Optimal Player Substitution Strategy

PROJECT SUPERVISOR

Dr. Fahad Sherwani

PROJECT TEAM

Usama Baloch	K191459
Muhammad Sameer	K191526
Shubair Raza	K191300

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

May 2023

Project Supervisor	Dr. Fahad Sherwani	
Project Team	Usama Baloch	K19-1459
	Muhammad Sameer	K19-1526
	Shubair Raza	K19-1300
Submission Date	May 2, 2023	

Dr. Fahad Sherwani _____

Supervisor

Ms. Farah Sadia _____

Co-Supervisor

Dr. Zulfiqar Ali Memon _____

Head of Department

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

Acknowledgement

I would like to express my deepest gratitude to Dr. Fahad Sherwani, our project supervisor, for his invaluable guidance, support, and expertise throughout the entire duration of our final year project, "Optimal Player Substitution Strategy." His insights, constructive feedback, and constant encouragement were instrumental in helping us achieve our goals and complete this project to the best of our abilities. Without his unwavering dedication and commitment, this project would not have been possible. I am truly thankful for the opportunity to work under his mentorship and for his unwavering support throughout this journey. In addition, the team would also like to express their appreciation for the constructive feedback and helpful suggestions offered by the jury, which enabled them to evaluate their work critically and make appropriate enhancements to the project.

Abstract

Increasing demand for sports has raised the challenge of automated analysis of players' performance in live games. Decision-making on substituting players between matches is done only manually by coaches using their experience. With the help of deep learning, we can achieve the challenge of automated analysis of players' performance in live games. This Final Year Project (FYP) explores the application of deep learning models like YOLOv7, YOLOv8, DEEPSORT, and ByteTrack. The aim of this project is to present a model that will track the performance of every individual player in the team and will show a table of players' performance. Our Proposed system contains multiple entities first starting by detecting as well as tracking the players and ball for further analysis, then by classifying the players, and at last calculating the performance of every classified player on parameters (stamina and ball possession), this solution helps coaches to identify which player to substitute on real facts.

Contents

List of Figures

List of Tables

1. Introduction

In recent years, the popularity of sports has surged, leading to a growing interest in the analysis of player performance during live games. To meet this challenge, researchers have turned to the fields of image and video processing, as well as computer vision and deep learning algorithms. These technologies have enabled the development of automated models capable of analyzing sports videos and extracting valuable insights into player performance.

One sport that has seen significant progress in this area is soccer, which has increasingly embraced deep learning models and artificial intelligence (AI) to enhance team performance. With advanced technologies such as these, teams can gain a competitive edge by tracking the performance of individual players in real-time and making informed decisions based on that data.

A number of studies have applied deep learning models in the area of soccer. In [1] YOLOv2 is used for object detection. They compared the introductory structure of YOLOv2 and a modified interpretation of YOLOv2 by adding one- by- one convolutional subcaste. Although YOLOv2 is formerly outstanding in terms of speed and delicacy, it still falls short to detect small objects. To break this problem, the network structure of YOLOv2 is modified. In [2], YOLOv3 is used to detect and classify objects in soccer videos, like players, soccer balls, and backgrounds, SORT algorithm is used for player tracking. In [7], they proposed a Sports Action Mining (SAM) framework on the usage of positional data for recognizing actions, e.g., dribbling.

This study aims to present a model for tracking the performance of each player on a soccer team. By defining a threshold for performance points, the model will show the performance of every player on the foot of the player. This approach promises to provide coaches and team managers with a powerful tool for optimizing team performance and maximizing their chances of success.

2. Related Work

Object detection and object tracking are one of the difficult tasks. But deep learning models help us to do this more accurately than others. YOLO (you only look once) is used to detect person class objects [1,2,4]. In [1] YOLOv2 is used for object detection. In this paper, they compared the basic structure of YOLOv2 and a modified version of YOLOv2 by adding a one-by-one convolutional layer. Although YOLOv2 is already outstanding in terms of speed and accuracy, it still falls short of detecting small objects. To solve this problem, the network structure of YOLOv2 is modified. The model is trained and tested on the VOC2007 dataset, which has a total of 9963 images with 20 categories. YOLO is made up of four parts: input image, CNN, Non-maximal Suppression method, and outputs. The activation methods used in [1,2] were linear and relu or leaky relu. For the final layer, they used linear activation as the last layer predicts category probabilities and bounding

boxes and for the rest of the layers, they used relu or leaky relu. To check the error sum-squared error was used as the loss function because it is easy to be optimized [1].
Some updates to YOLO! of little design changes to make it better. YOLOv3 is used to detect and classify objects in soccer videos, such as players, soccer balls, and backgrounds. This model is especially useful for distinguishing objects in the playfield and outside the playfield, which is a highly challenging task for conventional object-tracking approaches. YOLOv2 model does not directly predict the size and position of the object (in terms of the bounding box) in the frame. YOLOv3 was obtained by making further improvements to the YOLOv2 model, adding several convolutional layers after the base extractor with the last layer predicting a three-dimensional tensor encoding bounding box, class predictions, and objectness. For more than one class inside the bounding box multilabel classification is used and for the binary class, binary cross-entropy is used during the training process. YOLOv3 predicts an objectness score for each bounding box using logistic regression. This should be one of the bounding boxes prior that overlaps a ground truth object by more than any other bounding box prior. If the bounding box prior is not the best but does overlap a ground truth object by more than some threshold, we ignore the prediction. After detection, we need to keep track of players and the ball. It is difficult to track objects. After obtaining the detected and classified object bounding boxes from the YOLOv3 detection model, SORT is used to predict the position of the object in the next frame by estimating the object's velocity from KF using the bounding box of the detected object. Sort applies data association on successive frames using the Hungarian method and measures bounding box overlap which is used as the association metric. SORT framework consists of two stages; features are extracted by the first stage while the second stage classifies the required object. The proposed model achieves a tracking accuracy of 93.7% on multiple objects tracking accuracy metrics with a detection speed of 23.7 frames per second (FPS) and a tracking speed of 11.3 FPS [2].

Identifying players is also difficult to do. In a live match there is an audience as well, it will be difficult to identify which one is the player and which one is the audience. In [4] it is mentioned that deep learning models to detect, track and identify soccer players. To identify/classify players and audience two newly developed filters, spatial feature filter and bounding box location filter have been described that help in classifying players and audience. A new method of identifying every player by detecting a player t-shirt number has been developed and illustrated. This method provides tracks with high confidence and identity to most of the players corresponding to individual t-shirt numbers. Instead of tracking both audience and players it will take a lot of time. Using static bounding box location and spatial features, the detected person class objects are classified into players and non-players (audience) which helps us to reduce the number of objects to be tracked. There are 4 sections, and these are player detection, player-audience classification, player tracking and number tracking. For player detection the YOLOV3 algorithm is used. For player-audience classification the Spatial Feature Filter and Bounding Box Filter are used. For Spatial Feature Filter we make use of the spatial differences between players and audience. We collected over 10K images of players and non-players to train a deep learning model based on ResNet-50 architecture to classify the images into 2 categories: player and audience. For the bounding box filter a default value of 0.5 can be used, which indicates a bounding box has the same location during 50% of the entire game and thus the location can be considered as a non-player bounding box [4].

The world of sports intrinsically involves fast and complex events that are difficult for coaches, trainers, and players to analyze, and for audiences to follow. In sports, talent

identification and selection are imperative for the development of future elite-level performers. We propose an approach that automatically generates visual analytics from videos specifically for soccer to help coaches and recruiters identify the most promising talents. Following are the approaches that are used:

- Convolutional Neural Networks (CNNs) to localize soccer players in a video and identify players controlling the ball.
- Deep Convolutional Generative Adversarial Networks (DCGAN) for data augmentation.
- Histogram-based matching to identify teams and (d) frame-by-frame prediction and verification analyses to generate visual analytics.

We compare our approach with state-of-the-art approaches and achieve an accuracy of 86.59% in identifying players controlling the ball and an accuracy of 84.73% in generating the game analytics and player statistics. Computer vision is also used behind the scenes, in areas such as training and coaching, and providing help for the referee during a game. In team-based sports, such as soccer, talent identification is a complex process due to the different qualities associated with performance, including personal and tactical attributes. Personal attributes refer to how well the player can keep the ball possession with him/herself, and tactical attributes refer to how successful the player is in passing the ball to teammates and adapting to different strategies. To solve the talent identification problem there is an approach and system that automates the talent identification problem by generating visual analytics and player statistics for soccer from a video using traditional algorithms and deep learning techniques for computer vision. The dataset used consists of 49,952 images that are annotated into two classes namely: players with the ball (12,586 images) and players without the ball (37,366 images) [5].

Computer vision plays a crucial role in exploiting the information given by football videos. The automatization of tracking and detection of objects on the videos facilitates the analysis of the active entities or the identification of infringement during a match. Computer vision can help many applications such as shot classification, ball possession, or event detection. A piece of crucial information is needed for all these applications, the ball position. The automatization of the ball position in long-shot football sequences is a challenging task for many reasons. Different methods have been used to produce automated ball-tracking systems. In recent years, the use of deep learning improved the performance of the tracker. Ball tracking in a video sequence is a huge challenge that is extremely hard to implement simple methods of ball detection. To achieve better performance, it could be interesting to use the information given by the set of frames of the video instead of considering each frame independently. The works which only use classical methods of image processing do not need a dataset to train their model.

However, the generation of an accurate dataset without inconsistency is a crucial step for models using a convolutional neural network. The work is divided into two parts: a ball detector and a tracking system. Detecting the ball from a long-shot sequence of a football game is a complex task to implement. Several visual features of the ball can differ from one match to another one, such as the colors of the ball or the distance of the camera to the pitch, varying the size of the ball in the image. During a match, the circular shape of the ball can become blurry and elliptical due to high speed, as well as located in front of a player or a line that has often the same color as the ball [6].

To evaluate the performance of the individual player, dribbling is one of the most crucial

factors which can be considered. Recent advances in Computer Vision and Machine Learning empowered the use of image and positional data in several high-level analyses in Sports Science, such as player action classification, recognition of complex human movements, and tactical analysis of team sports. In the context of sports action analysis, the use of positional data allows new developments and opportunities by considering players' positions over time. The proposed Sports Action Mining (SAM) framework is grounded on the usage of positional data for recognizing actions, e.g., dribbling. The experimental results pointed out the Random Forest classifier achieved a balanced accuracy value of 93.3% for detecting dribbling actions, which are considered complex events in soccer. Scouting plays a key role in soccer analysis since this task provides high-level information related to the tactical and technical aspects of the match under analysis, supporting the decision-making process of coaches and their assistants. Nowadays, there are some companies in soccer analysis that provide scouting analysis on demand such as Stats Perform. However, this service is quite expensive and there are no open-source solutions for supporting sports performance analysis. In this context, the development of methods and tools to support automatic scout analysis has attracted a lot of attention from both academia and the sports industry. We propose a novel framework for detecting sports actions from positional data, called the Sports Action Mining framework (SAM). SAM was designed to provide general profiling and useful patterns to recognize sports-related actions based on the sequence of movements. SAM receives positional data from a tracking algorithm and provides insights on different sport action goals, such as recognition tasks able to support automatic scouting and skill modeling based on clustering computing. These outcomes are achieved by using a pipeline, grounded on association rule mining and machine learning algorithms, comprising data pre-processing, trace extraction, generation of association rules, action profiling, and recognition tasks. The method we proposed takes advantage of association rule mining algorithms for modeling actions in soccer. In other words, this framework can address the problem of automatically representing players and their displacements, which allows the extraction of patterns for further scouting and action recognition [7].

3. Requirements

Following are the hardware assumptions for using the system

- **I3** core processor or above
- **4** GB RAM or above
- **128** GB SSD space
- Have access to Google Colab if the system doesn't have GPU.
- Keyboard and Mouse
- Internet Browser (any)

4. Design

In this section, we outline the design of our FYP consists of the following steps:

1. **Prepare dataset:** In this step, we read the video and convert it into the frames (images) for further analysis.

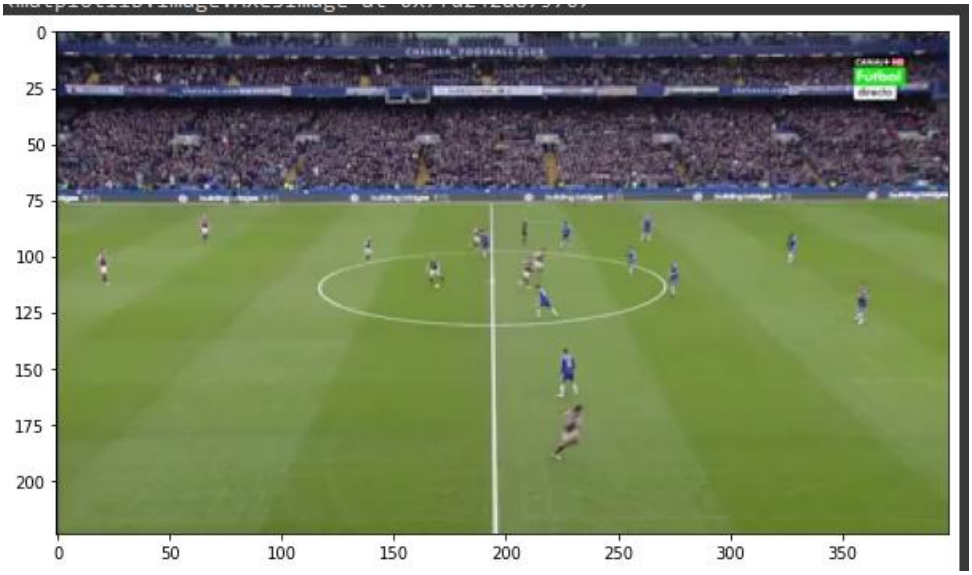


Figure 0.1 Video to Frame conversion

2. **Object detection model:** To detect players, ball, and referee, three different object detection models were trained and evaluated on the dataset: yolov5, yolov7, and yolov8.

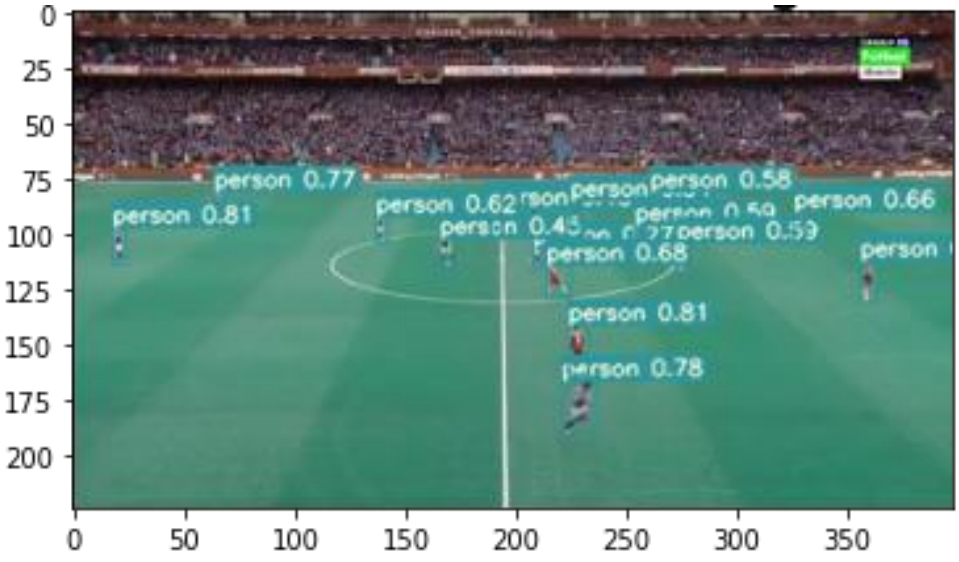


Figure 0.2 Player detection

3. **Model comparison:** Based on performance such as loss, speed, and efficiency, yolov5 was selected as the best model for this project.

4. **Object Tracking model:** The results of object detection are passed to ByteTrack for player tracking.

5. **Player Tagging:** After tracking, ByteTrack will also tag players in order to classify/recognize individual players

6. **Parameters calculation:** After tagging, we calculated two parameters: distance traveled by a player (stamina) and ball possession.

7. **Performance calculation:** To calculate the performance of the individual player, we use the above-calculated parameters and calculate the performance of the individual player.

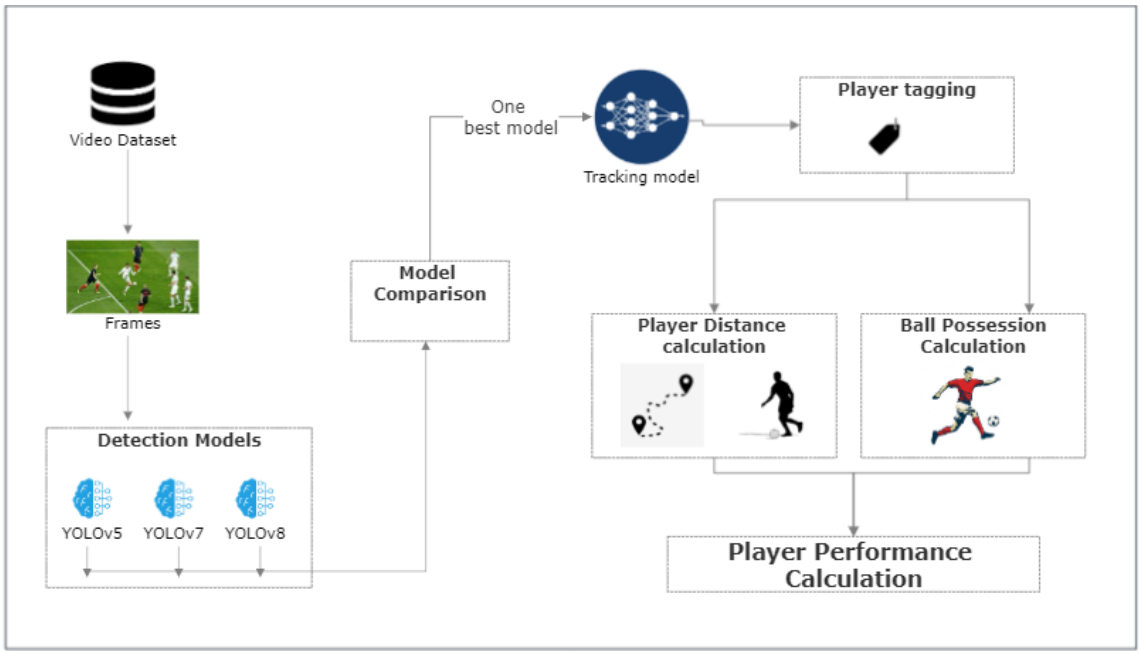


Figure 0.3 System architecture

4.1 Overall Workflow

First, we have video as our input and then we convert it into frames which will be used for further processing. These frames will be fed into our detection system which will use three Famous detection models YOLOv5, YOLOv7, and YOLOv8. Detection models will detect the usable characters from the frames like (players, referees, and balls). Then there will be a comparison between the detection models and the best model (YOLOv5 in our case) will be attached to another system called ByteTrack which tracks the detection frames and presents it as a video in front of us. Moreover, in parallel, it also classifies the players by assigning unique numbers to the players during the tracking and that identification will be used for further analysis, After that, we will calculate two parameters that are distance traveled by the player (stamina) and how much time the player was in ball possession, Finally, we will calculate the performance of the player based on these two parameters.

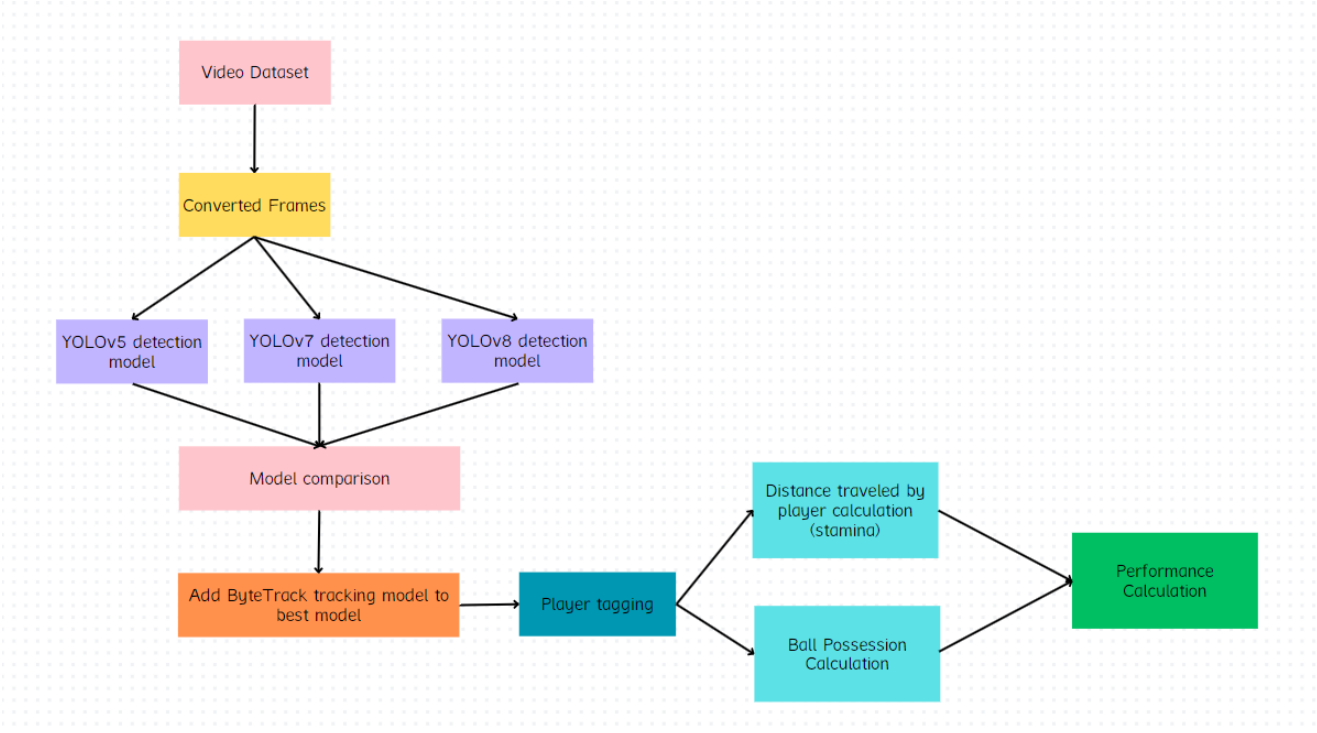


Figure 0.4 Generic workflow of the system

5. Implementation

5.1 Dataset:

Dataset is an open-source data set that is used for computer vision problems in football matches. The dataset has multiple videos with a size of 45 minutes (both half), there are eight videos that we are working on, and the whole size of the folder of the dataset is approximately 37.55 GB. Data quality is 1080 and some of them are also 720. For training the dataset we also needed some data preprocessing that like feature extractions from the frames, and analyzing those features (finding differences between team players and referee)

5.2 YOLOv5 For Players Detection

Popular object identification method YOLOv5 expands on the idea of YOLO (You Only Look Once). Compared to its predecessors, it features a more streamlined and effective architecture. The following elements make up the YOLOv5 architecture:

1. **Backbone:** A variation of the EfficientNet serves as the backbone network for YOLOv5. The backbone is in charge of extracting features at various sizes and degrees of abstraction from the input picture.

2. **Neck:** AN (Path Aggregation Network) is a neck network that is incorporated into YOLOv5. The PAN module collects low-level and high-level characteristics from multiple backbone network levels, allowing the identification of objects at various scales.

3. **Head:** The YOLOv5 head network resembles earlier YOLO versions. It comprises a series of detection layers followed by convolutional layers. The bounding box coordinates, objectness ratings, and class probabilities for the discovered items are predicted by the detection layers.

In this project, the first object detection model that we used is YOLOv5. We used this because, with its lightweight and effective architecture, YOLOv5 offers competitive accuracy while enabling quicker inference. Additionally, depending on the requirements, it provides small, medium, big, and extra-large model sizes to manage the trade-off between speed and accuracy.

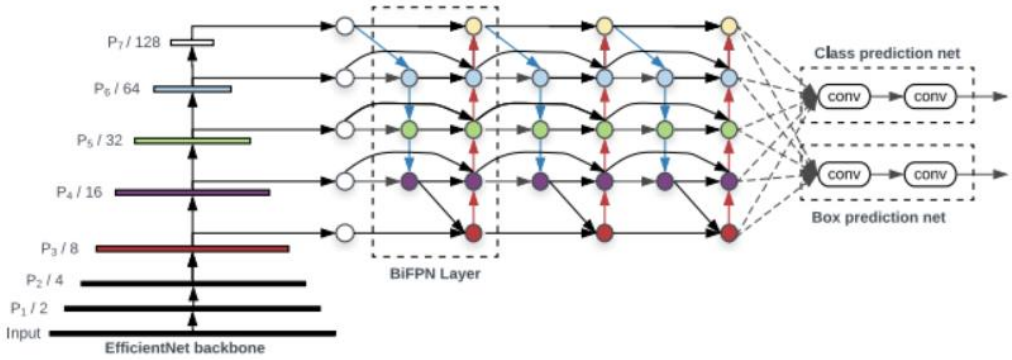


Figure 0.5 Architecture of YOLOv5

5.3 YOLOv7 For Players Detection

YOLOv7 is an object detection model that uses a deep neural network to identify objects within an image. It is designed to provide improved detection accuracy and faster inference speed.
The architecture of YOLOv7 is composed of several key components:

1. **Backbone:** YOLOv7 uses a backbone network to extract features from the input image, the same as YOLOv3 in [2]. However, YOLOv7 uses CSPDarknet as its backbone, which is an improvement over the Darknet-53 architecture used in YOLOv3. CSPDarknet uses cross-stage partial connections to improve the flow of information between the different layers of the network, which helps to reduce the number of parameters in the network while maintaining a high level of accuracy.

2. **Neck:** YOLOv7 uses a neck network to combine features from different levels of the backbone network to produce a set of features that can be used for object detection, the same as YOLOv3 in [2]. YOLOv7 uses a Spatial Pyramid Pooling (SPP) module as its neck network, while YOLOv3 [2] uses a feature pyramid network (FPN).

   FPN in YOLOv3 [2] uses top-down paths and lateral connections to combine functions from different layers of the backbone network. This approach can be slower and less efficient than the SPP module used in YOLOv7.

3. **Head:** Another component in YOLOv7 architecture is the head of YOLOv7. It is responsible for predicting the location, class, and confidence score of each object in the image. In [2] the head of YOLOv3 works the same as the YOLOv7. However, the head used in YOLOV7 is the modified version of the YOLOv5 head network, which has been shown to be more accurate and faster than the head

network used in YOLOv3 [2].

In this project, the second object detection model that we used is YOLOv. We used YOLOv7 as it provides better detection accuracy and faster inference speed than its predecessors. We used the pre-trained weights for YOLOv7 that were trained on the COCO dataset.

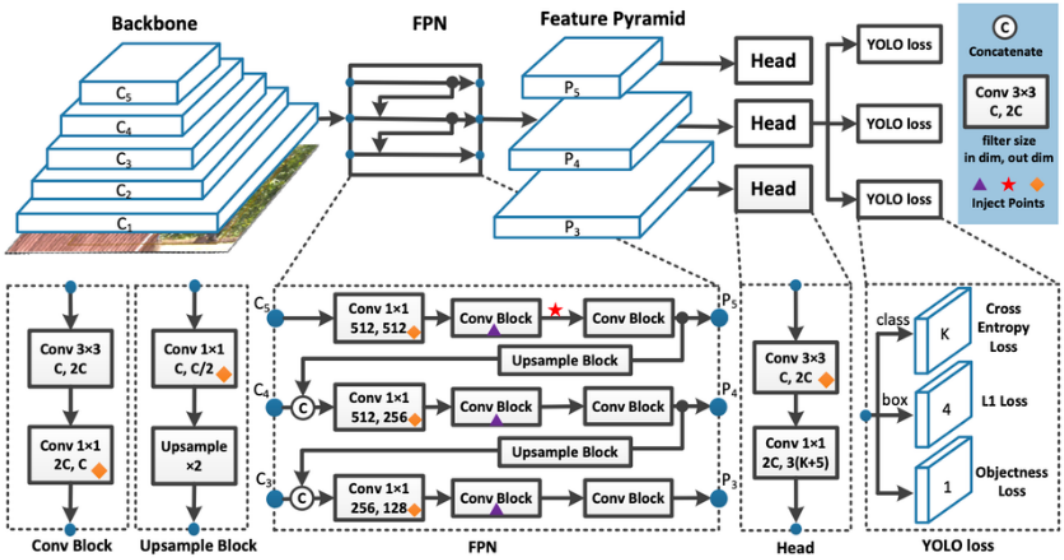


Figure 0.6 Architecture of YOLOv7

5.4 YOLOv8 For Players Detection

The most recent and cutting-edge YOLO model, YOLOv8, may be utilized for applications including object identification, picture categorization, and instance segmentation. Ultralytics, who also produced the influential YOLOv5 model that defined the industry, developed YOLOv8.

In this project, the third object detection model that we used is YOLOv8.

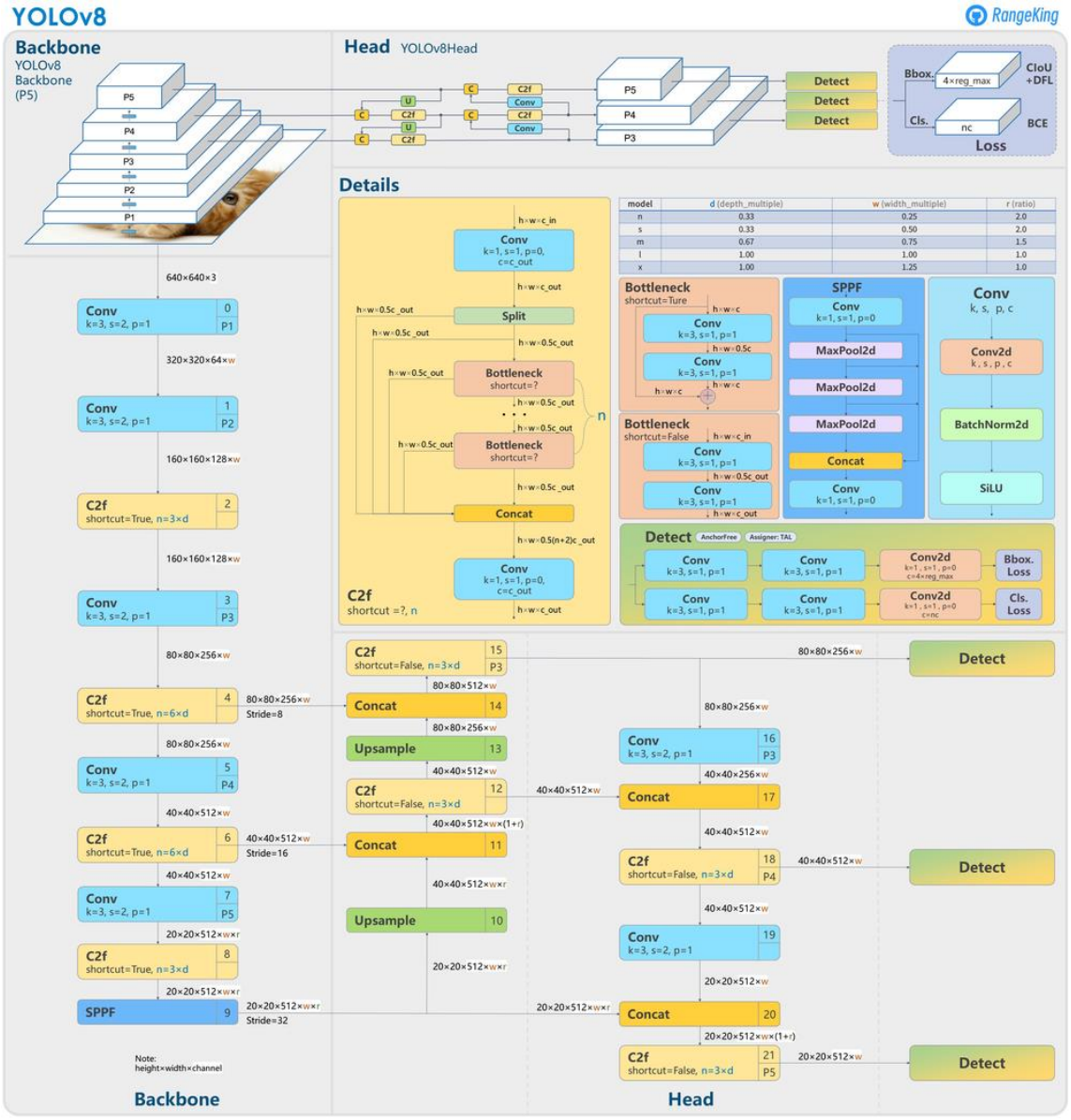


Figure 0.7 Architecture of YOLOv8

5.5 ByteTrack For Player Tracking and Tagging

A paradigm for tracking objects called ByteTrack was released in October 2021. You may give each individual a distinct ID by applying ByteTrack to the bounding box of persons that YOLOX has recognized. At the moment, ByteTrack is more effective than transformer-based tracking methods like SiamMOT.

ByteTrack uses a motion model that manages a queue called tracklets to store objects being tracked and conducts tracking and matching between bounding boxes with low confidence values to address the issue of disregarding bounding boxes with low confidence values. [8]

The matching procedure makes use of the BYTE algorithm. Using the Kalman filter, the locations of objects in the tracklets in the following frame are first predicted, and then using motion similarity, they are matched with bounding boxes that have high detection scores. Interaction over Union (IoU), which measures how much overlap there is between objects, is used to calculate the score for motion similarity. The algorithm then does a second matching. Lower confidence bounding boxes are then used to match objects in the tracklets that could not be matched. [8]

5.5.1 Comparison with Deepsort

In order to link the bounding boxes of identified persons between frames, DeepSort uses the ReID identification model. For those who could not be linked, Sort uses the forecast of bounding box movement determined by the Kalman filter. Only bounding boxes with high confidence values are used for this, though. [8]

ByteTrack tracks individuals between frames without the use of ReID, instead relying only on bounding box movement predictions. As a result, it is conceptually comparable to the DeepSort Sort phase. However, by dividing the processing into two steps—the first focusing on the bounding boxes with high confidence values and the second on the ones with low confidence values—performance has been improved. [8]

In this Project, for player tracking and to identify between players assigning unique IDs to them we used ByteTack model. To accurately track the movements of individual players over time, a state-of-the-art tracker called ByteTracker was added to the system. This tracker assigned a unique ID to each player, allowing their movements to be tracked and analyzed over time.

5.6 Parameters Calculations

We now reach our second last phase which is parameter calculation for every individual player during the football match. We worked on two parameters for every player, one is the stamina parameter and the other one is the ball possession parameter.

5.6.1 Stamina Calculation

A script was developed to calculate the stamina of each player based on the distance the player moved on the screen. The stamina value for each player was calculated by summing the distances for a chosen number of frames and comparing it to a threshold value. The stamina threshold and the number of frames required to detect a drop in stamina were customizable by the user, allowing them to fine-tune the script for their specific use case. If the distance traveled was less than the previous threshold value for three consecutive frames, the stamina value was reduced by 1%.

5.6.2 Ball Possession Calculation

A script was developed to calculate the ball possession of each player based on the number of frames the player was in possession of the ball. The proximity for ball detection was set to 30 pixels, which allowed the script to detect when a player was in possession of the ball. if the player was the first or second in terms of ball time possession his stamina is reduced by 0.5%.

5.7 Performance Calculation

Now we have two parameters: stamina and ball possession. Now these two parameters will help update the performance of every player during the match.

5.8 Display and Storage of Results

The performance values for each player were displayed in real-time next to their unique tracker ID. Additionally, the entire history of each player's stamina was saved to a CSV file for further analysis.

5.9 Main Algorithm

Step 1: First, we load the input video from the dataset

Step 2: After loading the video, then we preprocess the data using by converting it Into a list of frames, we save frames on list_frames[]

Step 3: we feed the frames into the pre-trained Yolov5 Model and get the results from it (detected features: ball, players, and referee).

Step 4: Apply a state-of-the-art tracker called ByteTrack to accurately track the

movements of individual players over time.

Step 5: Tag Every player with a unique ID so we further analyze the players from the match.

Step 6: Create a head pointer on the football to show that there is a point on the ball specifically, players are also differentiated by different colors from the referee.

Step 7: Applied stamina calculation on every individual player:

For player in list_players:

Stamina[player.id] = stamina_cal(player)

Ball_possession[player.id] = possession_cal(player)

Def stamina_cal(player):

-> we firstly set the threshold_value for a video which helps in further analysis

-> check if this player: distance traveled was less than the previous threshold value for three consecutive frames, then we reduce the stamina by 1% approx.

Def possession_cal(player):

-> We first have to set a value for checking how far a player is from ball then it can be calculated as ball that player has, which indirectly count ball possession of that player.

-> So, we set no of pixel = 30 which we tested on multiple test cases and find out that 30 pixels is the best one among all the test cases.

-> Now if a player is no of pixels near then ball => will be counted as ball possession of that player. So, if some player's ball possession is decreasing iteratively 3 times, then we reduce the performance by 0.5% which now updated.

Step 9: show the performance on the foot of every player during the match and it fluctuates for every player during the match. Convert the frames into one video and save the video for output purpose, in addition, the entire history of every player is also saved to a CSV File for further analysis.

6. Results and Evaluation metrics

In this project, to evaluate the models we used different evaluation metrics like precision,

recall and mean average precision (MAP). For mean average precision we checked two metrics that are: mAP50 and the mAP50-95.

6.1 YOLOv5 Results

We trained the YOLOv5 model for object (player, ball, referee and goalkeeper) detection. After the last epoch the mean average precision (mAP50) for the YOLOv5 model was 0.87, for mAP50-95 we got 0.551, we got precision of 0.934 and recall of 0.841.

Figure 0.8 YOLOv5 Loss

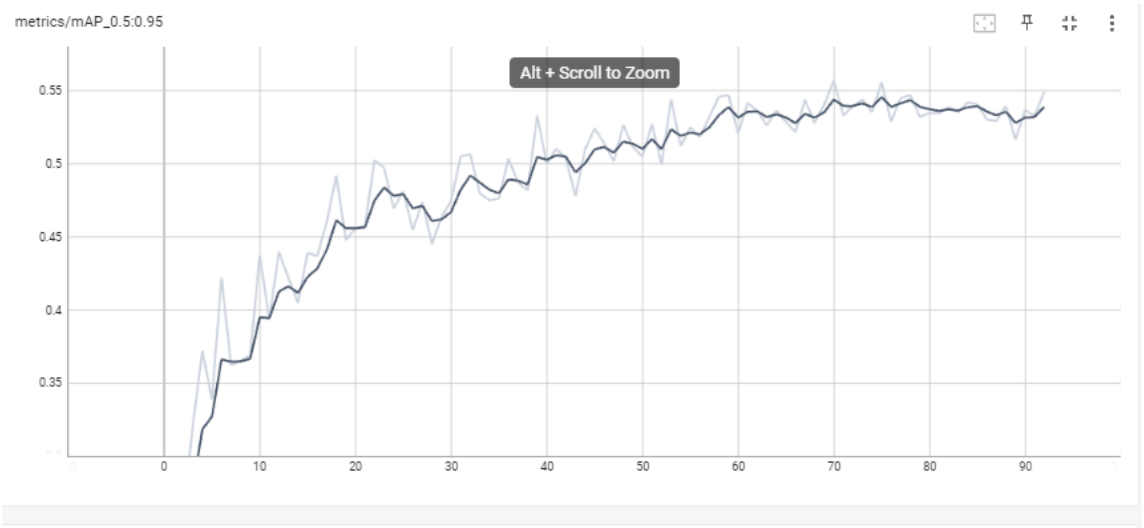


Figure 0.9 mAP50-90 metric for YOLOv5

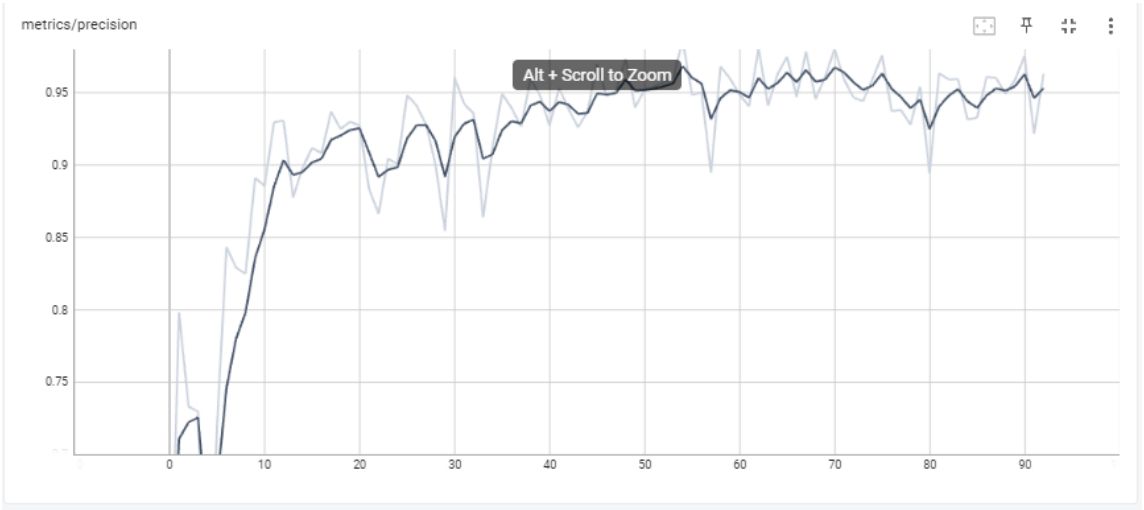


Figure 1.0 Precision metric for YOLOv5

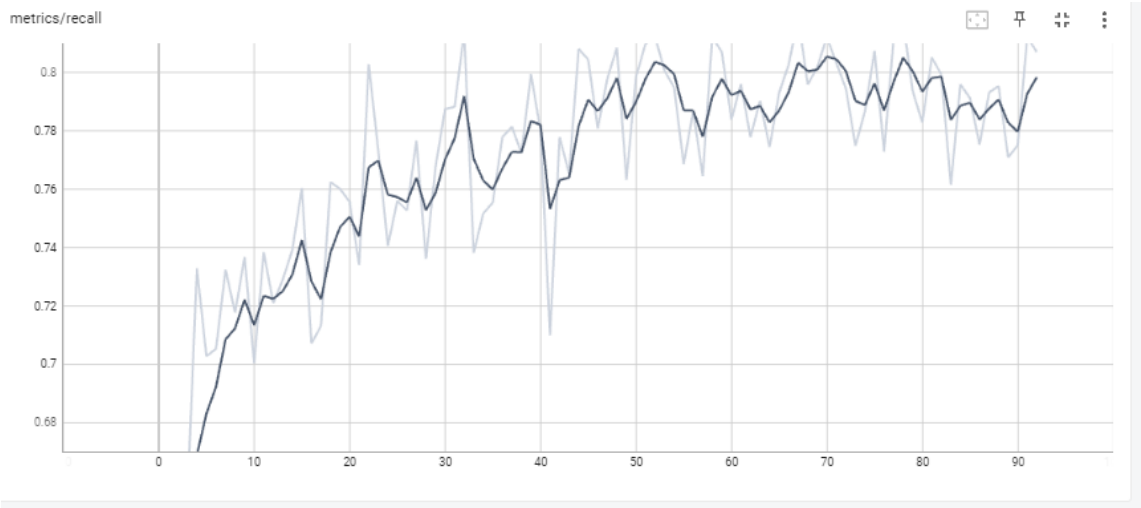


Figure 1.1 Recall metric for YOLOv5

6.2 YOLOv7 Results

We trained the YOLOv7 model for object (player, ball, referee and goalkeeper) detection. After the last epoch the mean average precision (mAP50) for the YOLOv7 model was 0.86, for mAP50-95 we got 0.551, we got precision of 0.878 and recall of 0.842.

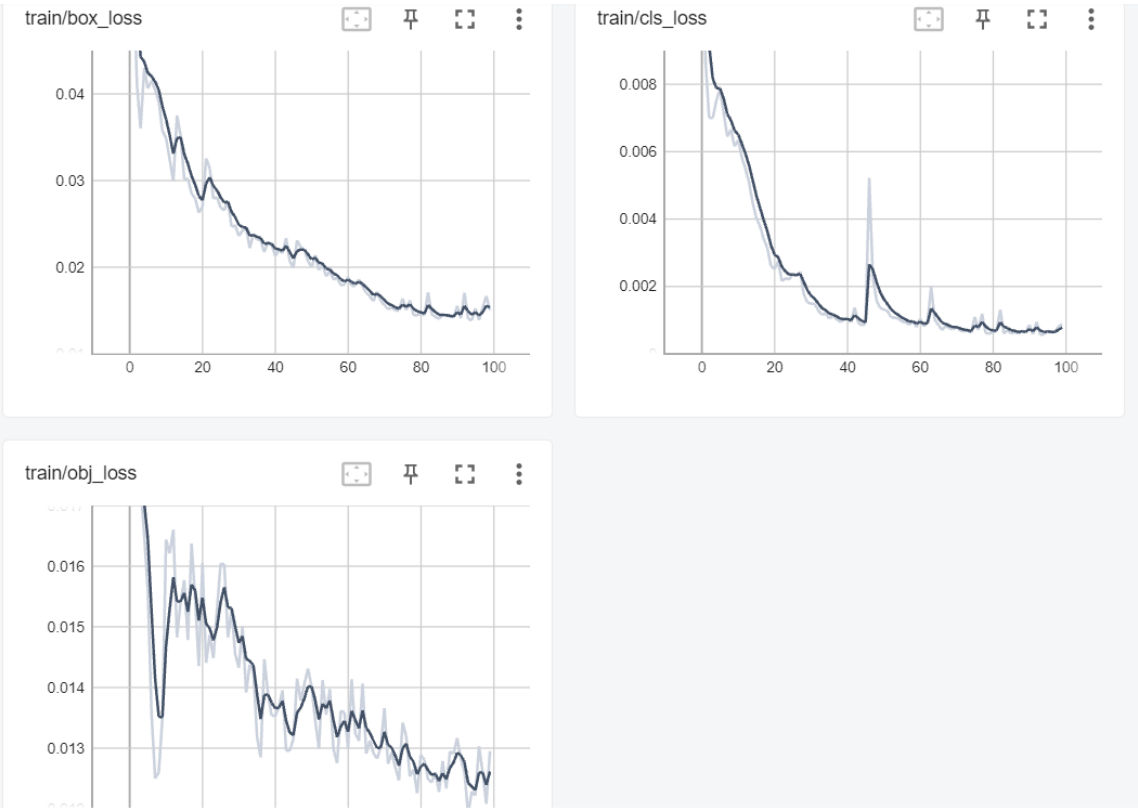


Figure 1.2 YOLOv7 Loss

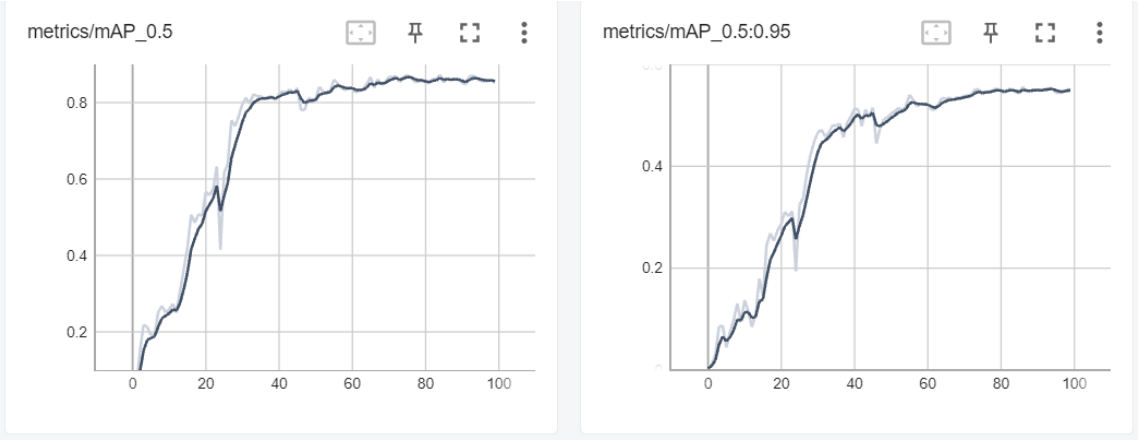


Figure 1.3 mAP50 and mAP50-95 metrics for YOLOv7

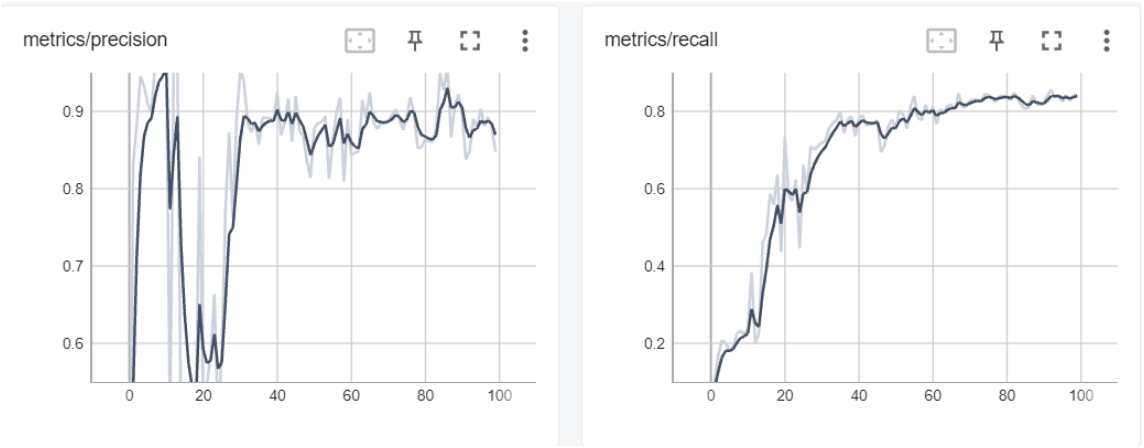


Figure 1.4 Precision and Recall metrics for YOLOv7

6.3 YOLOv8 Results

We trained the YOLOv8 model for object (player, ball, referee and goalkeeper) detection. After last epoch the mean average precision (mAP50) for the YOLOv8 model was 0.785, for mAP50-95 we got 0.554, we got precision of 0.824 and recall of 0.719.

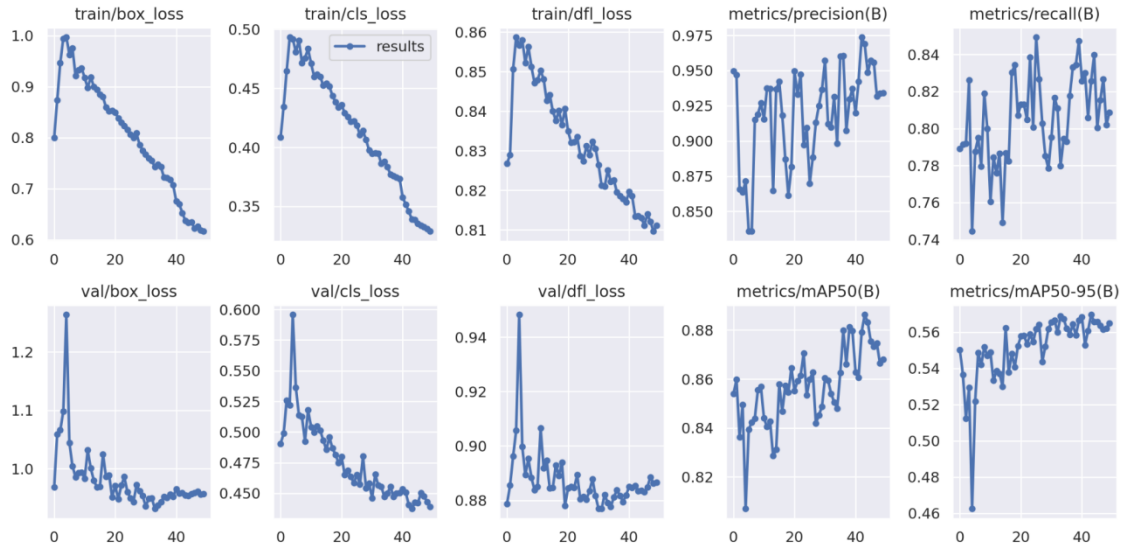


Figure 1.5 Metrics for YOLOv8

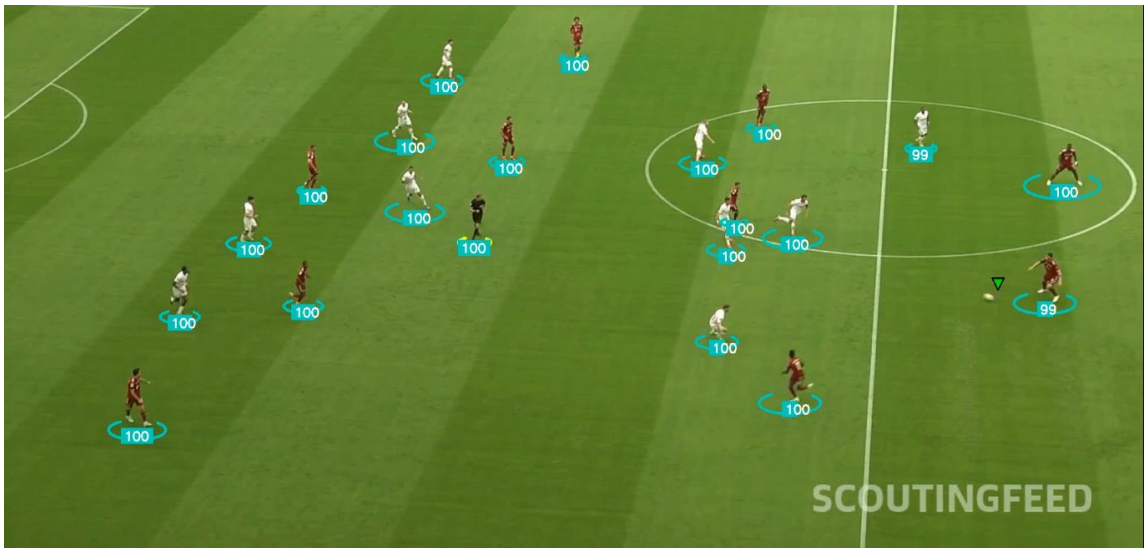
6.4 Detection Models Comparison

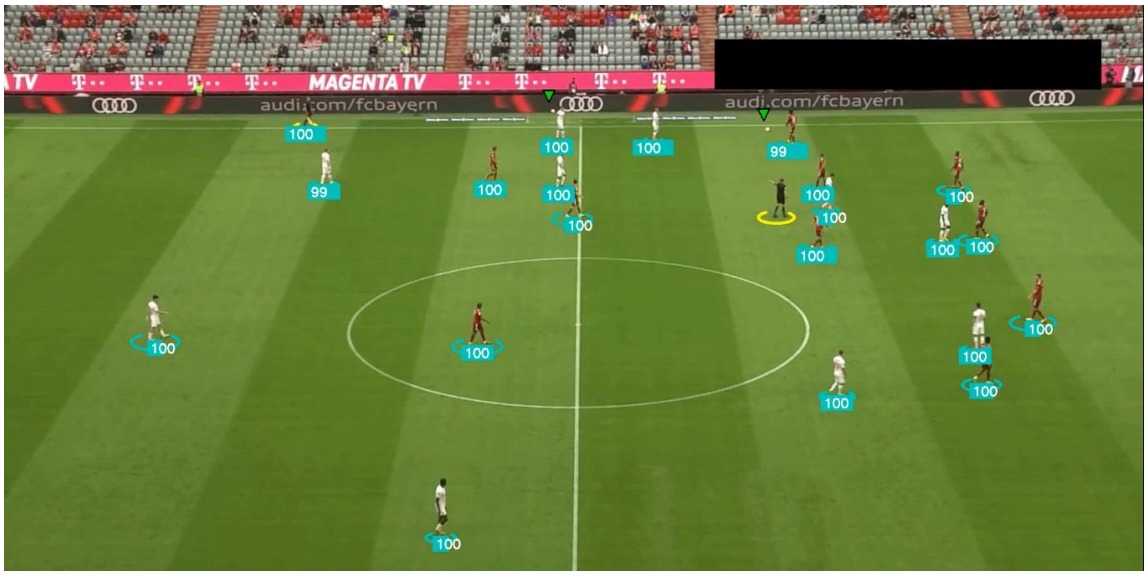
Below table 1.0 shows the comparison of the detection models. We see that YOLOv5 performed among the other two detection models.

Models	**YOLOv5**	**YOLOv7**	**YOLOv8**
mAP50	0.87	0.86	0.785
mAP50-95	0.551	0.551	0.554
Precision	0.934	0.878	0.824
Recall	0.841	0.842	0.719

Table 1.0 Models comparison

6.5 Main Results





7. Conclusion and Future Work

In this paper we propose a new architecture for object detection and recognition as well as individual player performance on the basis of two parameters: stamina and ball possession. Furthermore, we did a comparison between detection models YOLOv5, YOLOv7 and YOLOv8 and we found that YOLOv5 performed well in our dataset. ByteTrack is also a useful model for tracking with the additional technique of assigning unique IDs to the player which helps in distinguishing between the players. For player performance, two parameters were calculated: stamina and ball possession. These parameters were used to calculate the performance of every individual player using.

In the Future this project can be extended by adding additional parameters like shoot, dribbling parameter, etc. Also, one can try different ways of classifying players.

References

[1] An Improved Convolution Neural Network for Object Detection Using YOLOv2.

[2]. Naik, B. Thulasya, and Md Farukh Hashmi. "YOLOv3-SORT: detection and tracking player/ball in soccer sport." Journal of Electronic Imaging 32.1 (2022): 011003.

[3]. Gomez, Miguel-Angel, Carlos Lago-Peñas, and L. Adam Owen. "The influence of substitutions on elite soccer teams' performance." International Journal of Performance Analysis in Sport 16.2 (2016): 553-568.

[4]. Gerke, Sebastian, Antje Linnemann, and Karsten Müller. "Soccer player recognition using spatial constellation features and jersey number recognition." Computer Vision and Image Understanding 159 (2017): 105-115.

[5]. R. Theagarajan et al. "Soccer: who has the ball? Generating visual analytics and player statistics," in Proc. IEEE Conf. Comput. Vision and Pattern Recognit. Workshops, pp. 1749–1757 (2018).

[6]. Alexandre Lhoest. "Deep learning for ball tracking in football sequences". Link: http://hdl.handle.net/2268.2/9074 (2019-2020).

[7]. Barbon Junior, Sylvio, et al. "Sports action mining: Dribbling recognition in soccer." Multimedia Tools and Applications 81.3 (2021): 4341-4364.

[8].https://medium.com/axinc-ai/bytetrack-tracking-model-that-also-considers-low-accuracy-bounding-boxes-17f5ed70e00c