Development and Evaluation of a Real-Time Tracking System for Handball Penalty Shots

Jacob Attard

Supervisor: Christopher Farrugia

September - 2025

A dissertation submitted to the Institute of Information and Communication Technology in partial fulfilment of the requirements for the degree of BSc (Hons) Software Development

Authorship Statement

This dissertation is based on the results of research carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

The research was carried out under the supervision of Mr. Christopher Farrugia

....

Date   Signature

Copyright Statement

. . . .
Date

. . . .
Signature

Acknowledgements

I would like to thank my supervisor Mr. Christopher Farrugia, for his feedback and support, and I would like to also thank my family and friends for the constant support given during the course of this dissertation.

Abstract

TO DO

This section should clearly state what the study is about, summarizing how it was carried out and what the results were. References are not to be included in the abstract. It should present only the essentials of the work in general.

Keywords: YOLO Model, Ball Tracking, Computer Vision

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CNN	Convolutional Neural Network
YOLO	You Only Look Once

Chapter 1: Introduction

1.1 Significance of the Problem

Handball lacks dedicated technical solutions that address its gameplay dynamics, in contrast to other well-known sports like basketball and football, where sophisticated monitoring and performance analysis systems are well-established. This prevents athletes and coaches from using analytics to inform their decisions, which could improve both performance and training methods.

The analysis of ball trajectories is made possible by this study's application of computer vision to a field with limited technological integration. By providing information on technique and placement, this may change the way penalty shots are seen and managed.

The findings of this study will give coaches and athletes the resources they need to improve their tactics, increase training effectiveness, and boost overall performance. Furthermore, by filling in the gaps in sports-specific performance analytics, this work can provide motivation for future research into other sports that do not yet have comparable technology developments.

1.2 Research Motivation

The motivation for this research comes from a strong interest in computer vision from the leading researcher, combined with experience in handball spanning over eight years. Computer vision has the ability to completely change how coaches and athletes perceive and enhance performance, therefore the idea of using it into sports analysis is appealing. Beyond its academic significance, this research aims to provide more research to the handball community, offering a tool that enables data-driven training and strategic advancements.

1.3 Research Hypothesis and Questions

The hypothesis for this study is that a real-time tracking system can accurately track the trajectory of a handball during penalty shots and provide performance data that benefits both players and coaches. By integrating object detection, tracking, and pose estimation, the system is expected to deliver objective feedback on penalty shot execution and contribute to performance improvement.

To evaluate this hypothesis, the following research questions were formulated:

1. **How accurately can a program analyse handball penalty shots?** This question focuses on the system's overall effectiveness in tracking the ball and player movements, classifying shots, and providing reliable performance feedback.

2. **Which ball detection model is best for this scenario?** Given that accurate ball detection is central to the analysis, this question compares different state-of-the-art models, including YOLOv8, YOLOv11, and Faster R-CNN, to identify the most suitable model for handball penalty shot detection.

3. **What are the most significant statistics and metrics to consider when analysing handball penalty shots?** Beyond detection, this question explores which performance indicators—such as shot placement and player technique are most useful for coaches and athletes in improving training and strategy.

1.4 Document Structure

This dissertation is organised into five main chapters, followed by references and appendices.

Chapter 1: Introduction introduces the research problem, outlines the motivation behind the study, and presents the research hypothesis and questions.

Chapter 2: Literature Review provides an overview of relevant work in computer vision, real-time object detection, and sports analytics. It discusses algorithms such as YOLO, SSD, and Faster R-CNN, as well as applications of computer vision in sports, including ball tracking, player pose estimation, and real-time tracking systems.

Chapter 3: Research Methodology details the dataset preparation, technologies used, and implementation choices. It explains how object detection models and MediaPipe were applied, describes the model training process, and outlines the evaluation metrics used to assess system performance.

Chapter 4: Analysis of Results and Discussion presents and evaluates the results obtained from the experiments. It compares the performance of YOLOv8, YOLOv11, and Faster R-CNN models, highlights cross-model insights, and discusses the contribution of MediaPipe to pose analysis.

Chapter 5: Conclusions and Recommendations revisits the research questions, summarises the findings, and provides recommendations for future research directions while acknowledging limitations of the study.

Chapter 2: Literature Review

2.1 Overview

Handball is a fast-paced team sport involving two teams of seven. It involves players passing a ball using their hands with the aim of throwing it into the goal of the opposing team. Researchers have been looking into more and more ways into using data-driven strategies and certain techniques which involve different technologies to improve player performance and tactical decision-making. Acording to the EHF (European Handball Federation) glossary of handball terms and expressions the definition for a penalty shot is a free-throw taken from the 7m line with only the goalkeeper of the defending team between the penalty taker and the goal [4]. A penalty shot is awarded when a clear scoring opportunity is illegally destroyed by the defending team. The IHF (International Handball Federation) Rulebook specifies various scenarios under Rule 14 that warrant a penalty, including fouls or violations that impede a legitimate scoring chance [5].

2.2 Real-Time Object Detection

Real-time object detection is a key area of computer vision research that focuses on identifying and localizing objects in images or video streams [6]. Convolutional Neural Networks (CNNs), such as Mask R-CNN, SSD, and R-FCN, have become the standard techniques for these tasks, offering varying trade-offs between speed and accuracy. YOLO (You Only Look Once), a single-stage detection model, stands out for its ability to predict class probabilities and bounding boxes simultaneously, making it well-suited for real-time applications.

2.2.1 The different algorithms

You Only Look Once (YOLO)

YOLO's architecture, starting with 24 convolutional layers in its original form and evolving through versions like YOLOv2, has been adapted for efficiency, including the use of anchor boxes and max-pooling layers [7]. YOLO's capacity to manage these difficulties is demonstrated by its adaptation for identifying people and sports balls in handball photos. It also serves as a basis for action detection systems, providing real-time insights into game settings [8]. Significant development has been made in real-time object detection, which makes it a great tool in certain situations such as sports analytics. By treating object detection as a single regression problem, YOLO (You Only Look Once) transformed the field and allowed for real-time processing at up to 45 frames per second, and in optimised settings, up to 150 frames per second. YOLO is very good for time-sensitive applications like sports tracking, even though it compromises some localisation accuracy for speed [9].

YOLOv8 and YOLO11 are two different versions in the YOLO object detection series, each made for specific performance and efficiency needs. YOLOv8, introduced by Ultralytics in January 2023, emphasizes versatility and ease of use, supporting tasks such as object detection, segmentation and classification. With models ranging from YOLOv8n to YOLOv8x, it balances speed and accuracy, making it well-suited for various real-time applications. YOLO11, released in September 2024, is a more recent version of YOLO with a focus on further optimizing inference speed while maintaining competitive accuracy. Benchmark results show that YOLO11 offers marginal accuracy improvements over YOLOv8 but significantly enhances inference speed, particularly in CPU-bound environments. While YOLOv8 remains a strong general-purpose model with extensive community support, YOLO11 is ideal for real-time applications requiring the fastest possible detection performance.

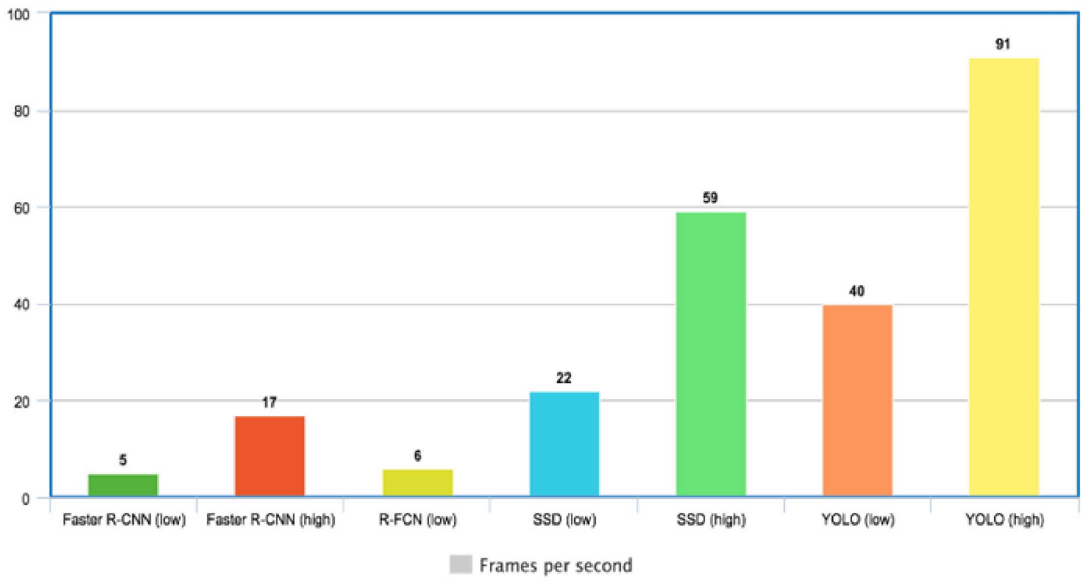


Figure 2.1: *Comparison of frames per second implementing the Faster R-Cnn, R-FCN, SSD, YOLO models using input images with different resolutions [1]*

Researchers have built upon YOLO by making improvements such as, Fast YOLO, inspired by GoogLeNet it uses smaller convolutional procedures. These changes will still manage to perform accurately whilst lowering computational demands. Fast YOLO works especially well in dynamic settings like sports since preprocessing methods like frame differentiation and Gaussian background subtraction are also used to reduce noise in video streams. According to experiments, this method produces competitive recall and precision rates, which makes it appropriate for real-time applications involving the detection of fast moving objects [10]. RTMDet built upon these foundations by introducing improvements such as dynamic label assignment and large-kernel depth-wise convolutions, which led to state-of-the-art results with over 1000 frames per second and good precision, indicating its potential for ultra-low latency applications in sports. [11].

Convolutional neural network (CNN)

Convolutional Neural Network (CNN)s are Artificial Neural Networks that include some convolutional layers. A Convolutional Neural Network typically consists of a series of convolutional layers and pooling layers. A convolutional layer consists of several kernels (also called filters) that are used to capture spatial information within the data. The kernel slides over the input with a predefined stride, and at each step it performs a convolution with the kernel and the patch. Each kernel produces a feature map which is then passed to the next layer.

In a study comparing YOLO and Mask R-CNN for sports ball recognition, YOLO demonstrated greater speed and versatility when trained on a custom dataset, improving its F1 score from 6% to 34%. Nonetheless, Mask R-CNN offered better segmentation skills and increased precision, especially in occlusion situations [12]. Mask R-CNN's accuracy can be useful for in-depth post-game analysis, but YOLO's faster speed makes it appropriate for real-time situations. A lightweight CNN-based approach was also designed for small humanoid robots. It used a region proposal algorithm that narrows down candidate regions before classification [13].

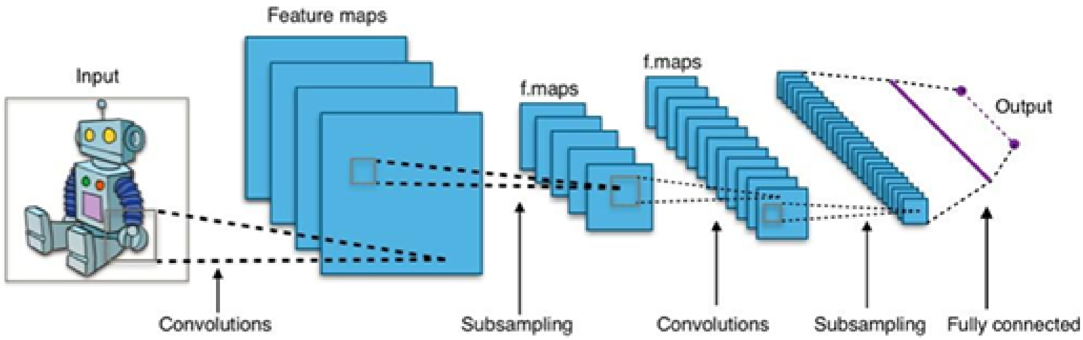


Figure 2.2: *convolutional neural networks [2]*

Single Shot Detector (SSD)

Another popular algorithm for object detection is the Single Shot MultiBox Detector (SSD). Unlike YOLO, which employs a single-stage detection method, SSD predicts item categories and locations while producing a fixed set of bounding boxes at various scales and aspect ratios. One study demonstrated an improved SSD model that demonstrated effectiveness in identifying tennis balls in real-world situations by incorporating AlexNet for initial object classification, reducing computing overhead prior to using SSD for localisation [14]. Another study that compared SSD, YOLO, and Faster R-CNN discovered that SSD provided a fair balance between detection speed and accuracy, which made it a strong choice for sports ball tracking in real time [15]. Researchers have enhanced activation functions, optimised loss functions, and feature pyramid networks to boost SSD performance, which has resulted in higher recall and precision rates [16]. Additionally, lightweight variants such as MobileNet-SSD have been developed for embedded applications, also being effective in real-time sports tracking and robotics [17]. These advancements highlight SSD's adaptability across various scenarios, particularly in tracking fast-moving objects such as sports balls, where rapid and accurate detection is essential.

2.2.2 The Challenges in Real-Time Object Detection

Small object sizes, object appearance variability, occlusion, and different player movement are all particular problems for object detection in sports. Small object detection, in which models struggle to identify small, fast-moving objects such as handballs, particularly when the ball appears in different lighting conditions or scales. Occlusion occurs when the ball is partially obscured behind players or becomes blurred from motion [18]. For models such as YOLO and SSD to function well in real time, a substantial amount of resources are needed. Another crucial issue is the trade-off between accuracy and speed, with quicker models frequently compromising accuracy [19]. To be able to solve certain challenges, it is best to have a wide range of images in different settings such as lighting and angles in the dataset [2]. The model will perform best when it is subjected to the conditions it will face during deployment thanks to the usage of authentic photos from the target location. Furthermore, taking pictures in a variety of lighting conditions and from a range of perspectives improves the model's adaptability and allows for dependable detection in a range of situations [20].

2.3 Sports Analytics and Performance Enhancement

Team performance is being improved with the use of sports analytics into strategy and training. With an emphasis on enhancing particular areas like offensive efficiency and defensive coordination, the study highlights how game-related information can help coaches with making the best decisions for the team [21]. It also demonstrates how performing analytics on previous games may improve training sessions by connecting the statistical data, giving athletes a better idea on what needs to be worked on during training. This is consistent with the growing trend of using data-driven insights to improve sports strategy for both individuals and teams.

In sports analytics, data visualisation is important because it helps analysts and coaches with complex data sets. Data from competitive sports can be divided into two categories: statistical data, which records game events and performance measurements, and spatiotemporal data, which monitors player and ball movement. An effective way to understand these types of data is to visualise them using different techniques. These can be heat maps, trajectory analysis, and network graphs. Analysts can evaluate player effectiveness, team configurations, and game plans by using these tools, which in the end will improve performance and decision-making. [22]

2.4 Statistics in Sports

In sports research, statistical analysis must be used correctly, particularly to ensure the validity and dependability of results. Statistical power analysis, which determines the probability of effectively rejecting a false null hypothesis, is an important but occasionally overlooked element [23]. A false null hypothesis is a null hypothesis which is incorrect or untrue in a population. A null hypothesis is frequently interpreted in hypothesis testing as a claim that there is no influence, distinction, or connection. Rejecting the null hypothesis suggests that there is an effect, difference, or association in the population that might be missed because of sample variability or issues with analysis [24]. The value of game-related data in affecting match results is assessed throughout the study [21]. These statistics include shot accuracy, possession time, turnovers, and defensive measures. The results show that while defensive statistics such as interceptions and goalkeeper stops have an influence on team performance, offensive statistics, shot accuracy and goals scored have a higher correlation with success. These results highlight the benefit of statistical analysis as a method for both assessing historical results and planning for upcoming contests.

In sports, where sample sizes tend to be smaller due to logistical restrictions, doing a power analysis during the planning phase can assist researchers in designing studies that are both robust and resource efficient. [23]

Match results are impacted by key performance factors such as positional play, interceptions, counterattacks, and shooting efficiency. Trends such as the growing emphasis on quick attacking plays and counterattacks have been identified through the analysis of competitive activity over several championships, highlighting the importance of agility and fast decision-making in competitive handball [25].

2.5 Applications of Computer Vision in Sports

2.5.1 Ball Detection and Tracking

Ball detection and tracking is the process of determining the ball's location throughout a video. This can be used to collect information such as shot speed and shots on target. The use of object identification models such as YOLOv8 has improved the precision and efficiency of ball tracking. Researchers utilised the YOLOv8m model for hockey, and the model was able to detect small, fast-moving balls with a precision of 0.752 and an F1-score of 0.686 [26]. This was accomplished by training the model on a sport-specific dataset, allowing it to adapt to obstacles such as occlusions and changing lighting conditions.

Due to the fast ball movement, different player formations, and multiple occlusions, handball makes it challenging to detect and track the ball. By separating the ball and players from cluttered backdrops, better segmentation methods were used in this work to increase detection accuracy [27]. This allowed for a better analysis of player locations and ball control. By combining powerful object identification capabilities with real-time processing, the YOLOv8 approach has shown success in addressing these issues.

2.5.2 Player Pose Detection

Pose estimation is a key area in computer vision that involves identifying and tracking the position of human body joints in an image or video stream. This allows for real-time analysis of posture and movement, offering valuable insights in fields like healthcare, robotics, and most notably, sports.

MediaPipe is an open-source framework developed by Google, specifically designed for building cross-platform applications that process perceptual data in real time, such as video, audio, and sensor input. Its architecture revolves around the concept of pipelines, where developers connect modular components called calculators to form a directed graph capable of handling complex tasks like pose estimation, face detection, object tracking, and more [28].

Several works have demonstrated how pose data can be used not only for recognition but also for predictive modeling, feedback generation, and event detection. For instance, pose forecasting has been applied to anticipate motion in handball [29], and MediaPipe has been used to guide exercise form in real-time [30] [31]. These applications illustrate how pose detection extends beyond visualization to actively support training, evaluation, and injury reduction in sports.

2.6 Real-Time Tracking Systems for Sports

The accuracy and effectiveness of real-time systems have been greatly increased by combining deep learning with conventional object identification methods. With an emphasis on the value of Faster R-CNN and YOLO in addressing practical problems, this study examines the development of detection models from manually constructed feature-based techniques to contemporary CNN-based architectures [32]. In particular, these models are better able to detect small objects and adjust to changing conditions thanks to the utilisation of advanced frameworks.

A multi-sensor radio-frequency (RF) system designed for real-time player and ball tracking and detection in team sports is described in this paper. This system combines Doppler radar and Radio Frequency Identification (RFID) technology to increase tracking accuracy and efficiency, in contrast to conventional video-based systems that suffer from accuracy and latency problems. With its multistatic configuration, the Doppler radar subsystem provides 3D Cartesian tracking of the ball, allowing it to be distinguished from players by velocity differences. In the meantime, the RFID subsystem uses resonance antenna tags attached to participants to use the Time Difference of Arrival (TDOA) method to estimate their positions very accurately. A complete game image that supports real-time analysis and game statistics can be created by integrating multiple subsystems and using data fusion techniques. Future advancements in this system are intended to enhance its capabilities even more, potentially enabling applications in live game evaluations and real-time video transmissions, thereby establishing it as a competitive alternative or supplement to visual tracking systems. [33]

The Handball.ai app acts as an example of how real-time tracking technologies may change sports, particularly handball. Through the use of artificial intelligence and observational techniques, it tackles the problems of accurately recognising and evaluating tactical behaviours during games. The system's ability for in-depth game analysis is demonstrated by its capacity to automatically track variables including player placements, tactical formations, and possession dynamics. The combination of both automatic and human variable tracking sets Handball.ai apart from other methods, enabling quicker and more thorough data collection. [34]

A cheap ball tracking system built on a Raspberry Pi with HSV colour space and motion tracking revealed the potential for real-time applications in limited environments. While limited by sensitivity to lighting and colour fluctuations, it demonstrates how simple detection methods can contribute to user-friendly sports tracking solutions. [35]

2.7 Key Takeaways

Advanced tracking technologies and data-driven methods could improve player performance and tactical decision-making in the fast-paced sport of handball. Key takeaways from the reviewed literature highlight the role of object detection models such as YOLO, SSD, and Mask R-CNN in real-time sports applications. Due to its speed, YOLO is well-suited for real-time processing, however Mask R-CNN provides superior segmentation and accuracy in situations with a lot of occlusion. SSD offers a balance between accuracy and detecting speed. The quality of the dataset is important to increase the precision of ball identification and tracking algorithms. Small object sizes, occlusion, and changing lighting conditions continue to be problems despite progress, especially in fast-paced sports including handball. Deep learning enhances the accuracy and efficiency of tracking systems, especially when combined with conventional object identification techniques. By integrating deep learning models like YOLO and SSD, tracking systems can better recognize and adapt to these challenges.

Chapter 3: Research Methodology

3.1 Dataset Preparation

The dataset was constructed by extracting frames from recorded handball penalty shot videos using Roboflow. Videos were captured during training sessions, and frames were sampled at regular intervals (e.g., 1 frame every 5 frames) to avoid redundancy while still capturing the motion of the ball across different stages of the shot. This ensured a balance between dataset size and diversity. Extracting images from video allowed for a variety of ball positions and player movements to be represented in the dataset, which improved the model's ability to generalize to unseen penalty shots.

Including the augmented images the dataset was divided into 2058 training images (85%), 185 validation images (8%), and 78 test images (7%). The training set was intentionally kept the largest to allow the models to learn robust feature representations. The validation set was used for hyperparameter tuning and to monitor overfitting during training. Finally, the test set, which remained completely unseen during training, was reserved for evaluating the final performance. This split was chosen to maintain a balance between sufficient training data and reliable evaluation, while also considering the relatively limited dataset size compared to large-scale benchmarks like COCO.

Accurate annotations were critical for this project, as the objective was to detect and track the ball during handball penalty scenarios with a high degree

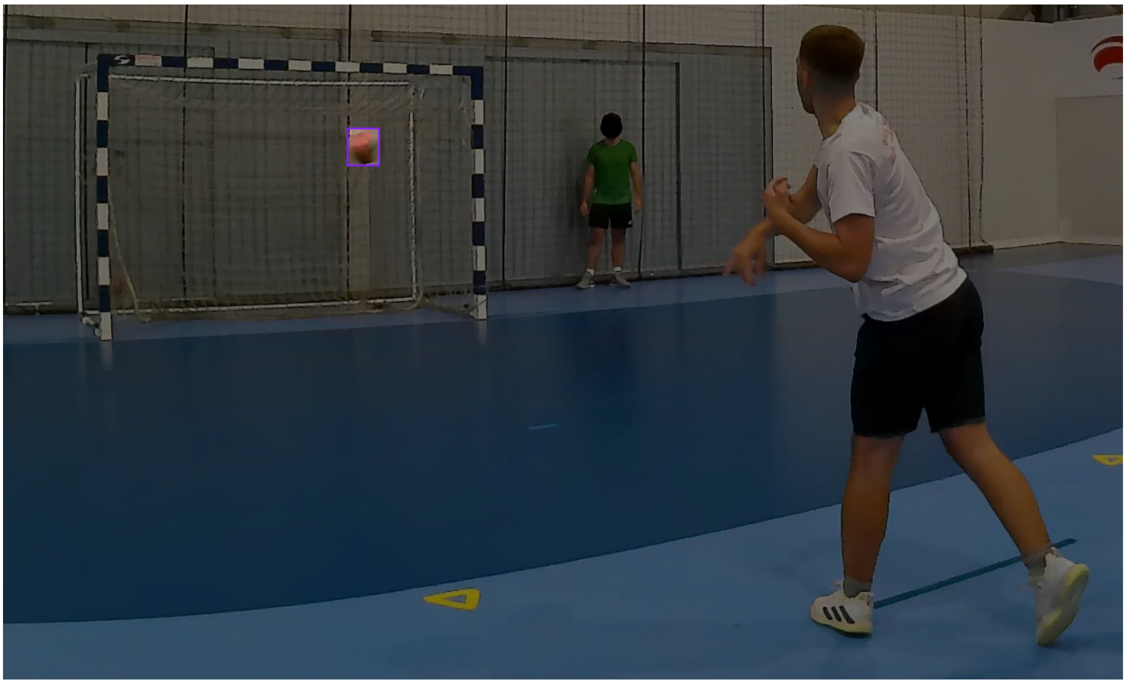


Figure 3.1: *Image used in Custom Dataset*

of precision. To achieve this, Roboflow was used for dataset preparation and annotation. Roboflow provides a web-based interface that simplifies the process of drawing bounding boxes, applying augmentations, and exporting datasets in formats compatible with frameworks such as YOLO and other models.

Each image in the dataset was manually annotated with bounding boxes around both the handball and the goalpost. The handball was annotated regardless of its position or state, including cases where it was partially occluded by a player's hand, blurred due to motion, or very small within the frame. The bounding boxes remained tight around the object to minimise background noise. The point of annotating the goalpost was to enable classification of shot outcomes.

To maintain consistency, a clear annotation schema was defined, with two classes: ball and goalpost. Every image contained at least one bounding box for the goalpost, ensuring that detection models could reliably learn its structure, while ball annotations were carefully checked to confirm they appeared in the correct position even when small or distorted by motion blur.

More images were added after the first training phase. The next set of images were, balls of varying colors to simulate diverse gameplay conditions, images captured from two different angles, left of the player and right of the player [36]. Adding these images to the dataset aims to improve the model's performance across a wider range of scenarios.

Camera placement was chosen to provide the best visibility of both the player's throwing motion and the goal. Cameras were positioned slightly behind the player, on either the left or right side. This angle offered a clear perspective of the player's arm and hand movement during the throw, which was for pose estimation, while also keeping the ball and goalpost within the frame for trajectory tracking. The rear-side perspective minimized occlusion from the player's body, ensured the hand and elbow remained visible, and maintained an unobstructed view of the goal. Although frontal or overhead views could provide other data, the chosen positioning offered the most practical balance between capturing player movement and tracking the ball toward the goal.

Augmentations were also applied to the images creating different copies of the same image but with different features. Such augmentations were a horizontal and vertical flip, rotation between -15 degrees and 15 degrees, brightness between -15% and 15%, a blur up to 2.8px and noise of up to 0.41% of pixels. By doing these augmentations the dataset increased by 80%, from 1292 images to 2321 images.

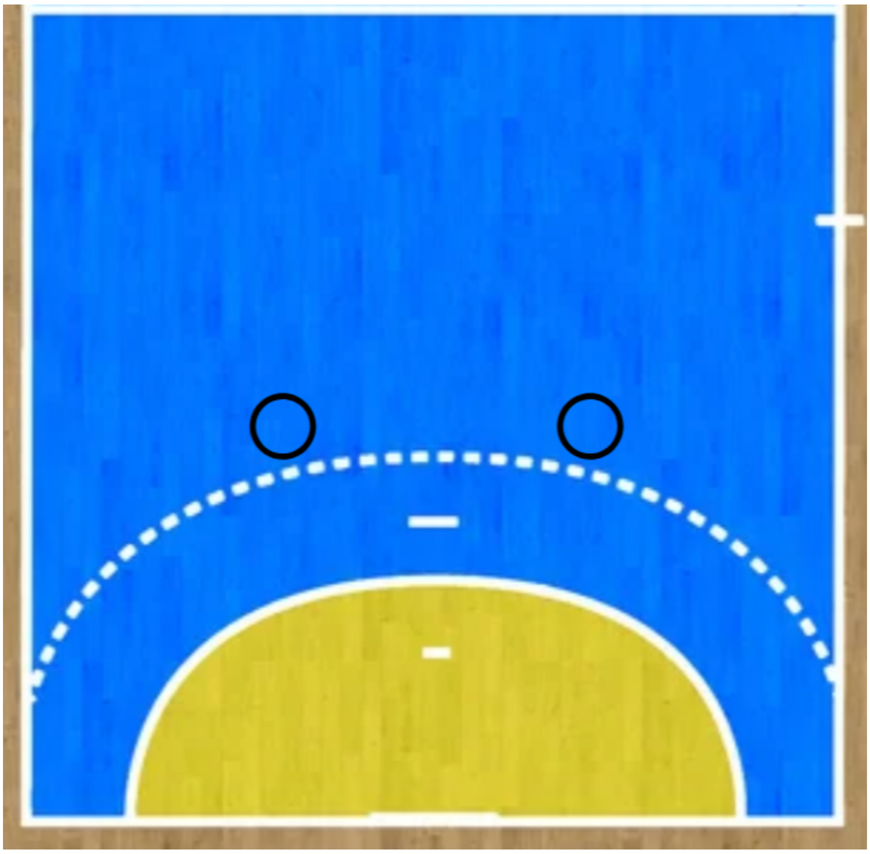


Figure 3.2: *Camera positions for images*

The augmentations simulate real-world variations such as different lighting conditions and slight changes in camera positioning. By augmenting the dataset, the models became more robust to variations in match scenarios and less dependent on the exact conditions present in the training data.

All individuals appearing in the dataset provided informed consent prior to image collection and annotation. Consent forms were signed to ensure that participants were aware of the purpose of the study and how the data would be used. This step was essential to uphold ethical standards and ensure that the dataset was created and used responsibly.

3.2 Technologies Used

In this study, multiple variants of both YOLOv8 and YOLO11 were tested to explore the trade-off between model size, accuracy, and efficiency. Specifically, the n (nano), s (small), and m (medium) configurations of each version were trained and evaluated. These progressively larger models represent increasing network depth and width, offering higher accuracy at the cost of computational resources and longer training time. By including both smaller models (YOLOv8n, YOLO11n) and mid-range models (YOLOv8m, YOLO11m), the evaluation captures how model complexity impacts real-time feasibility and precision in detecting a fast-moving handball.

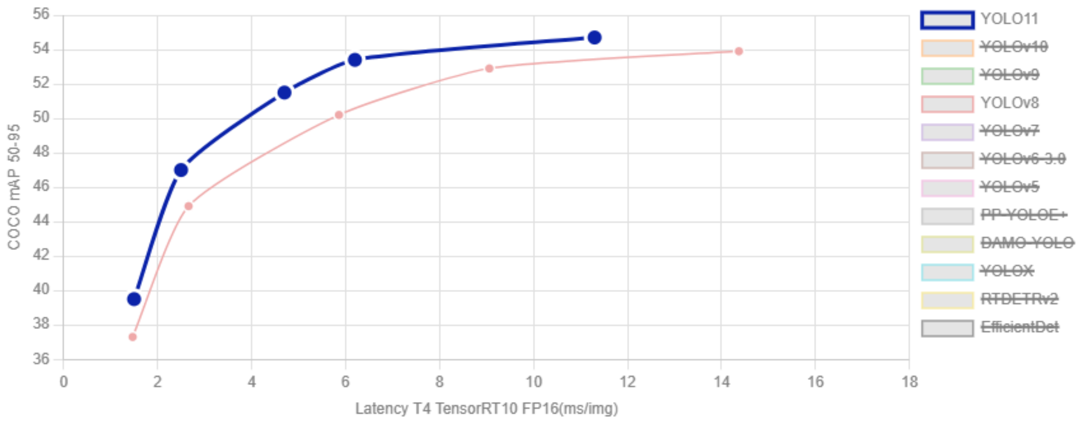


Figure 3.3: *Difference in latency between YOLO11 and YOLOv8 [3]*

The graph in Figure 3.3 compares the performance of YOLO11 and YOLOv8 in terms of accuracy (COCO mAP 50–95) and inference speed (latency per image). YOLO11, shown by the blue line, consistently achieves higher accuracy than YOLOv8 at similar or lower inference times. This means YOLO11 is not only more precise but also faster, making it a better choice for real-time applications where speed and accuracy are crucial.

Faster R-CNN is a two-stage detector, meaning it first proposes regions in the image that might contain objects, and then analyzes each of those regions in more depth to classify the object and make the bounding box more accurate. This added step generally makes Faster R-CNN more accurate than single-stage detectors like YOLO but at the cost of speed. Its architecture is built around two key stages: a Region Proposal Network (RPN), which suggests areas where objects might be located, and a second stage that classifies these areas and refines the predictions. This layered approach helps the model detect small objects, handle cluttered scenes, and maintain high accuracy even when parts of the image are obscured, a common challenge in handball footage where players and objects often overlap.

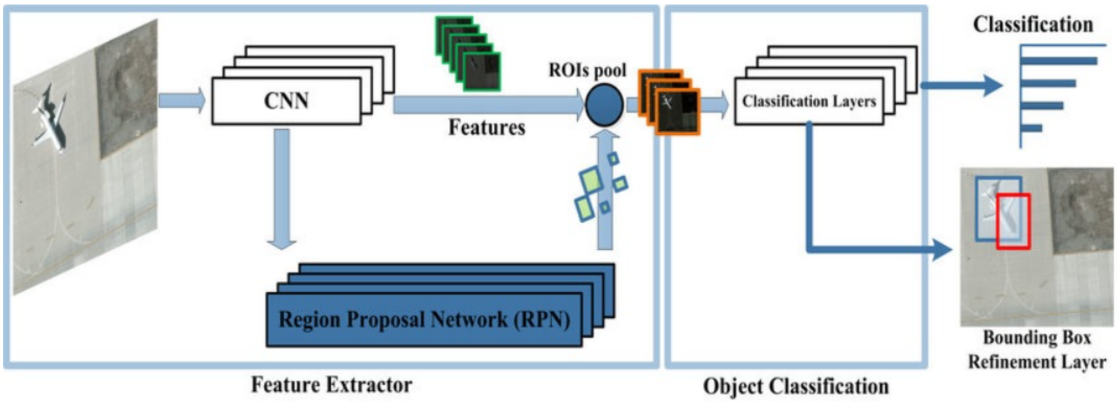


Figure 3.4: *Overview of Faster R-CNN Architecture*

Single Shot MultiBox Detector (SSD) was also included in the evaluation as an additional single-stage detector. Unlike Faster R-CNN, SSD skips the region proposal step and instead directly predicts bounding boxes and class probabilities from feature maps at multiple scales. This makes SSD faster than two-stage models, though generally less accurate at detecting small objects. In the context of handball, SSD provides a useful baseline for assessing how simpler architectures compare to more recent YOLO versions and the region-based Faster R-CNN.

By including a diverse set of models — YOLOv8 (n, s, m), YOLO11 (n, s, m), Faster R-CNN, and SSD — this study was able to comprehensively compare state-of-the-art single-stage and two-stage detectors. This variety provides valuable insights into the trade-offs between detection accuracy, recall, robustness at different IoU thresholds, and computational efficiency, all of which are critical for building a real-time handball penalty shot tracking system.

3.3 Pose Estimation using MediaPipe

In addition to tracking the handball and goalpost, the movement of the player is being tracked during a penalty shot, using MediaPipe, an open-source tool developed by Google that tracks human body landmarks in real time.

The idea behind this addition is to check whether the player is using proper throwing form. In handball, one key indicator of good technique is that the elbow should be raised above the shoulder during the throw. This helps generate more power and control over the shot, also reducing the risk of injury to the shoulder.

So by using MediaPipe the system keeps an eye on the positions of the elbow and shoulder throughout the shot. It then checks, frame by frame, whether the elbow is actually above the shoulder. This simple yes-or-no check gives quick insight into whether the player is following good form.

3.4 Model Training

The models were trained on a local machine equipped with an NVIDIA GeForce RTX 4060 GPU, providing the necessary computational power for the object detection task. Training for the YOLO models was conducted using the Ultralytics framework, which is built on PyTorch. The training process consisted of 200 epochs with a patience of 10 for each model. Various augmentations were applied to the dataset using Roboflow prior to training, further enhancing the model's ability to generalize across different shot scenarios. These details of the training environment ensured the model's performance was optimized for real-time object detection tasks.

In addition to the YOLO models, a Faster R-CNN model and an SSD model were trained using the same dataset, for 200 epochs with a patience of 10. The implementation for the Faster R-CNN model was based on the torchvision library's Faster R-CNN ResNet-50, made to recognize two object classes: handball and post. The input images and their corresponding annotations in Pascal VOC format were preprocessed and split into training, validation, and test sets consistent with those used for YOLO.

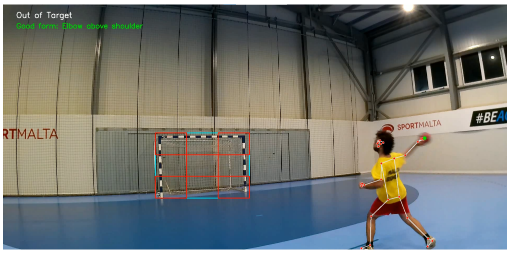
Chapter 4: Analysis of Results and Discussion

Table 4.1 compares the seven trained models — YOLOv8 (n, s, m), YOLOv11 (n, s, m), and Faster R-CNN — their results were compared across precision, recall, mAP@0.5, and mAP@0.5:0.95. These metrics provide a holistic view of detection quality, ranging from the ability to correctly identify true positives (recall), to the reliability of predictions (precision), and overall detection accuracy at different IoU thresholds (mAP). For consistency, results are reported from the final training epoch with early stopping applied, ensuring each model is represented at convergence.

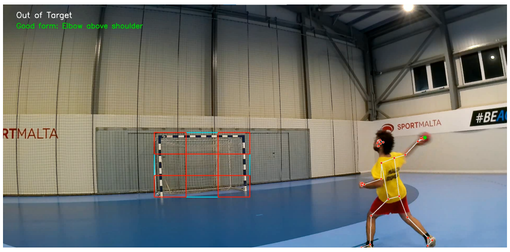
Model	Epoch	Precision	Recall	F1	mAP@0.5	mAP@0.5:0.95	Time (Min)
YOLOv8n	60	0.9263	0.9256	0.9260	0.9620	0.7626	19.73
YOLOv8s	77	0.9520	0.9615	0.9567	0.9775	0.8151	58.25
YOLOv8m	95	0.9500	0.9676	0.9587	0.9793	0.8135	772.13
YOLOv11n	94	0.8944	0.9463	0.9196	0.9552	0.7762	33.19
YOLOv11s	71	0.9397	0.9686	0.9539	0.9799	0.8109	53.62
YOLOv11m	71	0.9543	0.9529	0.9536	0.9797	0.7927	755.95
Faster R-CNN	19	0.8866	0.9780	0.9301	0.8866	0.7068	206.97
SSD	25	0.8300	0.6600	0.7350	0.8520	0.5680	48.54

Table 4.1: *Comparison of model performance across key metrics*

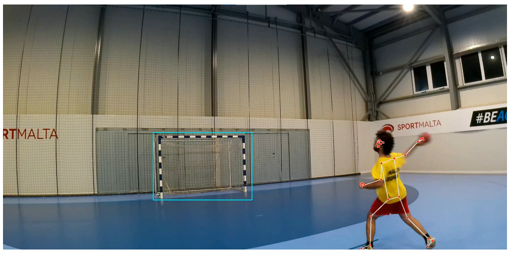
(a) *YOLOv8n*



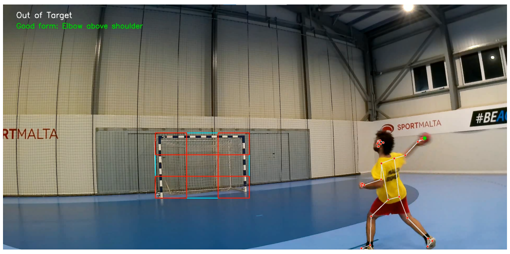
(b) *YOLOv8s*



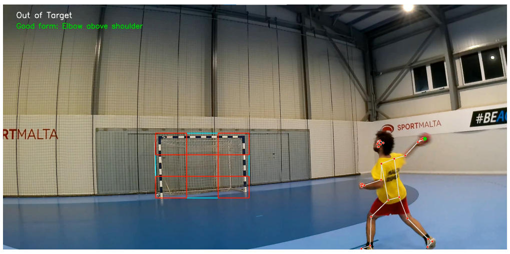
(c) *YOLOv8m*



(d) *YOLOv11n*

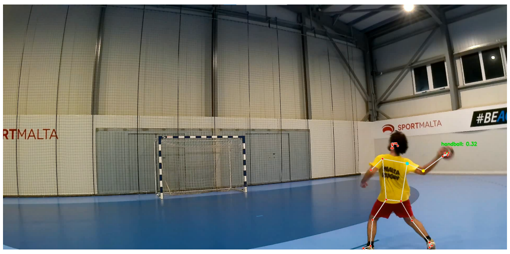


(e) *YOLOv11s*



(f) *YOLOv11m*

(g) *Faster R-CNN*



(h) *SSD*

Figure 4.1: *Qualitative comparison of detections across all eight trained models on the same penalty shot frame.*

4.1 Evaluation Metrics

To evaluate and compare the performance of the models, these metrics were used: precision, recall, F1 score, and mean average precision at different thresholds (mAP@0.5 and mAP@0.5:0.95).

For the YOLO models (YOLOv8 and YOLOv11), the Ultralytics framework automatically produced precision, recall, F1, and mAP metrics at the end of training. For Faster R-CNN and SSD, metrics were computed using the COCO evaluation toolkit integrated with PyTorch, which outputs detailed average precision (AP) and average recall (AR) scores at multiple IoU thresholds and object sizes.

By extracting results in a consistent way across all models, it was possible to directly compare performance under the same dataset and evaluation conditions. These metrics were analyzed to determine which model provides the most reliable performance for real-time handball penalty shot analysis.

Precision measures how many of the model's detections were actually correct. It reflects the model's ability to avoid false positives—detecting something as a ball or goalpost when it's not. High precision is important so the system remains reliable during real-time analysis.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall indicates how many actual objects the model successfully detected. This is important because if the ball is going fast during a penalty shot, missing the ball briefly could impact the system's usefulness.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1 Score combines both precision and recall into a single value. It's useful when trying to balance the trade-off between catching all relevant objects and being accurate about them.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

mAP@0.5 (mean Average Precision at 50 percent IoU) considers a prediction correct if the overlap between the predicted and actual bounding boxes is at least 50 percent. It gives a good general idea of how well the model detects and localizes objects.

$$\text{mAP@0.5} = \frac{1}{C}\sum_{c=1}^{C} AP_c\Big|_{\text{IoU}=0.5}$$

mAP@0.5:0.95 is more strict and it calculates the average precision across multiple IoU thresholds from 0.5 to 0.95 in steps of 0.05. This provides a more detailed picture of the model's ability to precisely detect and localize objects under varying conditions.

$$\text{mAP@0.95} = \frac{1}{C}\sum_{c=1}^{C} AP_c\Big|_{\text{IoU}=0.95}$$

These metrics will be analyzed in the Results section to compare how well each model performs in detecting and tracking the handball and goalpost. This will help identify which model is most suitable.

4.2 YOLOv8 Model

The YOLOv8 Model demonstrated a strong relationship between model size and performance.

YOLOv8n delivered precision and recall values above 0.92, with a mAP@0.5:0.95 of 0.763. Despite being the smallest and fastest model in the family (training completed in just under 20 minutes), it was able to reliably detect the handball across most frames. This makes it a compelling candidate for highly resource-constrained environments, although its lower fine-grained accuracy at stricter IoU thresholds indicates some difficulty in precise localization.

YOLOv8s represented a significant step up, achieving a recall of 0.961 and mAP@0.5:0.977, while still training in a manageable time of around 58 minutes. The high recall shows that YOLOv8s rarely missed detections, an important quality in tracking a fast-moving ball.

YOLOv8m was the strongest performer in this series, achieving the highest mAP@0.5 of 0.979 and maintaining excellent recall at 0.968. However, this came at a heavy computational cost, with training taking over 12 hours. The improvements over YOLOv8s were relatively small compared to the increase in time, raising questions about whether the performance gain justifies the additional resources.

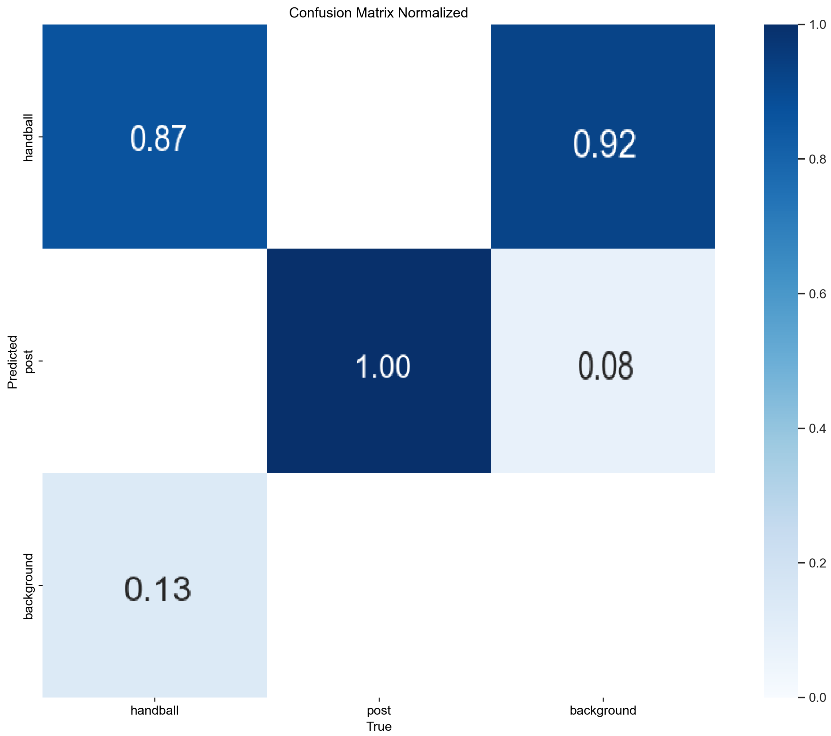


Figure 4.2: *YOLOv8n*

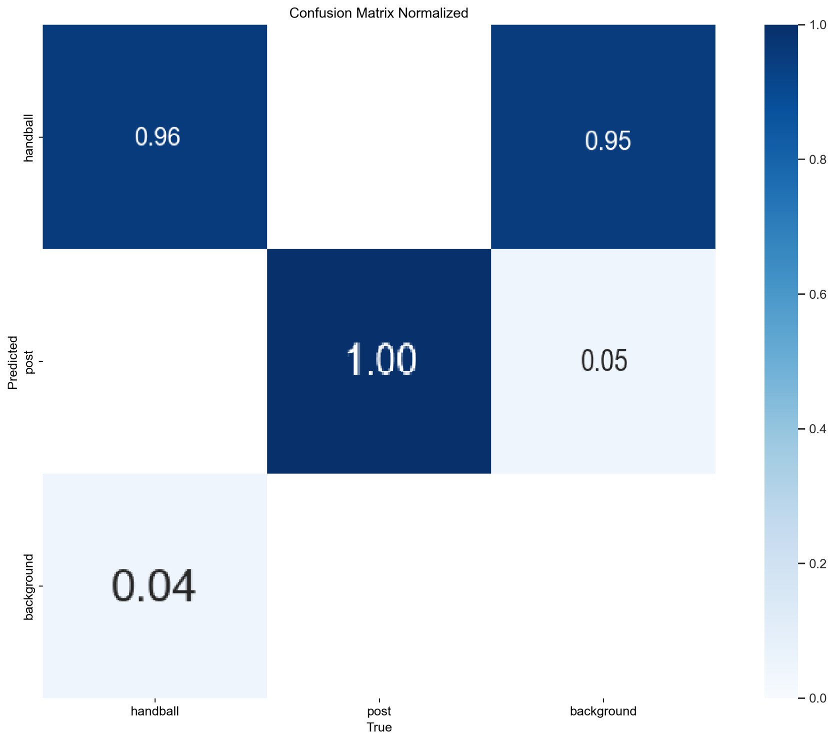


Figure 4.3: *YOLOv8s*

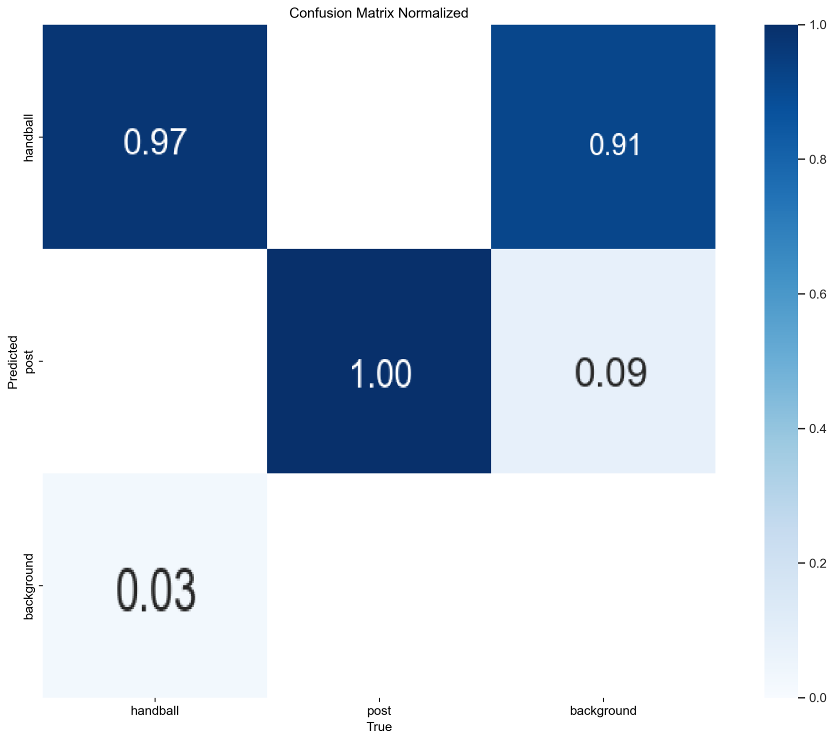


Figure 4.4: *YOLOv8m*

The confusion matrices for YOLOv8 models (Figures X–Y) highlight the differences in classification performance across scales. YOLOv8n shows notable confusion between handball and background, resulting in false negatives and false positives. YOLOv8s significantly reduces these errors, demonstrating strong recall and reliability in detecting the ball. YOLOv8m achieves the highest handball detection accuracy but with increased computational demands. Across all models, the post class was perfectly detected, indicating that the goalpost is an easier feature to identify compared to the small, fast-moving ball.

Overall, the YOLOv8 models illustrate the expected trade-off: larger models provide higher accuracy and robustness but at the cost of efficiency. For real-time applications, YOLOv8s and YOLOv8n are more attractive, while YOLOv8m is best suited for offline, high-accuracy analysis.

4.3 YOLOv11 Model

The YOLOv11 models showed competitive results with some notable differences compared to YOLOv8.

YOLOv11n achieved strong recall (0.946) and mAP@0.5 (0.955), but its precision of 0.894 was lower than YOLOv8n. This suggests that while YOLOv11n was effective at detecting most true objects, it generated more false positives compared to its YOLOv8 counterpart.

YOLOv11s delivered one of the strongest performances overall, with recall of 0.969 (the highest of all models) and mAP@0.5 of 0.980, comparable to YOLOv8m but with a fraction of the training time (54 minutes vs. 12+ hours). This combination of efficiency and accuracy makes YOLOv11s particularly well-suited to real-time sports analytics.

YOLOv11m reached precision of 0.954 and mAP@0.5 of 0.980, very close to YOLOv8m. However, its stricter mAP@0.5:0.95 was slightly lower (0.793 vs. 0.815 for YOLOv8s), indicating reduced robustness at fine-grained localization. Training time was also high (over 12 hours), placing it in the same category as YOLOv8m as more suitable for offline use.

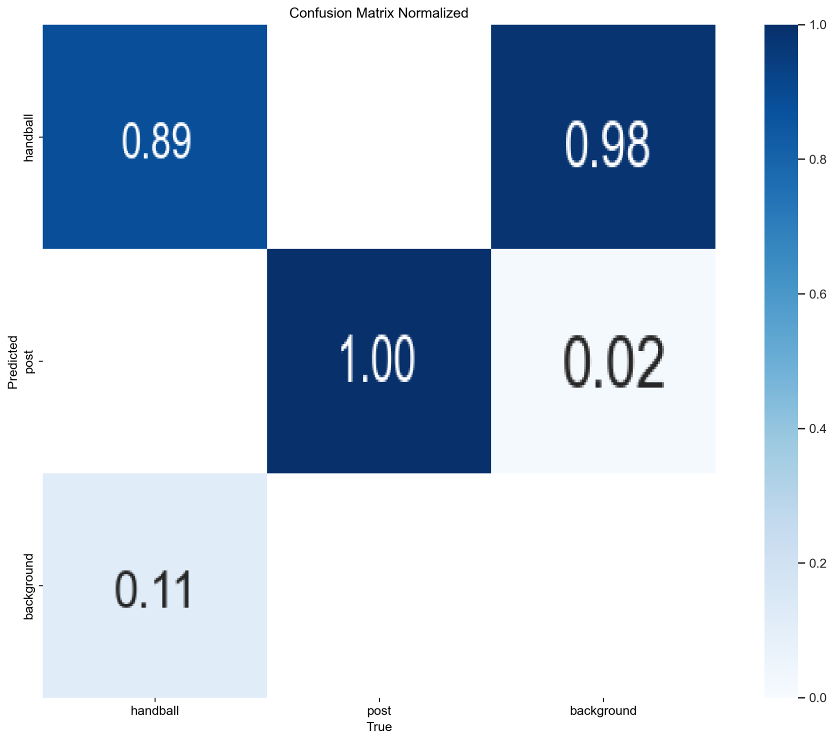


Figure 4.5: *YOLOv11n*

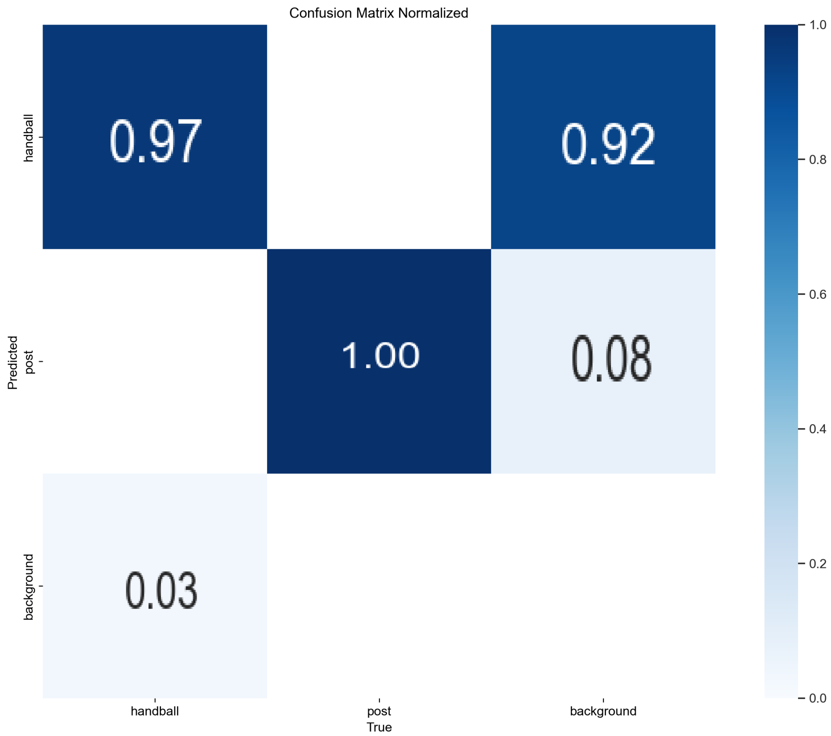


Figure 4.6: *YOLOv11s*

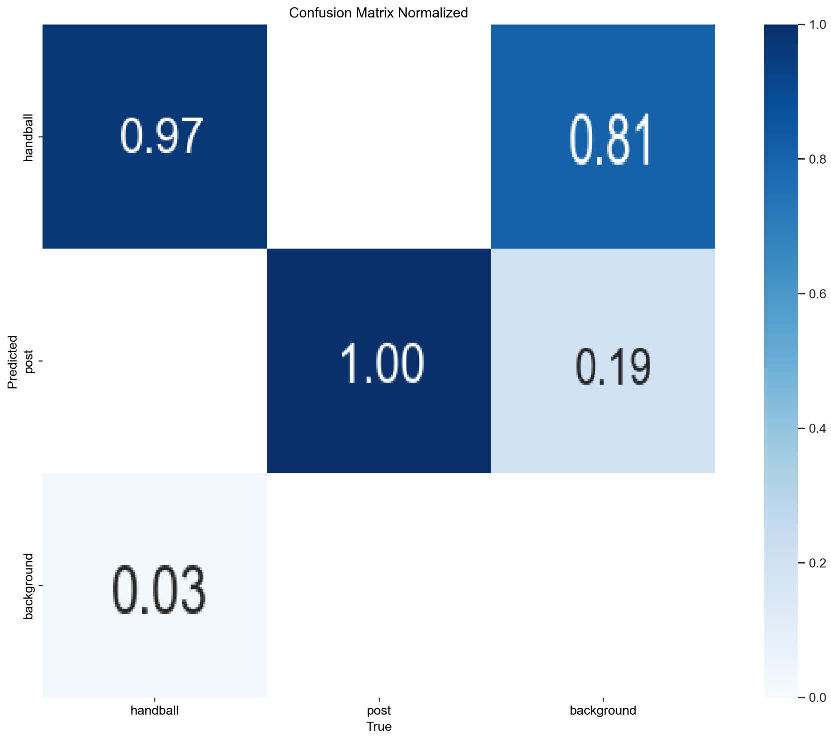


Figure 4.7: *YOLOv11m*

The confusion matrices for YOLOv11 models (Figures X–Y) further highlight the trade-offs between scale and accuracy. YOLOv11n achieved strong recall but suffered from a relatively high rate of false positives, consistent with its lower precision values. YOLOv11s provided the best overall balance, with minimal false positives and high handball detection rates, confirming its suitability for real-time applications. YOLOv11m reached high accuracy but showed a noticeable increase in misclassification of handballs as background, reflecting weaker fine-grained localization. Overall, the matrices reinforce the earlier quantitative findings: YOLOv11s is the most efficient and balanced model for handball tracking.

Overall, YOLOv11 showed that its efficiency gains are not always consistent across scales. The n variant was weaker than YOLOv8n in precision, but the s variant stood out as one of the best-balanced models across all experiments.

4.4 Faster R-CNN

Faster R-CNN behaved differently compared to the YOLO families due to its two-stage architecture.

It achieved the highest recall of all models (0.978), meaning it almost never missed the handball. This is a critical advantage in sports analysis, where even one missed detection could compromise shot tracking. However, this came at the cost of precision (0.887), indicating more false positives than the YOLO models. The mAP@0.5 of 0.887 and mAP@0.5:0.95 of 0.707 were also significantly lower, reflecting difficulties in bounding box precision and generalization across IoU thresholds.

Training time was long (over 3.5 hours), and inference is also slower compared to YOLO. While Faster R-CNN may be better suited for applications where missing a detection is unacceptable, its lower precision and efficiency limit its practicality for real-time use in handball.

4.5 SSD

The SSD model produced noticeably weaker results compared to YOLO and Faster R-CNN. With a precision of 0.83 and a recall of 0.66. Its mAP@0.5 of 0.852 was competitive at a lower IoU threshold, but its mAP@0.5:0.95 dropped sharply to 0.568, highlighting limitations in precise localization.

In terms of efficiency, SSD trained faster than two-stage detectors like Faster

R-CNN and was less resource-demanding than YOLOv8m and YOLOv11m. However, its reduced detection accuracy — especially at stricter IoU thresholds — makes it the weakest candidate among the models evaluated for real-time penalty shot analysis.

4.6 Cross-Model Insights

When comparing across all models, several important themes emerge:

Trade-off between accuracy and efficiency. YOLOv8m and YOLOv11m achieved the best mAP@0.5 values but required extensive training resources. In contrast, YOLOv8n and YOLOv11n trained quickly and were efficient but showed slightly weaker accuracy.

Balance of performance. YOLOv11s stood out as the most balanced model, combining high recall (0.969) and precision (0.940) with fast training time. This makes it ideal for real-time deployment.

Robustness at stricter IoUs. YOLOv8s achieved the highest mAP@0.5:0.95 (0.815), indicating it was the most reliable at precise localization — a valuable trait in small object detection tasks like tracking a handball.

Recall vs. precision trade-off. Faster R-CNN's extremely high recall but lower precision highlights a classic trade-off: it will catch nearly everything, but at the cost of producing more false detections. For penalty analysis, false positives may disrupt tracking pipelines, making YOLO's balance of precision and recall more desirable.

Suitability for research objectives. Since the dissertation emphasizes real-time detection and analysis of penalty shots, the YOLO models — particularly YOLOv8s, YOLOv8m, and YOLOv11s — align more closely with this goal than Faster R-CNN or SSD.

In summary, YOLOv8m delivered the highest detection accuracy but at an impractical computational cost, while YOLOv11s offered the best trade-off between accuracy, recall, and efficiency. Faster R-CNN demonstrated value in recall-focused scenarios but is less appropriate for real-time deployment. SSD, although efficient, underperformed significantly in both recall and small-object detection. Therefore, for the purpose of this dissertation, YOLOv11s and YOLOv8s emerge as the most suitable models, balancing performance with efficiency to meet the demands of real-time handball penalty shot analysis.

4.7 Media Pipe

To extend the analysis beyond ball detection, MediaPipe Pose was integrated into the pipeline to evaluate the throwing technique of players during penalty shots. While the primary research focus was on accurately tracking the ball's trajectory, pose estimation provides an additional layer of performance analysis that connects directly to training applications.

The system specifically evaluated whether the elbow position rose above the shoulder line during the throwing motion. This is the proper technique when throwing a handball, if not done this way it can lead to less throwing power and accuracy, also with improper technique it can lead to injury.

By also using pose estimation it strengthens the research objective by showing that real-time analysis of penalty shots can encompass both outcome-based metrics (ball trajectory) and technique-based metrics (player posture and movement).

4.7.1 Successful Implementation

Beyond quantitative evaluation of the models, the complete pipeline was successfully implemented and tested on real penalty shot scenarios. The system integrated ball and goalpost detection with pose estimation, producing real-time outputs that aligned with the objectives of this study.

The trained models were deployed in a Python-based pipeline, where detections were drawn as bounding boxes over video frames. This allowed the ball to be continuously tracked during penalty shots, and its trajectory was calculated from frame-to-frame positions.

In addition to ball and post tracking, MediaPipe pose estimation was integrated to analyze the player's throwing technique. Specifically, the system was able to verify whether the elbow rose above the shoulder during the throwing motion, a key indicator of proper technique. This provided a second layer of analysis beyond ball trajectory alone.

The implementation was evaluated on test video data, demonstrating smooth tracking and accurate event recognition under real-time conditions. Models such as YOLOv11s and YOLOv8s provided the most stable performance, achieving both reliable detections and fast inference speeds. Importantly, the combined pipeline showed that both the technical and practical objectives of the research were met: penalty shots could be tracked and analyzed, offering actionable insights for athletes and coaches.

Chapter 5: Conclusions and Recommendations

5.1 Research Questions and Hypothesis

The hypothesis for this study was that a real-time tracking system could accurately track the trajectory of a handball during penalty shots and provide performance data for both players and coaches. The experimental results support this hypothesis, showing that computer vision techniques are effective in capturing both outcome- and technique-based insights.

RQ1: How accurately can a program analyse handball penalty shots?

The system achieved reliable performance in tracking penalty shots by combining ball detection, trajectory analysis, and pose estimation. YOLO-based models proved capable of real-time detection, while MediaPipe added insights into throwing technique. Together, these components demonstrated that penalty shots can be analysed both in terms of outcome (shot placement, accuracy, and speed) and execution (player form).

RQ2: Which ball detection model is best for this scenario?

Among the evaluated models, YOLOv11s emerged as the most balanced choice, achieving high recall (0.969) and precision (0.940) while maintaining efficiency suitable for real-time use. YOLOv8s also performed strongly, achieving the highest mAP@0.5:0.95 (0.815), making it highly reliable for precise localization. Larger models (YOLOv8m, YOLOv11m) offered only marginal gains at the cost of training efficiency, while Faster R-CNN and SSD showed limitations that make them less suitable for real-time deployment.

RQ3: What are the most significant statistics and metrics to consider when analysing handball penalty shots?

The study found that both technical and outcome-oriented metrics are essential. From the outcome perspective, shot trajectory, on/off-target classification, and shot speed provide performance indicators for coaches and players. From a technical perspective, pose estimation—particularly monitoring elbow and shoulder alignment—provides actionable insights into throwing technique. Combined with precision, recall, and mAP values for detection reliability, these metrics form a comprehensive framework for penalty shot analysis.

Overall, the results confirm the hypothesis: a computer vision-based system can accurately and meaningfully analyse handball penalty shots, offering both technical and tactical value.

5.1.1 Limitations

While this research demonstrated the feasibility of using computer vision to analyse handball penalty shots, several limitations should be acknowledged.

First, the dataset size was relatively small (around 2,000 images after augmentation) and limited to controlled training environments. This may restrict the model's generalisability to real match conditions, where lighting, backgrounds, player styles, and occlusions vary significantly.

Second, the camera setup was restricted to a single rear-side perspective. Although this angle was effective for tracking the ball and evaluating throwing form, alternative viewpoints such as frontal or overhead shots were not captured, limiting contextual richness.

Additionally, while pose estimation was incorporated, only a single biomechanical indicator (elbow position relative to the shoulder) was analysed, leaving more detailed multi-joint kinematic analysis unexplored.

Finally, computational constraints restricted experimentation with larger models and prolonged training times even for medium-sized architectures.

5.1.2 Future Work

Several directions for extending this research can be identified.

Future studies could build larger and more diverse datasets, including footage from competitive matches and multiple environments, to improve robustness and generalisability.

Expanding the camera setup to incorporate multiple synchronized viewpoints (frontal, side, and overhead) would enable 3D trajectory reconstruction and deeper biomechanical insights.

From a modelling perspective, testing emerging architectures such as YOLOv12 or other advanced real-time detectors could further enhance performance.

Beyond detection, integrating tracking algorithms (e.g., SORT, ByteTrack) would allow consistent player and ball identity across frames, while applying advanced pose estimation frameworks could support full-body kinematic analysis.

Additionally, combining computer vision with wearable sensors (e.g., accelerometers, gyroscopes) could provide complementary data to handle ball occlusions and improve reliability in dynamic conditions.

Finally, deploying the system in real time during training sessions, with live feedback to players and coaches, would represent a significant step toward practical adoption in the sport.

List of References

[1] H Romero and Alex Morales Acosta. A review: Comparison of performance metrics of pretrained models for object detection using the tensorflow framework. *IOP Conference Series: Materials Science and Engineering*, 2020.

[2] Niloofar Abed and Ramu Murugan. Strategies for improving object detection in real-time projects that use deep learning technology. In *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, pages 1–6, 2023.

[3] ultralytics. *Model Comparisons: Choose the Best Object Detection Model for Your Project* [Online]. Available: `https://docs.ultralytics.com/compare/`, 2024.

[4] European Handball Federation. *EHF GLOSSARY OF HANDBALL TERMS AND EXPRESSIONS*, 2022.

[5] International Handball Federation. *Rules of the Game – Indoor Handball*, 2022.

[6] Surbhi Parganiha Vishesh Tamrakar and Saurabh Kumar Pandya. Real-time object detection and recognition with computer vision, 2024.

[7] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.

[8] Miran Pobar Matija Burić and Marina Ivašić-Kos. Adapting yolo network for ball and player detection. 2019.

[9] Ross Girshick Ali Farhadi Joseph Redmon, Santosh Divvala. YOLO: You only look once: Unified, real-time object detection. 2016.

[10] Hongji Wang Lihao Chen Ma Linjian Xiaoyan Zhang Shengyu Lu a, Beizhan Wang. A real-time object detection algorithm for video. 2019.

[11] Haian Huang Yue Zhou Yudong Wang Yanyi Liu Shilong Zhang Kai Chen Chengqi Lyu, Wenwei Zhang. Rtmdet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*, 2022.

[12] Marina Ivasic-Kos Matija Buric, Mirian Pobar. Ball detection using yolo and mask r-cnn. 2018.

[13] Meisam Teimouri, Mohammad Delavaran, and Mahdi Rezaei. *A Real-Time Ball Detection Approach Using Convolutional Neural Networks*, pages 323–336. 12 2019.

[14] Shenshen Gu, Lu Ding, Yue Yang, and Xinyi Chen. A new deep learning method based on alexnet model and ssd model for tennis ball recognition. In *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*, pages 159–164, 2017.

[15] R Deepa, E Tamilselvan, E.S. Abrar, and Shrinivas Sampath. Comparison of yolo, ssd, faster rcnn for real time tennis ball tracking for action deci-

sion networks. In *2019 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pages 1–4, 2019.

[16] Aman Sharma. Object detection model: An enhanced ssd. *International Journal for Research in Applied Science and Engineering Technology*, 12:1598–1612, 05 2024.

[17] Zheqi Zhu, Yingjia Gao, and Shenshen Gu. Tennis ball collection robot based on mobilenet-ssd. pages 300–307, 12 2021.

[18] Niloofar Abed and Ramu Murugan. Strategies for improving object detection in real-time projects that use deep learning technology. In *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, pages 1–6, 2023.

[19] Naif Alsharabi. Real-time object detection overview: Advancements, challenges, and applications. , 3:12, 2023.

[20] Imran Khan, Chen Tianhua, Khan Malak Abid Ali, Syed Aamir, and Waseef Menhaj. Object recognition in different lighting conditions at various angles by deep learning method. 01 2022.

[21] Francisco Jimenez Demetrio Lozano Hafrún Kristjánsdóttir Jose M. Saavedra, Miguel Pic. Relationship between game-related statistics in elite men's beach handball and the final result: a classification tree approach. 2019.

[22] Meng Du and Xiaoru Yuan. A survey of competitive sports data visualization and visual analysis. *Journal of Visualization*, 24, 08 2020.

[23] Janet Parks, Patricia Shewokis, and Carla Costa. Using statistical power analysis in sport management research. *Journal of Sport Management*, 1999.

[24] A. Field. *Discovering Statistics Using IBM SPSS Statistics*. SAGE Publications, 2013.

[25] Alexey Popovich, Lyudmila Bezukladnova, Alexander Bezukladnov, and Elena Goncharova. Competitive activity indicators in handball. statistics, analysis, forecast. *BIO Web of Conferences*, 26, 01 2020.

[26] Suhas Patel and Dipesh Kamdar. Accurate ball detection in field hockey videos using yolov8 algorithm. 9:411–418, 2023.

[27] Shohruh Begmatov, Mukhriddin Arabboev, Mokhirjon Rikhsivoev, Saidakmal Saydiakbarov, Zukhriddin Khamidjonov, Sardor Vakhkhobov, Xurshid Aliyarov, and Oybek Karimov. Ball and player detection in futsal videos using yolov8 model. 05 2024.

[28] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for perceiving and processing reality. In *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, 2019.

[29] Romeo Sajina and Marina Ivasic-Kos. Analysis of multi-person pose forecasting models on handball actions. In *2024 8th International Conference on Computer, Software and Modeling (ICCSM)*, pages 57–62, 2024.

[30] Deepak Rai, Anjali, Ankit Kumar, and Anjali Baghel. Pose detection using opencv and media pipe. In *2024 International Conference on Integrated Circuits, Communication, and Computing Systems (ICIC3S)*, volume 1, pages 1–6, 2024.

[31] Xinyan Xie. Yoga pose detection using mobilenetv3+pose comparison and instruction. In *2024 4th International Conference on Information Technology and Contemporary Sports (TCS)*, pages 7–12, 2024.

[32] Syed Irfan, Md Mia, Muhaiminul Islam, Sahidul Islam, and Tanjin Islam. Advancements in object detection and tracking algorithms: An overview of recent progress. *EPRA International Journal of Research Development (IJRD)*, pages 134–140, 2023.

[33] K. Kulpa P. Samczyński J. Misiurewicz D. Gromek M.Botwicz, W.Klembowski. The concept of an rf system for the detection and tracking of the ball and players in ball sports. 2017.

[34] Carlos García-Sánchez Sergio Sánchez-López Moises Marquina, Demetrio Lozano and Alfonso de la Rubia. Development and validation of an observational game analysis tool with artificial intelligence for handball: Handball.ai. 2023.

[35] Dharma Aryani, Kartika Dewi, Fery Ta.by, and Evita Sanggaria. Real-time ball detection and tracking using raspberry pi. *INTEK: Jurnal Penelitian*, 10, 04 2023.

[36] Brian K. S. Isaac-Medina, Chris G. Willcocks, and Toby P. Breckon. Multi-view vision transformers for object detection. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 4678–4684, 2022.

Chapter A: Introduction of Appendix

Interview summaries, sample questionnaires, and references should be placed in this section. For easier referencing, figures, tables, graphs, photos, diagrams, etc., should be inserted within the main text such as the literature review, the experimental process or procedure, the results and discussion chapters. Appendices are usually used to present further details about the results. Appendices may be a compulsory part of a dissertation, but they are not treated as part of the dissertation for purposes of assessing the dissertation. So any material which is significant to judging the quality of the dissertation or of the project as a whole should be in the main body of the dissertation (main text), and not in appendices.

Chapter B: Sample Code

This is the Github link for the code: https://github.com/jacobattard/Thesis.