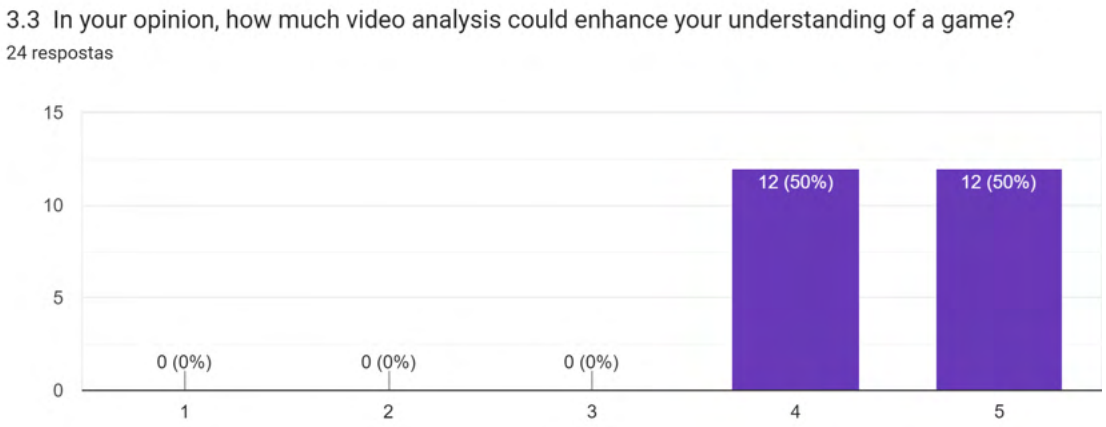


Figure A.18: In your opinion, how much video analysis could enhance your understanding of a game?

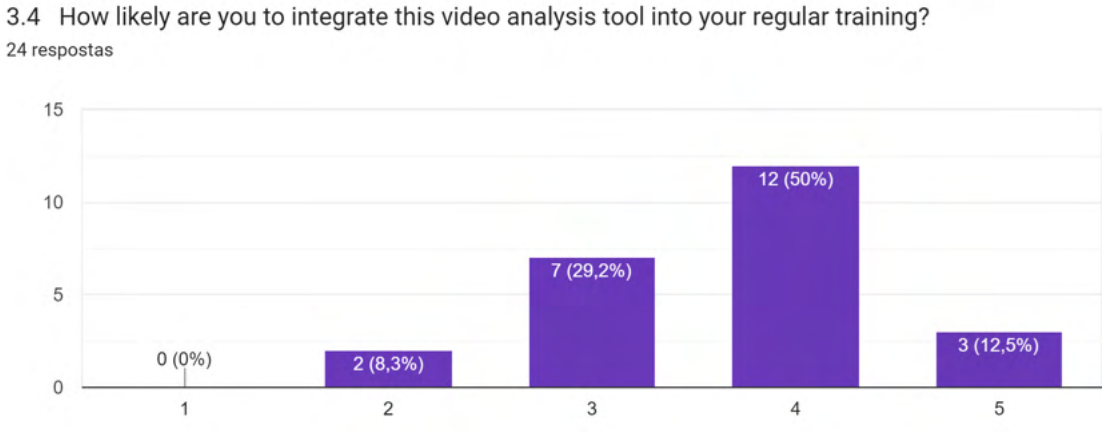


Figure A.19: How likely are you to integrate this video analysis tool into your regular training?

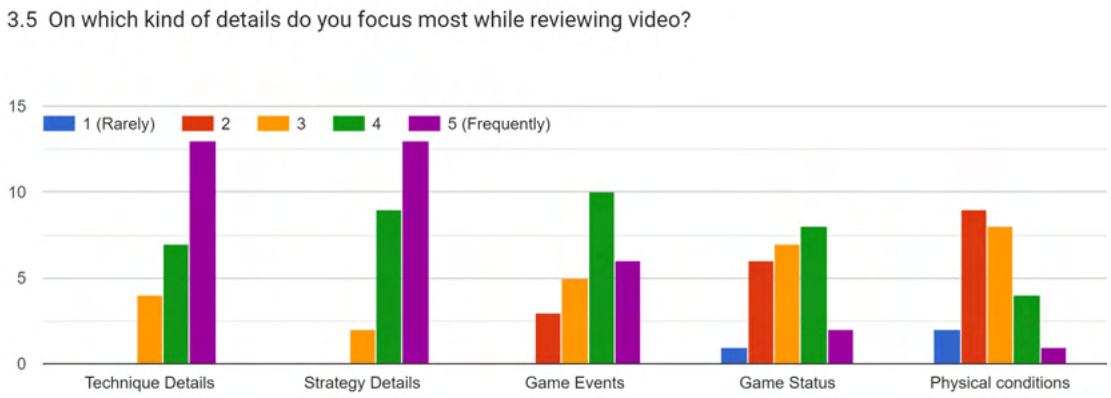


Figure A.20: On which kind of details do you focus most while reviewing video?

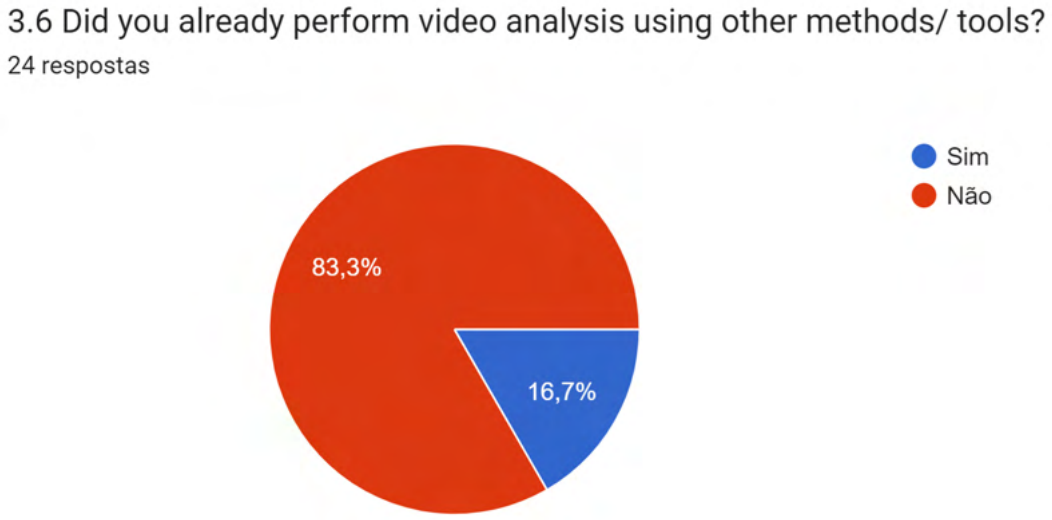


Figure A.21: Did you already perform video analysis using other methods/tools?

3.7 If yes, describe the other methods, tools and compare them with this prototype?

4 answers

Youtube video only

The coach records a video and then shows to the players

Another method for video analysis i have used in the past, is manual reviewing video footage in the context of sports, with a trainer. Compared to it, the impact this prototype would have on the analysis is clear and positive. It agelizes both the process of finding the moments you want to analyse and the examination of the moment.

A software that was used to record the match and was able to in a live environment perceive when there was a point and for each player and team

Figure A.22: If yes, describe the other methods, tools and compare them with this prototype?

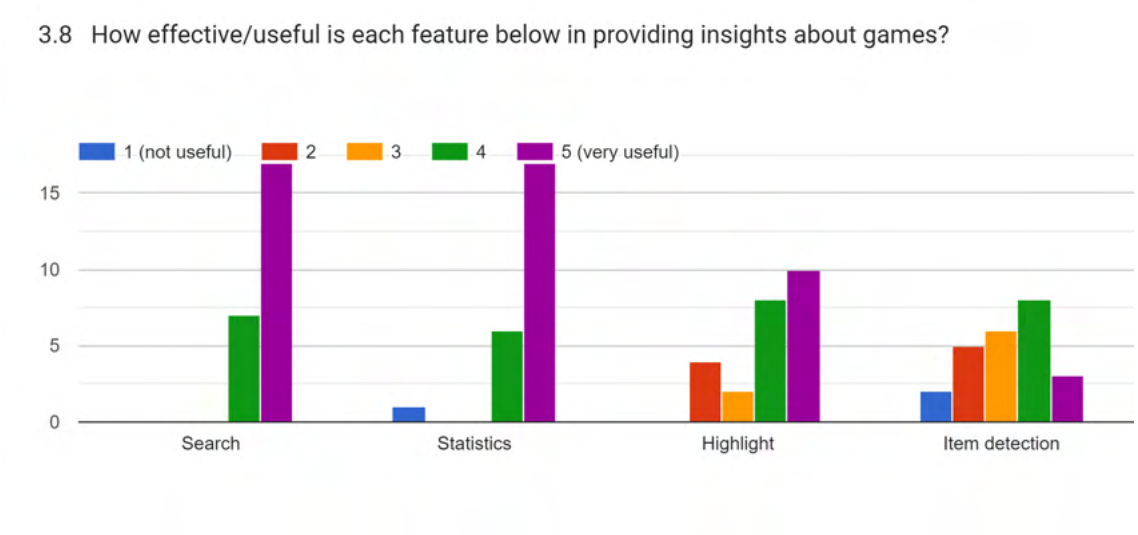


Figure A.23: How effective/useful is each feature below in providing insights about games?

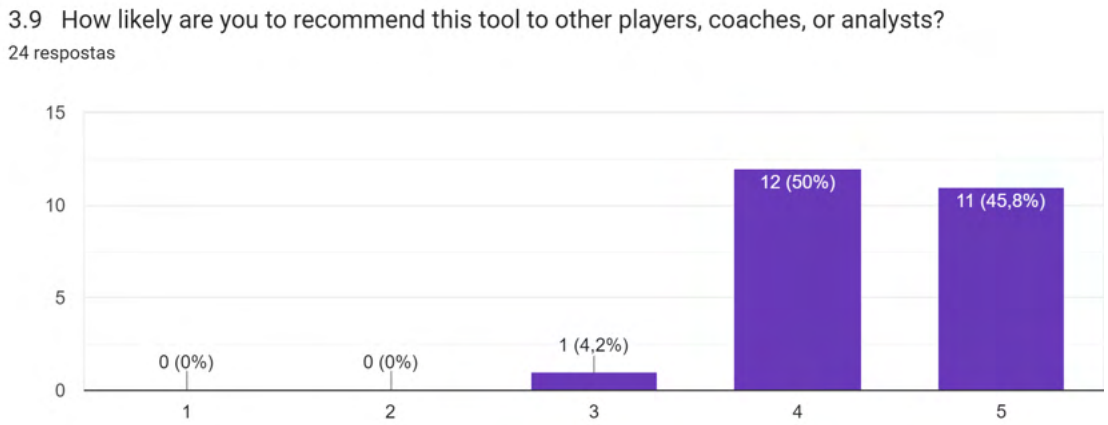


Figure A.24: How likely are you to recommend this tool to other players, coaches, or analysts?

B

Questionnaire Results of Final User Tests

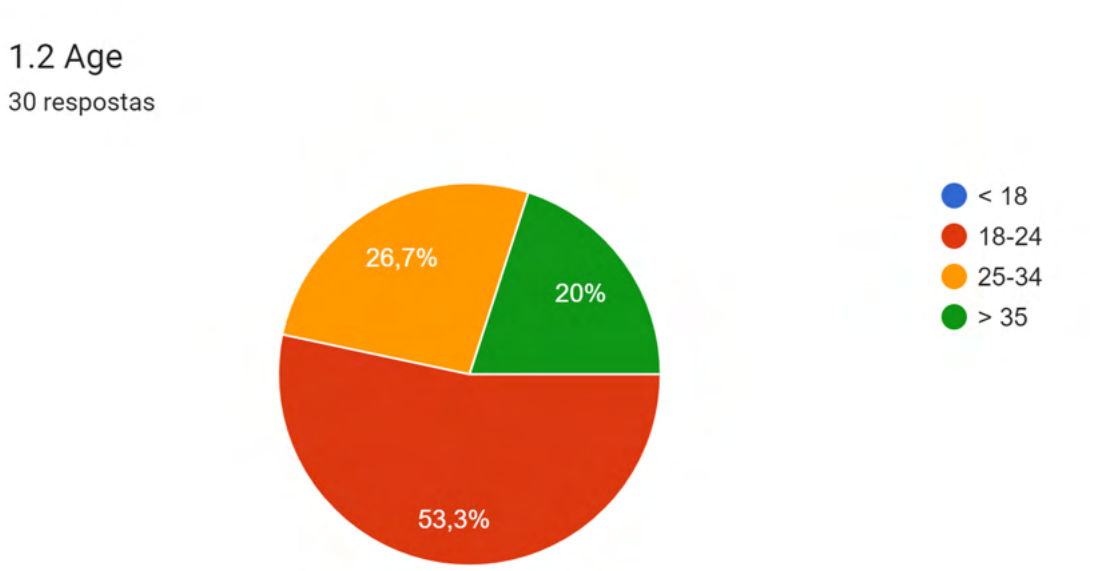


Figure B.1: Age.

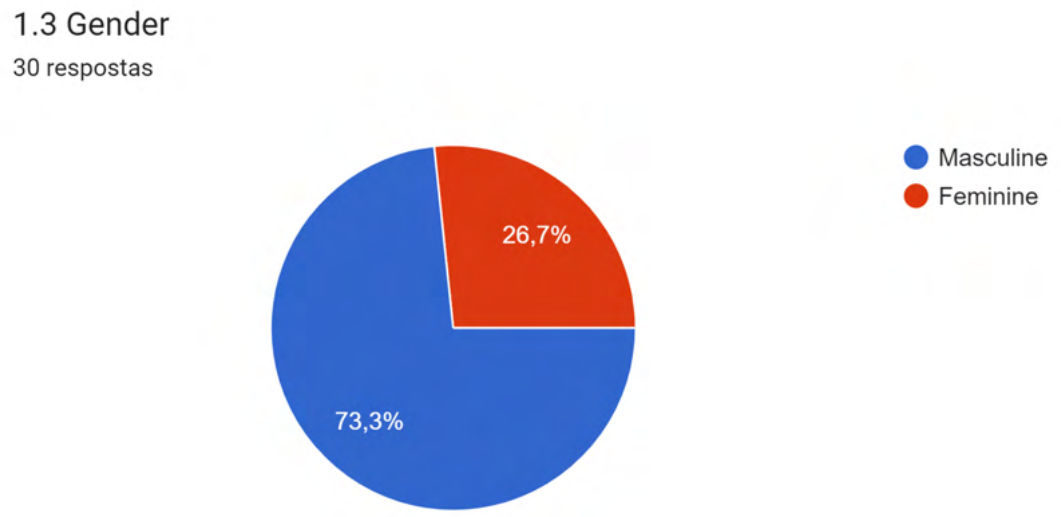


Figure B.2: Gender.

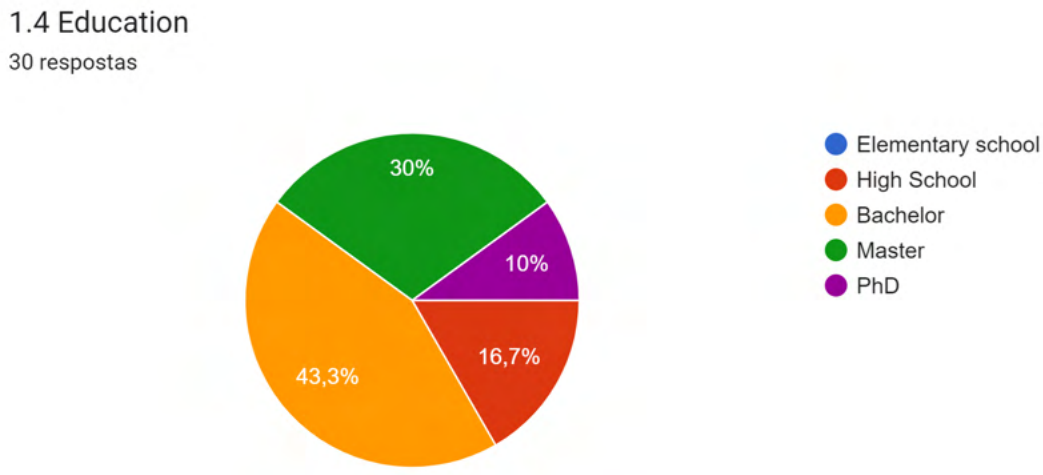


Figure B.3: Education.

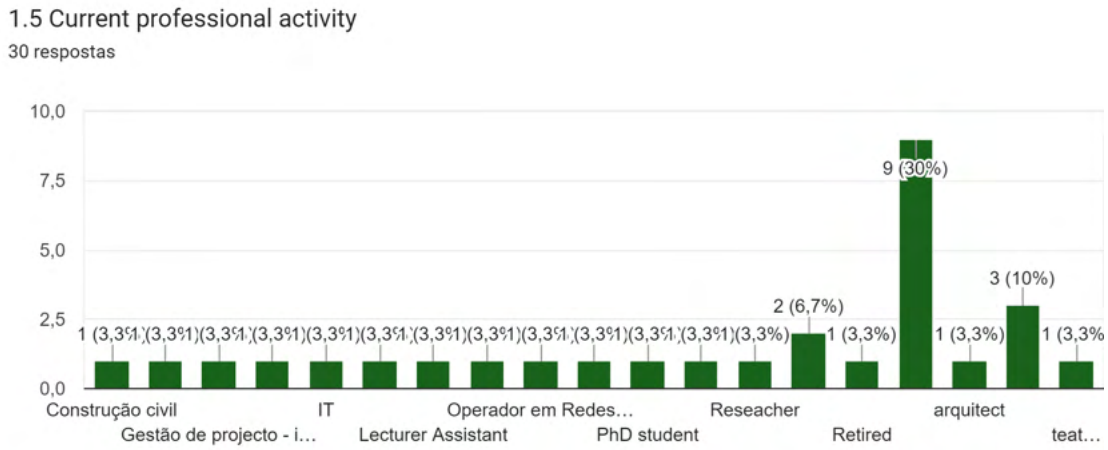


Figure B.4: Current professional activity.

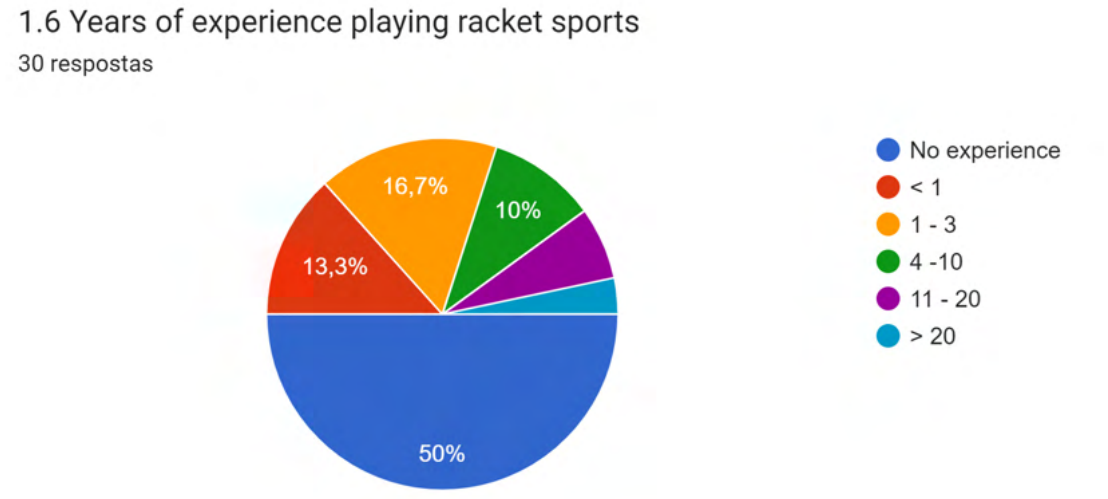


Figure B.5: Years of experience playing racket sports.

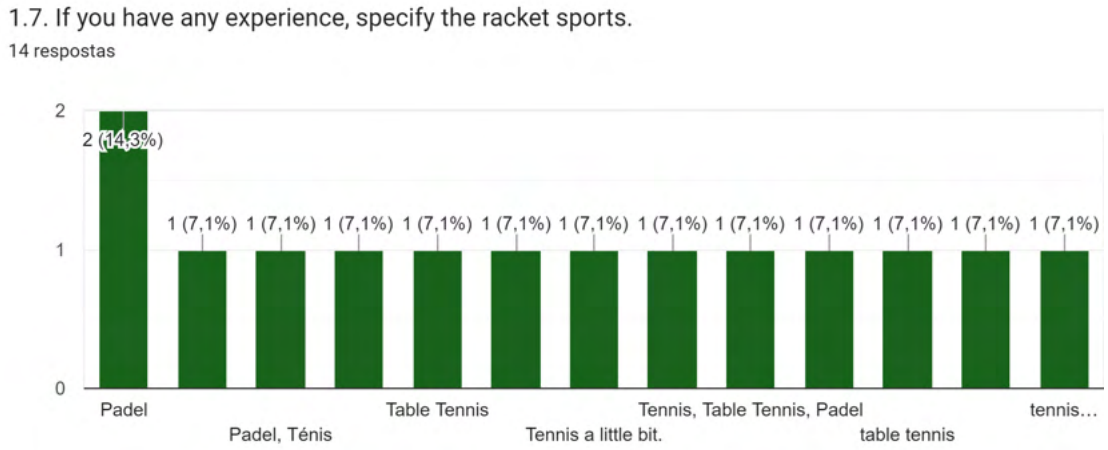


Figure B.6: If you have any experience, specify the racket sports.

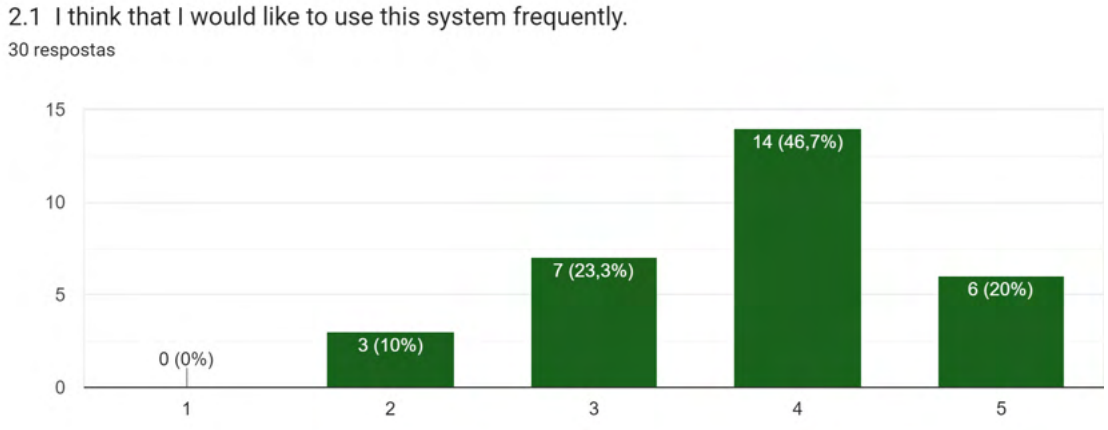


Figure B.7: System Usability Scale: I think that I would like to use this system frequently.

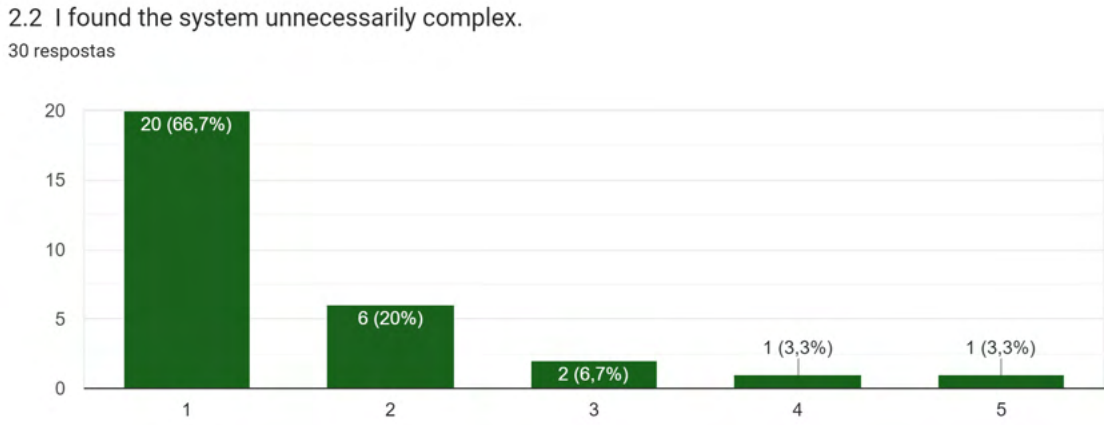


Figure B.8: System Usability Scale: I found the system unnecessarily complex.

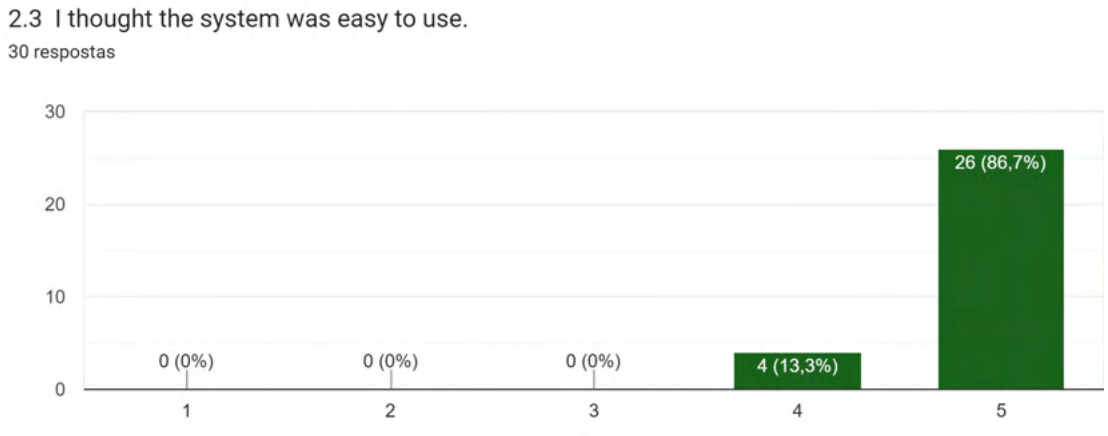


Figure B.9: System Usability Scale: I thought the system was easy to use.

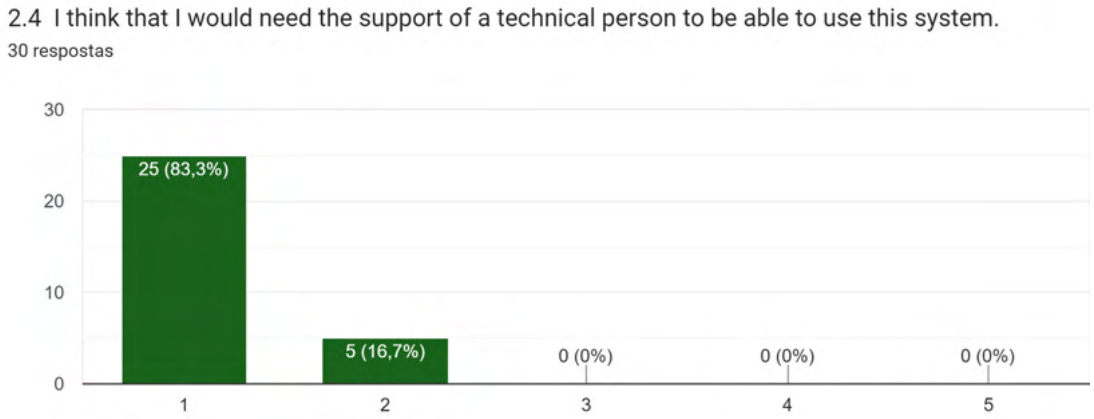


Figure B.10: System Usability Scale: I think that I would need the support of a technical person to be able to use this system.

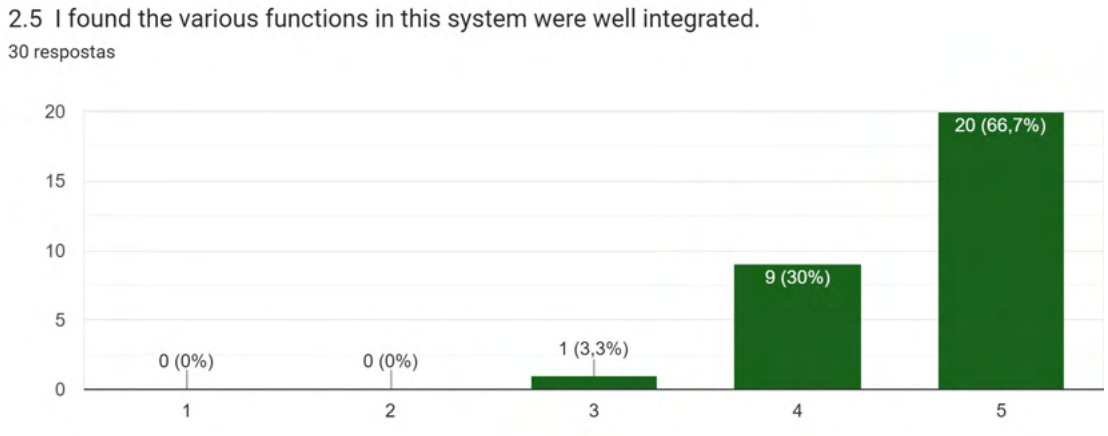


Figure B.11: System Usability Scale: I found the various functions in this system were well integrated.

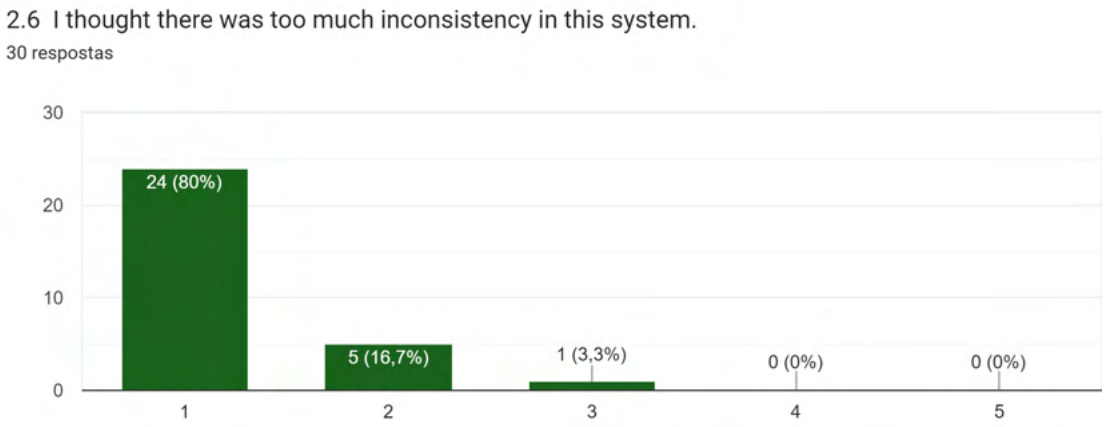


Figure B.12: System Usability Scale: I thought there was too much inconsistency in this system.

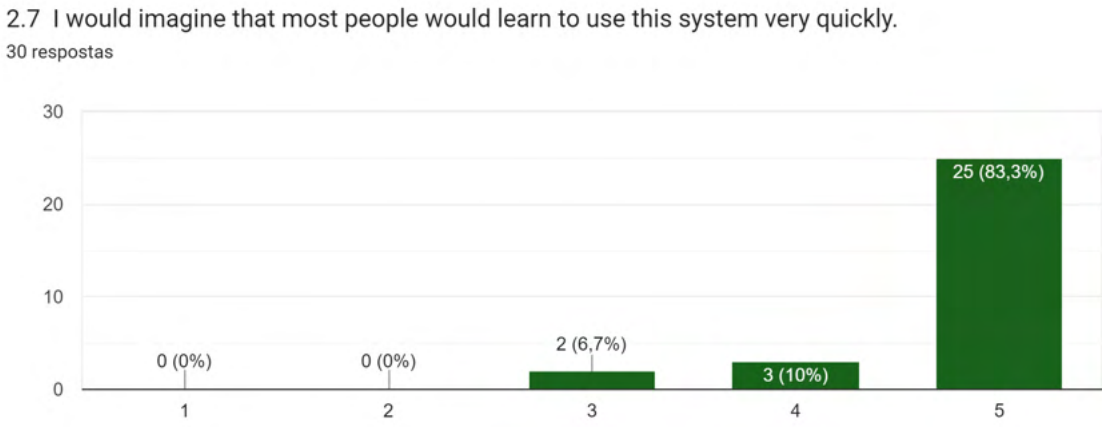


Figure B.13: System Usability Scale: I would imagine that most people would learn to use this system very quickly.

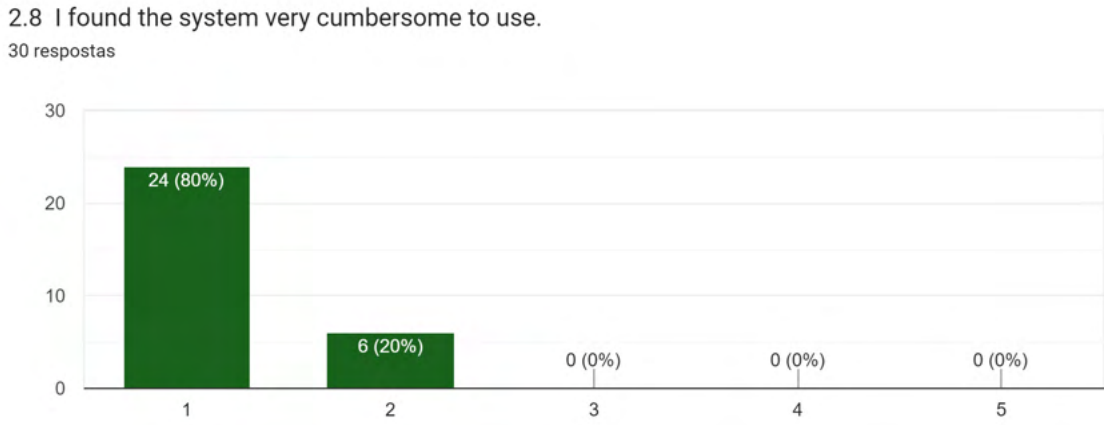


Figure B.14: System Usability Scale: I found the system very cumbersome to use.

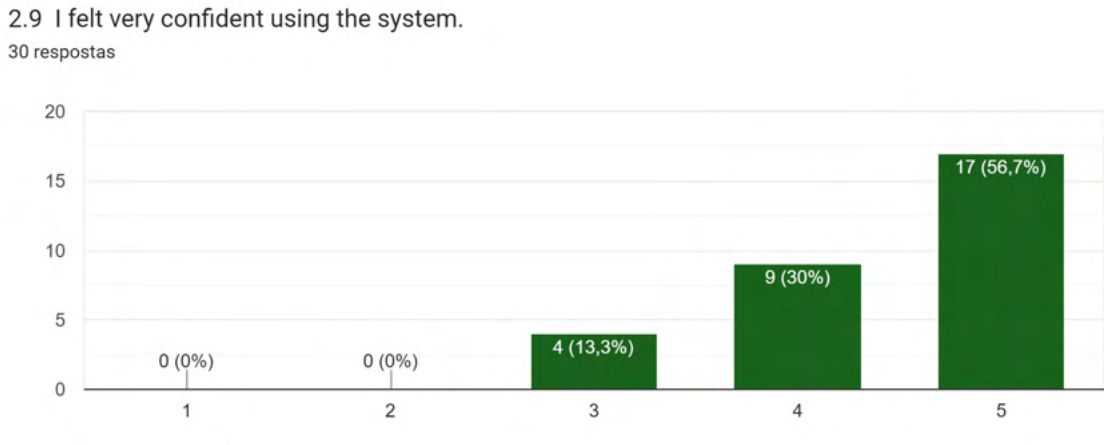


Figure B.15: System Usability Scale: I felt very confident using the system.

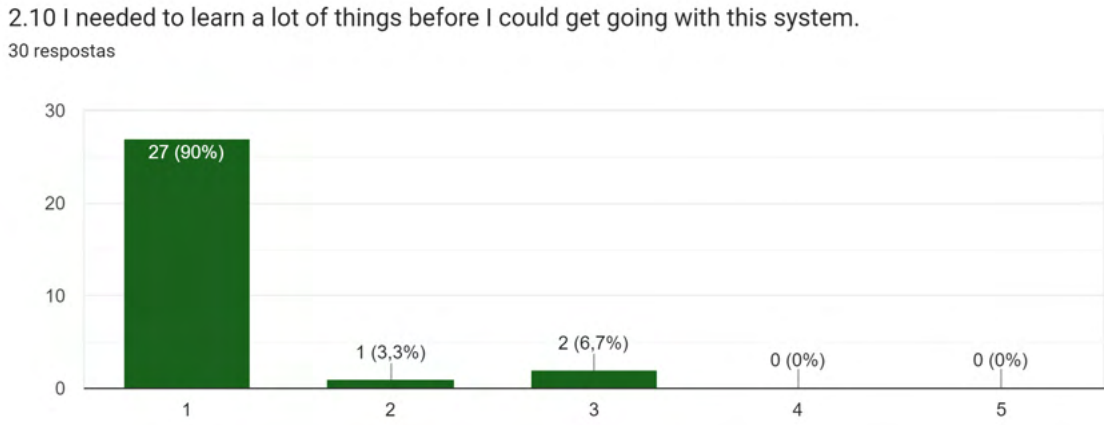


Figure B.16: System Usability Scale: I needed to learn a lot of things before I could get going with this system.

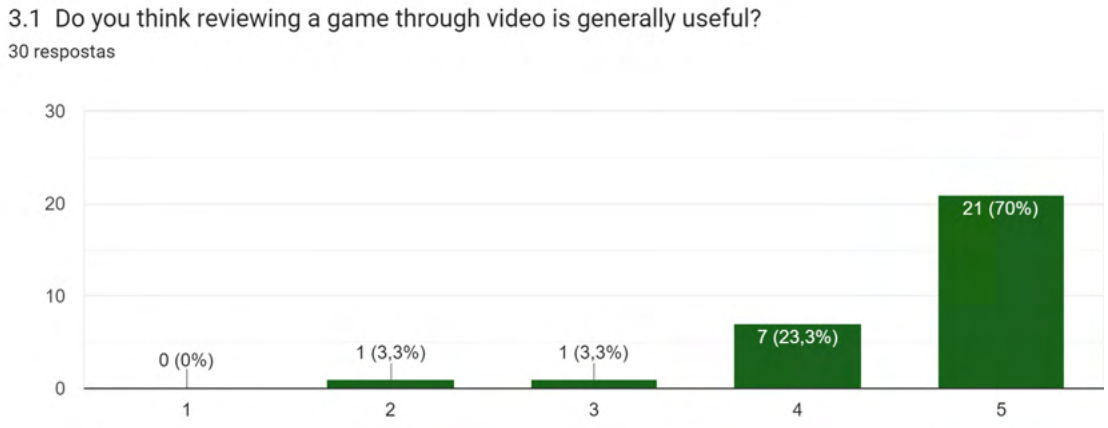


Figure B.17: Do you think reviewing a game through video is generally useful?

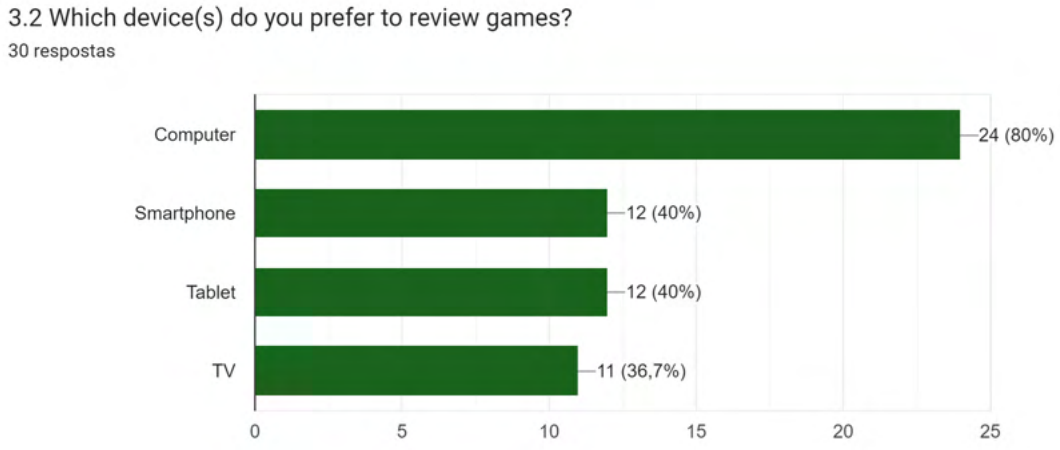


Figure B.18: Which device(s) do you prefer to review games?

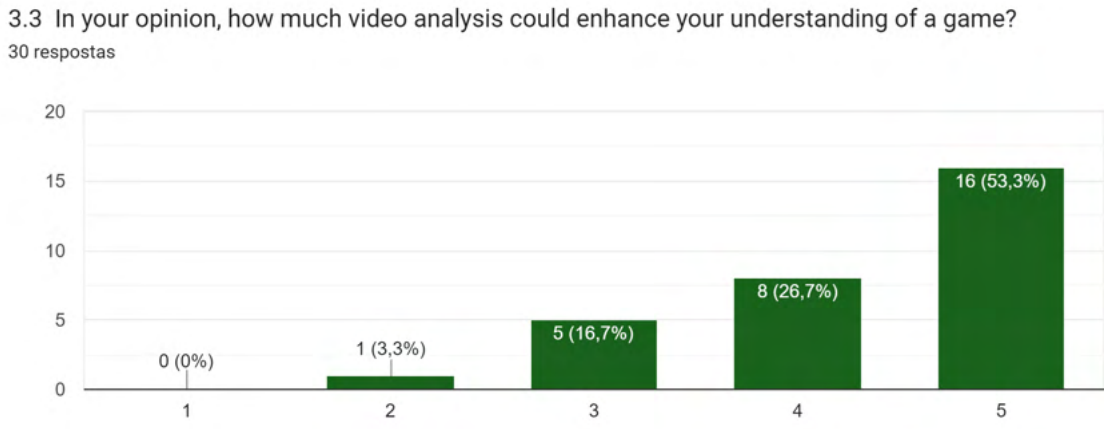


Figure B.19: In your opinion, how much video analysis could enhance your understanding of a game?

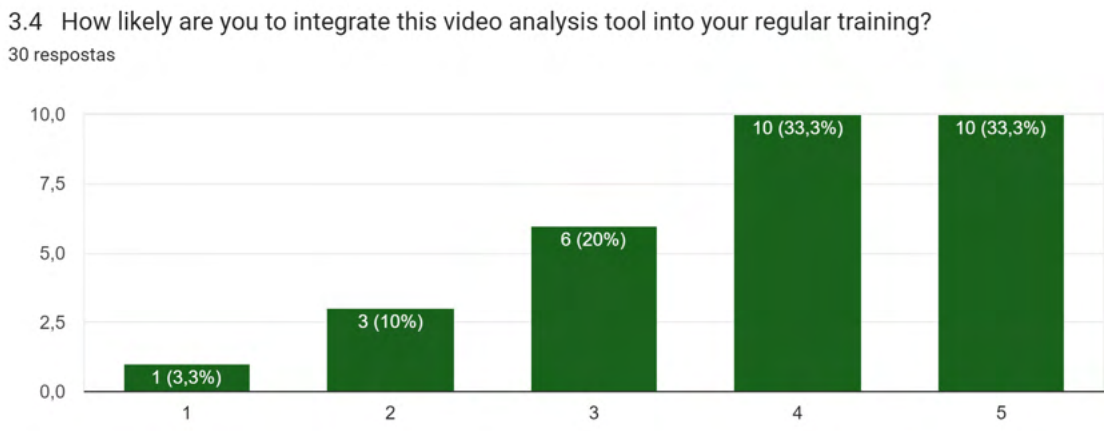


Figure B.20: How likely are you to integrate this video analysis tool into your regular training?

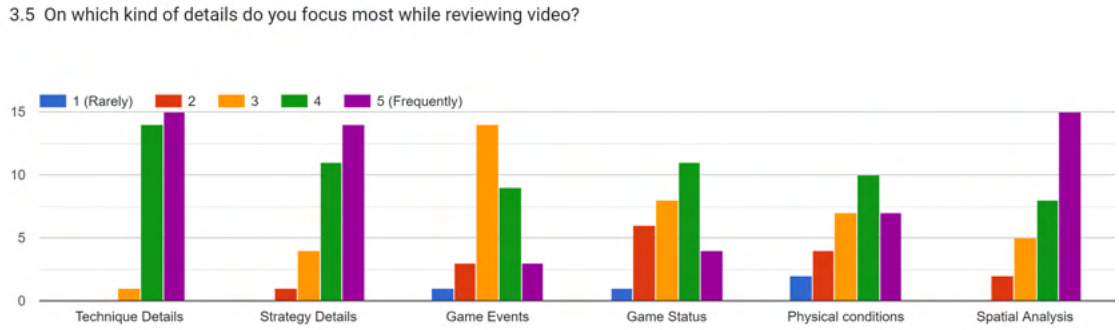


Figure B.21: On which kind of details do you focus most while reviewing video?

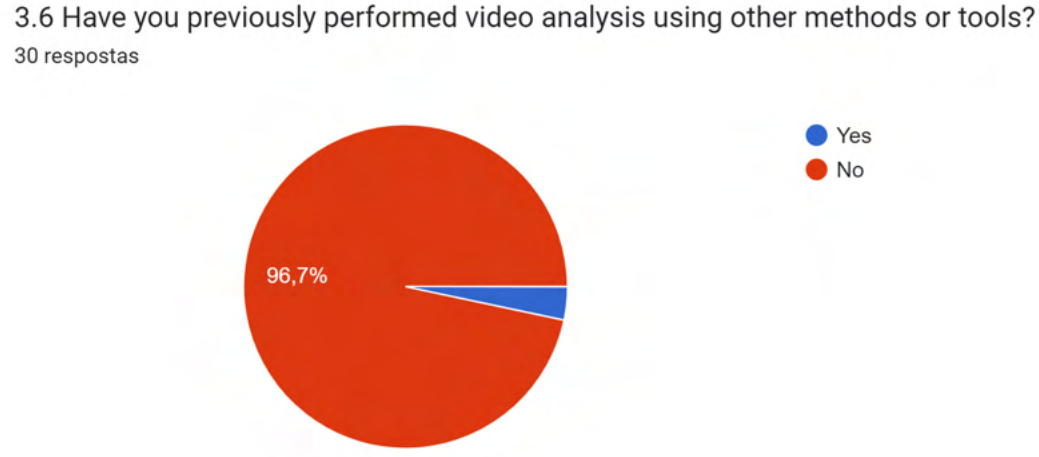


Figure B.22: Have you previously performed video analysis using other methods or tools?

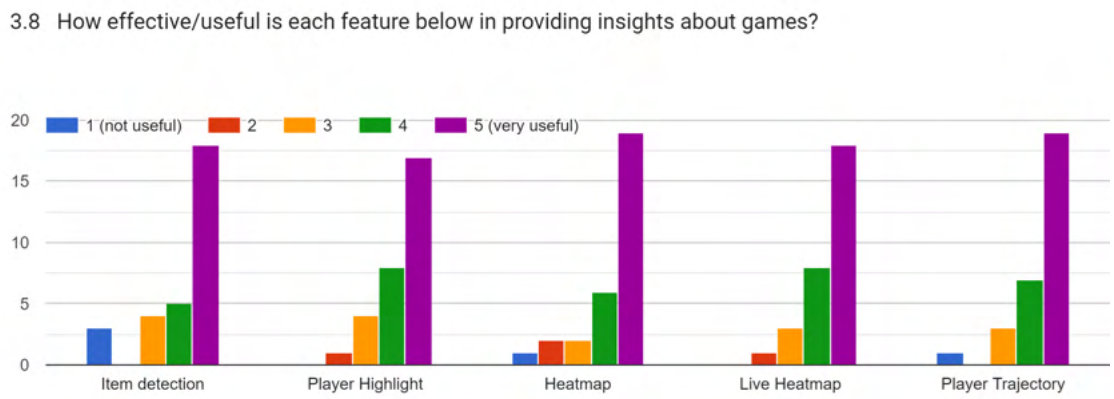


Figure B.23: How effective/useful is each feature below in providing insights about games?

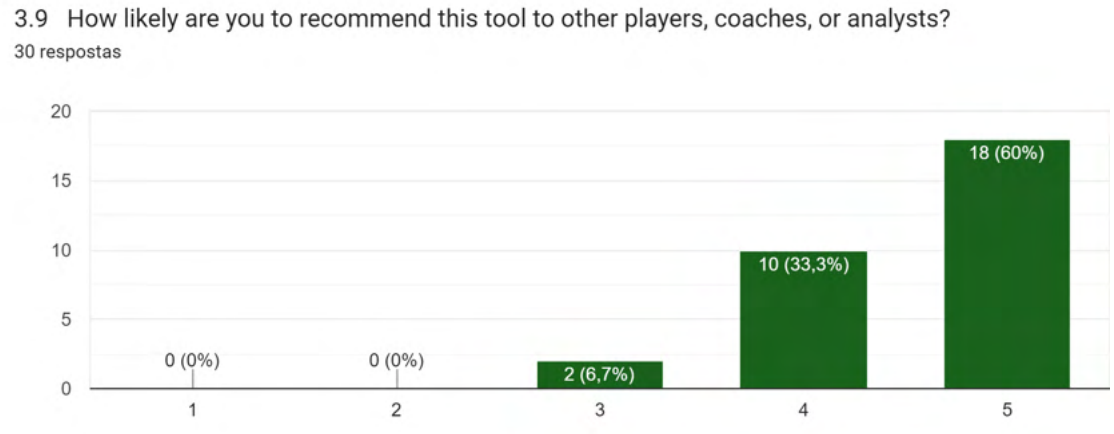


Figure B.24: How likely are you to recommend this tool to other players, coaches, or analysts?

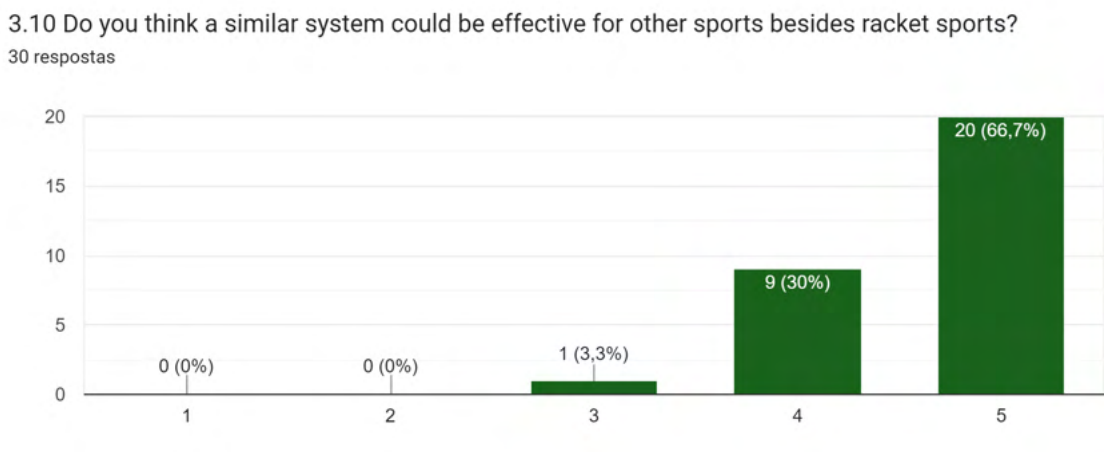


Figure B.25: Do you think a similar system could be effective for other sports besides racket sports?

C

User Experience Questionnaire Results

Items																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
6	7	4	1	2	5	6	6	2	5	6	1	7	6	6	5	2	3	1	6	1	6	1	3	2	3
5	6	3	2	2	5	6	4	2	2	6	2	6	5	5	5	3	2	2	7	2	6	2	2	3	6
6	6	1	2	2	5	6	4	2	2	6	1	6	6	7	7	1	2	1	7	2	5	1	2	1	7
7	7	2	1	1	7	7	7	1	2	7	1	7	7	6	7	1	1	2	7	1	7	1	1	1	6
7	7	1	1	1	6	7	4	2	2	6	1	7	7	6	7	1	2	1	7	1	7	1	2	1	7
6	6	4	2	2	5	6	2	4	3	4	3	6	6	2	6	2	3	3	2	7	4	6	5	5	4
6	6	2	1	2	6	6	4	2	2	6	1	6	6	6	7	4	3	1	6	3	6	2	2	2	6
5	7	2	1	2	3	5	4	1	2	6	3	7	6	7	4	1	4	1	6	2	6	1	1	1	6
6	7	4	1	2	5	6	4	3	3	5	2	7	5	6	6	3	4	2	7	2	5	2	2	3	6
7	7	1	2	1	7	7	4	2	1	7	1	7	7	7	7	1	1	1	7	1	7	1	1	1	7
6	7	2	1	1	7	7	4	2	1	6	1	7	6	6	7	1	1	1	6	2	7	1	2	1	7
7	7	2	1	1	6	6	4	1	2	6	1	7	7	6	6	1	1	2	6	1	7	1	1	1	6
7	5	1	2	1	6	6	5	1	2	7	1	7	6	5	6	1	2	1	6	1	7	1	2	2	6
7	7	3	1	2	6	7	5	1	2	6	1	7	5	5	6	1	2	2	7	6	6	2	3	1	5
6	7	1	1	1	6	7	5	2	2	7	1	7	7	6	6	1	2	1	7	2	6	1	2	1	6
7	7	2	1	1	5	7	7	1	3	7	1	7	6	6	7	1	2	1	7	1	7	1	2	1	6
7	7	2	1	1	7	7	7	1	3	7	1	7	7	5	7	4	1	1	6	1	7	1	1	1	6
7	7	2	1	1	7	6	1	7	2	7	1	7	7	6	7	4	1	1	7	1	7	1	1	1	6
6	7	4	1	5	3	5	3	4	4	4	3	7	5	3	5	1	3	1	4	2	5	2	3	2	5
7	7	2	1	1	4	6	4	2	3	7	1	7	5	6	4	2	4	2	6	1	7	1	3	1	6
5	6	2	1	2	6	6	4	2	3	6	2	6	5	6	5	2	3	2	6	2	6	3	2	1	6
6	7	3	2	3	5	5	6	3	5	6	2	7	6	5	6	2	3	1	6	2	6	1	2	2	6
5	7	4	1	1	5	5	7	1	2	7	1	7	6	4	5	4	4	2	4	1	6	2	1	2	5
6	6	2	2	2	5	6	3	1	2	6	2	6	6	5	6	4	2	2	6	1	7	1	2	2	7
6	6	3	1	2	6	6	7	1	5	6	2	7	5	6	6	4	2	2	6	2	5	2	2	2	5
5	6	1	2	2	6	6	5	1	3	5	2	7	6	6	6	2	2	2	7	1	6	2	1	3	7
6	6	2	2	2	6	6	6	4	5	5	2	6	5	4	5	2	2	2	6	2	6	4	5	2	6
6	7	2	1	2	5	6	4	4	2	7	2	6	6	7	7	1	1	1	6	1	7	1	3	1	6
5	7	1	1	2	5	6	7	1	2	7	2	7	6	6	6	1	2	2	6	1	5	1	2	1	6
7	7	2	1	1	6	6	4	2	3	6	1	7	6	5	7	1	1	1	7	1	7	1	1	1	6

Figure C.1: User Experience Questionnaire data.

Items																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
2	3	0	3	2	1	2	2	2	-1	2	3	3	2	2	1	2	1	3	2	3	2	3	1	2	-1
1	2	1	2	2	1	2	0	2	2	2	2	2	1	1	1	1	2	2	3	2	2	2	2	1	2
2	2	3	2	2	1	2	0	2	2	2	3	2	2	3	3	3	2	3	3	2	1	3	2	3	3
3	3	2	3	3	3	3	3	3	2	3	3	3	3	2	3	3	3	2	3	3	3	3	3	3	2
3	3	3	3	3	2	3	0	2	2	2	3	3	3	2	3	3	2	3	3	3	3	3	2	3	3
2	2	0	2	2	1	2	-2	0	1	0	1	2	2	-2	2	2	1	1	-2	-3	0	-2	-1	-1	0
2	2	2	3	2	2	2	0	2	2	2	3	2	2	2	3	0	1	3	2	1	2	2	2	2	2
1	3	2	3	2	-1	1	0	3	2	2	1	3	2	3	0	3	0	3	2	2	2	3	3	3	2
2	3	0	3	2	1	2	0	1	1	1	2	3	1	2	2	1	0	2	3	2	1	2	2	1	2
3	3	3	2	3	3	3	0	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	3	2	3	3	3	3	0	2	3	2	3	3	2	2	3	3	3	3	2	2	3	3	2	3	3
3	3	2	3	3	2	2	0	3	2	2	3	3	3	2	2	3	3	2	2	3	3	3	3	3	2
3	1	3	2	3	2	2	1	3	2	3	3	3	2	1	2	3	2	3	2	3	3	3	2	2	2
3	3	1	3	2	2	3	1	3	2	2	3	3	1	1	2	3	2	2	3	-2	2	2	1	3	1
2	3	3	3	3	2	3	1	2	2	3	3	3	3	2	2	3	2	3	3	2	2	3	2	3	2
3	3	2	3	3	1	3	3	3	1	3	3	3	2	2	3	3	2	3	3	3	3	3	2	3	2
3	3	2	3	3	3	3	3	3	1	3	3	3	3	1	3	0	3	3	2	3	3	3	3	3	2
3	3	2	3	3	3	2	-3	-3	2	3	3	3	3	2	3	0	3	3	3	3	3	3	3	3	2
2	3	0	3	-1	-1	1	-1	0	0	0	1	3	1	-1	1	3	1	3	0	2	1	2	1	2	1
3	3	2	3	3	0	2	0	2	1	3	3	3	1	2	0	2	0	2	2	3	3	3	1	3	2
1	2	2	3	2	2	2	0	2	1	2	2	2	1	2	1	2	1	2	2	2	2	1	2	3	2
2	3	1	2	1	1	1	2	1	-1	2	2	3	2	1	2	2	1	3	2	2	2	3	2	2	2
1	3	0	3	3	1	1	3	3	2	3	3	3	2	0	1	0	0	2	0	3	2	2	3	2	1
2	2	2	2	2	1	2	-1	3	2	2	2	2	2	1	2	0	2	2	2	3	3	3	2	2	3
2	2	1	3	2	2	2	3	3	-1	2	2	3	1	2	2	0	2	2	2	2	1	2	2	2	1
1	2	3	2	2	2	2	1	3	1	1	2	3	2	2	2	2	2	2	3	3	2	2	3	1	3
2	2	2	2	2	2	2	2	0	-1	1	2	2	1	0	1	2	2	2	2	2	2	0	-1	2	2
2	3	2	3	2	1	2	0	0	2	3	2	2	2	3	3	3	3	3	2	3	3	3	1	3	2
1	3	3	3	2	1	2	3	3	2	3	2	3	2	2	2	3	2	2	2	3	1	3	2	3	2
3	3	2	3	3	2	2	0	2	1	2	3	3	2	1	3	3	3	3	3	3	3	3	3	3	2

Figure C.2: User Experience Questionnaire transformed data.

Items																									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
2	3	0	3	2	1	2	2	2	-1	2	3	3	2	2	1	2	1	3	2	3	2	3	1	2	-1
1	2	1	2	2	1	2	0	2	2	2	2	2	1	1	1	1	2	2	3	2	2	2	2	1	2
2	2	3	2	2	1	2	0	2	2	2	3	2	2	3	3	3	2	3	3	2	1	3	2	3	3
3	3	2	3	3	3	3	3	3	2	3	3	3	3	2	3	3	3	2	3	3	3	3	3	3	2
3	3	3	3	3	2	3	0	2	2	2	3	3	3	2	3	3	2	3	3	3	3	3	2	3	3
2	2	0	2	2	1	2	-2	0	1	0	1	2	2	-2	2	2	1	1	-2	-3	0	-2	-1	-1	0
2	2	2	3	2	2	2	0	2	2	2	3	2	2	2	3	0	1	3	2	1	2	2	2	2	2
1	3	2	3	2	-1	1	0	3	2	2	1	3	2	3	0	3	0	3	2	2	2	3	3	3	2
2	3	0	3	2	1	2	0	1	1	1	2	3	1	2	2	1	0	2	3	2	1	2	2	1	2
3	3	3	2	3	3	3	0	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	3	2	3	3	3	3	0	2	3	2	3	3	2	2	3	3	3	3	2	2	3	3	2	3	3
3	3	2	3	3	2	2	0	3	2	2	3	3	3	2	2	3	3	2	2	3	3	3	3	3	2
3	1	3	2	3	2	2	1	3	2	3	3	3	2	1	2	3	2	3	2	3	3	3	2	2	2
3	3	1	3	2	2	3	1	3	2	2	3	3	1	1	2	3	2	2	3	-2	2	2	1	3	1
2	3	3	3	3	2	3	1	2	2	3	3	3	3	2	2	3	2	3	3	2	2	3	2	3	2
3	3	2	3	3	1	3	3	3	1	3	3	3	2	2	3	3	2	3	3	3	3	3	2	3	2
3	3	2	3	3	3	3	3	3	1	3	3	3	3	1	3	0	3	3	2	3	3	3	3	3	2
3	3	2	3	3	3	2	-3	-3	2	3	3	3	3	2	3	0	3	3	3	3	3	3	3	3	2
2	3	0	3	-1	-1	1	-1	0	0	0	1	3	1	-1	1	3	1	3	0	2	1	2	1	2	1
3	3	2	3	3	0	2	0	2	1	3	3	3	1	2	0	2	0	2	2	3	3	3	1	3	2
1	2	2	3	2	2	2	0	2	1	2	2	2	1	2	1	2	1	2	2	2	2	1	2	3	2
2	3	1	2	1	1	1	2	1	-1	2	2	3	2	1	2	2	1	3	2	2	2	3	2	2	2
1	3	0	3	3	1	1	3	3	2	3	3	3	2	0	1	0	0	2	0	3	2	2	3	2	1
2	2	2	2	2	1	2	-1	3	2	2	2	2	2	1	2	0	2	2	2	3	3	3	2	2	3
2	2	1	3	2	2	2	3	3	-1	2	2	3	1	2	2	0	2	2	2	2	1	2	2	2	1
1	2	3	2	2	2	2	1	3	1	1	2	3	2	2	2	2	2	2	3	3	2	2	3	1	3
2	2	2	2	2	2	2	2	0	-1	1	2	2	1	0	1	2	2	2	2	2	2	0	-1	2	2
2	3	2	3	2	1	2	0	0	2	3	2	2	2	3	3	3	3	3	2	3	3	3	1	3	2
1	3	3	3	2	1	2	3	3	2	3	2	3	2	2	2	3	2	2	2	3	1	3	2	3	2
3	3	2	3	3	2	2	0	2	1	2	3	3	2	1	3	3	3	3	3	3	3	3	3	3	2

Figure C.3: User Experience Questionnaire transformed data: scale means per person.

Item	Mean	Variance	Std. Dev.	No.	Left	Right	Scale
1	2.2	0.6	0.7	30	annoying	enjoyable	Attractiveness
2	2.6	0.3	0.6	30	not understandable	understandable	Perspicuity
3	1.8	1.0	1.0	30	creative	dull	Novelty
4	2.7	0.2	0.5	30	easy to learn	difficult to learn	Perspicuity
5	2.3	0.7	0.8	30	valuable	inferior	Stimulation
6	1.5	1.1	1.0	30	boring	exciting	Stimulation
7	2.1	0.4	0.6	30	not interesting	interesting	Stimulation
8	0.7	2.4	1.6	30	unpredictable	predictable	Dependability
9	1.9	1.9	1.4	30	fast	slow	Efficiency
10	1.3	1.3	1.1	30	inventive	conventional	Novelty
11	2.1	0.7	0.9	30	obstructive	supportive	Dependability
12	2.5	0.5	0.7	30	good	bad	Attractiveness
13	2.7	0.2	0.4	30	complicated	easy	Perspicuity
14	2.0	0.5	0.7	30	unlikable	pleasing	Attractiveness
15	1.5	1.3	1.1	30	usual	leading edge	Novelty
16	2.0	0.9	0.9	30	unpleasant	pleasant	Attractiveness
17	2.0	1.4	1.2	30	secure	not secure	Dependability
18	1.8	1.0	1.0	30	motivating	demotivating	Stimulation
19	2.5	0.3	0.6	30	meets expectations	does not meet expectations	Dependability
20	2.1	1.2	1.1	30	inefficient	efficient	Efficiency
21	2.2	2.0	1.4	30	clear	confusing	Perspicuity
22	2.2	0.7	0.8	30	impractical	practical	Efficiency
23	2.4	1.2	1.1	30	organized	cluttered	Efficiency
24	1.9	1.1	1.0	30	attractive	unattractive	Attractiveness
25	2.4	0.9	0.9	30	friendly	unfriendly	Attractiveness
26	1.9	0.8	0.9	30	conservative	innovative	Novelty

Figure C.4: User Experience Questionnaire results.

UEQ Scales (Mean and Variance)		
Attractiveness	2.156	0.35
Perspicuity	2.567	0.27
Efficiency	2.158	0.65
Dependability	1.842	0.41
Stimulation	1.942	0.53
Novelty	1.633	0.68

Figure C.5: User Experience Questionnaire scales (Mean and Variance).

Pragmatic and Hedonic Quality	
Attractiveness	2.16
Pragmatic Quality	2.19
Hedonic Quality	1.79

Figure C.6: User Experience Questionnaire: Pragmatic and Hedonic Quality.

Confidence interval (p=0.05) per item						
Item	Mean	Std. Dev.	N	Confidence	Confidence interval	
1	2.167	0.747	30	0.267	1.899	2.434
2	2.633	0.556	30	0.199	2.434	2.832
3	1.767	1.006	30	0.360	1.407	2.127
4	2.700	0.466	30	0.167	2.533	2.867
5	2.300	0.837	30	0.299	2.001	2.599
6	1.533	1.042	30	0.373	1.161	1.906
7	2.133	0.629	30	0.225	1.908	2.358
8	0.700	1.557	30	0.557	0.143	1.257
9	1.900	1.373	30	0.491	1.409	2.391
10	1.333	1.124	30	0.402	0.931	1.736
11	2.133	0.860	30	0.308	1.825	2.441
12	2.467	0.681	30	0.244	2.223	2.711
13	2.733	0.450	30	0.161	2.572	2.894
14	1.967	0.718	30	0.257	1.710	2.224
15	1.533	1.137	30	0.407	1.127	1.940
16	2.033	0.928	30	0.332	1.701	2.365
17	2.033	1.189	30	0.425	1.608	2.459
18	1.800	0.997	30	0.357	1.443	2.157
19	2.500	0.572	30	0.205	2.295	2.705
20	2.133	1.106	30	0.396	1.738	2.529
21	2.200	1.400	30	0.501	1.699	2.701
22	2.200	0.847	30	0.303	1.897	2.503
23	2.400	1.102	30	0.394	2.006	2.794
24	1.933	1.048	30	0.375	1.558	2.308
25	2.367	0.928	30	0.332	2.035	2.699
26	1.900	0.885	30	0.317	1.583	2.217

Figure C.7: User Experience Questionnaire: Confidence intervals per item.

Confidence intervals (p=0.05) per scale						
Scale	Mean	Std. Dev.	N	Confidence	Confidence interval	
Attractiveness	2.156	0.594	30	0.212	1.943	2.368
Perspicuity	2.567	0.521	30	0.186	2.380	2.753
Efficiency	2.158	0.808	30	0.289	1.869	2.447
Dependability	1.842	0.638	30	0.228	1.613	2.070
Stimulation	1.942	0.730	30	0.261	1.680	2.203
Novelty	1.633	0.825	30	0.295	1.338	1.928

Figure C.8: User Experience Questionnaire: Confidence intervals per scale.

Scale	Mean	Comparisson to benchmark	Interpretation
Attractiveness	2.16	**Excellent**	In the range of the 10% best results
Perspicuity	2.57	**Excellent**	In the range of the 10% best results
Efficiency	2.16	**Excellent**	In the range of the 10% best results
Dependability	1.84	**Excellent**	In the range of the 10% best results
Stimulation	1.94	**Excellent**	In the range of the 10% best results
Novelty	1.63	**Excellent**	In the range of the 10% best results

Figure C.9: User Experience Questionnaire Benchmark.

D

Final User Tests: Usability Test Guide

Usability Test Guide

NOVA LINCS, Departamento de Informática, Faculdade de Ciências e Tecnologias, Universidade NOVA de Lisboa, FCT

Brief description

This document aims to assist both the participant and the researchers in conducting an in-person workshop following the presentation of a new system focused on object detection and tracking functionalities. Before proceeding to the final questionnaire, complete the tasks described below with the help of a researcher if needed.

The object detection and tracking features, alongside video annotation, aim to provide innovative ways to highlight and analyze video content. Even though this system could be used in many different areas, this work focuses on racket sports as a use case. The developed features identify objects of interest in videos (e.g., player, ball, net, racket) and track the players over time. These mechanisms are used to generate additional features such as player highlighting, individual player heatmaps, and player trajectories throughout the video.

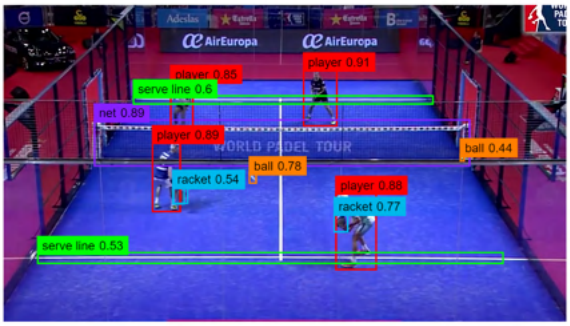
Please read each of the tasks described in the next section attentively while respecting the order assigned to each of them. All feedback provided either during the workshop or later is encouraged and appreciated by the development team, as new ideas and improvements may directly result from your participation.

Tasks

1. Explore the system UI and comment on it.
2. Open the available videos tab
   i. Identify CRUD operations.
   ii. Identify selection and rename operations.
3. Select one of the available videos.
   i. Watch a few seconds of the padel video.

4. Select the item detection option and comment on it.

   i. Turn the labels on and off.



5. Select the player highlight option and comment on it.

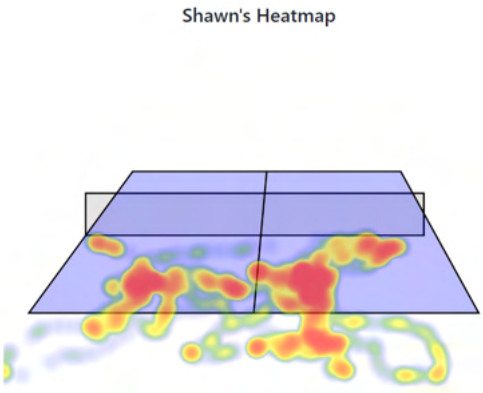
   i. Turn the labels on and off.

   ii. Change the highlighted player.

   iii. Change the name of one player.

6. Select the heatmap functionality and comment on it.

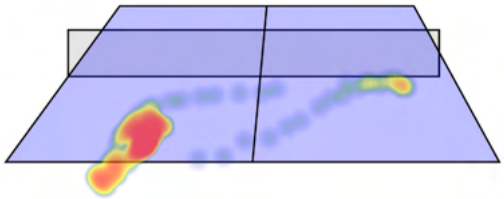
   i. Select different players to see their heatmaps.



7. Select the live heatmap functionality and comment on it.

   i. Select different players to see their heatmaps.

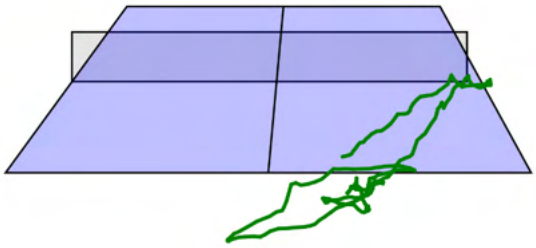
Shawn's Heatmap



8. Select the player trajectory functionality and comment on it.

   i. Select different players to see their trajectories.

Michael's Trajectory



E

Final User Tests: Consent Form for User Test

Information and Consent Form for User Test

Theme: Enhancing Racket Sports Video Analysis Through Object Detection and Tracking

Researchers: Prof. Nuno Correia and Tomás Martins

As a student for the Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, I am currently working on my master's thesis in Computer Science. Summarily, the main goal of this semi-structured interview is to present and discuss my system and its implications in the padel sport for practitioners, instructors, coaches, and analysts. In this interview, you will be asked to participate in a discussion regarding your experience in this sport focused on this research topic.

This study will allow this project's researchers to extract valuable feedback from our existing functionalities and their usefulness, as well as identify additional features or improvements that could be implemented.

Your participation must be voluntary. Refusing to participate in these tasks will not cause you harm or jeopardize any benefits you may already have. The lead investigator might remove you from the study. In that case, you will not be penalized in any way as a direct consequence of doing so.

If you have any questions regarding this workshop, please reach out to any of the following contacts:

Professor: Nuno Correia

Institution: Departamento de Informática, Faculdade Ciências e Tecnologia, UNL

Email: nmc@fct.unl.pt

Student: Tomás Martins

Institution: Faculdade Ciências e Tecnologia, UNL

Email: tmc.martins@campus.fct.unl.pt

I've read this document completely. Therefore, I fully understand the nature of this study, and I agree to be a participant. The lead researcher and respective associates have my permission to use the results of the mentioned experiments for academic use, such as in oral class presentations or others, thereby contributing to the scientific community as long as my identity remains anonymized.

I allow the recording of my voice and image to authorized researchers only.

_____ _____

PARTICIPANT'S SIGNATURE DATE (DD/MM/YY)