Dean&Francis

ISSN 2959-6157

# Basic Soccer Movement Detection Based on YOLO v8

**Lin Wu**

School of Computer Science, Beijing University of Posts and Telecommunications (BUPT), Beijing, China
h3615248979@outlook.com

**Abstract:**

The paper defines object detection and discusses the advantages and disadvantages of one-stage and two-stage detectors. A literature review demonstrates the excellent performance of You Only Look Once Version 8 (YOLO v8) in object detection and its application in soccer. Based on this, the paper uses YOLO v8 as a pre-trained model and leverages its object detection capabilities to detect basic soccer moves in images. With a manually collected and labeled training set, YOLO v8 is pre-trained on four moves: penalty kicks, shoots, dribbling and headers. Experiment results were obtained with a validation set. The results show that at a confidence level of 0.6, the average precision, the recall and the F1 score across all classes are approximately 0.78, 0.65 and 0.7 respectively. At the intersection over union ratio (IoU) threshold of 0.5, the mean average precision (mAP) across all classes is 78.5%, demonstrating good performance of the model. Future research may include human body modeling for soccer moves and human pose estimation to improve the detection accuracy of the model.

**Keywords:** YOLO v8; Object Detection; Artificial Intelligence; Soccer; Basic Movement Detection.

## 1. Introduction

With the development of artificial intelligence, object detection, a critical component of computer vision, has been applied to a wide range of industries. Amit et al. define the technology as detecting instances of objects from one or several classes in an image [1]. According to Goswami et al., object detection identifies objects in images and videos with complex backgrounds and provides accurate results through machine learning or deep learning [2].

Currently, popular object detection tools include two-stage detectors such as Faster Regions with Convolutional Neural Network (Faster R-CNN) and Mask Regions with Convolutional Neural Network (Mask R-CNN), and one-stage detectors such as You Only Look Once (YOLO). The two-stage methodology first generates region proposals and then performs classification and regression. Compared to two-stage detectors, one-stage detectors are faster while maintaining comparable accuracy. Swathi et al. compare the advantages and disadvantages of the two methods and conclude that one-stage detectors such as YOLO v8 currently outperform Faster R-CNN and Single

Shot MultiBox Detector (SSD), especially in reduced inference time without compromised accuracy[3].

From the first version released in 2015 to YOLO v8 in 2023, Vijayakumar et al. review the object detection function of different versions of YOLO and discuss the contributions of each version to different application scenarios. The review also summarizes three major features of YOLO v8: multiple backbones, data augmentation and versatile training [4]. The review believes that YOLO v8 has solved the long-standing problem of occlusion in object detection [5].

Object detection has also been applied to the sport of soccer. For example, Utsumie et al. used the technology to perform identification and tracking in videos to better describe soccer games [6]. Markappa et al. review the detection of soccer players, balls, goalkeepers and referees through machine learning and experimentally prove the superiority of YOLO v8 and YOLO v9 over previous versions [7]. Perkasa et al. conduct experiments to demonstrate that YOLO v8 performs better than its predecessors, especially in terms of precision, recall and F1 score, with an overall precision of 87.4% at mean Average Precision (IoU=0.5), mAP@0.5 in short. YOLO v8 outperforms YOLO v9 in object recognition at different confidence levels [8].

Undoubtedly, object detection has attained achievements after a long period of development. For example, Huang et al. study the training method to detect human movements and numbers, but research on human movement detection in specialized fields remains scarce [9]. Fang et al. propose that among various models, YOLO boasts faster training speed and higher accuracy with its training on Graphic Processing Units (GPU) [10]. Following the existing research, the paper conducts experiments based on YOLO v8 as a pre-trained model. With manually collected and labeled datasets, the experiment identifies basic movements in soccer images with object detection.

## 2. Research Methodology

### 2.1 Constructing Experiment Framework Based on YOLO v8

Based on the Anaconda platform, the paper used Pip commands to install the PyTorch AI learning framework, and integrated Compute Unified Device Architecture (CUDA) and CUDA Deep Neural Network Library(cuDNN) supported by PyTorch to optimize the deep learning framework. Then, the deep learning framework was configured by adding pytorch_cp312 interpreter to the pycharmproject1 of PyCharm. To directly use the preprocessed model of YOLO v8, the paper installed the Ultralytics library to include the model of YOLO v8n.pt.

### 2.2 Data Collection and Preprocessing

Images were collected from the Internet due to the lack of available datasets related to soccer moves. By using Baidu Image Library and capturing screenshots from videos, thousands of images from soccer games were collected. The images were screened based on their clarity and integrity, and classified into different moves: penalty kicks, shoots, dribbling and headers. Rebuffi S.A. et a l. prove that data augmentation can improve robustness [11]. Hence, Python scripts were used to adjust the image resolution, and later zoom and crop the images to train the model at different scales. As zooming and cropping destroyed the integrity of some images, a second screening was conducted to eliminate the problematic images.

### 2.3 The Establishment of Datasets

As Internet images had not been labeled, Anylabeling was used to label the images manually. The object move in each image was labeled with a rectangular box, and the corresponding class was recorded in the neighboring column. After labeling, the tool automatically generated a JSON file for each image, and the labels were stored in a nested format. In YOLO v8, the labels should be stored in TXT format with the class ID, center point coordinates, width and height in the same line. Therefore, Python was used to convert JSON files into the desired TXT files.

The YOLO v8 datasets were divided into the training set (80%) and the validation set (20%). The datasets were stored under primary directories named “train” and “val”, and secondary directories named “image” and “label”.

In order to navigate the YOLO v8 model and enable it to detect the object dataset, we generated a corresponding YAML file to record the absolute paths of the training set and the validation set, and labeled the movement classes to be trained.

### 2.4 Model Training

Regarding the basic principles of object detection, YOLOv8 utilizes a brand-new network architecture consisting of three components: Backbone, Neck, and Head. The Backbone typically employs efficient convolutional neural networks, such as EfficientNet or CSPNet. The Neck utilizes a Feature Pyramid Network (FPN) structure to integrate features at different scales. The Head is responsible for predicting the bounding box and class of the objects. YOLO v8 introduces an array of data augmentation techniques, such as Mosaic and MixUp to improve the generalization capability. Furthermore, YOLO v8 employs an anchor-free mechanism to directly predict the

ISSN 2959-6157

center point, width, and height of objects without relying on predefined anchors.

The simplified network of object detection in YOLO v8 is as shown in Fig.1 below.

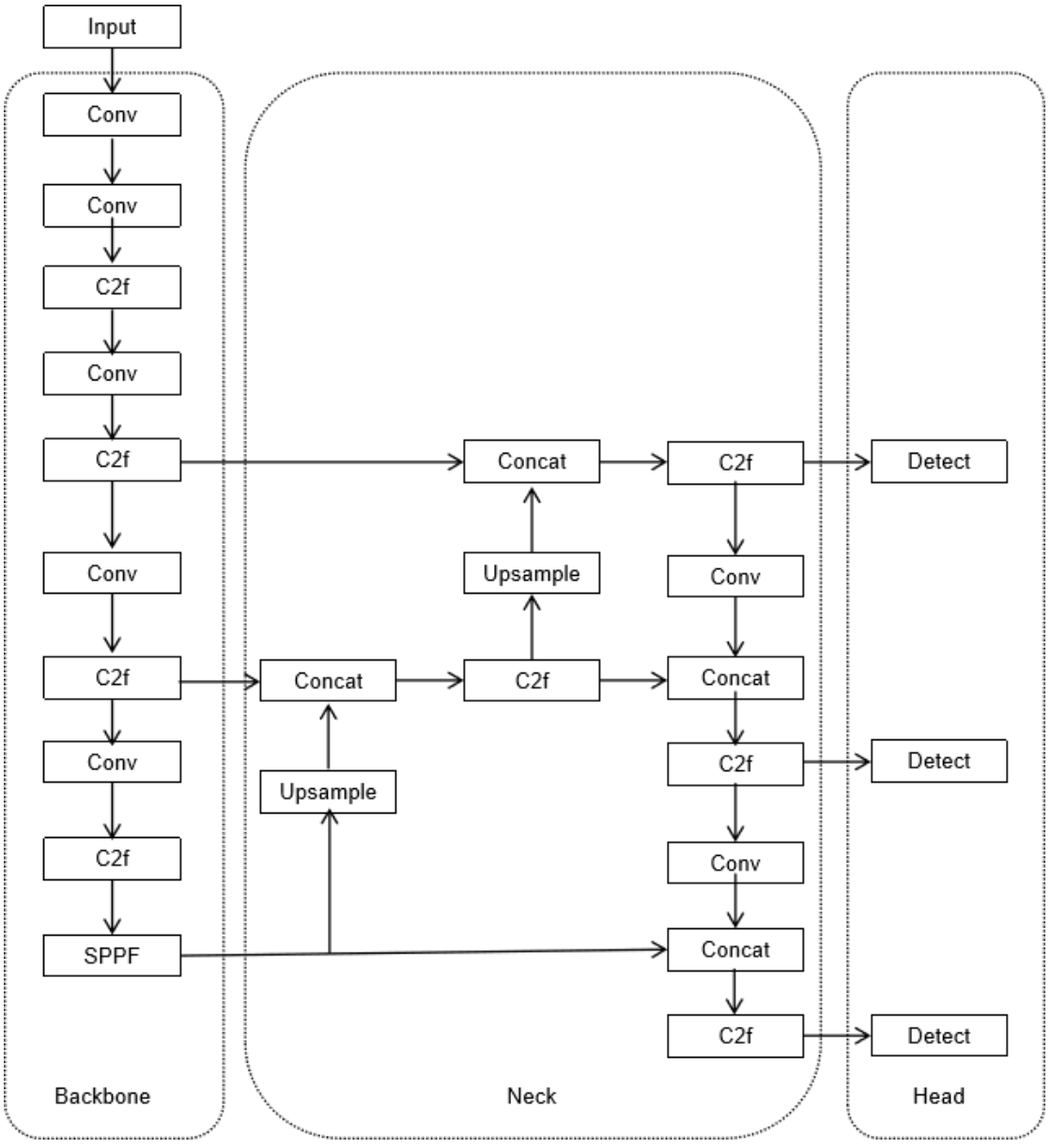


**Fig. 1 The network structure of object detection in YOLO v8**

Based on the previously established framework, YOLO v8 was trained with simple codes given that CUDA and cuDNN lowered the barrier for model training. As the images in the datasets were 300x300 pixels in size, the imgsz parameter was set to be 300 and the number of epochs to the default value of 100.

## 3. Experiment Results

### 3.1 Evaluation Metrics

The core metrics to evaluate the experiment results are as follows:

Precision: the proportion of samples that are correctly predicted among the samples predicted as positive

$$Precision = TP / (TP + FP) \quad (1)$$

where TP stands for True Positive, meaning the positive samples correctly predicted by the model; FP stands for False Positive, which indicates that the samples predicted as positive are actually negative.

Recall: among actually positive samples, how many are correctly predicted as positive.

$$Recall = TP / (TP + FN) \quad (2)$$

where TP is defined as in the previous paragraph; FN stands for False Negative, indicating that the samples predicted as negative are actually positive.

F1 score: the harmonic mean of precision and recall, with considerations on both the accuracy and coverage of the predictions. The F1 score ranges from 0 to 1, where 1 represents both perfect precision and recall, whereas 0 means either the precision or the recall is very underperforming.

$$F1 = 2*(Precision*Recall)/(Precision+Recall) \quad (3)$$

mAP @0.5: The average precision of all classes when the IoU threshold is 0.5. The formula is:

$$mAP@0.5 = \frac{\Sigma(AP50)}{No.ofClasses} \quad (4)$$

Of which:

$AP_{50}$ represents the average precision of the detection results of each class when IoU threshold is 0.5. (IoU is the intersection over union ratio, that is, the ratio of the intersection area to the union area between the predicted box and the ground truth box.)

The number of classes are those involved in the evaluation.

### 3.2 The Analysis of Experiment Results

The model automatically generated several files corresponding to the experiment results. The results are curves based on the metrics provided in 3.1.

Figure 2 shows the number of matches and confusions between predictions and the actual results. As shown in Figure 2, the diagonal line indicates 290 correctly predicted images out of 393 images in the validation set. Of these, 20 were correctly predicted for penalty kicks, 90 for shoots, 123 for dribbling, and 57 for headers.

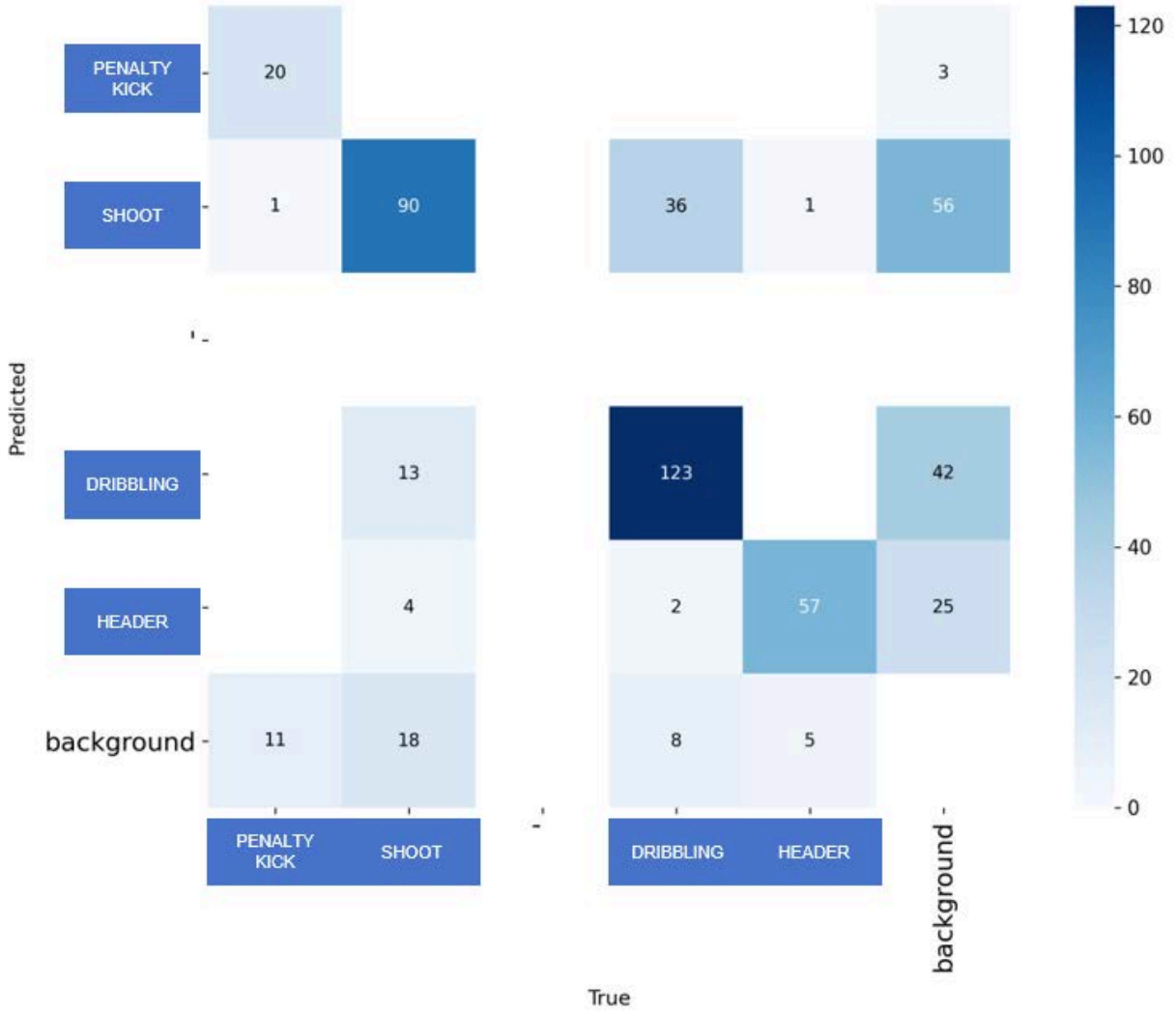


**Fig. 2 Confusion matrix**

ISSN 2959-6157

Figure 3 is a normalized confusion matrix. Through normalization, each cell's value is divided by the actual sample numbers in each class, generating a corresponding accuracy rate. The accuracy rate is 62% for penalties, 72% for shoots, 73% for dribbling, and 90% for headers.

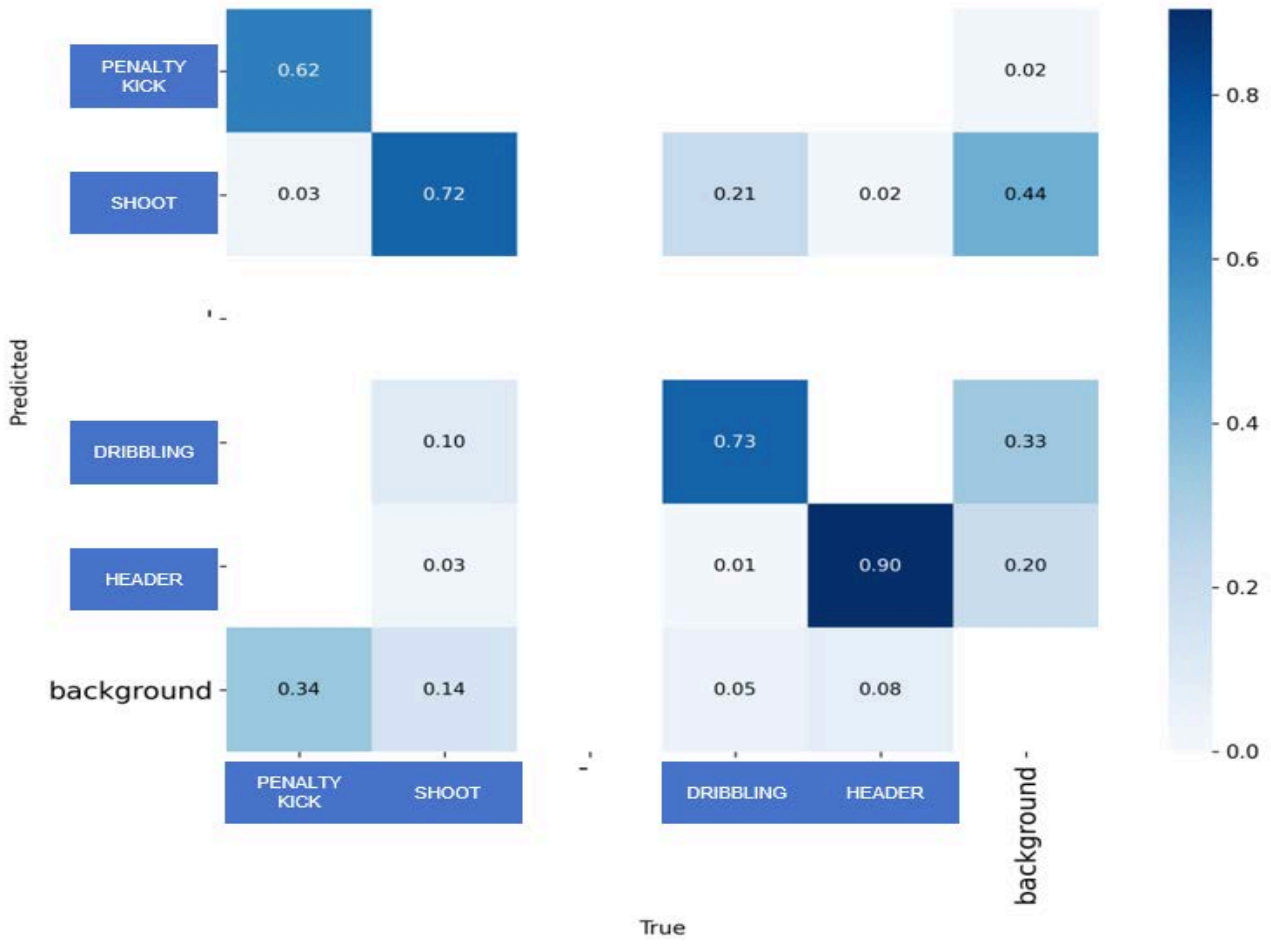


**Fig. 3 Confusion matrix normalized**

Figure 4 shows the precision at different confidence levels. When the confidence level is 1, the average precision across all classes is 1. When the confidence level is 0.6 (0.6-1.0 is generally considered as an interval of high confidence), the average precision across all classes is approximately 0.78, indicating that the prediction accuracy is superior.

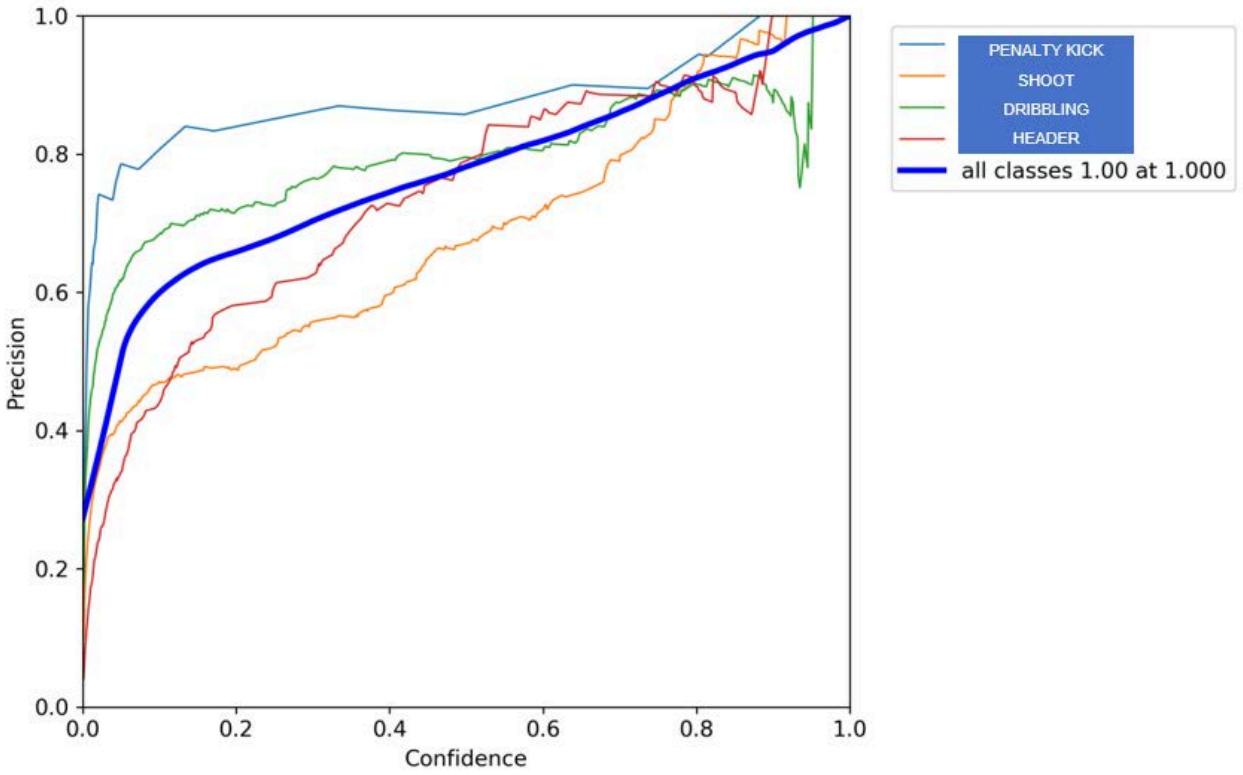


**Fig. 4 Precision-confidence curve**

Figure 5 shows the recall at different confidence levels. When the confidence level is set to 0.000 (all predictions accepted), the average recall across all classes is 0.91, indicating a maximum possible recall of 91%. The results indicate that the model has learned the features of most objects, but fails to detect the remaining 9%. When the confidence level is set to 0.6, the recall is 0.65, indicating that the model‘s recall performance is above average.

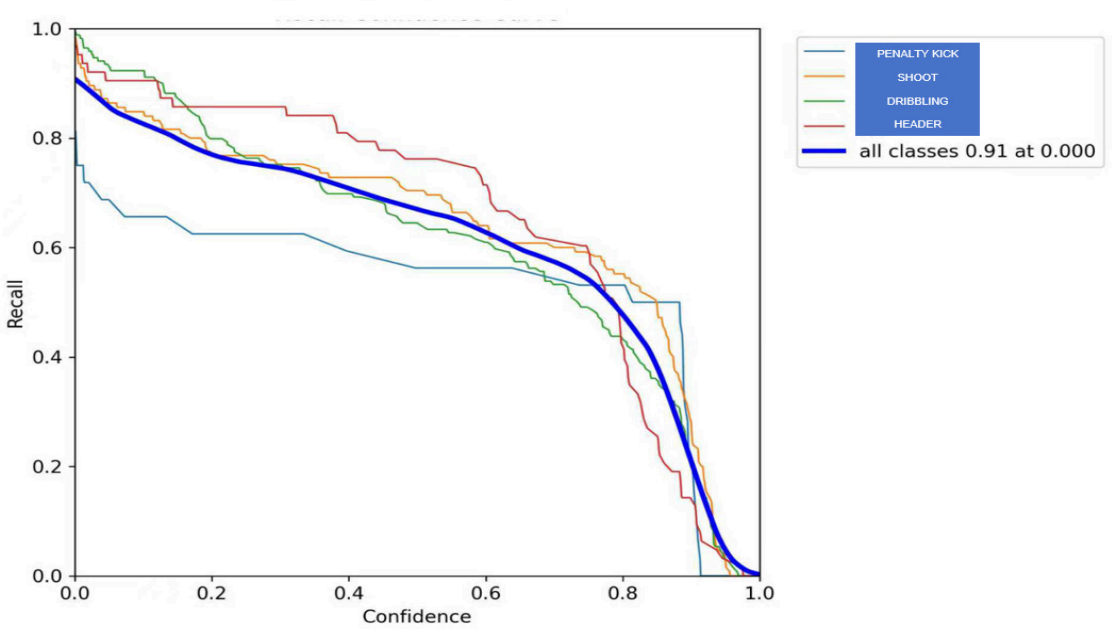


**Fig. 5 Recall-confidence curve**

ISSN 2959-6157

Figure 6 shows the F1 scores. As explained in 3.1, F1 is the harmonic mean of precision and recall. When the confidence level is 0.417, the F1 score across all classes is 0.72; when the confidence level is 0.6, the F1 score across all classes is 0.7. The results indicate that the model reaches a good balance between precision and recall.

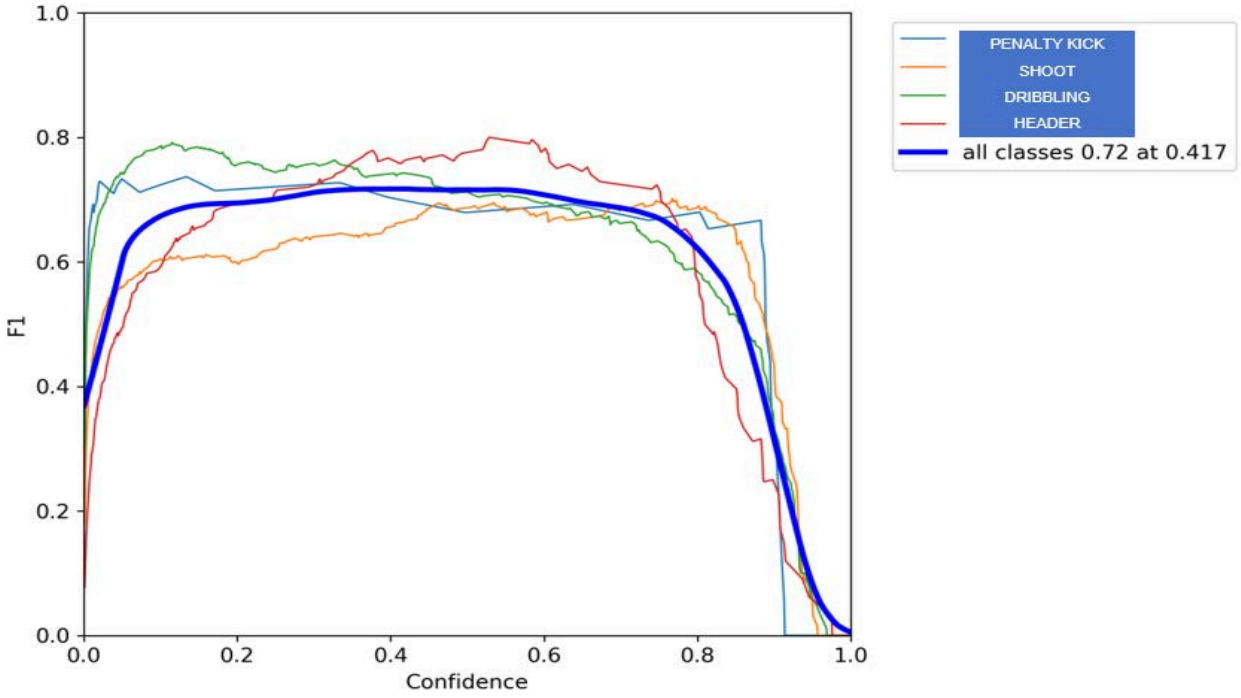


**Fig. 6 F1-confidence curve**

Figure 7 shows the trade-off between precision and recall. When recall = 0.7, precision = 0.78. When the IoU threshold is 0.5, the mAP across all classes is 78.5%, indicating that the model demonstrates good performance.

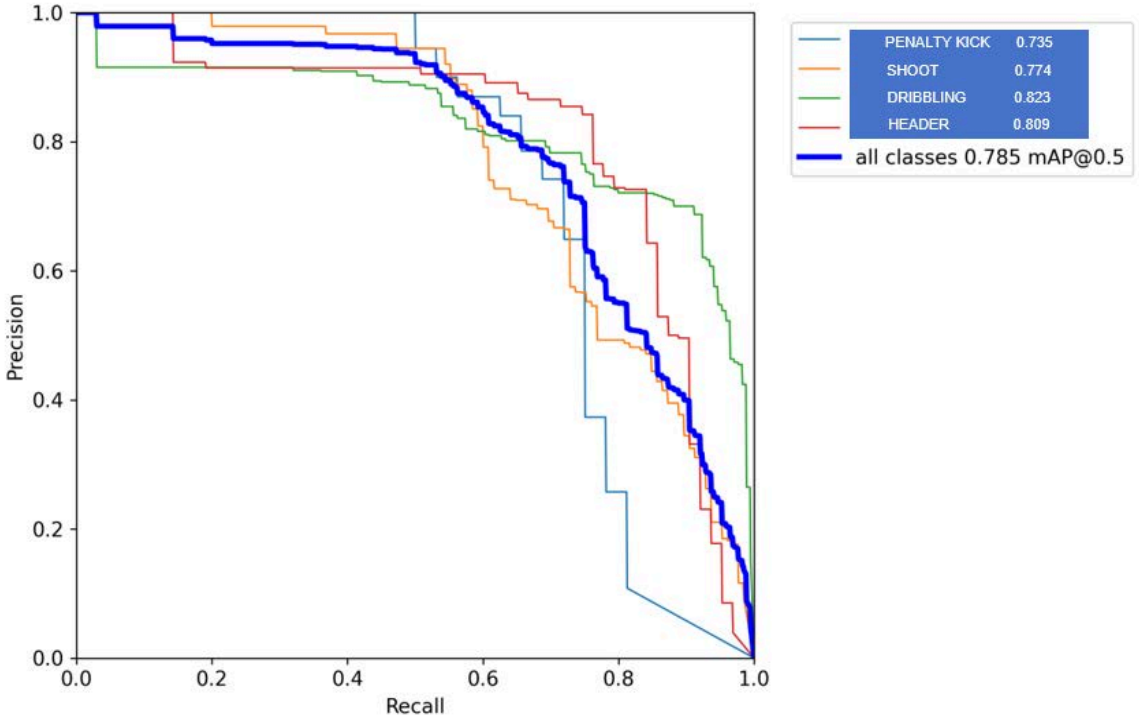


**Fig.7 Precision-recall curve**

**Figure 8 shows the curves of ten parameters automatically generated by the model.**

The train/box_loss curve indicates the localization error for predicted vs. ground truth boxes during training; the train/ cls_loss curve indicates the classification error for predicted vs. actual class labels during training; the train/ dfl_loss curve indicates the distribution focal loss during training. Distribution focal loss is used to deal with class imbalance in localization tasks and help the model to focus on hard-to-predict objects.

The Val/box_loss curve indicates the localization error for predicted vs. ground truth boxes during validation; the val/ cls_loss curve indicates the classification error for predicted vs. actual class labels during validation; the val/ dfl_loss curve indicates the distribution focal loss during validation. Distribution focal loss is used to deal with class imbalance in localization tasks and help the model to focus on hard-to-predict objects.

Regarding the following curves, "B" generally refers to bounding boxes. Precision (B) is the precision curve that indicates the proportion of actually positive samples in the samples predicted as positive by the model; recall (B) is the recall curve that indicates how many actually positive samples are correctly detected by the model; mAP50 (B) curve indicates the average precision with the IoU threshold of 0.5; the mAP50-95 (B) curve indicates the average precision with the IoU threshold between 0.5 and 0.95, to evaluate the detection performance at different IoU thresholds.

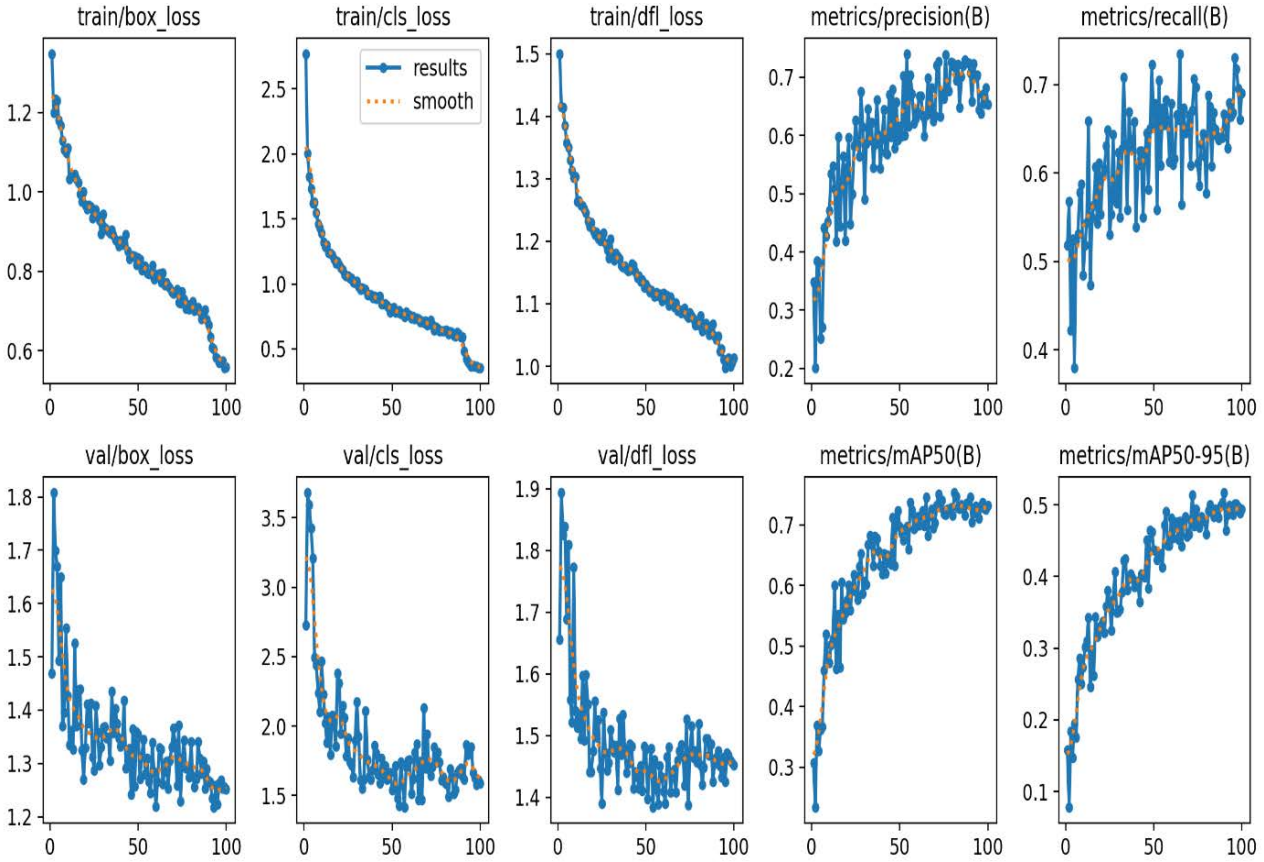


**Fig.8 Presentation of different curves**

## 4. Conclusions

This study trained the Yolov8 object detection model with a manually labeled training set and validated the model with a validation set. The validation results show that when the confidence level is 0.6, the average precision across all classes is approximately 0.78, demonstrating excellent performance in prediction accuracy. When the confidence level is set to 0.6, the recall is 0.65, indicating the above-average recall performance. When the confidence level is 0.6, the F1 score across all classes is 0.7, indicating that the model achieves a good balance between precision and recall. At an IoU threshold of 0.5,

ISSN 2959-6157

the mAP across all classes is 78.5%, demonstrating good performance of the model. It can thus be concluded that after training, the model achieves good accuracy and performance level when detecting soccer moves in images.

While the model achieves good performance in object detection after training, the research also has limitations. For example, the insufficient images of penalty kicks need to be supplemented in further research. Furthermore, in regard to motion recognition and motion analysis, pose estimation can be applied to provide human body modeling for soccer moves and improve the detection accuracy of the model in the future.

## References

[1] Amit Y, Felzenszwalb P, Girshick R. Object Detection. In: Ikeuchi, K. (eds). Computer Vision. Springer, Cham. 2021. https://doi.org/10.1007/978-3-030-63416-2_660.

[2] Goswami P K, Goswami G. A Comprehensive Review on Real Time Object Detection Using Deep Learning Model. 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2022, pp. 1499-1502, doi: 10.1109/SMART55829.2022.10046972

[3] Swathi Y, Challa M. YOLOv8: Advancements and Innovations in Object Detection. In: Senjyu, T., So–In, C., Joshi, A. (eds) Smart Trends in Computing and Communications. SmartCom 2024. Lecture Notes in Networks and Systems, vol 946. Springer, Singapore. 2024. https://doi.org/10.1007/978-981-97-1323-3_1, Page 9.

[4] Vijayakumar A, Vairavasundaram S. YOLO-based Object Detection Models: A Review and its Applications. Multimed Tools Appl 83, 83535–83574, 2024, Page 27.

[5] Vijayakumar A, Vairavasundaram S. YOLO-based Object Detection Models: A Review and its Applications. Multimed Tools Appl 83, 83535–83574 , 2024, Page 29.

[6] Utsumi O, Miura K, Ide I, Sakai S, Tanaka H. An Object Detection Method for Describing Soccer Games from Video. Proceedings. IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, 2002, pp. 45-48 vol.1, doi: 10.1109/ICME.2002.1035714.

[7] Markappa P. S. S, O'Leary C, Lynch C. A Review of YOLO Models for Soccer-Based Object Detection. 2024 Sixth International Conference on Intelligent Computing in Data Sciences (ICDS), Marrakech, Morocco, 2024, pp. 1-7, doi: 10.1109/ICDS62089.2024.10756443.

[8] Althaf Pramasetya Perkasa M, El Akbar R. R, Al Husaini M., Rizal R. Visual Entity Object Detection System in Soccer Matches Based on Various YOLO Architecture. J. Tek. Inform. (JUTIF), vol. 5, no. 3, 2024, pp. 811–820, Jun.

[9] Huang L, Liu G. Functional Motion Detection Based on Artificial Intelligence. J. Supercomput., 78 (3) pp. 4290-432.2022.

[10] Fang W, Wang L, Ren P. Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments. in IEEE Access, vol. 8, 2020, pp. 1935-1944.

[11] Rebuffi S A, Gowal S, Calian D A. Data Augmentation can Improve Robustness. Advances in Neural Information Processing Systems, 34, 2021, pp. 29935-29948.