International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 03 | Mar 2024 www.irjet.net p-ISSN: 2395-0072

Basketball Virtual Referee

Janhavi Banarase[1], Mayuri Gawande[2], Shreyash Kale[3], Shashank Mahalle[4]

[1]Student, Dept. of CSE Engineering, PRMIT&R College, Maharashtra, India
[2]Student, Dept. of CSE Engineering, PRMIT&R College, Maharashtra, India
[3]Student, Dept. of CSE Engineering, PRMIT&R College, Maharashtra, India
[4]Student, Dept. of CSE Engineering, PRMIT&R College, Maharashtra, India

---***---

Abstract - *Sports are evolving day by day and the technology that supports these sports is evolving at an exponential rate. Many sports have implemented computer vision to improve referee calls and the overall fairness of the game. Basketball is a fast-paced and dynamic sport that relies heavily on accurate and consistent officiating for fair play and a smooth game flow. However, traditional officiating methods depend on human referees, who are susceptible to errors in judgment due to factors like limited viewing angles, fatigue, and the pressure of split-second decisions. These inconsistencies can lead to frustration among players, coaches, and fans, potentially impacting the game's outcome. Virtual Basketball Referees use artificial intelligence that can distinguish between two major infractions in basketball games: double dribbling and travel. The system tracks players and the basketball in real time by utilizing the YOLOv8 object detection methodology. The system can precisely identify critical body spots on players, allowing for detection of double dribble violations, by combining pose estimate techniques with object identification. By combining these technologies, a reliable method for automating the identification of these frequent basketball infractions is offered, improving the accuracy and fairness of game officiating.*

Key Words: **You Only Look Once (YOLO), Convolutional Neural Networks (CNN), Computer Vision, Object Detection, Pose Estimation**

1. INTRODUCTION

Basketball is a very dynamic game where the movements the players make can be fast and subtle, yet fine-grained. As such the difference between an event occurring and not occurring can be small and occasionally unnoticeable, and the sequence of frames of which the dribble is done matters tremendously to determine whether a violation takes place. As such, a spatio-temporal action verification model is needed to assist referees in making decisions, as it is almost instantaneous in nature. The challenge is to achieve the balance between speed and accuracy since the violation detection has to be fast enough for real-time use. The Basketball Virtual Referee is a system that uses a YOLOv8 (You Only Look Once) machine learning model trained on annotated images to detect basketball, players in real-time. Additionally, it utilizes YOLOV8 pose estimation to detect key points on the body of the players. By combining these two techniques, the Basketball Virtual Referee is capable of accurately identifying travels and double dribbles in basketball games.

2. LITERATURE REVIEW

2.1 Computer Vision

Computer vision is a field of artificial intelligence that enables machines to interpret and understand visual information from the real world. Over the past few decades, significant advancements have been made in computer vision algorithms and technologies, leading to a wide range of applications in various industries. Computer vision aims to replicate the human visual system's ability to interpret and understand the visual world. It encompasses a broad range of tasks, including image recognition, object detection, image segmentation, and scene understanding. Recent advancements in deep learning have significantly improved the performance of computer vision systems, enabling them to achieve human-level performance on many visual recognition tasks. The key concepts in computer vision include image preprocessing, feature extraction, feature representation, and machine learning algorithms for classification and regression tasks. Convolutional Neural Networks (CNNs) have emerged as a dominant approach in computer vision, achieving state-of-the-art performance on tasks such as image classification and object detection.

2.2 OpenCV (Open Source Computer Vision Library)

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It has a comprehensive collection of algorithms and tools for a wide range of applications, making it one of the most popular libraries in the field of computer vision. OpenCV provides a rich set of features for image and video analysis, including image processing algorithms such as filtering, edge detection, and morphological operations. Feature detection and description algorithms such as SIFT, SURF, and ORB. Object detection algorithms such as Haar cascades and HOG (Histogram of Oriented Gradients). Machine learning algorithms for classification, regression, clustering, and dimensionality reduction. Deep learning capabilities through integration with frameworks like TensorFlow and PyTorch. OpenCV has made significant contributions to the field of computer vision research, including the development of new algorithms and techniques for image and video analysis. Benchmarking and evaluation of existing algorithms and datasets. Integration with other libraries and frameworks to enhance functionality and performance. OpenCV has a rich set of features, a wide range of applications, and an active community make it a valuable tool for researchers, developers, and practitioners in the field of computer vision.

2.3 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) have emerged as a powerful class of models for various tasks in computer vision, achieving state-of-the-art performance in image classification, object detection, and image segmentation. CNNs are a type of deep neural network designed to process and analyze visual data. They are inspired by the organization of the animal visual cortex and consist of multiple layers of neurons that learn hierarchical representations of the input data. The basic architecture of a CNN consists of convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to the input image to extract features while pooling layers reduce the spatial dimensions of the features. Fully connected layers are used for classification or regression tasks. CNNs can have multiple layers of each type, with deeper networks often achieving better performance but requiring more computational resources.

2.4 YOLOv8 (You Only Look Once)

YOLO (You Only Look Once) is an object detection algorithm that utilizes a single neural network to simultaneously predict bounding boxes along with class probabilities for each object in an image. YOLOv8 is the latest version of the YOLO (You Only Look Once) family of object detection models, developed by Ultralytics, which outperforms previous versions by introducing various modifications such as spatial attention, feature fusion, and context aggregation modules. These improvements result in faster and more accurate object detection, making YOLOv8 one of the key object detection algorithms in the field. YOLOv8 introduces new features and optimizations that make it an ideal choice for various object detection tasks in a wide range of applications.

YOLOv8 is a state-of-the-art deep learning model designed for real-time object detection in computer vision applications. It leverages advanced architecture and algorithms to enable accurate and efficient object detection. A modified version of the CSPDarknet53 architecture forms the backbone of YOLOv8. This architecture consists of 53 convolutional layers and employs cross-stage partial connections to improve information flow between the different layers. The head of YOLOv8 consists of multiple convolutional layers followed by a series of fully connected layers. These layers are responsible for predicting bounding boxes, confidence scores, and class probabilities for the objects detected in an image. One of the key features of YOLOv8 is the use of a self-attention mechanism in the head of the network. This mechanism allows the model to focus on different parts of the image and adjust the importance of different features based on their relevance to the task. Another important feature of YOLOv8 is its ability to perform multi-scaled object detection. The model utilizes a feature pyramid network to detect objects of different sizes and scales within an image. This feature pyramid network consists of multiple layers that detect objects at different scales, allowing the model to detect large and small objects within an image. Here's a brief overview of the YOLOv8 architecture:

Backbone: YOLOv8 uses a backbone network, typically based on the CSPDarknet53 architecture. This backbone is responsible for extracting features from the input image.

Neck: YOLOv8 features a neck module that further refines the features extracted by the backbone. This module helps in improving the accuracy of object detection.

Detection Head: The detection head of YOLOv8 is responsible for predicting bounding boxes and class probabilities for objects in the input image. It typically consists of several convolutional layers.

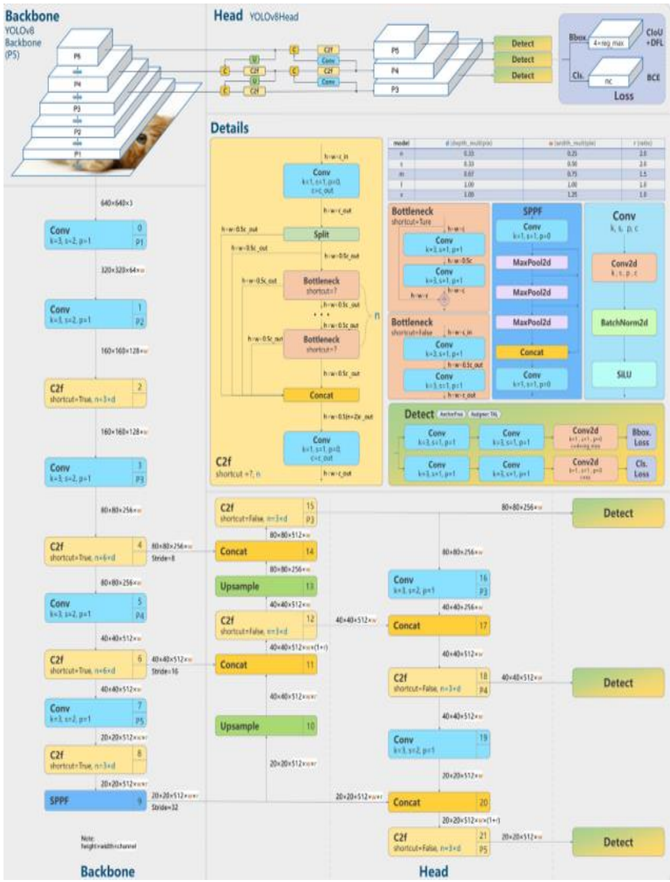
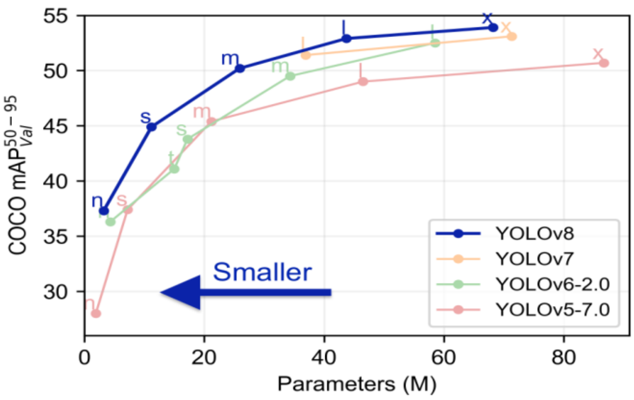


Fig-1: YOLOv8 Architecture

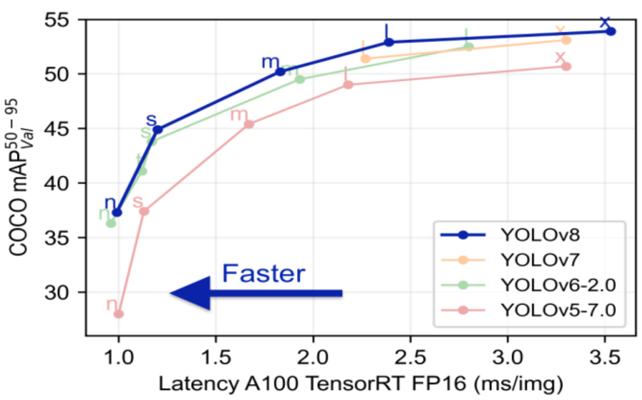
YOLO Layer: YOLOv8 utilizes a YOLO layer to generate the final output predictions. This layer processes the detections from the detection head and outputs the bounding boxes along with their associated class labels and confidence scores.

Anchor Boxes: YOLOv8 uses anchor boxes to improve the detection of objects at different scales. These anchor boxes are predefined bounding boxes of various sizes and aspect ratios that are used to predict object locations.

Loss Function: The model uses a combination of loss functions, including localization loss, confidence loss, and classification loss, to train the network and improve its accuracy in detecting objects.



Graph-1: COCOmAP$^{50-95}$ vs Parameters for different versions of YOLO



Graph2: COCOmAP$^{50-95}$ vs Latency A100 TensorRT FP16 (ms/img) for different versions of YOLO

3. METHODOLOGY

Object Detection and Tracking: Detect basketball, players, and rim using YOLOv8 object detection. Track their movements in real time.

Pose Estimation: Estimate player poses to identify key body points. Use these points for a double dribble.

Double Dribble Detection: Use pose estimation to detect when a player double-dribbles.

Foul Detection & Feedback: Implement logic to detect fouls based on player interactions and movements. Provides real-time feedback on detected violations and fouls. Display warnings or penalties on the screen as necessary.

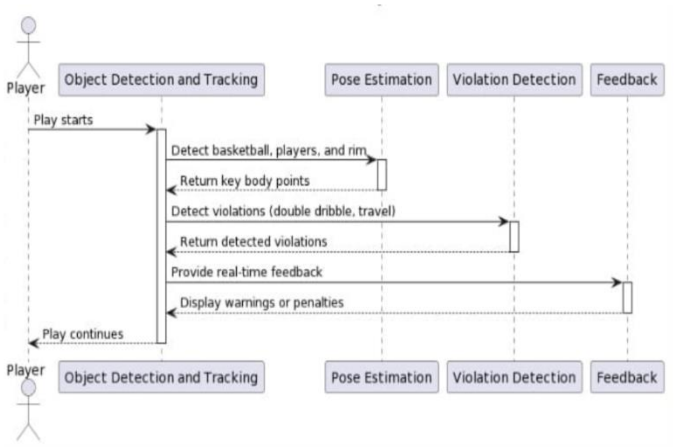


Fig-2: Basketball Virtual Referee System Sequence Diagram

3.1 Basketball and Player Detection

The first step in the Basketball Virtual Referee's system is basketball detection and player detection. A pre-trained YOLOv8 model specifically designed for object detection tasks is loaded. This model has been trained on a dataset of images containing basketball games, allowing it to recognize players and basketballs within the frames. The video frame is fed into the YOLOv8 model. The model analyzes the frame and identifies objects that resemble players and basketballs based on the learned patterns from the training data. For each detected object, the model outputs a bounding box indicating the object's location in the frame and a confidence score representing the certainty of the detection being a player or basketball. The system generates a list of bounding boxes along with their corresponding confidence scores for each detected player and basketball in the frame. This information provides the foundation for subsequent steps in the virtual referee system, such as tracking the movement of

Fig-3: Basketball and Player Detection

players and the basketball throughout the game. Analyzing player positions to identify potential fouls or violations.

3.2 Pose Estimation

Pose estimation refers to the process of determining the position and orientation of an object or a person in a given space, typically from sensor data. In the context of computer vision, pose estimation often refers to estimating the pose of a human body. In a basketball virtual referee system, pose estimation plays a crucial role in tracking the movements of players and the basketball to analyze the game and make referee-like decisions. In a basketball game, player movements such as traveling (moving without dribbling the ball) and double dribbling (dribbling the ball with both hands or restarting dribbling after a pause) are considered violations. Detecting these violations accurately can be challenging, but it's crucial for fair gameplay. Once the players are detected, the system uses YOLO pose estimation to identify and track key points on the body of the players. These key points correspond to body joints such as the ankles, knees, hips, elbows, and wrists. For example, if a player's hand key points move in a way that suggests they are dribbling the ball, but their foot key points indicate they are also moving, this could be flagged as a potential traveling violation. There are typically 16 key points of the human body commonly used in pose estimation and related applications. These points are often referred to as "key points" or "joints" and are used to represent the pose or position of a person. There is a list of the 16 key points in Fig5. These key points are used in various computer vision tasks, such as human pose estimation, action recognition, and gesture recognition, to understand and analyze human movements and poses.

Index	Key point
0	Nose
1	Left-eye
2	Right-eye
3	Left-ear
4	Right-ear
5	Left-shoulder
6	Right-shoulder
7	Left-elbow
8	Right-elbow
9	Left-wrist
10	Right-wrist
11	Left-hip
12	Right-hip
13	Left-knee
14	Right-knee
15	Left-ankle
16	Right-ankle

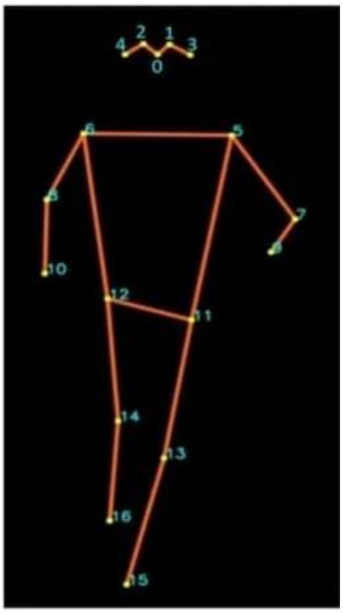


Fig-4: Keypoints of Human body

Fig -5: Players and Keypoints Detection

3.3 Double Dribble Detection

By analyzing the movement of the basketball and the player's key points, the system can determine when a player is dribbling the ball. By tracking the position and movement of the player's key points and analyzing the interactions with the basketball, the system can detect situations where a player dribbles the ball, stops, and then starts dribbling again without another player touching or possessing the ball in the meantime. a double dribble is a violation that occurs when a player illegally resumes dribbling. This can happen in several ways:

- A player ends their dribble by catching or causing the ball to come to rest in one or both hands and then dribbles it again with one hand.

- A player touches the ball before the ball hits the ground.
- A player dribbles the ball with two hands simultaneously.
- A player stops dribbling and then starts again without passing or shooting the ball.

3.4 Real-Time Feedback:

The Basketball virtual Referee provides real-time feedback on double dribble violations during basketball games. It highlights the detected violations on the video feed, making it easy for referees or users to identify and assess the accuracy of the system's decisions. Additionally, the system can generate logs or alerts to record detected violations for further analysis or review.

4. RESULT

4.1 User Interface using Streamlit: We've developed a Basketball Virtual Referee system using the Streamlit web application framework that accepts input in the form of video, or webcam. OpenCV is utilized for image processing. When the "Predict with YOLOv8" button is selected (fig. 6), our YOLOv8 model runs in the background and detects objects within the frame.

There are 2 modes:

- Video Mode
- Webcam Mode

4.1.1 Video Mode

The video mode accepts various formats such as MP4, AVI, MKV, and MPEG4. We used a video as input (as seen in fig. 6), By clicking "Predict Using YoloV8-Pose" activated the Yolov8 model in the background, resulting in the detection violation called double dribble (as shown in the table in fig. 6). The video had a frame rate of 25 fps. Users can upload basketball gameplay videos directly. The uploaded video will be displayed within the application, allowing users to play the video as needed. The application utilizes computer vision algorithms to analyze the video frames and detect instances of double dribble violations. This detection is done by identifying patterns of dribbling and hand movements typical of double dribble violations. When a double-dribble violation is detected, the application highlights the corresponding segment of the video where the violation occurred, making it easy for users to identify and review.

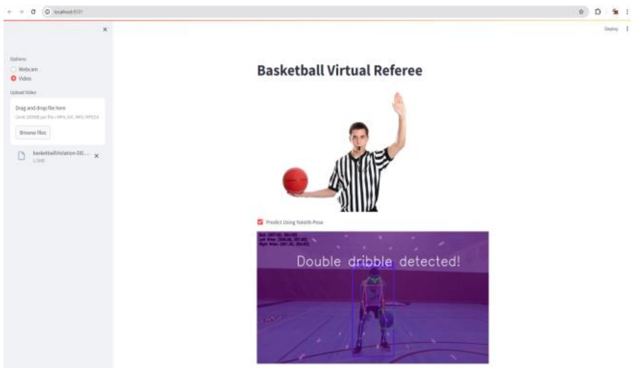
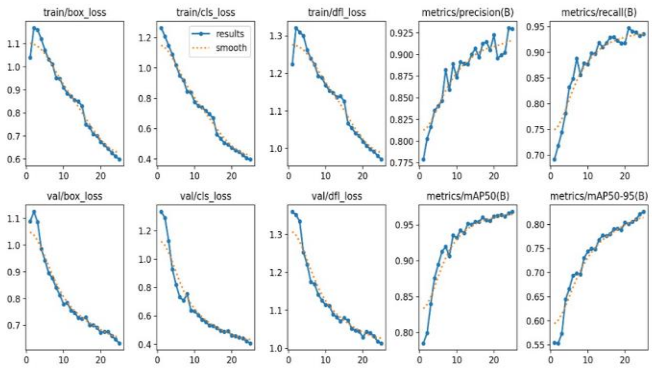


Fig -6:Video Mode

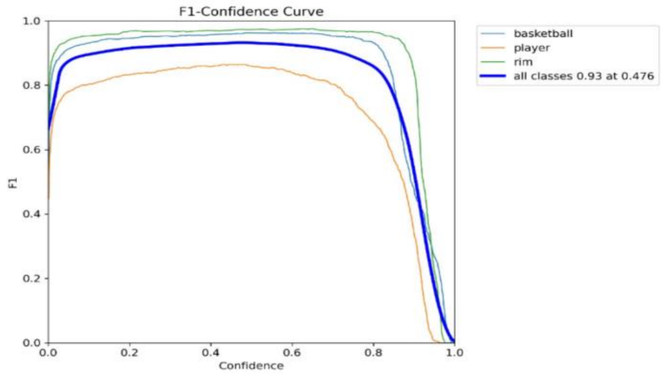
4.2 Model Evalution

The Graph-3 shows graphs of training results of object detection using YOLOv8. Train/val box_loss: the bounding box loss of the training dataset or validation dataset; the smaller the box is, the more accurate. Train/val cls_loss: train or validation is speculated to be the mean of classification loss, and the smaller the classification is, the more accurate. The F1 confidence curve(as seen in Graph 4) represents the relationship between the F1 score and the confidence threshold used to make predictions. The F1 score is a metric that combines precision and recall, providing a



Graph-3: Graph of YOLOv8 Object Detection training results

balance between these two measures. The confidence threshold is the threshold above which a model predicts the positive class. The confidence level of a model detecting a basketball increases as the confidence level increases. The confidence level increases until it reaches 0.93 at a confidence threshold of 0.476. The curve shows the precision-recall for all classes of objects, with a mAP (mean Average Precision) of 0.93 at an IoU (Intersection over Union) threshold of 0.476.



Graph-4: F1 -Confidence Curve for basketball, player and rim detection

In the normalized confusion matrix(as seen in Fig-7), the counts of true positive, true negative, false positive, and false negative predictions are converted into rates or percentages, providing a more interpretable view of the classifier's performance. The normalized confusion matrix shows that the classifier has correctly predicted 97% of the images of basketball, and 93% of the images of players that belong to the positive class.

Graph -5 shows the Pose Estimation training results of the YOLOv8 model. Train/val box_loss: the bounding box loss of the training dataset or validation dataset; the smaller the box is, the more accurate. Train/val cls_loss: train or validation is speculated to be the mean of classification loss, and the smaller the classification is, the more accurate. The blue line on Graph 6 represents the F1 score for identifying a "Basketball Player."The F1 score remains at 1.0 (perfect) until the confidence threshold reaches approximately 0.908. Beyond this threshold, the F1 score drops sharply to zero.It states that for "all classes," the F1 score is 1.00 when the confidence level is up to 0.908.

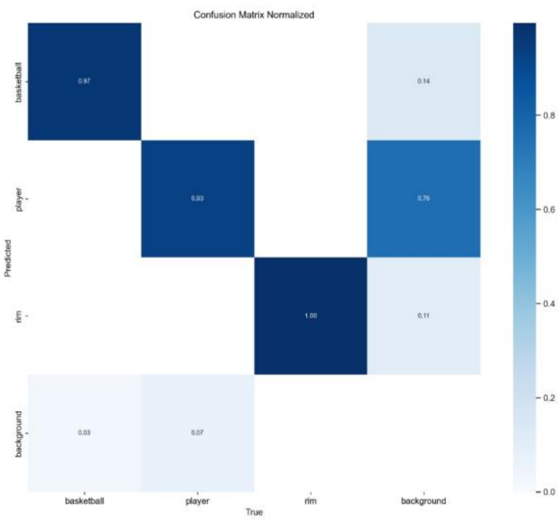
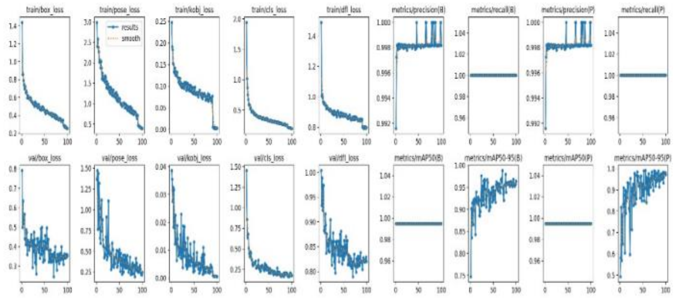
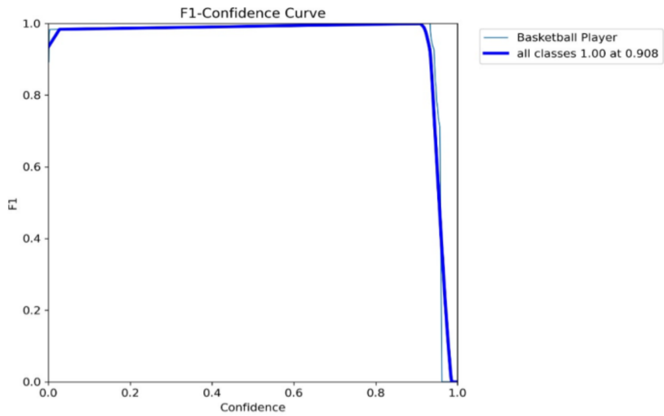


Fig-7: Normalized Confusion Matrix for object detection



Graph-5 Graph of YOLOv8 Pose Estimation training results



Graph-6: F1 -Confidence Curve for basketball player pose estimation

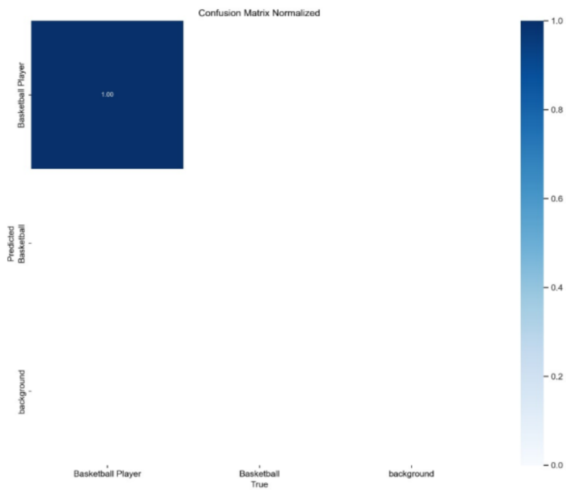


Fig-8: Normalized Confusion Matrix Pose Estimation

5. CONCLUSIONS

The development of the AI Basketball Virtual Referee represents a significant advancement in the intersection of computer vision and sports technology. Through the utilization of Python programming and cutting-edge computer vision algorithms, our project aimed to tackle the intricate task of detecting fouls and violations in basketball games. Throughout the process of this project, we have successfully designed and implemented a robust system capable of real-time foul detection with a high degree of accuracy. By leveraging computer vision techniques, our AI referee is capable of analyzing live video feeds of basketball games, swiftly identifying fouls, and alerting officials accordingly. This not only streamlines the officiating process but also enhances the fairness and integrity of the game. Looking ahead, the AI Basketball Virtual Referee holds immense potential for further refinement and expansion. With continued research and development, our system could evolve to encompass a broader range of fouls and violations, thereby enhancing its utility and impact within the realm of basketball officiating.

REFERENCES

[1] L. Jing, K. Kitahama, M. Yamata and Y. Hoshino, ”Basketball Foul Model and Judgment System Proposal”, 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), Kyoto, Japan, 2021, pp. 988-991.

[2] Zemgulys, Julius Raudonis, Vidas Maskeli¯ unas, Rytis Damaˇ seviˇcius, Robertas, ”Recognition of basketball referee signals from real-time videos”, Journal of Ambient Intelligence and Humanized Computing,2020, pp. 979-991

[3] W. Fang, L. Wang and P. Ren, ”Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments,” in IEEE Access, vol. 8, pp. 1935-1944, 2020, doi: 10.1109/ACCESS.2019.2961959.

[4] Weikun Li ”Research on the application of artificial intelligence in basketball”, Proc. SPIE 12156, International Conference on Algorithms, High-Performance Computing, and Artificial Intelligence (AHPCAI 2021), 1215605 (21 December 2021)

[5] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, ”A brief introduction to OpenCV,” 2012 Proceedings of the 35th International Convention MIPRO, Opatija, Croatia, 2012, pp. 1725-1730.

[6] Chakraborty, Bodhisattwa Meher, Sukadev, ”A real-time trajectory-based ball detection-and-tracking framework for basketball video”, Journal of Optics,2013, pp.156-170.