An Integrated Players' Collision Prediction System in Football Using Deep Learning Algorithms

Students Names:
Reem S. Almualem
Warood K. Alzayer
Ghala M. Alkhaldi
Fida M. Alelou
May M. Alotaibi
Supervisors:
Dr. Huda D. Althumali
Dr. Asma A. Alshammari

A project submitted in fulfilment of the requirements for the degree of Bachelor of Science

in the

Computer Science Department
College of Science and Humanities in Jubail

June 9, 2024

Abstract

Football is widely recognized as the most popular sport globally, with an estimated 265 million registered players and fan participation within billions. Among major sporting events, the Football World Cup stands out as one of the largest and most popular. However, like any other sport, football exposes players to certain injury risks, which tend to escalate with increased participation. In fact, compared to many other sports, football carries a higher incidence of injuries, with player collisions ranking among the most common causes. To mitigate these risks and enhance game strategy, players detection, collision prediction, and the integration of advanced data extraction techniques is essential. Recently, Artificial Intelligence (AI) techniques have been employed in football analysis due to a huge increase in data collection by professional teams, increased computational power, and advancements in Machine Learning (ML) and Computer Vision (CV). This project introduces an effective and efficient system for predicting player collisions in football. Initially, a comparison was conducted between different versions of YOLO (You Only Look Once). YOLOv8, a state-of-the-art real-time object detection framework and a core Deep Learning (DL) task, that was developed on the Roboflow platform. The results demonstrated that YOLOv8 outperformed the other versions, leading to its selection for the player detection model. Following the player detection process, an integrated algorithm for collision prediction was integrated into the model to complete the system's aim. Various top-view videos of football matches from different sources were used as the dataset. The proposed model was implemented using Python programming language, yielding promising results with an 88% accuracy rate. Additionally, post-prediction, a data file was extracted that provided insights into the players' occurrences and patterns, aiding game-play optimization strategies. Applying this system is expected to positively impact the financial aspects of sport organizations and the players themselves.

المستخلص

تُعرف كرة القدم على نطاق واسع بأنها الرياضة الأكثر شعبية على مستوى العالم، حيث يقدر عدد اللاعبين المسجلين فيها بنحو ٥٦٢ مليونًا، ويقدر عدد المشجعين المشاركين فيها بالمليارات. من بين الأحداث الرياضية الكبرى، تبرز بطولة كأس العالم لكرة القدم باعتبارها واحدة من أكبر الأحداث وأكثرها شعبية. ومع ذلك، مثل أي رياضة أخرى، تعرض كرة القدم اللاعبين لبعض مخاطر الإصابة، والتي تميل إلى التصاعد مع زيادة المشاركة. في الواقع، مقارنة بالعديد من الرياضات الأخرى، تحمل كرة القدم نسبة أعلى من الإصابات، حيث تعد اصطدامات اللاعبين من بين الأسباب الأكثر شيوعًا. للتخفيف من هذه المخاطر وتعزيز استراتيجية اللعبة، يعد نموذج الكشف عن اللاعبين والتنبؤ بالتصادم و تقنيات استخراج البيانات المتقدمة أمرًا ضروريًا. في الآونة الأخيرة، تم استخدام تقنيات الذكاء الاصطناعي في تحليل كرة القدم بسبب الزيادة الهائلة في جمع البيانات من قبل الفرق المحترفة، وزيادة القوة الحسابية، والتقدم في تعلم الآلة والرؤية الحاسوبية. يقدم هذا المشروع نظامًا فعالًا للتنبؤ باصطدامات اللاعبين في كرة القدم. يولو ٨، هو إطار عمل متطور للكشف عن الكائنات في الوقت الفعلي ويعد احد المهام الأساسية للتعلم العميق، تم تطويره على منصة الروبفلو. في البداية، تم إجراء مقارنة بين الإصدارات المختلفة من يولو. أظهرت النتائج أن يولو ٨ تفوق على الإصدارات الأخرى، مما أدى إلى اختياره كنموذج للكشف عن الاعبين. بعد عملية الكشف عن اللاعبين، تم دمج خوارزمية مخصصة للتنبؤ بالتصادم في النموذج لإكمال هدف النظام. تم استخدام العديد من مقاطع الفيديو ذات الرؤية العلوية لمباريات كرة القدم من مصادر مختلفة كمجموعة بيانات. تم تنفيذ النموذج المقترح باستخدام لغة البرمجة بايثون، مع حصوله على نتائج واعدة بمعدل دقة ٨٨ %. بالإضافة إلى ذلك، بعد التنبؤ، تم استخراج ملف بيانات يوفر نظرة ثاقبة لأحداث اللاعبين وأنماطهم، مما يساعد في تحسين استراتيجيات اللعب. علاوة على ذلك، من المتوقع أن يؤثر هذا النظام بشكل إيجابي على الجوانب المالية للمنظمات الرياضية واللاعبين أنفسهم.

Acknowledgements

First and foremost, we express our heartfelt gratitude to Allah SWT, the most gracious and the most merciful, for granting us the strength, guidance, and blessings throughout the journey of completing this project. We extend our deepest thanks to our families, especially our mothers, for their unwavering support, love, and encouragement during this endeavor. We are immensely grateful to our supervisors Dr. Huda Althumaly and Dr. Asma Alshammari for their invaluable guidance, expertise, and continuous support. Our sincere appreciation also goes to all members of the Computer Science Department at the College of Science and Humanities for their collaboration, insights, and contributions. With the blessings of Allah, the support of our families, the guidance of our supervisors, and the collaboration of the department and college, we have successfully created and completed this project that we are truly proud of.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ANNs	Artificial Neural Networks
AP	Average Precision
CIoU	Complete Intersection over Union
CNNs	Convolutional Neural Networks
COCO	Common Objects in Context
CSS	Cascading Style Sheets
CV	Computer Vision
C2f	Cross Stage Partial Networks
DL	Deep Learning
DFL	Distribution Focal Loss
ELAN	Efficient Layer Aggregation Network
FIFA	Fédération internationale de football association
GAN	Generative Adversarial Network
GELAN	Generalized Efficient Layer Aggregation Network
GPS	Global Positioning System
GPU	Graphics Processing Unit
HTML	Hypertext Markup Language
IAR	Image-Assisted Routing
IBM	International Business Machines
ICCV	Improved Color Coherence Vector
ID	Identity

IDF1	Identification F1
IDP	Identification Precision
IDR	Identification Recall
IoU	Intersection over Union
k-NN	KNearest Neighbor
KCF	Kernelized Correlation Filter
KF	Kalman Filtering
LSTM	Long-Short Term Memory
mAP	Mean Average Precision
MARS	Motion Analysis and Reidentification Set
ML	Machine Learning
MOT	Multiple Object Tracking
NBA	National Basketball Association
OpenCV	Open Source Computer Vision Library
PGI	Programmable Gradient Information
PIL	Python Imaging Library
ReLU	Rectified Linear Unit
RFID	Radio Frequency Identification
RNNs	Recurrent Neural Networks
SAFF	Saudi Arabian Football Federation
SORT	Simple Online and Real-Time Tracker
SOT	Single Object Tracking
SPA	Single Page Application
SSR	Server-Side Rendering
UI	User Interface
VR	Virtual Reality
YOLO	You Only Look Once

Chapter 1

Introduction

This chapter introduces the research topic and its significance in the field of sports. It then presents the problem statement, followed by the project's aim and objectives. The chapter also provides an overview of the project methodology and its importance in the field of sports. Moreover, the scope, target beneficiaries, and the contributions of the project are discussed. Lastly, the project organization is presented.

1.1 Introduction

Football is undoubtedly the world's most popular sport, with five billion football fans around the world [1]. The recent growth can be attributed to the increasing participation of females and the expansion of the game to countries with less historical involvement [2]. The Football World Cup is considered one of the most important and popular sporting events worldwide. People of all ages show immense love for football, supporting and cheering for their favorite local and international teams. In football, there are two teams consisting of 11 players each. They play for a total of 90 minutes, divided into two halves of 45 minutes, separated by a 15-minute break in between. Players in football can use any part of their body, except their arms and hands, to stop the opposing team from scoring goals. The goalkeeper is the only player allowed to touch the ball inside the penalty area.

However, like most sports, football poses a certain level of injury risk to its players. The rate of injuries typically appears to be greater in men when compared to women who are participating in the same kind of tournament

[3], [4]. In international men's football tournaments, injuries occur at a rate of 51 to 144 per 1000 match hours, which is roughly equivalent to 2 to 3 injuries per match [4], [5]. Players contacting each other account for the majority of soccer injuries, with 15% to 50% resulting from foul play [6]–[9]. Collisions between players have been a recurring issue in football, leading to player absences, increased medical expenses, and decreased team performance. It is essential to accurately identify players that have a higher risk of injury in order to effectively manage the risks in football. Additionally, it is important to conduct a thorough data extraction process on the players and the match to enhance the accuracy of identifying high-risk situations. This not only helps in managing risks but also contributes to the overall enhancement of the game.

To address these issues, several solutions have been introduced including the use of Frequency Identification (RFID) tracking to monitor player positions, Machine Learning (ML) algorithms to predict real-time collisions, and haptic feedback to alert players. Additionally, an ML-based approach was proposed to predict the severity of head collisions, helping to maintain player safety and prevent injuries [10], [11]. Incorporating advanced data extraction processes can further enhance player safety and performance. The Catapult One smart Global Positioning System (GPS) vests track metrics like speed, distance, and sprints, helping optimize player workload and reduce injury risks. International Business Machines (IBM)'s Power Index and Match Insights, utilize Artificial Intelligence (AI) to analyze players' performance and predict outcomes based on various factors. The National Basketball Association (NBA) Global Scout app, powered by AI, allows players to analyze their own videos to assess their different skills and identify improvement areas. Finally, the deep learning-based scoring system from the Gymnastics World Championships tracks athletes' movements and visualizes performance via 3D models in order to extract main scoring data [12].

The fast progress in AI has created new opportunities and benefits in various individual sports, such as analyzing games and developing strategies, predicting match outcomes, improving athletes' performance and health, scouting and recruiting players, preparing training and diet plans, enhancing sports journalism, selling tickets, maximizing broadcasting and streaming, improving fan engagement, optimizing ad opportunities, and even creating AI referees [13], [14]. This study aims to present an efficient and effective system to predict player-to-player collisions in football using You Only Look Once Version 8 (YOLOv8) framework for players detection and an integrated algorithm for collision prediction. Additionally, the system extracts data about the players and the match which provide insights into the players' occurrences and patterns, aiding game-play optimization strategies. The proposed system holds great potential for the development of intelligent player collision warning technology in football, ultimately benefiting the sport as a whole.

1.2 Problem Statement

In football, numerous factors impact players and the outcome of matches, from player movements and interactions to strategic game-play decisions. Collisions, which pose a high risk of injury, can lead to prolonged player absences, increased medical costs, and decreased team performance. Injuries ranging from minor discomfort to career-ending problems. These collisions disrupt the flow of the game, interfere with player performance, and negatively affect the performance of the entire team. Despite the advancements in technology, current approaches to manage players collisions are mostly very expensive [15].Additionally, there is a significant lack of adoption of data extraction technologies in football. Instead, many teams and coaches continue to rely on manual data collection and analysis methods, such as reviewing match footage by hand and making subjective assessments based on video observations [15]. This traditional approach is time-consuming, prone to human error, and often lacks the precision needed to effectively analyze player movements, predict collisions, and optimize strategies. The continued preference for these outdated methods underscores the urgent need for more sophisticated data extraction systems that can provide accurate insights to enhance player safety and performance. Therefore, it is essential to manage and analyze player collisions and match data to develop effective game-play strategies and evaluate team performance for achieving success.

1.3 Project Importance (Motivation)

The Saudi Arabian Football Federation (SAFF) has recently initiated a comprehensive endeavor aimed at securing the hosting rights for the Fédération Internationale de Football Association (FIFA) World Cup in the year 2034. In light of this significant undertaking, it would be prudent to develop a robust system capable of predicting player collisions and extracting data pertaining to the collisions. This system will be able to reduce injuries and enhance the players' safety. It will allow proactive measures to be taken and it will enable coaches and medical staff to accurately identify players who are most vulnerable to collisions and provide them with appropriate training and guidance. This can result in a reduction in the number and severity of player injuries during matches, enhancing long-term player health and well-being leading to reducing medical costs and sustaining players' availability. Moreover, this system can be used in analyzing the performance of the players. By utilizing the extracted data, coaches can analyze players' positioning and reaction times to improve game strategies. The data provided by this system can also offer valuable insights that influence the modification of game rules to make sports safer, as the rapid data extraction frees analysts from manual processing.

1.4 Project Aims and Objectives

In this study, the focus is on addressing the critical issue of player collisions in football. This study aims to develop a collision prediction system utilizing Deep Learning (DL) techniques. Specifically, the objectives of this study are:

Objective 1: To develop an object detection model that detects football players using DL algorithms.

Objective 2: To develop an integrated collision prediction system using the DL algorithm to effectively track and predict football players' collisions.

Objective 3: To build a web-based system capable of extracting data regarding players' collisions from football matches videos.

1.5 Project Methodology

The development of this project relies on the following steps:

- **Literature Review**
  The related research works are reviewed and classified into two main categories: Players Tracking and Detection, and Players Collision Prediction.

- **Data Collection**
  The dataset will be collected from several top-view football match videos obtained from various sources.

- **Players Detection**
  -The players Detection models will be developed using DL algorithms such as YOLOv7, YOLOv8, and YOLOv9.
  -The models will be trained and implemented using the Roboflow and Google Colab platforms.

- **The Integrated Collision Prediction Algorithm**
  The integrated collision prediction algorithm will be developed by integrating it with the best DL algorithm using the Python programming language.

- **Web-based System**
  A web-based system will be built using Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS) to provide a user-friendly interface for the application.

1.6 Project Scope

The scope of this research is the prediction and analysis of players' collisions in football on a global scale. The system will be able to process top-view football match videos, accurately detect players, predict a potential collision, extract post-match data, as well as ensure a smooth and intuitive experience for the end-users.

1.7 Project Target

Football collision prediction system leverage DL and data extraction to enhance the overall football experience. This system assists several target beneficiaries, who can benefit from them differently. These beneficiaries can be:

- **Athletes**: Players themselves can benefit from the collision prediction system by gaining insights into their positioning and movement patterns. They can adjust their game-play, anticipate collisions in advance, and take preventive measures to avoid injuries.

- **Coaches**: Utilize the collision prediction system to develop strategies to minimize collisions, and analyzing collision data and identifying patterns.

- **Sports Officials and Referees**: Utilize the collision prediction system to review incidents and make accurate, timely calls. The system can help identify borderline or controversial plays, providing objective data to support decision-making.

- **Medical Staff and Team Physicians**: Leverage collision data to understand the forces and impacts experienced by players, creating protocols and rehabilitation programs based on data. Early collision alerts can facilitate faster emergency response and medical attention.

- **Sports Scientists and Researchers**: Use the collision data to study the impact of collisions on player performance, injury rates, and long-term health. This data can also aid in the development of more efficient protective equipment and safety protocols.

- **Sports Broadcasters and Analysts**: Use collision prediction data to provide more informative and engaging coverage of games, enhancing the viewing experience for fans. Data can generate graphics, replays, and analyses that highlight key moments and strategic elements of the sport.

- **Game Developers and Virtual Reality (VR) Companies**: Leverage collision data can be valuable for game developers and VR companies involved in creating sports simulations. It can improve the realism and accuracy of virtual sports experiences, making them more immersive and safer.

1.8 Project Contributions

This project presents several important advancements in the field of sports analytics and players' technologies. The key contributions of this project are:

- Undertake a theoretical analysis of current research pertaining to the detection and tracking of players, as well as players' collision prediction, within the context of sports.
- Aggregate football match top-view videos from diverse sources to construct a comprehensive dataset, partition the video recordings into segments, and meticulously annotate individual frames.
- Develop a collision prediction system for players utilizing a DL algorithm (YOLOv8) in conjunction with an integrated algorithm.
- Develop a web-based system that extracts data regarding players' collisions.

1.9 Project Organization

This project is organized as follows:

- **Chapter 1** provides an introduction to the project topic, where the problem statement is discussed with the project importance, as well as the objectives of the project are clarified with a brief discussion of the methodology, and the project scope, project target, and project contributions are discussed.
- **Chapter 2** reviews the research works related to this study; it contains an analysis of previous research works related to the research area. Also, the gap of knowledge is identified in this chapter.

- **Chapter 3** describes the methodology of the study, where the software platforms used in this project and the programming language are provided. Furthermore, the YOLO algorithm and its various versions were discussed. Moreover, the performance metrics are presented.
- **Chapter 4** illustrates the implementation of the collision prediction system. It starts with the dataset construction and configurations, then the detection models implementation and their results, followed by the integrated prediction algorithm implementation and the system's results.
- **Chapter 5** describes the development of the web-based collision prediction system; it clarifies the functionalities, the architecture, the front and back ends, the design, and the interfaces of the system.
- **Chapter 6** concludes the whole project and identifies potential areas for future work.

Chapter 2

Background and Literature Review

This chapter discusses the background of the football sport, as well as an overview of AI and its uses in football. Moreover, this chapter reviews the research related to players detection and tracking, and players' collision prediction using AI algorithms and other technologies. The gap of knowledge is presented at the end of this chapter.

2.1 Background

Millions of fans worldwide are united by the dynamic sport of football, which demands talent, passion, skill, and strategic prowess. Different AI techniques can be leveraged in this sport that help enhance it. In this section, the background of football sport, an overview of AI, as well as AI in football will be discussed.

2.1.1 An Overview of The Football Game

A rectangular field with goals at each end is the field where football is played. It is also known as association football or soccer. In this game, the objective is to score more goals than the other team by using any part of the body except the hands and arms to move the ball into the opponents' goal [16]. The game is divided into three phases: attacking, defending, and transitional. Teams can use various tactical and strategic approaches to gain an advantage, such as pressing, counter-attacking, and possession-based football [16]. It is characterized by a number of interactions, principles, and rules between players. In addition, football is a demanding sport that requires players to commit to regular training, fitness, and tactical awareness. Over the years, football has attracted the attention of the entire world, regardless of age. Its appeal has been growing since the 2022 FIFA World Cup in Qatar.

2.1.2 An Overview of AI

One of the computer science fields is AI [17], it is the development of intelligent machines that are capable of carrying out activities that have traditionally needed human intellect [18]. AI has several subfields, one of them is ML which includes DL as a subset [19]. Also, Computer Vision (CV) is one of the AI subfields that uses ML algorithms [20]. These subfields of AI have seen tremendous growth and attention in recent years. ML models are computer algorithms that utilize data to make predictions or decisions [21]. Popular ML algorithms include decision trees and random forests.

2.1.2.1 Deep Learning (DL)

DL is a subfield of ML that utilizes algorithms called Artificial Neural Networks (ANNs) which have self-learning capabilities, and their structure and function are inspired by the human brain [22]. DL algorithms such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have the capacity to process large amounts of data, derive intricate patterns, and produce extremely precise predictions or classifications [19].

2.1.2.2 YOLO (You Only Look Once)

A popular DL-based one-stage object detection model is YOLO which is a real-time object detection algorithm that locates and classifies objects in images.[23], [24]. It was introduced in 2015 by Joseph Redmon and colleagues [25]. It uses CNN to predict bounding boxes in images and it is known for its rapidity in detecting objects in images [24]. The YOLO framework has undergone several modifications, with each version building on the preceding one to overcome constraints and improve performance [26]. YOLO models have been widely employed in varied industries, such as self-driving vehicles, traffic applications, and robotics [26].

2.1.2.3 Computer Vision (CV)

Computer vision is the field of study that focuses on enabling computers to interpret and understand visual information from the world, similar to how humans do. It involves developing mathematical techniques to reconstruct the 3D shapes and appearances of objects from images. Despite significant advances, computers still struggle with tasks that are easy for humans, such as identifying all animals in a picture.

Key applications of computer vision include:

- Optical Character Recognition (OCR) for reading handwritten and printed text.
- Machine inspection for quality assurance in manufacturing.
- Automated object recognition in retail checkout systems.
- 3D model building from aerial photographs.
- Medical imaging for diagnostics and monitoring.
- Automotive safety systems for detecting obstacles.
- Motion capture for animation and special effects.
- Surveillance for security and traffic monitoring.
- Biometric authentication, such as fingerprint recognition.

In consumer-level applications, computer vision is used for photo stitching, exposure bracketing, image morphing, 3D modeling, video stabilization, and face detection [27].

2.1.3 AI in Football

No other sport can compare to the way football unites and enthralls the human family. Its remarkable ability to unite people and spread joy over cultural boundaries is evidenced by its global following of over 4 billion[1]. Football, the most popular game in the world, has enormous potential for utilizing technology to advance human progress. AI has a significant impact on the football game, revolutionizing several aspects of the sport. AI systems are capable of processing enormous amounts of data from many different sources, such as player statistics, past performance, in-game analysis in real-time, and post-game analysis. AI can produce insights and predictions to help coaches and players make educated decisions on the field by merging this data with ML algorithms. Some AI technologies, such as CV, DL, and ML, are utilized to improve video analysis through player movement tracking in real-time, pattern recognition, and scenario prediction [15].

2.2 Literature Review

Reviewing the research works related to this study plays a crucial role in understanding the current state of knowledge on this subject, identifying any shortcomings or limitations in past research, and proposing potential areas for further investigation. The reviewed literature has been mainly classified into two classes, as shown in Figure 2.1 players' detection and tracking techniques, and players' collision prediction techniques. The first class includes several aspects of players' detection and tracking using different techniques to achieve high accuracy in detecting and tracking players in different sports. The second class reviews literature related to collision prediction based on AI methods.

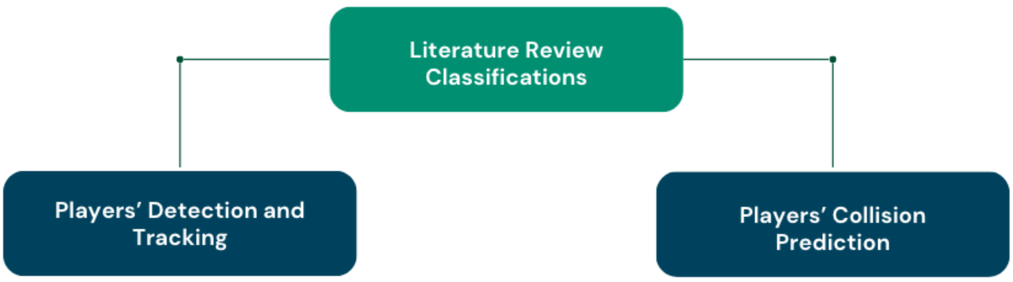


FIGURE 2.1: Literature Review Classification

2.2.1 Players Detection and Tracking Techniques

Player detection can be defined as the identification of the player in a certain frame or image and providing a bounding box around them [28], while player tracking can be defined as following the players' movements where the identification of each player is maintained across different frames [29]. The players detection and tracking are crucial as they are the foundation of the advanced analysis of the game and the players' performance. Coaches can create strategies and boost team performance by having a solid understanding of game dynamics, which depends on accurate player detection and tracking [30]. Furthermore, player detection and tracking offer comprehensive data that enhances sports analysis and facilitates more thoughtful and strategic decision-making [31]. In this class, 22 studies were analyzed and reviewed. Most of the studies utilized the YOLO algorithm in its different versions along with other tracking algorithms in order to align with the studies' objectives. As well as other AI techniques.

Figure 2.2 illustrates the statistics of the different detection techniques that were studied in the literature. As shown in the chart most studies utilized YOLOv3 in detecting the players.

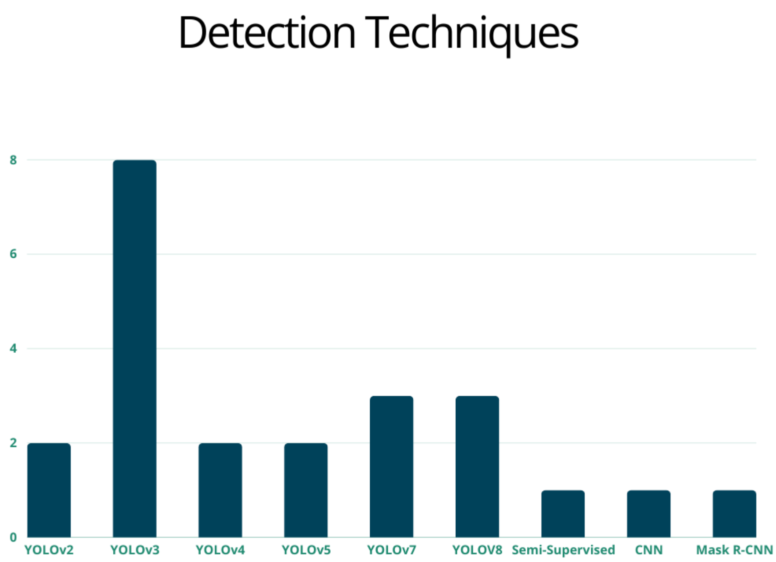


FIGURE 2.2: Detection Techniques Chart

Figure 2.3 illustrates the statistics of the different tracking techniques that were studied in the literature. As shown in the chart, most of the studies utilized Simple Online and Real-Time Tracker (SORT) and its extended version Deep Simple Online and Real-Time Tracker (DeepSORT) in tracking the players. Note that some studies did not mention the name of the tracking method explicitly.

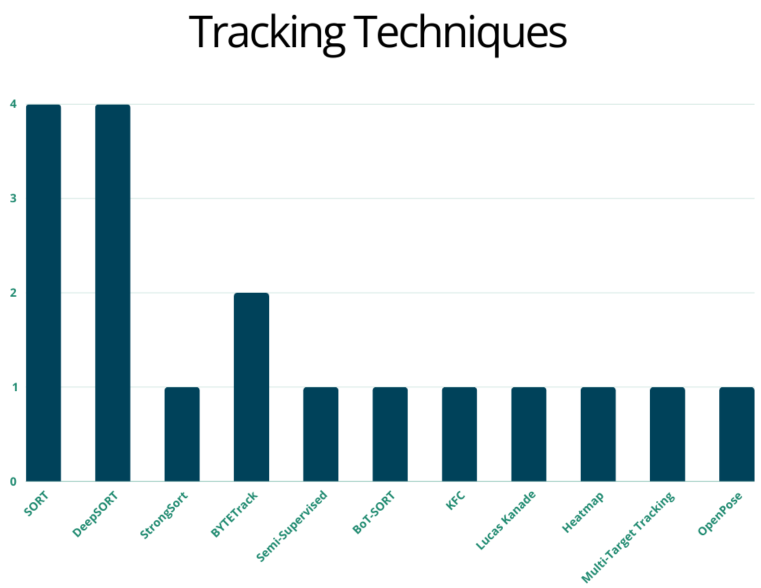


FIGURE 2.3: Tracking Techniques Chart

Studies such as [32] and [33] utilized the YOLOv2 along with tracking algorithms such as SORT and DeepSORT.

The study [32] focused on the challenge of real-time detection and tracking of basketball players in broadcast videos. This research aims to develop a framework that can precisely identify and monitor numerous players in a dynamic, occlusion-prone sporting environment. To achieve this, two algorithms were used in this study. The SORT algorithm is used for tracking, while the YOLOv2 detector is used for player detection. The results demonstrate that the framework performs well in player detection and tracking tasks, achieving a high Average Precision (AP) of 89% for standard criteria and 63% for stricter criteria.

The authors in [33] the authors studied the problem of tracking the players in handball videos using a single video source. The YOLOv2 object detection method is utilized in this study which predicts bounding boxes and class probabilities in a single pass to provide real-time detection capabilities as well as other algorithms such as the traditional Hungarian assignment algorithm, the SORT algorithm with a motion model, and its extension DeepSORT. The results reveal that YOLOv2 has the potential for player detection. The DeepSORT showed the highest accuracy in player tracking.

Other studies [34], [35], [36], [37], [38], [39], [40], and [41] utilized tracking methods such as SORT, DeepSORT, Mask R-CNN, and other methods along with YOLOv3.

Both [34] and [35] focused on detecting and tracking in dynamic sports environments such as handball sport. In [34] the authors considered some factors such as rapid motions, frequent alterations in the path, obstructions, and players entering and exiting the camera's field of view. High-quality video recordings were collected to create the dataset, these videos were analyzed using YOLOv3 and DeepSORT to generate initial annotations. After that, the DeepSORT algorithm was implemented to track the players. The tracking results were evaluated using Identity switches (ID), Identification Precision (IDP), Identification Recall (IDR), and Identification F1 (IDF1). According to these measures, 24.7% of the time, players were accurately recognized.

Where in [35] the authors aim to employ DL techniques to detect and track players, as well as to assess player motions in difficult circumstances. Players and balls were detected using YOLOv3 and Mask R-CNN, players were tracked using DeepSORT, and actions were recognized using Long-Short Term Memory (LSTM). The Hungarian algorithm was used in player tracking tests with the PT-Handball dataset. The YOLOv3 algorithm and Mask R-CNN algorithm were evaluated using different performance metrics such as recall, precision, and F1. Both algorithms got the highest percentage in precision, where the YOLOv3 got 95% and the Mask R-CNN got 98%.

The studies in [36] and [37] focused on detecting players and balls in football videos using YOLOv3. The purpose of [36] is to enhance sports performance evaluation by applying DL algorithms for accurate tracking. Employing the SORT algorithm, videos are processed to identify relevant frames, while Kalman Filtering (KF) and bounding box overlap are used for tracking. The results show a high percentage of detecting the players with a precision of 97%.

In [37] the authors aimed to improve the detection and tracking of balls and players in soccer videos, aiming to aid coaches in team evaluation, referees in decision-making, and identify talented players. The YOLOv3 architecture was used for ball and player detection, with a pre-trained version for player detection and a model trained on self-annotated frames for ball detection. The SORT algorithm was used for tracking, a fusion of the KF and bounding box overlap. The study found that the YOLOv3 model and SORT algorithm effectively detected and tracked both balls and players in soccer videos. However, the dataset used was limited, containing only 50 images from mobile camera captures and various sources.

According to [38] the authors tackle the challenge of detecting football players accurately from various viewpoints using aerial images. They propose the Image-Assisted Routing (IAR) scheme which estimates the location of sensor nodes using images captured from uncrewed aerial vehicles. The primary method employed involves utilizing informed filters to detect players and harness statistical shape information. The study explores the accuracy of the detection from different viewpoints and then compares the informed filters with YOLOv3, showing superior accuracy. The main findings include the effectiveness of IAR and informed filters in dynamic sporting environments and using optimal camera locations for player detection. The level of detecting accuracy was excellent. 0.005524 miss rate with a false positive (FP) of 0.01 per image.

While the work in [39] tackles the challenge of developing an application using CV techniques for amateur football analytics to provide inexpensive solutions accessible to football clubs that are on a low budget. The primary applied technique involves combining existing research in football, DL, and CV to develop a TV broadcast system, that can analyze football games from a single point of view. The system contains modules for court detection, object detection, tracking, and team classification using techniques like Conditional Generative Adversarial Network (GAN) Pix2Pix, Lucas-Kanade method, and YOLOv3 CNN. The main findings of this work demonstrate the feasibility of extracting main conclusions from football games, such as ball and player detection and their location on the court.

In the study of [40] the authors considered the problem of detecting players in continuous long-shot broadcast videos, especially football matches. They presented a novel transductive approach to player detection using YOLOv3 and other algorithms, leveraging a limited number of domain-labeled samples to achieve substantial performance enhancements. The main method employed involves introducing a technique to collectively assign domain labels at the instance level, which is essential for handling target data such as broadcast videos that contain considerable domain noise. Additionally, they created and made available a comprehensively annotated dataset consisting of soccer broadcast videos to assess player detection tasks. The main findings highlight significant improvements in player detection accuracy achieved through the proposed transductive approach compared to current state-of-the-art methods. By annotating domain labels for approximately 100 samples per video, the authors showcased a recall of 85% for football broadcast videos.

Where the study [41] focused on the domain of DL-based multi-target trajectory tracking in multi-frame basketball game video images improving trajectory tracking accuracy and resolving the issue of occlusion interference in the tracking process. The study's suggested methodology is divided into tracking and detection. Target detection is accomplished by the YOLOv3 algorithm, with improvements made to the backbone network to enhance feature extraction. Target location prediction is achieved by using the KF, and multi-target tracking based on appearance feature similarity is accomplished by utilizing a hierarchical data association technique. Video pictures are the sort of data used in this investigation. The proposed method got the highest convergence comparing the other algorithms with a percentage of 84.4% under the non-occluded conditions.

The studies in [42] and [43] utilized YOLOv4 in detecting the players in different sports with different tracking methods.

The work in [42] addressed the necessity for efficient and reliable athlete tracking in sporting competitions. The objective of this research is to develop a tracking system that utilizes DeepSORT and YOLOv4 to track each athlete in real-time while also delivering important track data for analysis and strategy formulation. The DeepSORT algorithm was utilized for tracking, while YOLOv4 was utilized for detection. YOLOv4 was successfully trained on the INRIA person dataset, yielding a maximum Intersection over Union (IoU) of 0.97673. Furthermore, the Market1501 and Motion Analysis and Reidentification Set (MARS) datasets were used to train DeepSORT, which resulted in a mean Average Precision (mAP) of 68.2%.

Moreover, the work in [43] addressed the challenge of biased and subjective video analysis in tackle-collision sports by proposing an automated system for evaluating the tackle risk in rugby union in-game matches. The primary technique applied involves utilizing the YOLOv4 technique for player detection and ball detection within tackle segments, followed by using the KF for tracking and using OpenPose for pose estimation. The main findings of this paper show an accuracy of 62.50% on a different collection of tackles in rugby, which aims to help referees make more decisions regarding foul play.

YOLOv7 was utilized in the following studies in addition to other tracking techniques such as StrongSORT.

The study [44] aims to automate object detection and classification in soccer fields, including players and the ball, to enhance game analysis, player performance evaluation, and fan engagement. The goal is to develop a system that accurately detects and tracks the ball, and classifies players, and referees, thereby improving object detection and classification accuracy. To identify the ball and players on the field, the authors utilized the YOLOv7 real-time object detection model. In addition, they utilized a K-Nearest Neighbor (k-NN) classifier and the Improved Color Coherence Vector (ICCV) features to categorize people into five categories. High accuracy was shown by the study in both the detection and classification models. The classification model was able to achieve 96% accuracy in the last match, whereas the detection model could achieve 91% precision for both players and the ball.

The work in [45] addressed the problem of a lack of available and relevant data for data analysis in the field of eSports. The objective of the study was to develop a method for efficiently detecting and tracking players in eSports videos, generating a dataset from the minimap. The objective was achieved by using a GAN to artificially generate a dataset, training an object detection model YOLOv7, and refining it with a multi-object tracking model StrongSORT. The results showed 98% accuracy in inference on real eSports videos. The study also discussed the limitations, such as the reliance on minimap information and the need for further evaluation of different game scenarios. The performance metrics used to evaluate the algorithm were not explicitly mentioned in the provided document.

The study [46] utilized YOLOv7 and YOLOv5 to compare the performance of the two algorithms. The purpose of the study is to automate the process of producing sports recordings for indoor sports like futsal and handball. The main goal is to use AI technology to reduce the costs related to recording these games. For object detection, the system utilizes the YOLOv5, YOLOv7. These algorithms identify each frame of the recording's relevant objects, like players or the ball. After that, an expert system analyzes this data to find important game occurrences. The system processes video data, particularly frames from sports records. The results show that YOLOv7 performed better in detecting the objects with the highest mAP of 96% in detecting the handball players.

The studies in [47], [48], and [49] utilized YOLOv8 algorithms along with various tracking methods such as BoT-SORT and BYTETrack.

In [47] the authors used the TeamTrack dataset which is a pioneering benchmark dataset designed for Multi-Object Tracking (MOT) in sports. Its goal is to address the challenges in tracking objects that have similar appearances but move differently, which frequently occur in team sports. The collection offers extensive and varied full-pitch videos from a variety of sports, such as handball, basketball, and football. The technique steps involve object detection Using YOLOv8, reidentification, and temporal association using appearance-based features. The dataset contains metrics that can be utilized for evaluating MOT algorithms which are BoT-SORT and BYTETrack. The detection algorithm was tested in different sports, and it got the highest mAP of 71% in handball sport.

The study in [48] the study presented a new dataset and compared the results of the YOLO8n and FootAndBall algorithms to build a baseline for real-time detection in an effort to address the problem of inadequate datasets for player and ball detection in soccer movies. For training and evaluation, the authors used the SoccerNetv3 and SoccerNetv3H250 datasets. They used several input resolutions to train the FootAndBall model and fine-tune the YOLO8n model. Using the SoccerNetv3H250 dataset, five models—both the original and trained versions—were evaluated. Outperforming the FootAndBall model, the YOLO8n model showed better real-time detection performance with the highest AP of 90.58% of player detection on the football field.

Furthermore, other more studies utilized YOLOv8 to compare its performance with YOLOv5, such as in [49] the study focuses on the task of determining ball possession using object detection and tracking techniques in football sports data analytics. Using methods for CV, the primary aim is to enhance sports analytics. The research compares YOLOv5 and YOLOv8. YOLOv8 was selected due to its better performance. To detect and track objects, the method combines BYTETrack and YOLOv8. Videos from tactical cameras recording football games provide input, and identified items (ball, players, and referees) are tracked individually.

Player tracking and detection can be applied using different AI techniques such as in [50], [51], and [52] studies.

In [50], the authors proposed DL-based multicamera multitarget tracking systems for football player tracking. The aim is to accurately track football's and various players' trajectories in football game videos. The technique uses DL algorithms such as the Kernelized Correlation Filter (KCF) algorithm and an improved version of the KCF algorithm for data fusion, image processing, and camera calibration. Imagery data from multiple cameras is the type of data utilized. The two algorithms were evaluated twice, the first experiment was on single target tracking where the KFC algorithm showed higher accuracy than the improved KFC algorithm with 88%. The second experiment was on multitarget tracking where the improved KFC algorithm showed higher accuracy with 98%.

The work in [51] focused on football player detection, which focuses on football video analysis. Using a deep CNN, the main goal is to detect players in real-time. The five successive convolutional layers in the suggested algorithm use leaky Rectified Linear Unit (ReLU) activation functions and batches of normalization to process frames from football videos. Exact detection is improved by a residual connection surrounding the first three convolutional blocks. The model was evaluated using AP on two datasets ISSIA-CNR and soccer player detection, the results show that the AP of the ISSIA-CNR dataset is 91.5% and the soccer player detection is 93.2%.

Moreover, in [52], the authors tackle the challenge of observing the occupancy of football fields and public sports facilities. The main technique applied involves developing a system that detects players using a single narrow-angle thermal camera and a cheap fisheye camera. They employ a knowledge distillation approach, training a network with different views and modalities of the same scene and custom motion detection and data augmentation algorithms. The main findings demonstrate the effectiveness of the proposed solution in detecting players around the entire field, with changing video conditions in real-time adaptation through online distillation. The system was evaluated using AP and showed a satisfying performance with AP >70% for threshold Intersection over Union (t_IoU) = 0.25.

The study [53] addresses the problem of detecting and tracking players in videos captured by multiple cameras by developing a semi-supervised system that can detect and track players in football and basketball and provides statistics based on analysis of multi-camera sports videos. The system is built around a monocular camera object detection and tracking system that was modified to function in the sports industry after being created for video surveillance applications. The system was evaluated using precision and recall and they revealed a percentage of 89.6% with a threshold value of 2.8 for region fusion.

2.2.2 Players' Collision Prediction Techniques

Collisions have significant effects on players' performance and safety in various sports. Accurate prediction and avoidance of collisions in various situations is critical to protecting the safety of the players and increasing the players' performance, which leads to more achievements on the sporting level. The following studies illustrate the techniques used in predicting the collision between players in different sports.

The authors in [10] argued that the high rate of injuries in contact sports is a major concern for athletes, coaches, parents, and fans. They noted that injuries can have a significant impact on athletes' physical and mental health, as well as on their careers. The main techniques applied were RFID to track the positions of players in the field. An ML algorithm was used to predict collisions in real-time, and haptic feedback was used to alert players of potential collisions. In general, this study found that the system could predict collisions with less than 14% of false alarms. However, the system has only been evaluated in a small pilot study, and it does not consider the high speeds of players.

In [11], the authors addressed a common problem in soccer: injuries, specifically head injuries caused by collisions between players. This problem is one of the most important problems that can lead to concussions and other serious injuries with severe consequences for players' health. The authors presented an ML-based approach to predict the severity of head collisions before they occur. This approach contributes to maintaining player safety, developing strategies to prevent player injuries, and identifying players at increased risk of injury. The model achieved an average accuracy of 73% for a dataset of male players and 70% for a dataset of male and female soccer players. Despite the presentation of a promising model to achieve the safety of soccer players, the study was based on a relatively small dataset, which limits the reliability of the study on an independent and large dataset.

A succinct summary of several studies on collision prediction, data tracking, and collection is provided in Table 2.1 This table provides an overview of the techniques employed, the objectives, and the results of each study. This table is a valuable asset for comprehending the developments in this area and pinpointing possible directions for further investigation.

Ref	**Technique Name**	**Objective**	**Results**
		Players Tracking and Detection	
[32]	SORT and YOLOv2	To develop a framework that can identify and monitor numerous players in a dynamic, occlusion-prone sporting environment	AP: 63% - 89%
[33]	YOLOv2, the traditional Hungarian assignment algorithm, and SORT	To track the players in handball videos using a single video source	N/A

[34]	YOLOv3 and DeepSORT	To detect and track players in dynamic sports environments	Accuracy: 24.7%
[35]	YOLOv3, Mask R-CNN, DeepSORT, and LSTM	To employ DL techniques to detect and track players, as well as to assess player motions in difficult circumstances	Precision: 95% - 98%.
[36]	YOLOv3, SORT, KF, and bounding box overlap	To enhance sports performance evaluation by applying DL algorithms for accurate tracking	Precision: 97%
[37],	YOLOv3, SORT, fusion of KF and bounding box overlap	To improve the detection and tracking of balls and players in soccer videos	N/A
[38]	Informed filters to detect players and harness statistical shape information	To detect football players accurately from various viewpoints using aerial images	• Accuracy: 0.005524 miss rate • FP: 0.01 per image

[39]	Conditional GAN Pix2Pix, Lucas-Kanade method, and YOLOv3 CNN	To develop an application using computer vision techniques for amateur football analytics to provide inexpensive solutions	N/A
[40]	YOLOv3	To detect players in continuous long-shot broadcast videos, especially football matches	Recall: 85%
[41]	YOLOv3, KF, and multi-target tracking	To improve trajectory tracking accuracy and resolving the issue of occlusion interference in the tracking process in multi-frame basketball game video	Convergence: 84.4%
[42]	DeepSORT and YOLOv.	To develop a tracking system for each athlete in real-time while also delivering important track data for analysis and strategy formulation.	mAP: 68.2%.
[43]	YOLOv4, KF, and OpenPose	To propose an automated system for evaluating the tackle risk in rugby union in-game matches.	Accuracy: 62.50%

[44]	YOLOv7, k-NN, and ICCV	To automate object detection and classification in soccer fields, including players and the ball	Accuracy: 91% - 96%
[45]	GAN, YOLOv7, and Strong- SORT	To develop a method for efficiently detecting and tracking players in eSports videos, generating a dataset from the minimap.	Accuracy: 98%
[46]	YOLOv5 and YOLOv7	To automate the process of producing sports recordings for indoor sports	mAP: 96% for YOLOv7
[47]	YOLOv8, re-identification, and temporal association using appearance-based feature	To address the challenges in tracking objects that have similar appearances but move differently	mAP: 71%
[48]	YOLO8n	To address the problem of inadequate datasets for player and ball detection in soccer movies	AP: 90.58%
[49]	YOLOv5 and YOLOv8	To detect and track ball possession techniques in football sports data analytics	N/A

[50]	KCF	To address the problem of DL-based multicamera multitarget tracking systems for football player tracking	Accuracy: • 88% for KFC algorithm on single target tracking • 98% for improved KFC algorithm on multitarget tracking
[51]	CNN	To track and detect football players and focus on football video analysis	AP: • 91.5% for the ISSIA-CNR dataset • 93.2% for the soccer player detection
[52]	Single narrow-angle thermal camera and a cheap fisheye camera	To observe the occupancy of football fields and public sports facilities	AP > 70%
[53]	Monocular camera	To detect and track players in videos captured by multiple cameras	Precision and Recall: 89.6%
Players Collision Prediction			

[10]	RFID and Haptic Feedback	To develop a system that can accurately predict collisions along with haptic feedback that can effectively warn athletes reducing the risk of injuries	False-alarm rate: less than 14% false alarms
[11]	Random Forest	To investigate whether prein-jury variables can be used to predict the severity of head collision events	Accuracy: 70% - 73%

TABLE 2.1: Literature Review Summary

2.3 Gap of Knowledge

In the field of sports science, it is crucial to investigate and analyze potential risks and collisions that athletes may face during game-play. Additionally, it is vital to extract and gather detailed information about the players and the game itself in order to comprehensively understand the dynamics at play.

In this project, 24 research articles have been reviewed and summarized in Table 2.1. As shown in Figure 2.4, previous studies had different objectives; 91.7% of the studies focused on player tracking and detection, while the remaining 8.3% focused on players collision prediction.

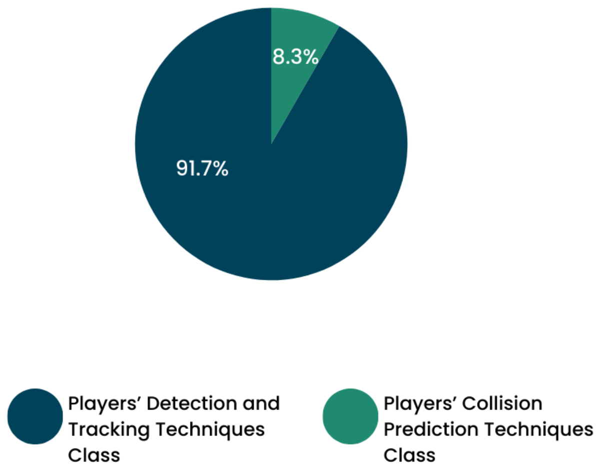


FIGURE 2.4: Percentages of Literature Review Classification

While there is a few research on collision prediction in various sports, an interesting observation arises, there is a noticeable lack of studies focusing on collision prediction specifically within the context of football. Despite its massive popularity, there is a lack of comprehensive exploration of data extraction techniques specifically tailored to the collisions in football. It has revealed several important aspects. Firstly, as shown in Figure 2.4, there's been a significant increase in papers exploring player tracking and detection classification, while player prediction classification has received little attention, with only two papers in this area. Importantly, there's a lack of research using player tracking, detection and collision prediction in football to gather and extract detailed match and player information, highlighting a significant gap in the existing literature. Lastly, although only two papers have tackled player prediction, there is a desire to achieve better results than those studies.

2.4 Summary

In this chapter, a background on football and AI was discussed. Also, a literature review of 24 studies was classified into two main classes, players detection and tracking and players' collision prediction. All these studies were presented and analyzed, and a gap of knowledge was deduced and derived from these studies.

Chapter 3

Methodology

This chapter describes the methodology of the project, including the programming language, software used, the selected DL algorithms, and performance evaluation metrics.

3.1 Software Platforms

The following platforms will be utilized in developing the collision prediction system. Two different platforms will be utilized each with different services provided.

3.1.1 Roboflow

Roboflow is a platform for developing CV applications that offers resources for managing datasets, training models, and deploying applications. It makes it possible to fine-tune foundation models, supports a variety of annotation formats, and makes it easier to distribute models across many devices [54]. The utilization of the Roboflow platform facilitated the development of the players detection models in this project. It will be utilized in data configuration such as object annotation, data augmentation, and data preprocessing, more details will be mentioned in (Chapter 4).

3.1.2 Google Colab

Google Colab, which stands for "Collaboratory" is an online platform that allows developing and running Python code from a browser. It requires no setup and provides free Graphics Processing Unit (GPU) access [55]. In this project, Google Colab will be utilized in developing the collision prediction system particularly, in creating the players detection models, developing the integrated prediction system, and combining the YOLOv8 detection model with the integrated prediction system.

3.2 Programming Language

The cornerstone of modern ML development, Python, in its (version 3.11) iteration, serves as the primary programming language for crafting the models discussed in this project."Python is powerful and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open. [56]", it has been in development since 1990 when its creator, Guido van Rossum, began working on it [57]. There are some reasons of why people prefer Python rather than other languages. Python's vibrant community fosters collaboration and knowledge exchange, further enriching the DL landscape [56]. Moreover, renowned for its simplicity, readability, and extensive library support, Python empowers developers and researchers alike to delve into complex DL tasks with ease. This mature and stable language is highly appealing to developers due to its cross-platform nature, object-oriented, high-level, and dynamic [57]. Additionally, effective Python programming relies not only on the language but also on the standard Python library and numerous extension modules, which are equally important [57]. The following libraries will be used in the model:

- **OpenCV**: OpenCV (Open Source Computer Vision Library) is used for CV tasks such as reading, processing, and displaying images and videos.
- **Pandas**: pandas is a data manipulation and analysis library. It provides data structures and functions needed to work with structured data seamlessly.
- **Datetime**: This module supplies classes for manipulating dates and times. The datetime class is used to work with date and time objects.

- **Ultralytics**: This is a custom library for the YOLO object detection model. It is used in the code to load and use a pre-trained YOLO model for detecting objects in the video.
- **Openpyxl:** openpyxl is a library to read, write, and manipulate Excel files in Python. It supports Excel 2010 xlsx/xlsm/xltx/xltm files.
- **Openpyxl.styles**: This module in openpyxl is used for styling Excel cells, such as changing fonts, filling cell colors, and aligning text.
- **Openpyxl.drawing.image**: This module in openpyxl is used for embedding images into Excel files.
- **Math**: This module provides mathematical functions. It is used in the code for calculations such as computing the Euclidean distance.
- **PIL (Python Imaging Library), specifically its fork, Pillow**: PIL/Pillow is used for opening, manipulating, and saving many images in different formats such as PNG, JPG.
- **Matplotlib**: matplotlib is a plotting library. pyplot is a module in matplotlib used for plotting and visualizing data, such as generating the pie chart for some data in this project.

3.3 Deep Learning Algorithms

Delving into DL algorithms, this section focuses on different versions of YOLO, as it is considered the most mentioned DL algorithm as shown in the literature review in Chapter 2:

3.3.1 YOLO Architecture

YOLO is an open-source object detection algorithm that utilizes a CNN. It is well-known for its rapid and effective object detection in images. YOLO divides the image into grids of various sizes, such as 3x3, 5x5, and 19x19. Each grid is in charge of evaluating whether an item exists in its region and estimating its attributes, including midpoint, length, height, and class. For each grid, an estimation vector is formed, which comprises the confidence score, the x and y coordinates of the object's midpoint ($B_x and B_y$), the object's width (B_w), height (B_h), and the likelihood of the object belonging to a certain class. Figure 3.1 illustrates the architecture of the YOLO algorithm [24].

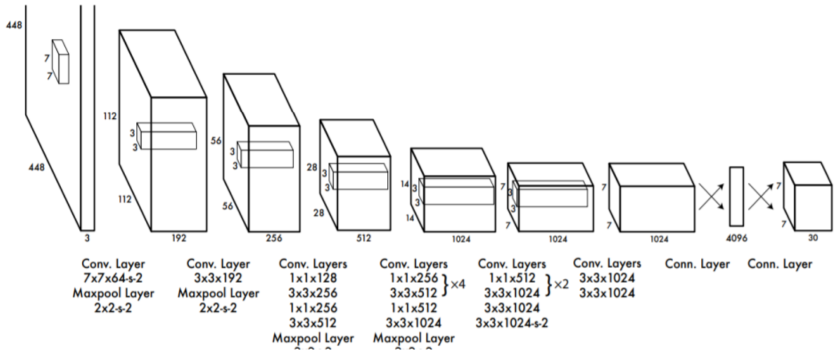


FIGURE 3.1: YOLO Algorithm Architecture [58]

Since its creation, the YOLO family has continually advanced, with each iteration building on the previous versions to enhance performance and overcome limitations. Figure 3.2 illustrates the timeline for the three versions of YOLO that will be utilized and compared in this project.

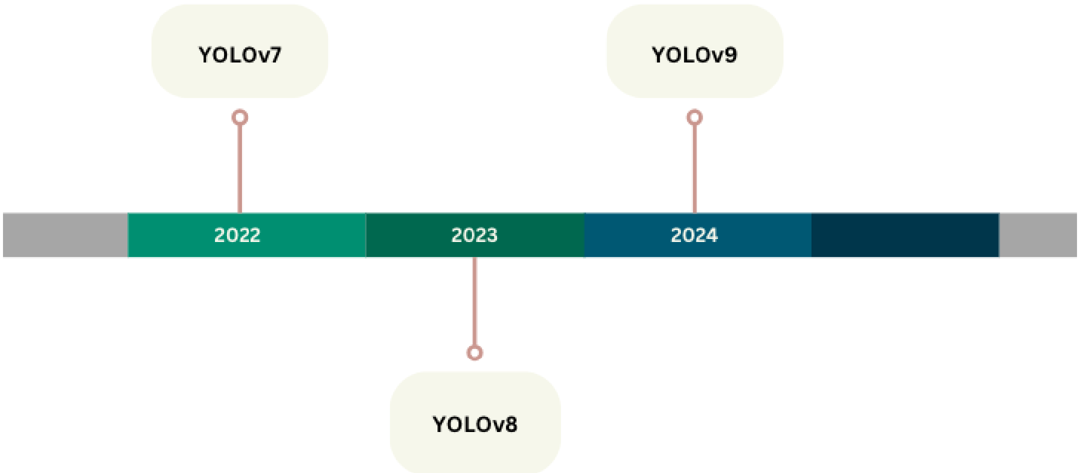


FIGURE 3.2: YOLO Versions Timeline

- **YOLOv7**
  YOLOv7 (2022), was a highly advanced object detection model that belongs to the YOLO family. It is known for its real-time performance and

accuracy in detecting objects within images or videos. YOLOv7 builds upon previous iterations of the YOLO model architecture, incorporating advancements in DL techniques to improve detection accuracy and efficiency [59].

- **YOLOv8**
  YOLOv8 (2023), an evolution of the YOLO series, represents a significant advancement in object detection technology. Building upon the strengths of its predecessors, YOLOv8 incorporates novel techniques to enhance both accuracy and efficiency in detecting objects within images or videos. YOLOv8 introduces refinements to the model architecture and training process, making it well-suited for various applications, including football player detection [60], [61].

- **YOLOv9**
  YOLOv9 (2024), represents the latest iteration of the YOLO series, incorporating advancements in DL techniques to achieve state-of-the-art performance in object detection tasks. Building upon its predecessors, YOLOv9 is designed to offer enhanced accuracy and efficiency. The introduction of PGI (Programmable Gradient Information) in YOLOv9 allows both deep models and lightweight models to achieve substantial improvements in accuracy [62].

Table 3.1 provides a detailed comparison of the different architectures of YOLOv7, YOLOv8, and YOLOv9, highlighting their unique features and improvements [59], [60], [61], [62], and [63].

Key Feature	YOLOv7	YOLOv8	YOLOv9
Architecture	E-ELAN (Extended Efficient Layer Aggregation) with enhanced CSP-Darknet53	Improved CSP-Darknet53	GELAN (Generalized Efficient Layer Aggregation Network) which is a combination between CSPNet and ELAN
Dataset	COCO (Common Objects in Context) Dataset	COCO Dataset	COCO Dataset
Mechanism	E-ELAN, Model Scaling	C2f module (Cross Stage Partial Networks), EfficientRep, CloU (Complete Intersection over Union), DFL (Distribution Focal Loss)	PGI
Additional Information	Similarity can be expected with scaled YOLOv4 because YOLOv7 is written by the same authors	Similarity can be expected with YOLOv5 because YOLOv8 is developed by the same company	YOLOv9 is a general and extended version based on YOLOv7

TABLE 3.1: YOLO Versions Comparison

3.4 An Integrated Prediction Algorithm

An integrated algorithm is a set of instructions that are developed specifically to address the needs and requirements of a particular project or application [64]. In this project, an integrated collision prediction algorithm was developed and then integrated with YOLOv8 to help accomplish the aim of this project and enable the completion of the journey of detecting players to predict collisions between them which is one of the most important stages for developing strategies and plans to prevent future injuries. This algorithm includes player tracking, players distances calculation, and some other relevant features.

3.5 Performance Evaluation Metrics

The effectiveness, accuracy, and quality of AI models are assessed and monitored using quantitative measures called performance metrics. In this project, the assessment of the models will be measured twice, for the detection phase and the prediction phase.

3.5.1 Players' Detection Evaluation

Precision, Recall, and Mean Average Precision (mAP) are the performance metrics that will be used in evaluating the performance of the detection models (YOLOv7, YOLOv8, YOLOv9).

- **Precision**

  It is a metric that measures a model's accuracy in predicting positive results [65]. Precision indicates how many items reported as positive are actually positive. It is calculated by the following formula [66]:

$$Precision = \frac{TP}{TP+FP} \times 100$$

  True positives (TP), False positives (FP).

- **Recall**

  It is a metric that measures a model's ability to detect all relevant records in a dataset [67]. Recall is calculated by the following formula [66]:

$$Recall = \frac{TP}{TP+FN} \times 100$$

  False negatives (FN)

- **Mean Average Precision (mAP)**

  It is a metric that evaluates the performance of object detection models [68]. It is computed by taking the Average Precision (AP) of each class [69].

  $$mAP = \frac{1}{Q}\sum_{q=1}^{Q} AP_q$$

  (Q) represents the total number of classes.

However, in this project, the players' detection models are evaluated automatically after training in Google Colab.

3.5.2 Collision Prediction Evaluation

Accuracy, F1-Score, precision, and recall are the performance metrics that will be used in evaluating the performance of the prediction system. In evaluating the collision prediction system, the TP represents the collisions predicted by the system that actually occurred, and the FN represents the actual collisions that occurred but the system did not predict them. The FP represents the collisions predicted by the system but no collisions occurred, and the TN represents the cases where no collision occurred and the system correctly predicted that no collision occurred.

- **Accuracy**

  The accuracy is a metric used to assess how accurate a model's predictions are. It is the ratio of correct prediction to all predictions produced. It is calculated by the following formula [67]:

  $$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100$$

  True negatives (TN).

- **F1-Score**

  It is a metric used for assessing how well classification models perform [67]. It is calculated by the following formula [66]:

  $$F_1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100$$

3.6 Summary

This chapter outlines the project's methodology, focusing on using Roboflow and Google Colab platforms with Python and its extensive libraries for DL tasks. It details the use of YOLO object detection algorithms (YOLOv7, YOLOv8, YOLOv9) and the integrated prediction algorithm. Additionally, the performance evaluation metrics such as precision, recall, mAP, accuracy, and F1-Score will be utilized to evaluate the system.

Chapter 4

Deep Learning System for Players' Collision Prediction

This chapter introduces a novel approach for building a system capable of detecting players on the field using top-down football footage. The system leverages YOLOv8 along with an integrated prediction algorithm to generate predictive alerts indicating potential collisions. The steps involved include dataset construction, detection models development, and an integrated prediction algorithm implementation. The system's results are presented and discussed at the end of this chapter.

4.1 An Overview of the Player Collision Prediction System

The football collision prediction system's development follows a two-phase approach. Initially, a specialized dataset is constructed by collecting and processing football match videos, segmenting them into individual frames. Subsequently, this dataset is refined using the Roboflow platform: splitting into training and validation sets, object annotation, data augmentation to enhance model robustness, and pre-processing. The second phase focuses on developing a YOLOv8-based framework for football collision prediction, concluding with a comprehensive evaluation of the system's ability to accurately predict player collisions on the football field.

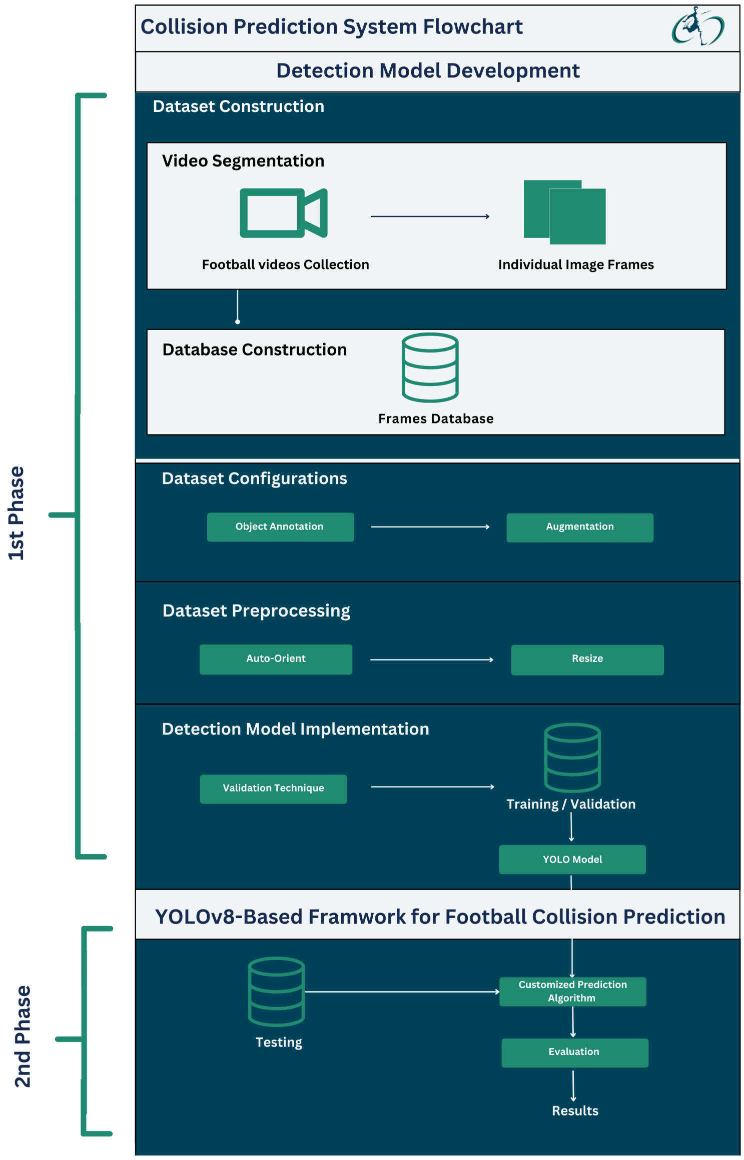


FIGURE 4.1: Football Collision Detection and Prediction Model Flow Chart

4.1.1 Detection Model Development

This section explores the performance differences between YOLOv7, YOLOv8, and YOLOv9 when applied to our project dataset. Through a comparative analysis, we aim to identify the YOLO model that offers the most efficient and accurate object detection capabilities. The efficacy of YOLO models is intrinsically tied to the nature of the object detection problem they are designed to solve, with performance varying significantly depending on factors like the number of object classes, object size, scale, and the complexity of the background environment [70]. To identify the most effective YOLO model for our player collision prediction task, we undertake a comprehensive model development process. This includes constructing a dataset through video segmentation and database creation, splitting the dataset, meticulously annotating objects, augmenting data for robustness, pre-processing for model compatibility, employing rigorous validation techniques, and ultimately comparing the performance of YOLOv7, YOLOv8, and YOLOv9.

- **Dataset Construction**
  The dataset utilized in this study was meticulously gathered from various online sources, focusing exclusively on football videos filmed from a top-down perspective. These videos were chosen to ensure consistency in viewpoint and facilitate accurate player tracking and collision detection. Only high-quality footage with minimal distortion and sufficient resolution was selected to ensure reliable model training. Careful consideration was given to include a diverse range of game scenarios, encompassing different playing styles, environmental conditions, and team compositions.

- **Video Segmentation and Database Creation**
  The process of creating a database involved segmenting football videos into individual images. This step was essential for preparing the dataset for annotation and subsequent model training. A custom Python script, leveraging libraries like OpenCV, was developed to efficiently extract frames from the videos, ensuring accuracy and consistency in the segmentation process. By converting the videos into a structured database of image frames, the dataset was transformed into a format suitable for annotation and subsequent utilization in training the collision detection and prediction model.

- **Dataset Splitting.**
  To ensure the effectiveness of training and evaluation for the model, the dataset underwent meticulous segmentation into three subsets:

  - The training set (70%): This portion of the dataset is used to train the model to recognize patterns and features. This enhances the model's abilities by exposing it to a large amount of data.
  - The validation set (15%): Essential for assessing model performance and avoiding overfitting during training, which results in adjustments for enhanced accuracy and generalization.
  - The testing set (15%): Reserved for the final evaluation of unseen data, providing a fair evaluation of the model's performance, and offering insights into its prediction capabilities.

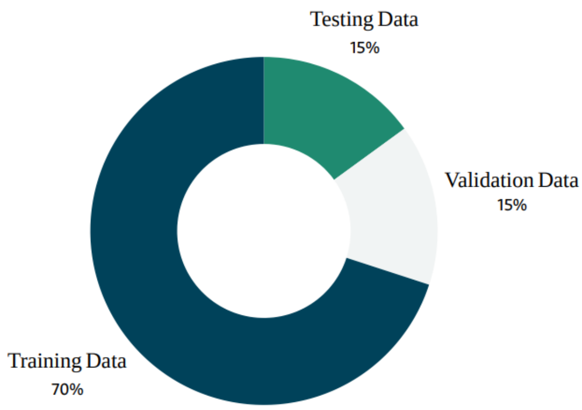


FIGURE 4.2: Dataset Splitting

- **Dataset Configurations**
  The Roboflow platform offers a suite of services that facilitate successful model configurations, and in this project, these services were leveraged for object annotation, augmentation, and dataset pre-processing.
  - **Object Annotation**
    Object annotation is a fundamental step in training object detection models, enabling them to recognize and localize specific entities within images. In this project, object annotation was conducted using the Roboflow platform, a comprehensive tool designed for efficient and accurate dataset labeling. As illustrated

in Figure 4.3, Roboflow provided a user-friendly interface for annotating football players within the image frames extracted during the video segmentation phase. Each player within the frames was meticulously annotated, specifying their bounding boxes to indicate their positions accurately. Additionally, Roboflow facilitated the management and organization of annotated data, ensuring consistency and completeness throughout the labeling process. The annotated dataset served as the foundation for training the YOLOv7, YOLOv8, and YOLOv9 models, enabling them to learn the intricate patterns associated with football player collisions and enhance their predictive capabilities.

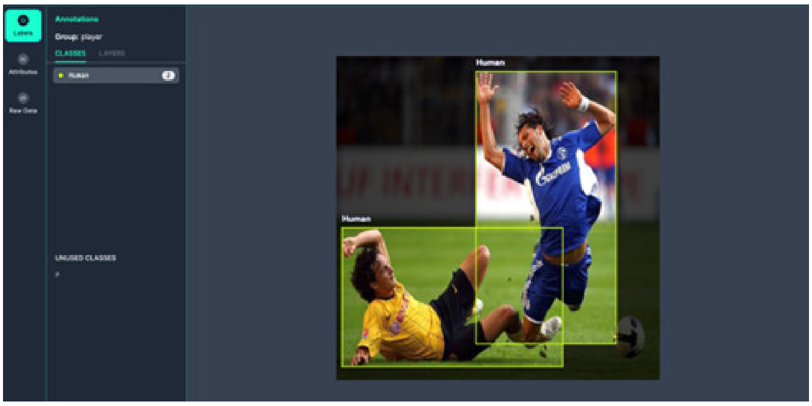


FIGURE 4.3: Annotations for Football Players

- **Dataset Augmentation**

  The augmentation process involved a series of transformations applied to the original dataset to increase its diversity and robustness:

  * 90° Rotate: Clockwise, Counter-Clockwise, Upside Down: This augmentation technique involves rotating the image by 90 degrees in three different directions: clockwise, counter-clockwise, and upside down. This simulates variations in camera angles and perspectives commonly encountered in football game footage.

* Rotation: Between -15° and +15°: Rotation augmentation is applied within a range of -15° to +15°. This introduces variability in player positions and orientations, mimicking the dynamic nature of player movements during a football match.
* Shear: ±10° Horizontal, ±10° Vertical: Shear transformations are applied horizontally and vertically within a range of ±10°. This introduces distortions to the image, representing perspective changes and camera movements.
* Grayscale Apply to 15% of images: Grayscale conversion is applied to 15% of the images in the dataset. This reduces the reliance on color information, making the model more robust to variations in lighting conditions and color distributions.
* Hue: Between -30° and +30°: Hue adjustment is made within a range of -30° to +30°. This changes the overall color tone of the image, simulating variations in environmental lighting conditions.
* Saturation: Between -44% and +44%: Saturation adjustment is applied within a range of -44% to +44%. This alters the intensity of colors in the image, adding variability to the color saturation levels.
* Brightness: Between -15% and +15%: Brightness adjustment is made within a range of -15% to +15%. This simulates changes in overall brightness levels, representing fluctuations in lighting conditions.
* Exposure: Between -10% and +10%: Exposure adjustment is applied within a range of -10% to +10%. This simulates changes in the exposure settings of the camera, affecting the overall brightness and contrast of the image.

These augmentations were strategically chosen to simulate various real-world conditions and challenges encountered during football games, such as changes in lighting, camera perspectives, and image distortions. Each transformation aimed to enhance the dataset's diversity while preserving the integrity of the underlying game footage.

The original dataset consisted of 2408 images capturing various football game scenarios. After augmentation, as shown in Figure 4.5 the dataset has grown to 4733 images. This represents an increase of 96% in the size of the dataset. The figure also shows how the dataset is split into training, validation, and test sets. The training set is the largest set, containing 4650 images (98% of the augmented dataset). The validation set is much smaller, containing 83 images (2% of the augmented dataset). As we mentioned before, the test set was 15% and was created manually outside of RoboFlow.

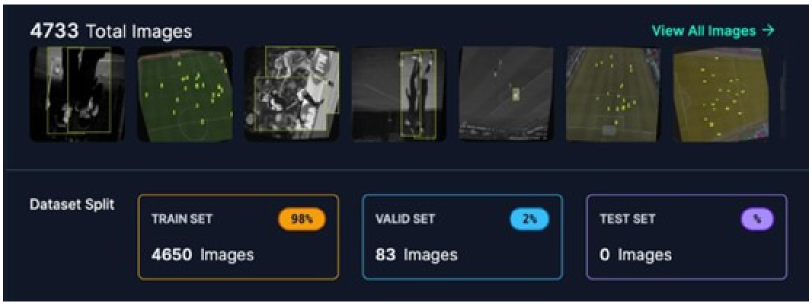


FIGURE 4.4: Dataset split after augmentations

- **Dataset Pre-processing**
  Prior to model training, the dataset underwent pre-processing steps to standardize its format and dimensions:
  - Auto-Orient: Auto-orientation is applied to standardize the orientation of all images in the dataset. This ensures consistency in image orientation, eliminating inconsistencies that may arise from different recording devices or angles.
  - Resize: Stretch to 640x640: Resizing is performed to standardize the dimensions of all images in the dataset to 640x640 pixels. This facilitates uniformity in image size, enabling efficient processing and model training.

These pre-processing steps were essential to ensure uniformity across the dataset, enabling seamless integration into the DL pipeline. Auto-orientation eliminated inconsistencies in image orientation while resizing ensured all images were uniformly scaled to facilitate efficient model training and inference.

- **Validation Technique for Robust Models:**
  In CV tasks like detecting players in sports videos, a model's ability to consistently and accurately interpret unseen footage is crucial. Validation techniques address this by evaluating the model's performance on data it wasn't exposed to during training. Similar to ML models, CV models depend on validation to ensure their effectiveness on unseen data. This process helps prevent the model from simply memorizing the training examples and instead focuses on learning patterns that translate to real-world scenarios. Several methods exist for allocating datasets for CV model validation, all aiming to ensure the model's accuracy and reliability in real-world scenarios. In this project, we employed the common and effective Hold-Out technique.

  Here is how the Hold-Out technique works:

  - Training on Training Set: This portion of the dataset is used to train the model to recognize player patterns and features. By exposing the model to a large amount of data, it enhances its detection abilities.

  - Evaluation on Validation Set: This step was essential for assessing the model's performance. We used precision, recall, and mAP metrics to evaluate how well the model detected the players in the validation set. This ensures that the model can detect unseen video footage, leading to more accurate player detection in real-world scenarios.

4.1.2 Detection Model Results

The findings of YOLOv7, YOLOv8, and YOLOv9 models for football player detection show significant differences. YOLOv8 had the greatest precision score, narrowly beating YOLOv7 and YOLOv9. On the other hand, YOLOv8 had the highest recall, indicating that it could catch a greater proportion of real positives than the other models. However, in terms of mAP, YOLOv8 once again outperformed YOLOv7 and YOLOv9, scoring somewhat higher. These findings indicate that, whereas YOLOv8 excelled in precision, recall, and mAP, YOLOv9 achieved competitive precision and mAP scores.

Figure 4.5 illustrates the precision score for all versions of YOLO, YOLOv7 achieved 93.3%, YOLOv8 achieved 93.4%, and YOLOv9 achieved 94.8%.

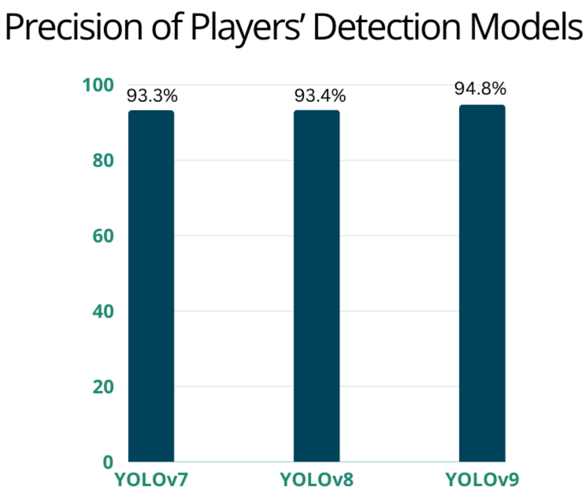


FIGURE 4.5: Precision of YOLOv7, YOLOv8, and YOLOv9

Figure 4.6 illustrates the recall score for all versions of YOLO, YOLOv7 achieved 92.4%, YOLOv8 achieved 95.3%, and YOLOv9 achieved 90.5%.

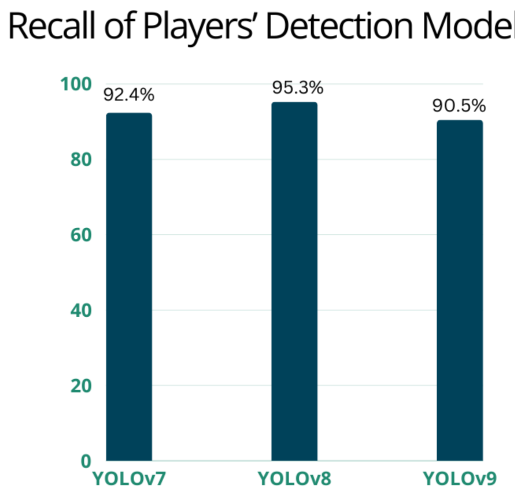


FIGURE 4.6: Recall of YOLOv7, YOLOv8, and YOLOv9

Figure 4.7 illustrates the mAP score for all versions of YOLO, YOLOv7 achieved 96.6%, YOLOv8 achieved 96.9%, and YOLOv9 achieved 96.4%.

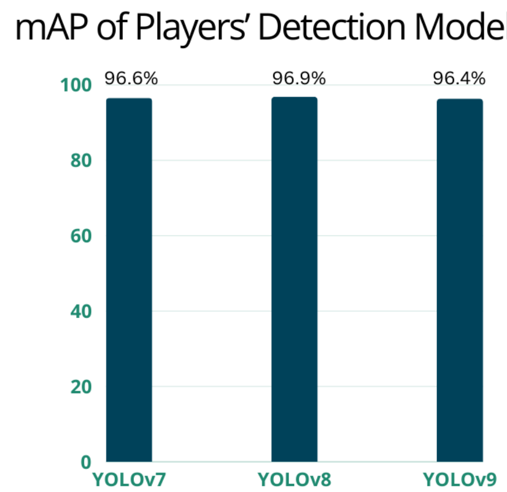


FIGURE 4.7: mAP of YOLOv7, YOLOv8, and YOLOv9

While the latest version of the YOLO object detection model, YOLOv9, incorporates novel features such as the PGI and the GELAN which have been shown to enhance model accuracy [62], the results of the current work indicate that the previous version, YOLOv8, demonstrated superior performance as illustrated in Table 4.1. As such, YOLOv8 was ultimately selected to drive the development of the collision prediction system.

Comparison Field	YOLOv7	YOLOv8	YOLOv9
Recall	3	1	2
Precision	3	2	1
mAP	2	1	3
Average	3	1	2

TABLE 4.1: Ranking of Algorithms in Terms of Different Comparison Field

4.1.3 YOLOv8-Based Framework for Football Collision Prediction

The integrated prediction algorithm introduced a new approach that integrates YOLOv8 to accomplish the prediction task. The core characteristics of the player collision prediction system are also discussed in this section. It concentrates on the following elements.

- **Object Tracking:**
  Object tracking involves detecting objects between frames by taking advantage of their temporal and spatial properties. The basic idea of object tracking is found in this method, which is as simple as gathering the first set of detection, assigning them unique IDs, and tracking them over frames.
  Object tracking has been used in several domains such as pedestrian tracking [71], vehicle tracking [72], and player tracking [33]. Object tracking can be classified into two categories: Single Object Tracking (SOT) and Multiple Object Tracking (MOT) [73]. In the player collision prediction system, we used frame-by-frame multiple-object detection.

- **Multiple Object Tracking**
  The main tasks of a multiple-object tracking system are to find several objects in a frame, assign and maintain their identities, and track an object's path across input frames. The following algorithm illuminates the multiple object tracking of players' collision prediction:

Algorithm 1 Multiple Object Tracking (MOT)

---

1. Initialization:
   - Create an empty dictionary: `center_points`.
   - Set counter: `id_count = 0`.
2. Update(`objects_rect`: list of bounding boxes):
   - Create an empty list: `objects_bbs_ids`.
3. For each `rect` in `objects_rect`:
   (a) Calculate center coordinates (`cx, cy`) of `rect`.
   (b) Check if object is already being tracked:
      - Set `same_object_detected = False`.
      - For each `id, pt` in `center_points`:
         * Calculate distance between (`cx, cy`) and `pt`.
         * If distance < threshold (35):
            · Update `center_points[id]` with (`cx, cy`).
            · Add [`x, y, w, h, id`] to `objects_bbs_ids`.
            · Set `same_object_detected = True`.
            · Break from loop.
   (c) If not `same_object_detected` (new object):
      - Add (`cx, cy`) to `center_points` using `id_count` as key.
      - Add [`x, y, w, h, id_count`] to `objects_bbs_ids`.
      - Increment `id_count`.
4. Remove IDs not being tracked:
   - Create a new dictionary: `new_center_points`.
   - For each `obj_bb_id` in `objects_bbs_ids`:
      * Extract `object_id` from `obj_bb_id`.
      * Add `center_points[object_id]` to `new_center_points`.
   - Set `center_points = new_center_points`.
5. Return `objects_bbs_ids`.

Algorithm 1 displays the list of tracked items, each with a distinct ID and related data, which is kept up to date using the multiple object tracking algorithm. It processes the bounding box of each identified object in each new frame. The information on an existing track is updated if the item matches it. Otherwise, the object is identified as new and given a new ID before being added to the tracking list. Tracks that no longer have matching objects are removed from the list once all the objects in the current frame have been processed. The revised list of tracked objects was then returned by the algorithm prepared for the following frame.

- **Prediction Algorithm Integrated with Yolov8**
  We treated collision prediction as a ternary classification problem in our study. This implies that our objective was to create a model that can classify every video frame as safe, in close proximity, or colliding with players. The integrated prediction algorithm takes as input the confidence and distance threshold, and the uploaded video, then executes some processing on them. lastly, predicts if there is a collision, close proximity, or safety. Therefore, we used a flowchart to clarify class labels.
  As shown in Figure 4.8 the flowchart outlines a decision process for detecting and predicting player collisions. It begins by calculating the Euclidean distance between players and their overlap ratio by the following formulas [66], [74]:

$$Distance = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

  - (x1) and (x2) are the x-coordinates of the first and second points, respectively.
  - (y1) and (y2) are the y-coordinates of the first and second points, respectively.

$$OverlapRatio = \frac{IntersectionArea}{BoundingBoxArea}$$

- (IntersectionArea) The intersection area between two bounding boxes is determined by calculating the overlapping region's width and height.

- (BoundingBoxArea) The area of each bounding box is computed by multiplying its width and height.

Based on these calculations, it determines if the situation is safe, a collision is likely, or the players are in close proximity.

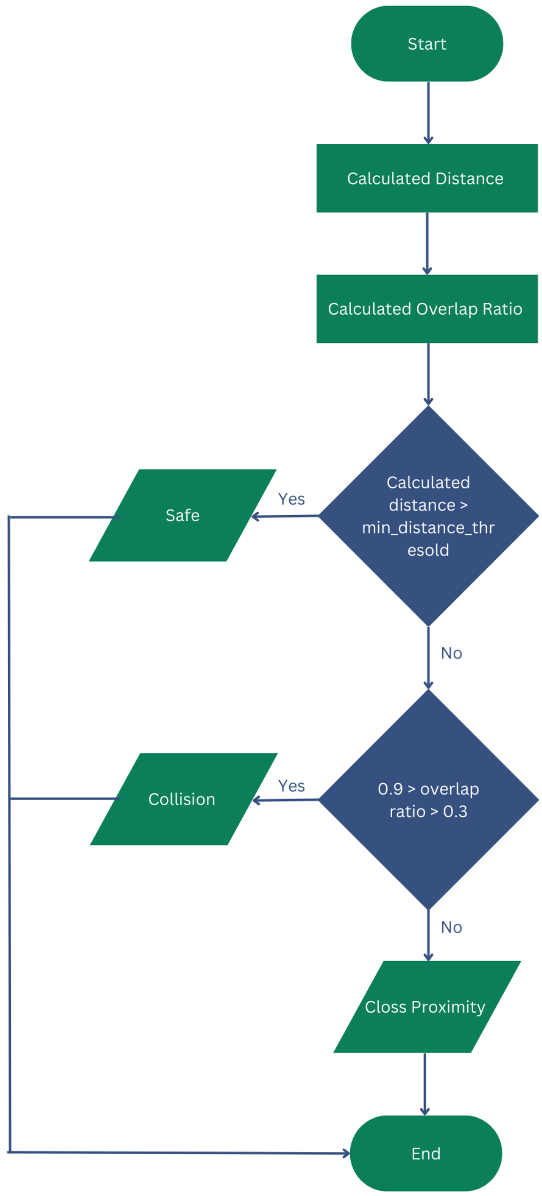


FIGURE 4.8: Integrated Prediction Algorithm Flowchart

The flowchart in Figure 4.8 determines the status of the player collision prediction system by analyzing the relative position and overlap. Initially, it calculates both the distance between players and the extent to which they overlap, expressed as a ratio. If the calculated Euclidean distance is greater than the distance the user enters, the players are considered safe. However, if the distance fell below this threshold, the flowchart evaluated the overlap ratio. An overlap ratio within a specific range (0.3 to 0.9) indicated that the players collided. Conversely, an overlap ratio is considered in close proximity if it is not within the specific range (0.3 to 0.9). The final outcome of the flowchart categorizes the situation as safe, close proximity, or collision, providing a clear assessment of the collision risk based on the calculated parameters.

After extensive testing on eight videos (more than 900 frames), the 0.3 to 0.9 range was chosen to balance between specificity (reducing false alarms) and sensitivity (predicting real collisions).

4.1.4 Collision Prediction System Results and Discussion

This section presents an evaluation of the collision prediction algorithm integrated with the YOLOv8 object detection model. Eight videos, comprising over 900 frames, were analyzed, and suitable parameters (distance threshold and confidence threshold) were determined for each video to optimize the algorithm's accuracy. Ground-truth collisions, observed directly in the videos, were compared to the algorithm's predictions by utilizing state information (safe, close proximity, collision). Algorithm performance was assessed using accuracy, precision, F1-score, and recall metrics (detailed in Chapter 3) to demonstrate its predictive capability. The results of each video are presented in Table 4.2.

No.	Video	Frame	Real Collision	Predicted Collision	Correctly Predicted	Recall	Precision	F1-score
1	test1	206	1	2	T	100%	50%	67%
2	test3	124	1	1	T	100%	100%	100%
3	test4	124	1	0	F	0%	0%	0%
4	test6	103	1	2	T	100%	50%	67%
5	test9	117	1	2	T	100%	50%	67%
6	test12	100	1	2	T	100%	50%	67%
7	test13	103	1	1	T	100%	100%	100%
8	test14	65	1	1	T	100%	100%	100%
Average				Accuracy	88%	88%	63%	71%

TABLE 4.2: System's Results

A satisfactory performance in collision prediction was demonstrated by this integrated algorithm employed in YOLOv8, with a high recall of 88% and accuracy of 88%, indicating its ability to accurately identify collisions in the majority of situations and capture the majority of actual collisions present in the video data. The algorithm can be oversensitive and produce alerts or predictions more frequently than necessary, depending on the supplied parameters, as suggested by the moderate F1-score of 71% and lower precision of 63%. These results suggest that there is still potential for improvement, especially in terms of reducing false positives.

In contrast to the [11] model's average accuracy of 73% for men and 70% for mixed-gender datasets, this integrated method exhibits a higher accuracy of 88% in collision prediction. While both models exhibit promise, the Head Collision Severity Prediction model's smaller dataset casts doubt on its generalizability compared to the algorithm's evaluation on a larger dataset. The additional metrics of this algorithm, recall and precision, offer a more complete understanding of its performance.

Despite the fundamental differences in the methodologies employed by the [10] model and this integrated prediction algorithm both predict collisions, their methods are significantly different. Using YOLOv8 for object detection and video analysis on a dataset higher than the others, this customized prediction system reports 88% accuracy and offers detailed performance metrics. On the other hand, in a smaller pilot study, RFID monitoring and an ML algorithm were used with the goal of real-time collision prevention via haptic feedback. This study did not have the additional metrics needed for a thorough comparison, but it reported a low false-alarm rate, which suggests high accuracy.

4.2 Summary

This chapter successfully achieved its primary objectives. Firstly, a YOLOv8 model was developed for object detection, capable of detecting and tracking football players. This model outperformed YOLOv7 and YOLOV9, as demonstrated in Table 4.1. Secondly, a specialized system was created to effectively predict collisions between football players. This was achieved by integrating the YOLOv8 model into an integrated prediction algorithm, yielding optimal results with an accuracy rate of 88%. Finally, to make this system more practical, a web-based system was built to extract player collision data from football match video footage which will be discussed in Chapter 5.

Chapter 5

The Development of The Web-Based Prediction System

The construction of the web-based prediction system is discussed in detail in this chapter. We begin by outlining the basic idea behind the system, as well as its features and technical details. Subsequently, the system's architecture is examined, with particular attention paid to how the front-end (user interface) and back-end (data processing and prediction creation) interact. A system flow diagram is used to graphically depict this, and it shows how the design progressed from low-fidelity to high-fidelity wire-frames. Finally, we demonstrate the system's potential by presenting the outcomes of the website-based player collision prediction system.

5.1 The Web-Based Prediction System

The primary objective of creating a football player collision prediction system is to enable coaches, performance analysts, and medical personnel to predict player collisions promptly and easily. This will enable them to proactively identify players who may be at risk of collision during practice and games by extracting pertinent data about the players during the game. following the model training with various YOLO algorithms, specifically YOLOv7, YOLOv8, and YOLOv9. With a maximum mAP of 96.9%, the YOLOv8 algorithm was selected as the foundational algorithm for the web-page prediction system.

The system allows the user to enter the confidence threshold and distance threshold, and upload the video that he/she wants to be predicted. Using the integrated prediction algorithm, the system will detect and predict the collision between players and will extract an Excel sheet that stores the data of that video and a pie chart image that shows the percentage of safe, close proximity, and collisions that occur in the video.

5.1.1 Functionalities of the Web-Based Prediction System

This system provides a service that uses a personalized prediction algorithm to examine the player's video data and identify possible collisions. Users can submit videos, choose their preferred confidence and distance criteria, and obtain data outputs in the form of an Excel file and a pie chart, along with the video that includes the predicted collisions.

Function	**Description**	**Side**
User Input	The user can specify the value of every required input	(client-side)
Video Processing	The system uses a customized prediction algorithm to detect potential collisions between players	(server-side)
Output Extraction	The system will extract the data file from the uploaded video with the predicted collisions	(server-side)

TABLE 5.1: Functionalities of the Web-Based Prediction System

5.1.2 Technical Aspects of the Web-Based Prediction System

This section discusses the technical aspects of our web application for predicting collisions between players. This application is written in Python, a programming language with a wide range of libraries that make it easy to read and understand [57].

Programming language	Python (3.12.2), HTML, CSS, JavaScript
Libraries used	Flask, OpenCv, Pandas, ultralytics

TABLE 5.2: Technical Aspects of the Web-Based Prediction System

5.2 Website Architecture

In this section, a Single Page Application (SPA) with server-side rendering (SSR) was used. The purpose of this architecture is to predict player collisions in videos. The system uses Python with Flask for the backend and OpenCV, Pandas, and YOLOv8 for detecting and predicting, video processing, and data analysis.

5.2.1 The Website Flow

As shown in Figure 5.1, the system comprises a back-end and front-end web-based user interface. Users upload a video file and configure settings such as confidence and distance thresholds on the front end. These data are transmitted to the back-end. The back-end recognizes and tracks the motion of players in the video using CV libraries. It computes the distance from one another and compares it with the set thresholds. The back-end records the incident and creates notifications on the video in the event of a violation. Additionally, a report containing analysis and visualization was produced by the back end. Ultimately, the video, report, and pie chart images that have been processed are compressed and returned to the user's browser for download. The Flask-SocketIO library enables two-way communication between the front and back ends.

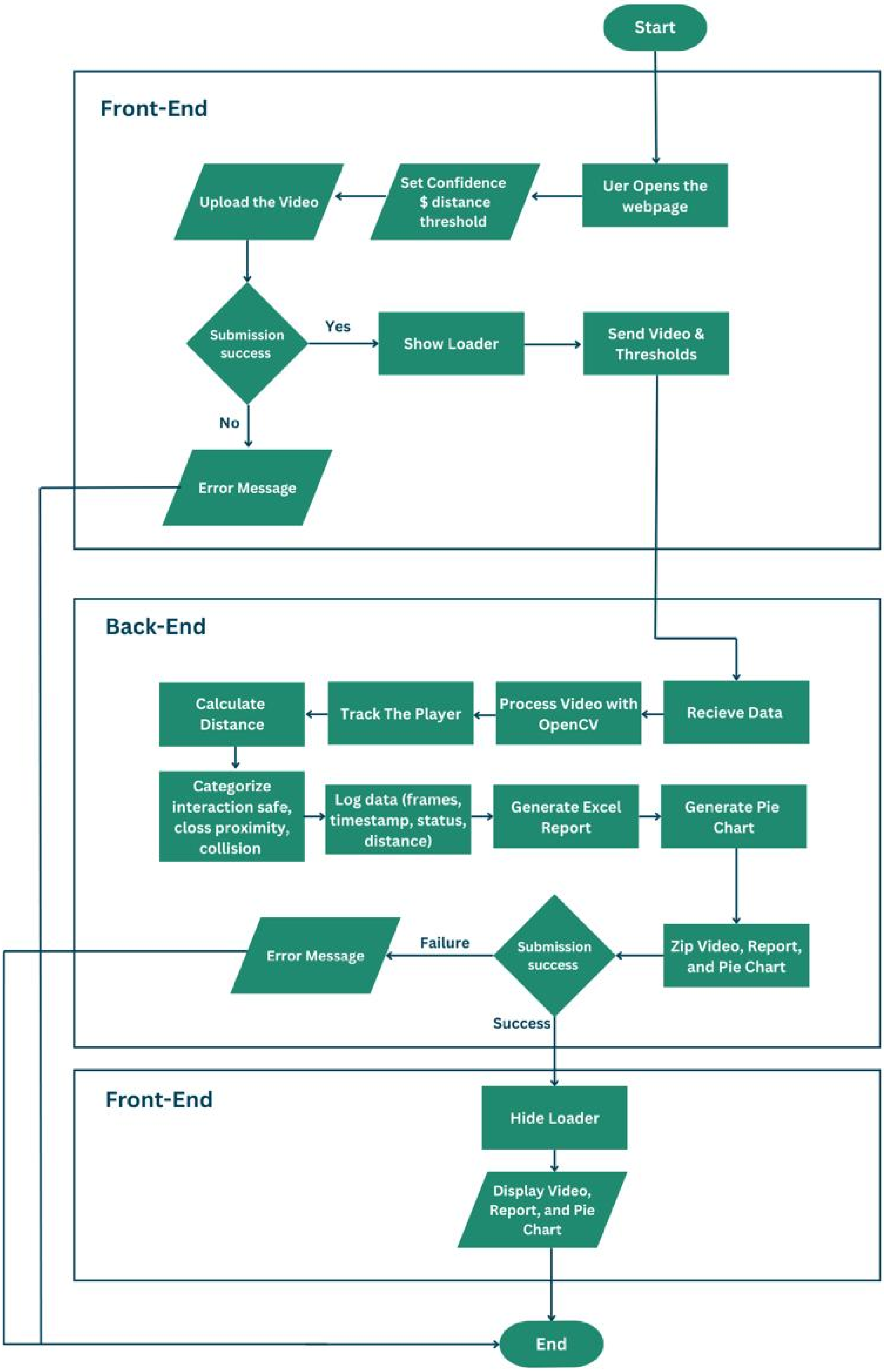


FIGURE 5.1: Website Flow

5.2.2 Front-End (SPA)

The front-end of the system was developed using the free and open-source bootstrap framework, which is used to create online applications and web-pages, serves as the user interface and manages interactive processes. The system operates as follows:

1. User Interface (UI) Creation	
Programming Language	**Description**
(HTML, CSS)	The web-page was initially viewed by the users in their browsers. This shows them the user interface, which has the logo of the website and spaces marked as forms for user input. Next, the user fills the appropriate input areas with the desired values for the confidence and distance thresholds. Subsequently, by selecting the file upload option, they choose the video file that they wish to predict a collision. Finally, the user hits the "Submit" button to initiate the prediction process.
2. Form of Submission and the Loader	
Programming Language	**Description**
(HTML, CSS, JavaScript)	A JavaScript event named onsubmit is triggered on the form when the "Submit" button is clicked. The showLoader function is then invoked by this event. The loader element, which contains a loading animation, becomes visible as soon as the showloader function is called. It does this by assigning the display style of the element to "block." Furthermore, the method uses a semi-transparent layer as an overlay element. The interaction between the user and the remainder of the page was blocked by this overlay during the prediction process.
3. Communication with Back-end	

Programming Language	Description
(Python, JavaScript)	It is not what you might think when you click "Submit" to start the player collision prediction video on the page. Instead, real-time communication between a web server and web clients is implemented via the Flask-SocketIO library. In other words, it allows for a two-way dialogue to occur between the web application and the user's browser, which was previously responsible for sending the data to the web server. The uploaded video file and the confidence and distance threshold values that the user previously entered were included in the data.
4. Processing on Backend	
Programming Language	**Description**
(Python)	The uploaded video file and user-specified thresholds are among the data that the front-end sends to the server. The server then processes the video frame by frame using strong libraries, such as OpenCV and YOLOv8. This enables it to recognize certain players in a video and to determine their respective distances from one another. The back-end separates collisions into safe, close proximity, and actual collision categories based on set thresholds and measured distances. It goes one step further and uses tools such as Pandas and Matplotlib to create pie charts, which are visual representations of this collision data.
5. Response from Back-end	
Programming Language	**Description**

(Python)	After processing is completed, the back-end uses a Socket.IO connection to transmit a response back to the front-end. This response includes the collision statistics as an Excel sheet, a pie chart that shows the percentage of the status of either safe, close proximity, or collision, and a video detected and predicted in a folder.
6. Front-End Receives Response and Updates	
Programming Language	**Description**
(Python, JavaScript)	The JavaScript code on the front end waits for the server's response using a Socket.IO event listener. The code initially examines the status message in a status_update event listener after receiving it. The loader and overlay elements disappear if the analysis is successful, as indicated by "data.status ===’success’" and the hideLoader method takes over. Next, in a folder, the server sends back the collision data (statistics), a pie chart that shows the percentage of the status of either safe, close proximity, or collision, and a video with the detection and prediction results.

TABLE 5.3: Front-End (SPA)

5.2.3 Back-End (SSR)

The Flask framework, a micro-framework created to quickly create a web application, was used to build the back-end of the system. Developers have the freedom to add features as needed while implementing the fundamental functionality, as it is only implemented for now [75]. The Back-End of the system handles video processing for player collision prediction and communicates back with the front end. Specifically, the code imports the following libraries: Matplotlib for visualization, pandas, OpenPyxl, libraries for CV (OpenCV, YOLO), libraries for working with files (Werkzeug, UploadSet), libraries for tracking objects (tracker.py, not provided), libraries for building web applications (Flask), and libraries for cross-origin requests (Flask-CORS, Flask-SocketIO). The system operates as follows:

1. Video Processing	
Programming Language	**Description**
(Python)	The method uses a pre-trained YOLOv8 model to identify players in each frame after receiving the user's input for the video filename, confidence threshold, and distance threshold. It measures the distance between players by tracking them across multiple frames. To determine proximity to one another, bounding box overlap is employed. Lastly, it logs pertinent data for analysis and establishes collision status (safe, close proximity, collision) depending on the overlap and distance thresholds.
2. Creating The Excel Report	
Programming Language	**Description**
(Python)	The code generated an Excel workbook using OpenPyxl. The log DataFrame was added to a sheet and conditional formatting was applied according to the collision state. A formatted title and header rows were added after the logo. The status counts—safe, in close proximity, or in a collision— were computed and recorded on an additional sheet. Finally, the workbook is saved after creating a pie chart representing the collision distribution using Matplotlib, saving it as an image, and inserting it into a specific "Graph" sheet.
3. Zipping Output Files	
Programming Language	**Description**

(Python)	The script generates a ZIP file that includes the video that has been processed, an Excel report, and a pie chart that illustrates the percentage of the status that is either safe, in close proximity, or colliding.
4. Sending Status Update	
Programming Language	**Description**
(Python)	Successful processing is indicated by the Back-End sending the Front-End client a message - status_update - through the SocketIO connection.
5. Returning the ZIP Archive	
Programming Language	**Description**
(Python)	The pie chart, processed video, and report are all compressed and returned as downloadable files to the Front-End.

TABLE 5.4: Back-End (SSR)

5.3 The System Design (website)

In this section, the web application is designed to offer a user-friendly, single-page interface built with Proto.io, which allows users to complete all tasks on one webpage. With this simplified method, customers are guided through a clear process that starts with setting collision prediction parameters, such as confidence and distance thresholds. These settings guarantee that the system identifies events that fit the user's requirements. Subsequently, the user can upload the footage for prediction and detection.

5.3.1 Low-Fidelity Wire-frame

The basic layout of our web application and user flow were established using low-fidelity wire-frames as guides. By concentrating on the essential structure and functionality, we were able to rapidly conceptualize and iterate several design approaches without becoming bogged down in the visual minutiae.

As shown in Figure 5.2, the key components of the wire-frames: a section labeled "Set Prediction Parameters" featuring separate input fields for "Confidence Threshold" and "Distance Threshold," which lets users adjust the prediction accuracy; an area labeled "Upload Video" where users could choose which videos to upload; a button labeled "Submit" to start the analysis; and a space set aside for the application logo.

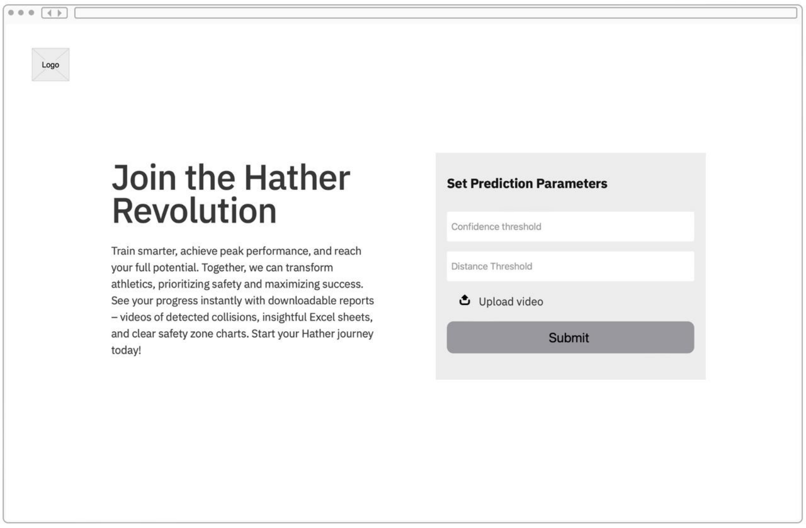


FIGURE 5.2: Low-Fidelity Wire-frame

5.3.2 High-Fidelity Wire-frame (Prototype)

A more sophisticated prototype was created to replicate the main features of the application by building on low-fidelity wire-frames. According to Figure 5.3.

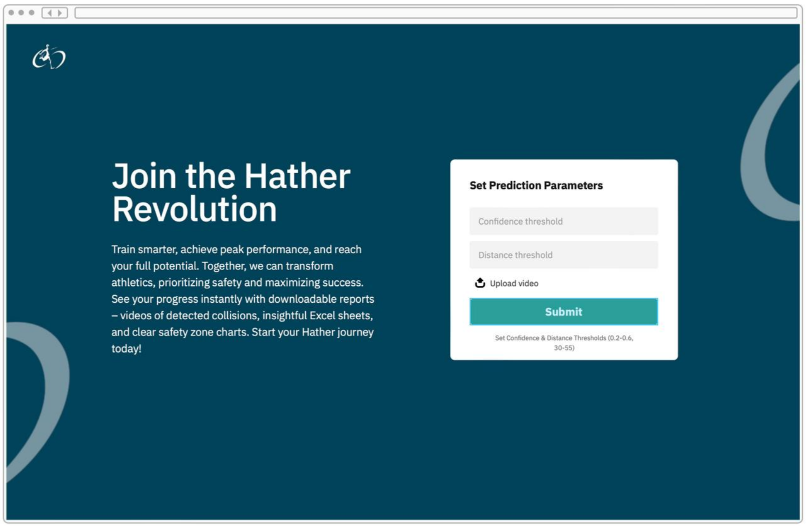


FIGURE 5.3: High-Fidelity Wire-frame

The prototype incorporates interactive input fields for confidence and distance thresholds, enabling users to fine-tune their preferences. Figure 5.4 illustrates the expected functionality for choosing and submitting video files using a simulated "Upload Video" button. Furthermore, as illustrated in Figure 5.5, the prototype included example visualizations to exhibit the kind of output the program would produce, providing users with an idea of the finished products.

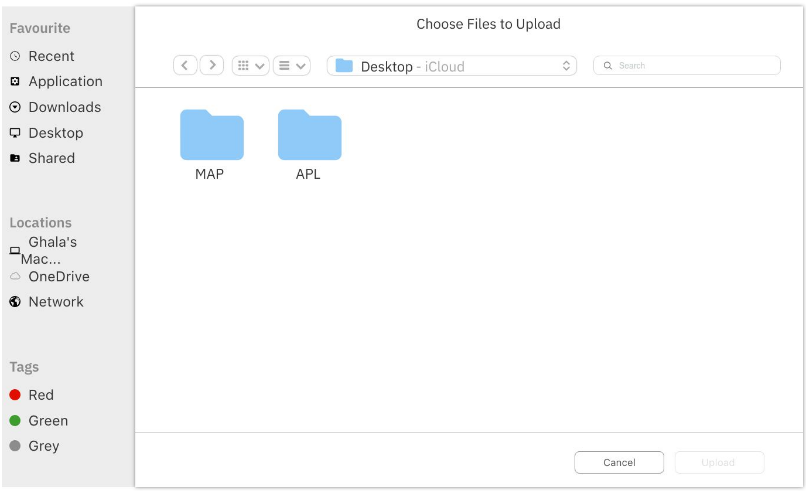


FIGURE 5.4: Upload Video

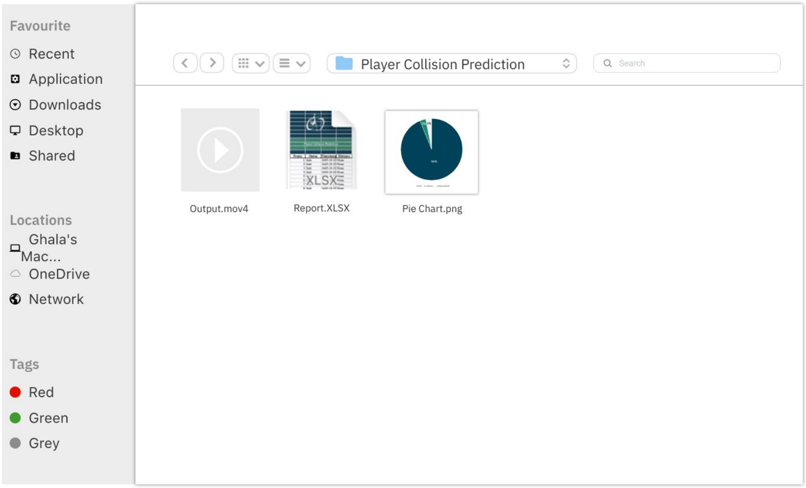


FIGURE 5.5: The Output

5.3.3 Player Collision Prediction System Outputs

In this section, the last objective, which was to construct a web application for predicting player collisions in videos, was successfully achieved. This system utilizes the YOLOv8 algorithm to identify players and evaluate their interactions according to pre-established thresholds. As shown in Figure 5.6, users can customize these thresholds and upload the video to be analyzed. A loading indicator appears following the submission, as shown in Figure 5.7. After processing the data on the Back-End and zipping it, a folder containing the output is delivered to the user's device, as illustrated in Figure 5.8.An example of the output generated by the proposed system is shown in Figure 5.9.

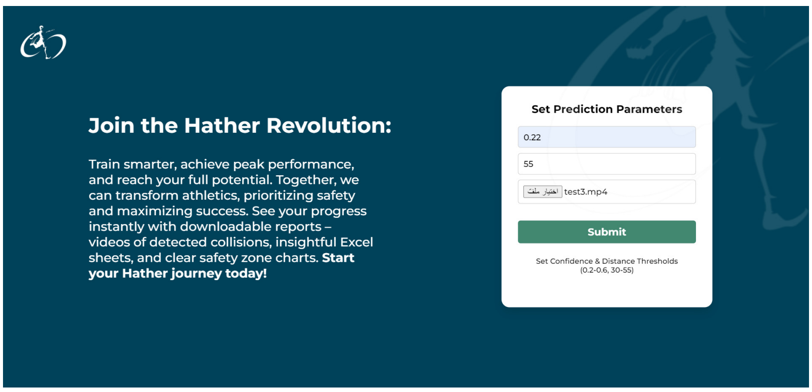


FIGURE 5.6: Set Prediction Parameters

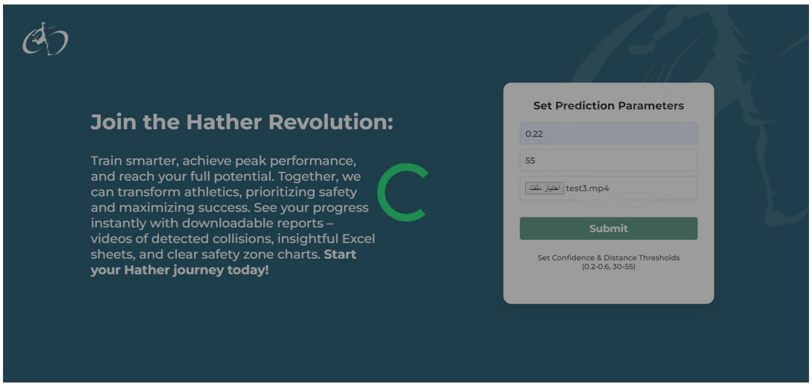


FIGURE 5.7: Loading Indicator

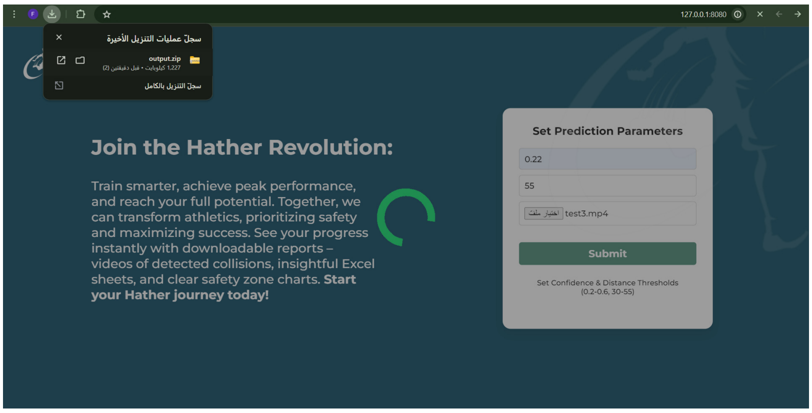


FIGURE 5.8: The Folder

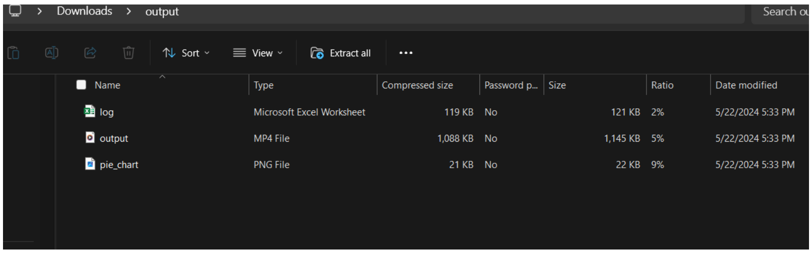
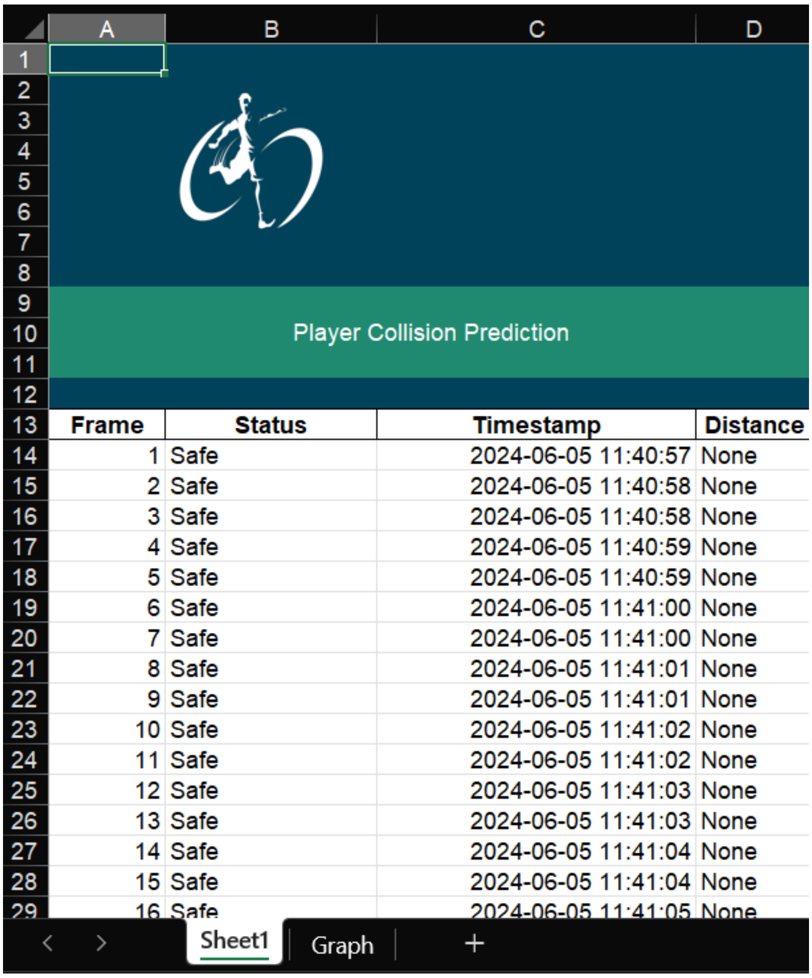


FIGURE 5.9: The Output

As shown in Figure 5.9 the folder contains 3 outputs. Figure 5.10 shows the first file is the Excel sheet, which contains the frames, status, timestamps, and the distance of the collision and close proximity. Figures 5.11, 5.12 and 5.13 show the video detected and predicted. Figure 5.14 shows the pie chart, which shows the percentage of the statue in the video.



Frame	Status	Timestamp	Distance
1	Safe	2024-06-05 11:40:57	None
2	Safe	2024-06-05 11:40:58	None
3	Safe	2024-06-05 11:40:58	None
4	Safe	2024-06-05 11:40:59	None
5	Safe	2024-06-05 11:40:59	None
6	Safe	2024-06-05 11:41:00	None
7	Safe	2024-06-05 11:41:00	None
8	Safe	2024-06-05 11:41:01	None
9	Safe	2024-06-05 11:41:01	None
10	Safe	2024-06-05 11:41:02	None
11	Safe	2024-06-05 11:41:02	None
12	Safe	2024-06-05 11:41:03	None
13	Safe	2024-06-05 11:41:03	None
14	Safe	2024-06-05 11:41:04	None
15	Safe	2024-06-05 11:41:04	None
16	Safe	2024-06-05 11:41:05	None

FIGURE 5.10: Excel Sheet

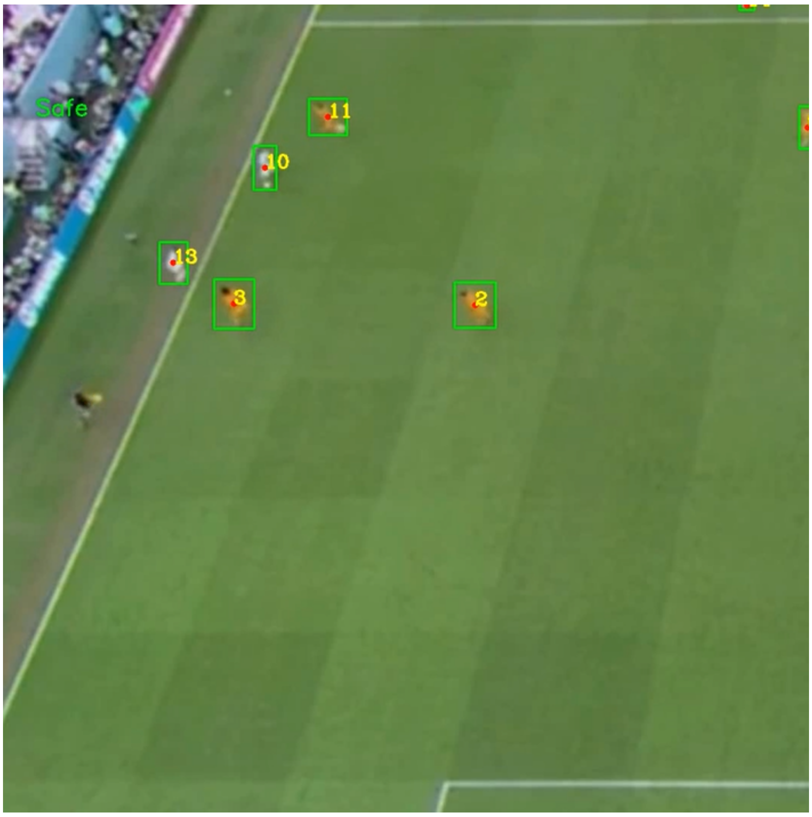


FIGURE 5.11: The output video status: safe

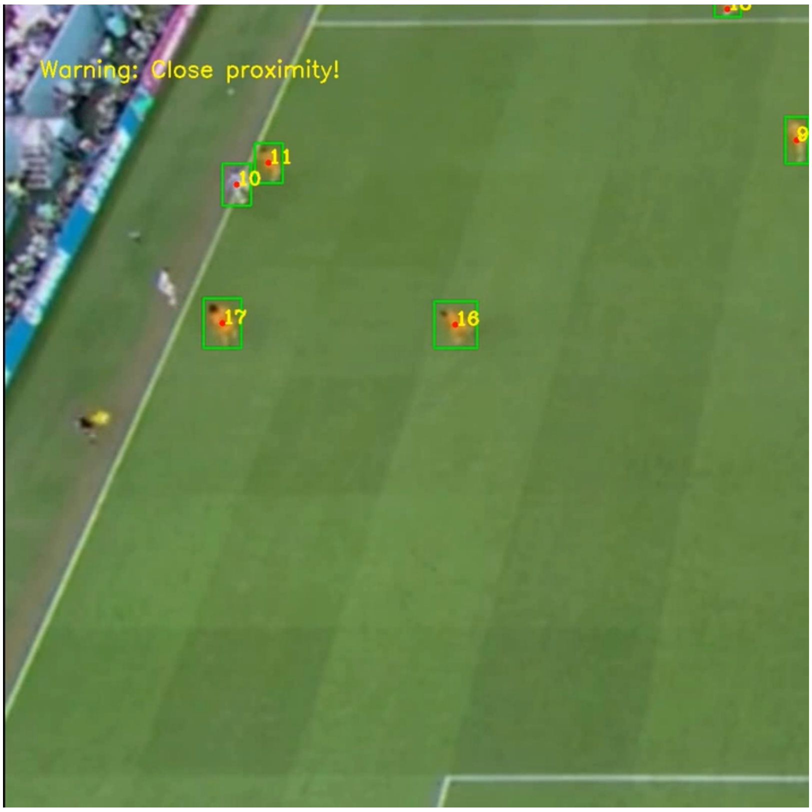


FIGURE 5.12: The output video status: warning

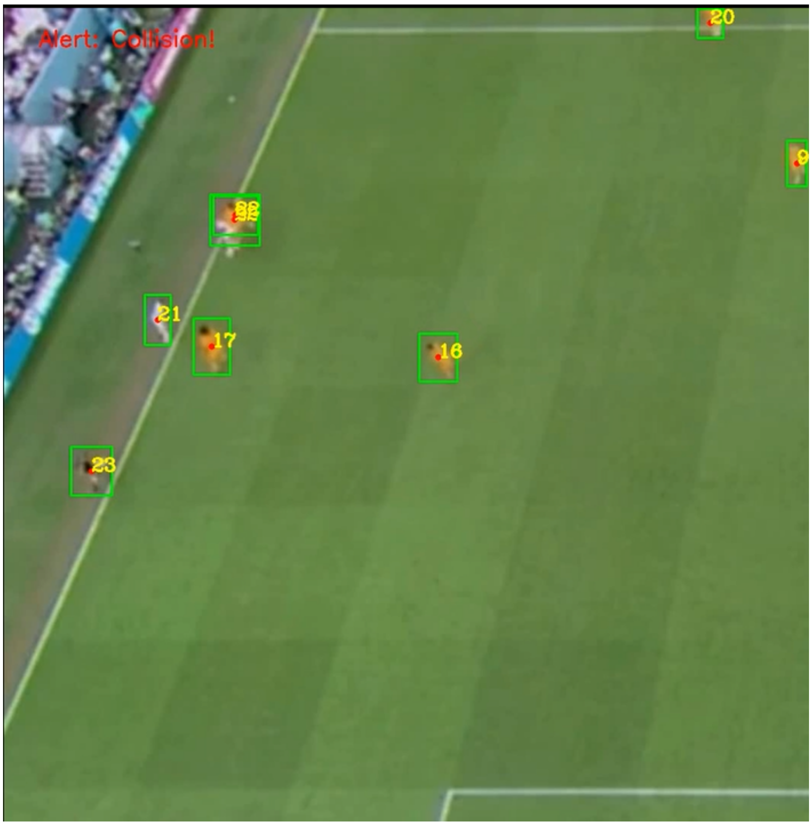


FIGURE 5.13: The output video status: collision

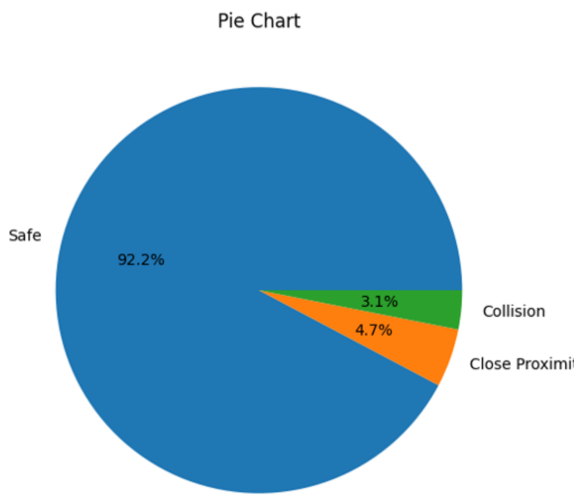


FIGURE 5.14: Pie chart

5.4 Summary

This chapter delves into the development of a web-based prediction system for football player collisions, emphasizing its architecture, functionalities, and technical aspects. It highlights the utilization of YOLOv8 for collision prediction, enabling users to input video files, configure thresholds, and receive data outputs including Excel files and pie charts illustrating collision predictions. Technical details encompass Python programming, Flask framework for back-end, and OpenCV for video processing. The system is designed for user-friendly interaction, progressing from low-fidelity wireframes to a high-fidelity prototype, ultimately achieving the goal of predicting player collisions effectively.

Chapter 6

Conclusion and Future Work

This chapter concludes the project and discusses further potential of this project.

6.1 Conclusion

Football's popularity and the associated risks of player injuries highlight the necessity for innovative solutions to enhance player safety and optimize game strategies. This project aims to address the critical issue of player collisions in football by developing an efficient and effective system for predicting such events. The system utilizes a DL algorithm and an integrated algorithm to analyze top-view videos of football matches. This system is able to predict the players' collision and extract a file that contains data about the collision. This data file can then be used for further analysis or to inform strategies for improving player safety and game management.

In this project, the various versions of YOLO were compared, determining that YOLOv8 provided superior performance for player detection tasks with the highest mAP of 96.9%. This selection was crucial as it formed the foundation for the collision prediction system. By integrating YOLOv8 and an integrated collision prediction algorithm to develop the collision prediction system, 88% accuracy was achieved in predicting player collisions, showcasing the potential DL in enhancing sports safety and performance. Roboflow and Google Colab platforms were used to implement this system, with Python programming language and its extensive libraries supporting the DL. In addition, a web-based prediction system was developed to enhance the practical application of this project.

This project underscores the significant benefits that AI can bring to sports science, particularly in football. By identifying players at higher risk of collisions, the system helps reduce injuries and optimizes team performance and strategic planning. Additionally, the financial implications for sports organizations are substantial, as reducing player injuries can lead to lower medical costs and sustained player availability, thus enhancing overall team performance and success.

6.2 Future Work

In conclusion, this project represents a meaningful advancement in integrating AI into sports, specifically football. The developed collision prediction system offers a practical solution to the players' collision problem. AI demonstrates a potential to transform this project's approach to player safety and game strategy by integrating the system with extracting more detailed data about players and matches, developing a real-time dataset, creating technology for warning players before collisions, and enhancing player identification. These advancements could enable early collision alerts, allowing for faster emergency response and medical attention, further transforming safety and performance in football.

References

[1] FIFA Publications, *The football landscape – the vision 2020-2023*, Available at: https://publications.fifa.com/en/vision-report-2021/the-football-landscape/, Accessed: 13 December 2023, 2020.

[2] S. Kuper and S. Szymanski, *Soccernomics: Why England loses, why Germany and Brazil win, and why the US, Japan, Australia, Turkey–and even Iraq–are destined to become the kings of the world's most popular sport*. Hachette UK, 2018.

[3] C. Cunningham and S Cunningham, "Injury surveillance at a national multi-sport event.," *Australian journal of science and medicine in sport*, vol. 28, no. 2, pp. 50–56, 1996.

[4] A. Junge, J. Dvorak, T. Graf-Baumann, and L. Peterson, "Football injuries during fifa tournaments and the olympic games, 1998-2001: Development and implementation of an injury-reporting system," *The American journal of sports medicine*, vol. 32, no. 1_suppl, pp. 80–89, 2004.

[5] R. D. Hawkins and C. W. Fuller, "Risk assessment in professional football: An examination of accidents and incidents in the 1994 world cup finals.," *British journal of sports medicine*, vol. 30, no. 2, pp. 165–170, 1996.

[6] J. Dvorak and A. Junge, "Football injuries and physical symptoms," *The American journal of sports medicine*, vol. 28, no. 5_suppl, pp. 3–9, 2000.

[7] C. Fuller, A. Junge, and J. Dvorak, "An assessment of football referees' decisions in incidents leading to player injuries," *The American Journal of Sports Medicine*, vol. 32, no. 1_suppl, pp. 17–22, 2004.

[8] R. D. Hawkins and C. W. Fuller, "An examination of the frequency and severity of injuries and incidents at three levels of professional football.," *British journal of sports medicine*, vol. 32, no. 4, pp. 326–332, 1998.

[9] H. Inklaar, "Soccer injuries: Ii: Aetiology and prevention," *Sports medicine*, vol. 18, pp. 81–93, 1994.

[10] M. Mostafavi, F. Nikseresht, J. E. Resch, L. Barnes, and M. Boukhechba, "Collision prediction and prevention in contact sports using rfid tags and haptic feedback," in *Advances in Usability, User Experience, Wearable and Assistive Technology: Proceedings of the AHFE 2021 Virtual Conferences on Usability and User Experience, Human Factors and Wearable Technologies, Human Factors in Virtual Environments and Game Design, and Human Factors and Assistive Technology, July 25-29, 2021, USA*, Springer, 2021, pp. 400–406.

[11] G. Tarzi, C. Tarzi, A. Saha, and M. D. Cusimano, "Predicting severity of head collision events in elite soccer using preinjury data: A machine learning approach," *Clinical journal of sport medicine*, vol. 33, no. 2, pp. 165–171, 2023.

[12] A. Ahramovich, *Ai in sports: Use cases, examples, and challenges*, `https://www.itransition.com/ai/sports`, Accessed: 2024-05-24, 2023.

[13] N. G. Erdil and M. Gül, *Spor Bilimleri Ve Egzersizde, Bütünsel Yaklaşımlar*. Ankara: Gazi Kitabevi, 2023, ISBN: 978-625-365-324-8. [Online]. Available: `www.gazikitabevi.com.tr`.

[14] C. Gray, "Top 10 applications for ai in the sports industry," *AI Magazine*, 2022, Accessed: 2024-05-24. [Online]. Available: `https://aimagazine.com/ai-applications/top-10-applications-for-ai-in-the-sports-industry`.

[15] V. R. Cossich, D. Carlgren, R. J. Holash, and L. Katz, "Technological breakthroughs in sport: Current practice and future potential of artificial intelligence, virtual reality, augmented reality, and modern data visualization in performance analysis," *Applied Sciences*, vol. 13, no. 23, p. 12 965, 2023.

[16] F. M. Clemente, F. M. L. Martins, R. S. Mendes, and A. J. Figueiredo, "A systemic overview of football game: The principles behind the game," *Journal of Human Sport and Exercise*, vol. 9, no. 2, pp. 656–667, 2014.

[17] A. Pannu, "Artificial intelligence and its application in different areas," *Artificial Intelligence*, vol. 4, no. 10, pp. 79–84, 2015.

[18] G. Musser, "The quest to build a truly intelligent machine helps us learn about our own intelligence," *Scientific American*, vol. 330, no. 4, 2024, Accessed: 2024-05-24. [Online]. Available: https://www.scientificamerican.com/article/what-the-quest-to-build-a-truly-intelligent-machine-is-teaching-us/.

[19] IBM, *Ai vs. machine learning vs. deep learning vs. neural networks*, https://www.ibm.com/think/topics/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks, Accessed: 2024-05-24, 2023.

[20] "Computer vision." (accessed June 8, 2024). (2024), [Online]. Available: https://www.ibm.com/topics/computer-vision#:~:text=Computer%20vision%20is%20a%20field,they%20see%20defects%20or%20issues.

[21] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN computer science*, vol. 2, no. 3, p. 160, 2021.

[22] G. M. Van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature communications*, vol. 11, no. 1, p. 4069, 2020.

[23] U Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical analysis of design aspects of various yolo-based deep learning models for object detection," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 126, 2023.

[24] M. E. Atik, Z. Duran, and R. ÖZGÜNLÜK, "Comparison of yolo versions for object detection from aerial images," *International Journal of Environment and Geoinformatics*, vol. 9, no. 2, pp. 87–93, 2022.

[25] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.

[26] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[27] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[28] M. Pobar and M. Ivasic-Kos, "Active player detection in handball scenes based on activity measures," *Sensors*, vol. 20, no. 5, p. 1475, 2020.

[29] A. Maglo, A. Orcesi, J. Denize, and Q. C. Pham, "Individual locating of soccer players from a single moving view," *Sensors*, vol. 23, no. 18, p. 7938, 2023.

[30] M. Şah and C. Direkoğlu, "Review and evaluation of player detection methods in field sports: Comparing conventional and deep learning based methods," *Multimedia Tools and Applications*, vol. 82, no. 9, pp. 13 141–13 165, 2023.

[31] M. Şah and C. Direkoğlu, "Evaluation of image representations for player detection in field sports using convolutional neural networks," in *13th International Conference on Theory and Application of Fuzzy Systems and Soft Computing—ICAFS-2018 13*, Springer, 2019, pp. 107–115.

[32] D. Acuna, "Towards real-time detection and tracking of basketball players using deep neural networks," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA*, 2017, pp. 4–9.

[33] M. Buric, M. Ivasic-Kos, and M. Pobar, "Player tracking in sports videos," in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, IEEE, 2019, pp. 334–340.

[34] K. Host, M. Ivasic-Kos, and M. Pobar, "Tracking handball players with the deepsort algorithm.," in *ICPRAM*, 2020, pp. 593–599.

[35] M. Ivasic-Kos, K. Host, and M. Pobar, "Application of deep learning methods for detection and tracking of players," *Deep Learning Applications*, 2021.

[36] B. T. Naik and M. F. Hashmi, "Ball and player detection & tracking in soccer videos using improved yolov3 model," 2021.

[37] B. Pawar, "Football detection with yolov3," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 05, no. 10, pp. 2520–2526, 2023. [Online]. Available: www.irjmets.com.

[38] T. Oki, R. Miyamoto, H. Yomo, and S. Hara, "Detection accuracy of soccer players in aerial images captured from several viewpoints," *Journal of Functional Morphology and Kinesiology*, vol. 4, no. 1, p. 9, 2019.

[39] P. Mavrogiannis and I. Maglogiannis, "Amateur football analytics using computer vision," *Neural Computing and Applications*, vol. 34, no. 22, pp. 19 639–19 654, 2022.

[40] C. A. Gadde and C. Jawahar, "Transductive weakly-supervised player detection using soccer broadcast videos," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 965–974.

[41] Y. Gong and G. Srivastava, "Multi-target trajectory tracking in multi-frame video images of basketball sports based on deep learning," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 10, no. 2, e9–e9, 2023.

[42] Y. Zhang, Z. Chen, and B. Wei, "A sport athlete object tracking based on deep sort and yolo v4 in case of camera movement," in *2020 IEEE 6th international conference on computer and communications (ICCC)*, IEEE, 2020, pp. 1312–1316.

[43] Z. Martin, S. Hendricks, and A. Patel, "Automated tackle injury risk assessment in contact-based sports-a rugby union example," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4594–4603.

[44] J. AbuKhait, M. Alaqtash, A. Aljaafreh, and W. Othman, "Automated detection and classification of soccer field objects using yolov7 and computer vision techniques.," *International Journal of Advanced Computer Science & Applications*, vol. 14, no. 11, 2023.

[45] K. Duay, "Players detection and tracking for esports videos analysis with dataset generation from minimap," M.S. thesis, Itä-Suomen yliopisto, 2023.

[46] S. Purtak, J. Lazarek, P. Gora, and Gsiorowski, "Automated system for indoor sport video production," in *Proceedings of the 11th International Conference on Sport Sciences Research and Technology Support (icSPORTS 2023)*, SCITEPRESS – Science and Technology Publications, Lda., 2023, pp. 23–33, ISBN: 978-989-758-673-6. DOI: 10.5220/0012159300003587.

[47] A. Scott, I. Uchida, N. Ding, *et al.*, "Teamtrack: A dataset for multi-sport multi-object tracking in full-pitch videos," *arXiv preprint arXiv:2404.13868*, 2024.

[48] K. Moutselos and I. Maglogiannis, “Setting a baseline for long-shot real-time player and ball detection in soccer videos,” in *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*, IEEE, 2023, pp. 1–7.

[49] V Shankara, S. Ahmed, M Sneha, and J Guruprakash, “Object detection and tracking for football data analytics,” in *Proceedings of the 1st International Conference on Artificial Intelligence, Communication, IoT, Data Engineering and Security, IACIDS 2023, 23-25 November 2023, Lavasa, Pune, India*, 2024.

[50] B. Zheng, “Soccer player video target tracking based on deep learning,” *Mobile Information Systems*, vol. 2022, pp. 1–6, 2022.

[51] T. Wang, T. Li, *et al.*, “Deep learning-based football player detection in videos,” *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[52] A. Cioppa, A. Deliege, N. U. Huda, R. Gade, M. Van Droogenbroeck, and T. B. Moeslund, “Multimodal and multiview distillation for real-time player detection on a football field,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 880–881.

[53] R. Martín and J. M. Martínez, “A semi-supervised system for players detection and tracking in multi-camera soccer videos,” *Multimedia tools and applications*, vol. 73, pp. 1617–1642, 2014.

[54] Roboflow, *Roboflow: Computer vision tools for developers and enterprises*, `https://roboflow.com/`, Accessed: 2024-05-26, 2024.

[55] Google, *Colaboratory frequently asked questions*, `https://research.google.com/colaboratory/faq.html`, Accessed: 2024-05-26, 2024.

[56] P. S. Foundation, *About python*, `https://www.python.org/about/`, Accessed: 2024-05-26, 2024.

[57] A. Martelli, A. M. Ravenscroft, S. Holden, and P. McGuire, *Python in a Nutshell.* " O’Reilly Media, Inc.", 2023.

[58] J Redmon and A Farhadi, *Yolo9000: Better, faster, stronger. in proceedings of the ieee conference on computer vision and pattern recognition (pp. 7263-7271)*, 2017.

[59] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.

[60] M. Hussain, "Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo," *IEEE Access*, vol. 12, pp. 42 816–42 833, 2024.

[61] A. Vijayakumar and S. Vairavasundaram, "Yolo-based object detection models: A review and its applications," *Multimedia Tools and Applications*, pp. 1–40, 2024.

[62] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," *arXiv preprint arXiv:2402.13616*, 2024.

[63] G. Boesch. "Yolov7: A powerful object detection algorithm (2024 guide)." (Jun. 1, 2024), [Online]. Available: https://viso.ai/deep-learning/yolov7-guide/ (visited on 06/01/2024).

[64] R. Kitchin, "Thinking critically about and researching algorithms," in *The Social Power of Algorithms*, Routledge, 2019, pp. 14–29.

[65] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.

[66] J. Han, M. Kamber, and J. Pei, "Data mining concepts and techniques third edition," *University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University*, 2012.

[67] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," *arXiv preprint arXiv:2008.05756*, 2020.

[68] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*, IEEE, 2020, pp. 237–242.

[69] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," *Digital Signal Processing*, vol. 132, p. 103 812, 2023.

[70] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[71] Z. Sun, J. Chen, L. Chao, W. Ruan, and M. Mukherjee, "A survey of multiple pedestrian tracking based on tracking-by-detection framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1819–1833, 2020.

[72] X. Hou, Y. Wang, and L.-P. Chau, "Vehicle tracking using deep sort with low confidence track filtering," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2019, pp. 1–6.

[73] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103 448, 2021.

[74] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

[75] G. Dwyer, S. Aggarwal, and J. Stouffer, *Flask: building python web services*. Packt Publishing, 2017.