The Referee's Watch
A Framework for Detecting and Classifying Player Movements in American Football

Gulecha Mitesh Moore Keegan Patil Harshwardhan Shingala Jainish

Abstract

The project aims to detect and factorize NFL fields and players using advanced computer vision techniques. Field detection is a fundamental task in sports analytics, as it can provide useful information such as ball possession, player movements, and game events. The proposed approach utilizes the Hough transform algorithm to detect the field lines, which is an effective method for detecting straight lines in images. The Hough transform can accurately detect the yard lines, end zones, and sidelines, which are crucial for field lines detection.Player detection is a challenging task due to the high variability in player appearance, pose, and occlusions. To address this issue, the project employs the state-of-the-art YOLOv8 model for player detection. Field mark detection is another important task for sports analytics, as it can provide information on the position and orientation of the ball and players. To tackle this task, the project uses YOLOv5 for training and detecting field marks. The project employs 3D localization to obtain an instance of field and player detection on a 2D plane. Overall, the proposed approach demonstrates the potential of computer vision in NFL, which could have applications in sports analytics, broadcasting, and game simulations.

1. Introduction

American football is a popular and dynamic sport that requires extensive analysis and tracking to uncover valuable insights. Coaches, statisticians, and medical professionals can benefit from a better understanding of player movements and game events. However, manual analysis can be time-consuming and prone to errors. Therefore, there is a need for automated systems that can perform player detection, tracking, and placement on the field.

The Referee's Watch project is an attempt to automate the analysis and comprehension of football games using advanced computer vision techniques. The project aims to detect and factorize NFL fields and players, which can reveal critical information on player effectiveness, team tactics, and gaming. Coaches, analysts, and broadcasters may use these technologies to provide real-time analysis and feedback to players and teams while enhancing spectator viewing.

To achieve this goal, the project utilizes several state-of-the-art computer vision algorithms such as the Hough transform, YOLOv8, and YOLOv5. The Hough transform algorithm detects the field lines accurately, while YOLOv8 is a deep learning-based object detection algorithm that can detect multiple objects in an image with high accuracy and speed. The YOLOv5 is a lightweight object detection algorithm that can achieve high accuracy with fewer parameters and faster inference time. The project also employs 3D localization to obtain ground truth data for field and player detection, which is a crucial step in evaluating the proposed approach's performance.

We believe that the proposed approach demonstrates the potential of computer vision in detecting NFL fields and players, which could have significant applications in sports analytics, broadcasting, and game simulations. The Referee's Watch project provides an automated and scalable solution to the complex problem of American football game analysis and has the potential to revolutionize the way we understand and analyze the game.

2. Data

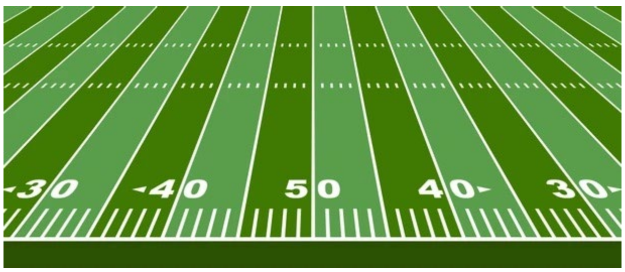
For this part, we are using dataset available on the kaggle competition: NFL 1st and Future - Impact Detection. Detect helmet impacts in videos of NFL plays. We are using the video clips in this dataset as our benchmark dataset. We have also used NFL-Player Number dataset from Kaggle for jersey number detections

2.1. NFL 1st and Future - Impact Detection Dataset

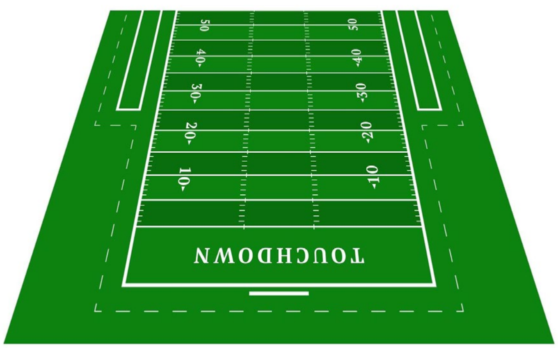
The dataset utilized in our approach for testing purposes has been obtained from the Kaggle competition. It comprises mp4 video recordings for each play, featuring two different views - sideline and endzone. The collection contains touchdown statistics as well as precise player labeling and includes details on each player's positions, speeds, jersey numbers, orientation, and contacts for other players. We intend to evaluate the performance of our model using the designated test data provided.

Files:

train/test .mp4 videos: These are videos of football plays shot from two different angles - endzone and sideline.
train_labels.csv: This file provides the helmet tracking and collision labels for the training set. It includes gameKey, playID, view, video, frame, label, and bounding box information.
sample_submission.csv: This is a sample submission file that can be used to submit predictions.
images: This is a set of still photo equivalents of the train/test videos for use making a helmet detector.
image_labels.csv: This file contains the bounding boxes corresponding to the images in the images folder.
(train/test)_player_tracking.csv: These files contain data about the position, speed, acceleration, distance traveled, orientation, and direction of the players during each play..



Sideline view (Ref: google images)



Endzone View (Ref. google images)

2.2. NFL Player Numbers Dataset

The dataset provided a collection of images that can be used to train models for jersey number recognition. The bounding box information in the image_labels.csv file can be used to associate each image with the player and play from which it was extracted. This dataset was used to detect jersey numbers from a football play.

Files:
Images: This dataset contains NFL player images with their jersey numbers. The images have a dimension of 64x64 pixels and were extracted from game plays.
Image_labels.csv: This file contains the bounding boxes used to extract each image in the dataset. The bounding boxes are specified using the left, width, top, and height values.

2.3. NFL Health & Safety - Helmet Assignment

This dataset contains additional file train and test baseline_helmets to NFL 1st and future - impact detection dataset. This dataset has been used to train model for helmet detection task.

Files:
(train/test) _baseline_helmets.csv: This file contains imperfect baseline predictions for helmet boxes.

3. Background & Related Work

3.1. "YOLOv3: Incremental Improvement" [2]

The YOLO (You Only Look Once) object detection method is revised in Joseph Redmon and Ali Farhadi's article, "An Incremental Improvement," and it performs at the cutting edge on several benchmark datasets.The article suggests a number of advancements, including feature extraction, multi-scale prediction, and enhanced training methodologies.

3.2. "Simple online and realtime tracking." [3]

A straightforward and effective online object tracking method that can monitor numerous objects in real-time is shown in this paper. The technique tracks objects by combining motion and appearance information, and it can deal with occlusions and changes in appearance. The study utilizes several tracking datasets to demonstrate the method's efficacy.

3.3. "Using CV & ML to Classify NFL Game" [4]

This passage discusses the time-consuming task that NFL coaches face when analyzing game film to prepare for upcoming matchups. The author suggests that computer vision techniques can automate the classification of game film from start to finish, including formations, player routes, and speeds. This automated player tracking system has significant implications for game planning, scouting, and player/coach evaluation. The ability to analyze player location data quickly and on a large scale will revolutionize how football coaches scout and analyze players and opposing coaches.

4. Methods

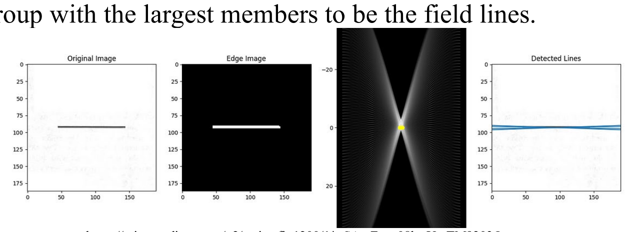
4.1. Player Detection

For player detection we used a YOLOv8 pretrained model from Ultralytics. We were unable to train our own because our data lacked bounding box data for the entire player, only helmets. We originally tried YOLOv3 and v4 but found the later models had large improvements in accuracy. YOLO models are generally used in detection in videos due to the relatively fast inference speed. While this model was very good at detecting players at around 90% accuracy, it also detected some extraneous things, like referees and field lines that it thought were sports balls.

V8:
https://drive.google.com/file/d/1_5_7TN689VT8sLGSHidVXU1t-yNfGctm/view?usp=sharing
V3:
https://drive.google.com/file/d/1vCx05xrr4Xgx12Pt6vpE5Z4D7KiHRA-C/view?usp=sharing
V4:
https://drive.google.com/file/d/1vCgtw7nIOPkXFcpC0zViPTJGQn1twUIp/view?usp=sharing

4.2. Line Detection

Line detection works using a mixture of Canny edge detection and Hough transformations in order to find straight lines. With the collection of lines found in the image, we cluster the found lines and take the median value as a non-maximal suppression. This gives one line per grouping, and with these we again cluster by the angle of each line from the picture's x axis in order to find only field lines. Because of the camera's perspective, their slopes will not be the same despite being parallel in reality, and so the clustering by the angles is necessary to group together lines whose slopes might cross the vertical axis. We take the group with the largest members to be the field lines.



https://miro.medium.com/v2/resize:fit:1200/1*pSAwZyau08leeYmTM9203Q.png

4.3. Helmet Detection

Helmet detection used YOLOv5 where we trained on top of a pretrained model, since our data had helmet bounding boxes included. The only reason we did not upgrade to v8 in this case was because we ran out of time to train the model. This model is found lacking in that when the camera zooms out and the players grow smaller it cannot detect many helmets. Despite that, when the camera is more focused in, it has fairly good accuracy and not too many false positives.

4.4. Field Number Detection

Number detection is also trained on a YOLOv8 model. Due to time constraints, we were not able to connect this part with the rest in order to succeed in 2d localization, mapping players onto the field. Some problems are noticeable in its inference, such as when a tens-place number gets cut off, it will read the whole as a 10 due to the field line being to the left of a 0 and when there are extra numbers on the field such as a design with a 100 in it on one field.

4.5. Localization

Localization is an important aspect of computer vision that enables machines to understand and interact with the visual world around them. It refers to the process of identifying and localizing objects or features within an image or video. This can involve detecting the location and boundaries of objects, identifying specific features within the image, or tracking the movement of objects over time.

Object detection is a common technique used in localization. This involves using computer vision algorithms to identify and localize specific objects within an image or video. Object detection can be used for a variety of applications, such as identifying and tracking vehicles in traffic, detecting defects in manufactured products, or identifying and tracking individuals in security footage.

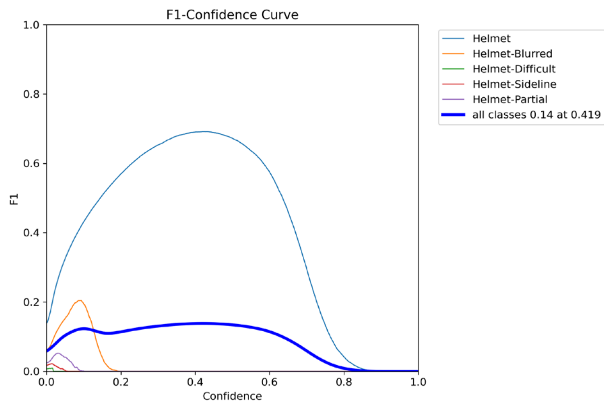
For the scope of the project, we have used YOLOv8 to detect players in the visible plane and then using Hough transform and some manual calculation we got the relative position of the players with respect to the field. The proposed model is to get the players relative co-ordinates in the field, their team status(home/away), player's position and his jersey number. Matching these features across multiple frames will help in determining their position and movement on the field.

4.6. Plans for full design

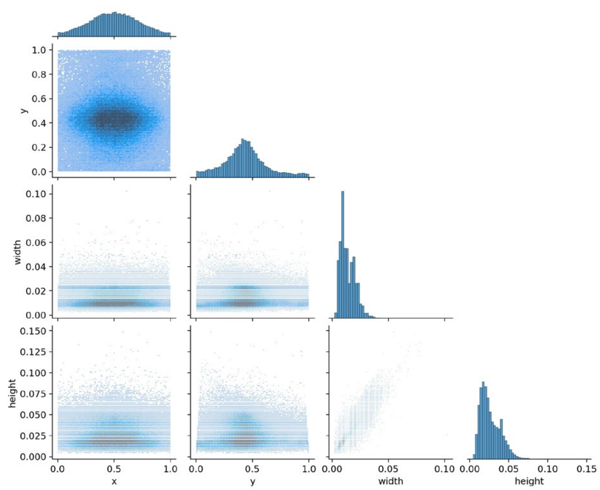
Our goal was to combine all of these parts into one system that could do 2D localization, mapping the players onto a 2D field representation, and then testing against the data in our training set for accuracy measurements. Due to a lack of time, we were unable to accomplish this, as the linchpin of field number detection came far behind the rest. Our goal was to use the bottom of each players' bounding box along with its two nearest field lines in order to figure out approximately what yard line the player was at. Without the field number detection, we could not know which line was what yard line, and so placing the players on the field was impossible.

5. Results

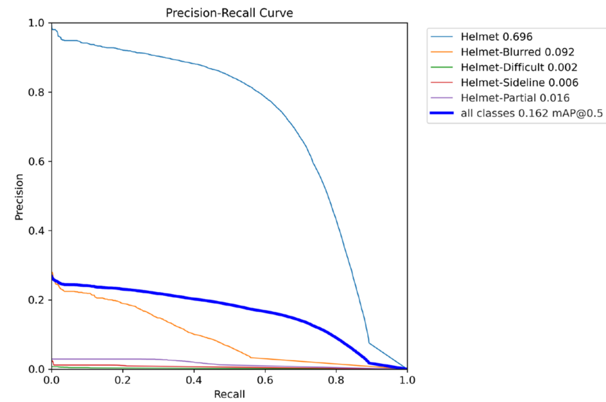
5.1. Training Results of Helmet Detection



F1 Score

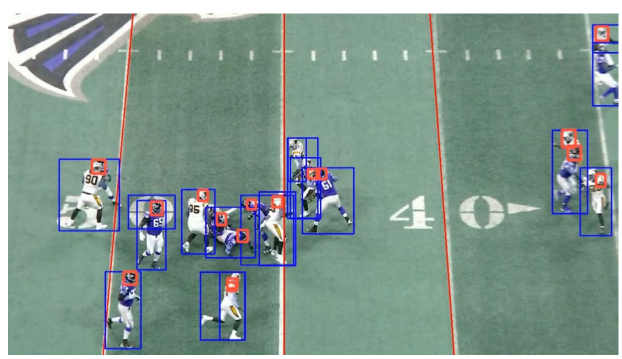


Labels Correlogram

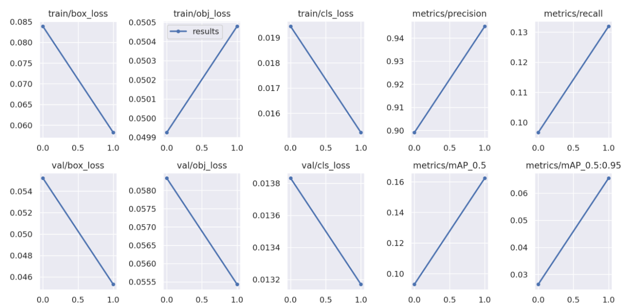


PR Curve

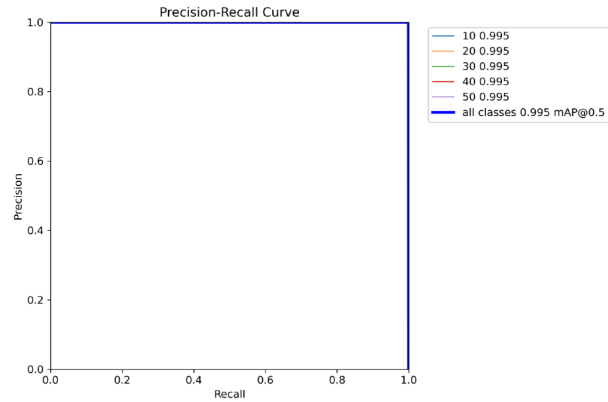
Batch Results for training Data



Final Result for Players(blue), Lines(Red) ,Helmets(red) Detection

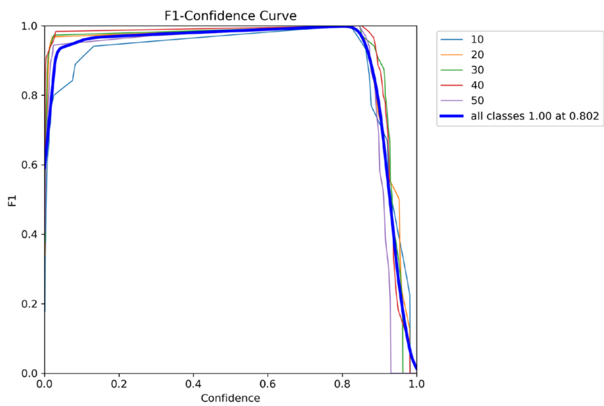


Results

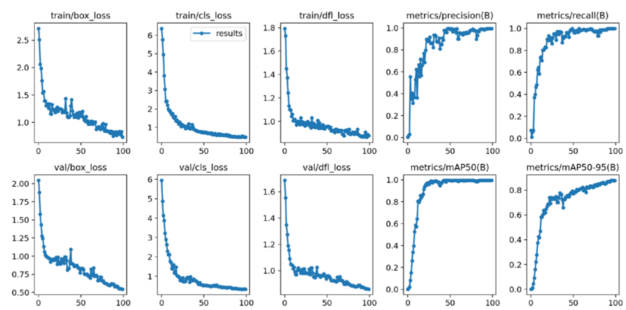


PR Curve

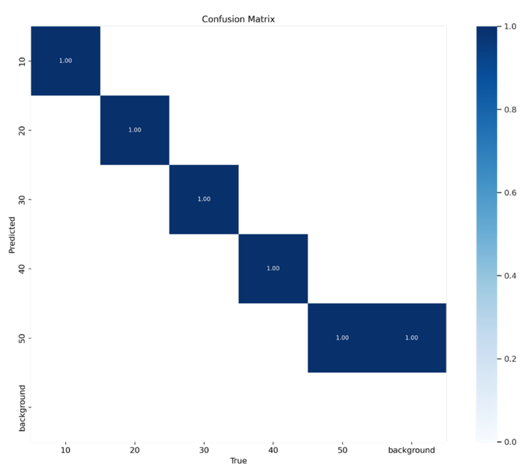
5.2. Training Results for Helmet Detection



F1 Curve

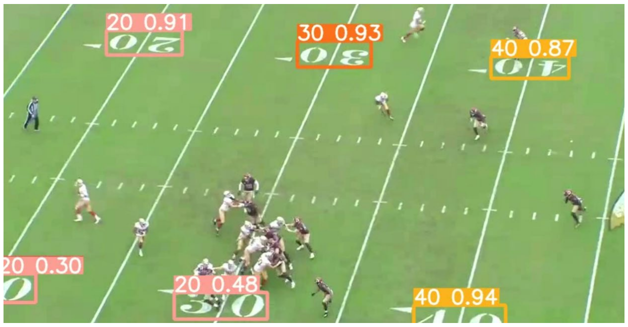


Results



Confusion Matrix

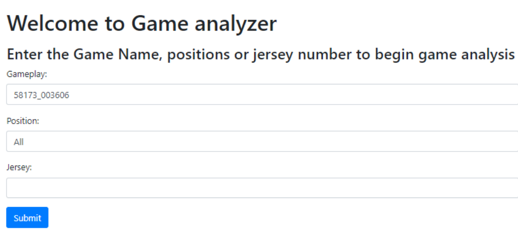
Batch Results for Training data



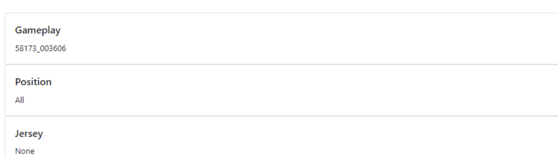
Final Result for Field Mark Detections

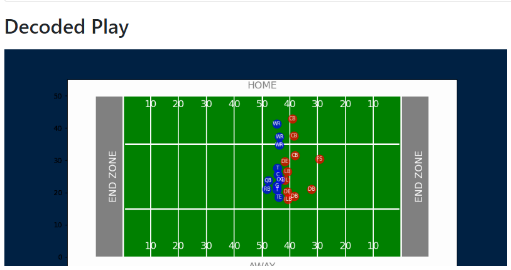
5.3. Localization

The application will then convert the positions of the players detected and tracked by the object detection model, to 2D positions on a localized plane, which represents the American Football field.



We can use the game analyzer page to enter the game name and the things that you want to track

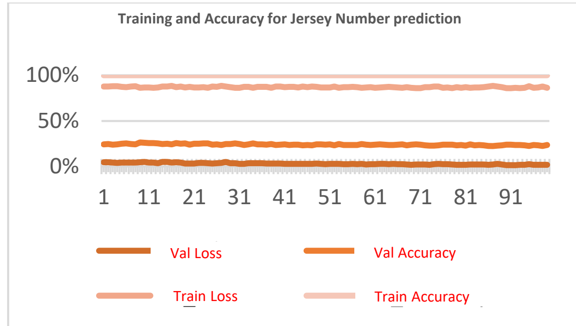




Output with filters

5.4. Jersey Number Detection

We trained a model for jersey number detection but the model didn't train well and learn at all



6. Conclusion

Our work has partially succeeded in our goal of tracking players automatically in football game footage. Though all of our parts work well individually, the late completion of some kept us from getting to the point of placing players on a 2D field representation and testing for overall accuracy with our labeled data. Our trial and error with different models and version cost of project time, and so models that should have been done earlier on were only completed very late in the project's timeline. Despite this, output images and videos show the success of the individual parts of the project. Moving forward, the first goal would be to complete localization by combining each part successfully. Furthermore, player number detection would be the next step to identify individuals in the frames. Further tasks include player speed and direction finding, which is possible with YOLOv8.

References

[1] https://github.com/ultralytics/ultralytics
[2] Redmon, Joseph and Ali Farhadi. "YOLOv3: An Incremental Improvement." ArXiv abs/1804.02767 (2018): n. Pag.
[3] Bewley, Alex, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos and Ben Upcroft. "Simple online and realtime tracking." 2016 IEEE International Conference on Image Processing (ICIP) (2016): 3464-3468.
[4] 6005a1646d2198b4b548347b_Using Computer Vision and Machine Learning to Automatically Classify NFL Game Film_2018poster.pdf (webflow.com)
[5] NFL player numbers - kaggle dataset https://www.kaggle.com/datasets/frlemarchand/nfl-player-numbers?select=train_player_numbers.csv
[6] NFL 1st and future: impact detection - kaggle dataset https://www.kaggle.com/competitions/nfl-impact-detection/data?select=train_player_tracking.csv
[7] NFL Health & Safety: Helmet Assignment - kaggle dataset https://www.kaggle.com/competitions/nfl-health-and-safety-helmet-assignment
[8] https://www.kaggle.com/code/eneszvo/yolov5-helmet-detection-train-and-inference/notebook
[9] https://github.com/danielazevedo/Football-Analytics/tree/master/player_detection
[10] https://github.com/ultralytics/ultralytics