Football Players & Ball Detection

Federico Cesare Cattò

Gennaio 2026

Contents

Chapter 1

Data Understanding

This section describes the initial data exploration and validation phase carried out before training the object detection model. The goal of this step is to ensure dataset integrity, understand its structure, and identify potential issues that could negatively affect the training process.

1.1 Dataset Structure

The Dataset is organized following the standard YOLO format, with separate directories for images and labels, and a clear split between training and validation sets. In particular, the structure consists of:

- `images/train` and `images/val` directories containing the input images;
- `labels/train` and `labels/val` directories containing the corresponding annotation files in YOLO format.

Each image has an associated text file with the same filename, ensuring a one-to-one correspondence between images and annotations. A preliminary inspection confirms that all folders are correctly populated and consistent.

1.2 Dataset Size and Split

The dataset contains a total of 600 images, divided into:

- 520 images for training;
- 80 images for validation.

This corresponds to an approximate 87%–13% train-validation split, which is reasonable for a supervised object detection task where the primary focus is model learning rather than hyperparameter tuning.

1.3 Class Definitions and Distribution

The task involves detecting three different object classes:

- **Player** (class ID 0),

- **Referee** (class ID 1),
- **Ball** (class ID 2).

The class distribution was computed by parsing all YOLO label files and counting the number of bounding boxes per class. The resulting distributions are:

- **Training set**:
  - Player: 9,916 instances
  - Referee: 844 instances
  - Ball: 384 instances
- **Validation set**:
  - Player: 1,467 instances
  - Referee: 129 instances
  - Ball: 61 instances

As expected in football-related scenarios, the dataset is highly imbalanced, with players being the dominant class and the ball representing the rarest object. This imbalance is typical in real-world sports footage and poses additional challenges for detecting small objects such as the ball.

Figure 1.1 shows a representative frame from the validation dataset annotated in YOLO format. Players and referees are marked with bounding boxes, while the ball is highlighted with a circle. This image emphasizes the scale differences between players and the ball, highlighting the inherent challenge of detecting small objects in football scenes.

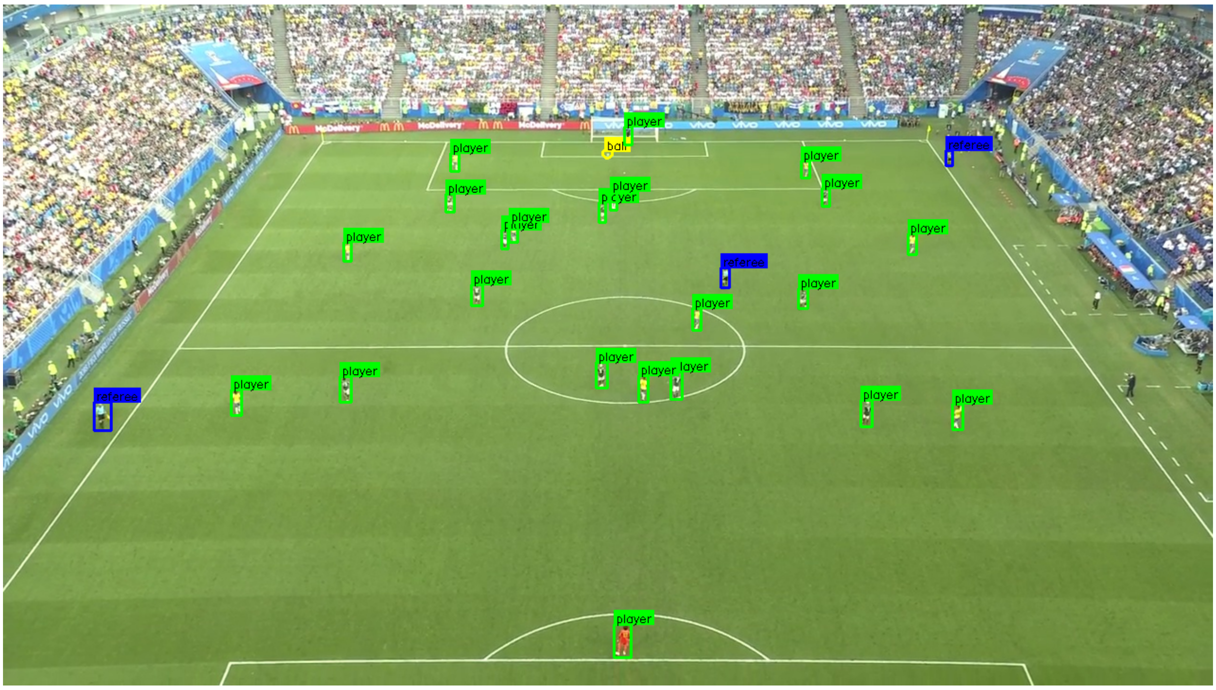

Figure 1.1: Example frame from the validation dataset annotated in YOLO format. Green boxes: players; Blue boxes: referees; Yellow circle: ball.

This figure provides a qualitative view of the dataset and the annotation quality.

1.4 Bounding Box Integrity Check

All annotations follow the YOLO format, where bounding boxes are represented by normalized values in the range $[0, 1]$ for center coordinates and dimensions.

A systematic validation was performed to verify that all bounding box values fall within the valid range. The results show:

- 11,144 bounding boxes in the training set;
- 1,657 bounding boxes in the validation set;
- **No invalid bounding boxes detected**.

This confirms the correctness of the annotation process and ensures that no malformed labels will interfere with the training procedure.

1.5 Image Resolution Analysis

An analysis of image dimensions was conducted on both training and validation sets. All inspected images share the same resolution:

$$1280 \times 720 \text{ pixels}$$

Both the minimum and maximum width and height values are identical across the dataset, indicating complete consistency in image size. This homogeneity simplifies preprocessing and avoids the need for complex resizing or aspect-ratio handling strategies during training.

1.6 Summary

The data understanding phase confirms that the dataset is:

- correctly structured and complete;
- consistently annotated using the YOLO format;
- free of invalid bounding boxes;
- uniform in image resolution.

Despite the presence of class imbalance, which reflects realistic football scenarios, the dataset is well-suited for training an object detection model focused on players, referees, and ball detection.

Chapter 2

Data Cleaning and Preprocessing

This section describes the data cleaning and preprocessing steps applied to the dataset before model training. The objective of this phase is to guarantee annotation consistency, remove corrupted samples, and correct minor labeling issues that could negatively impact the learning process of the object detection model.

2.1 Image–Label Consistency Check

The first preprocessing step consists of verifying the one-to-one correspondence between images and label files. For each image in the dataset, a corresponding annotation file with the same filename is expected, and vice versa.

This check was performed separately for the training and validation sets. The analysis confirmed that:

- all training images have a corresponding label file;
- all validation images have a corresponding label file;
- no orphan label files exist without a matching image.

This result confirms the structural integrity of the dataset and prevents runtime errors during training.

2.2 Detection of Empty Annotation Files

A common issue in object detection datasets is the presence of empty label files, which may arise from frames without detectable objects or errors during the annotation process.

The dataset was scanned to identify annotation files with zero size. The results show:

- 3 empty label files in the training set;
- no empty label files in the validation set.

Empty annotation files were considered unsuitable for training and were removed in a subsequent cleaning step.

2.3 Normalization and Format Validation

All YOLO annotations are expected to follow a strict format consisting of five values per line:

$$(\text{class\_id}, x, y, w, h)$$

where bounding box coordinates are normalized in the range $[0, 1]$.

Each annotation file was parsed line by line to verify both format correctness and value normalization. No formatting issues or normalization errors were detected in either the training or validation sets, confirming that all annotations comply with the YOLO specification.

2.4 Bounding Box Boundary Check

Even when normalized values lie in the $[0, 1]$ range, bounding boxes may still extend partially outside the image boundaries. For this reason, an additional geometric validation was performed by checking whether:

$$x \pm \frac{w}{2},\ y \pm \frac{h}{2} \in [0, 1]$$

This analysis revealed:

- 26 bounding boxes partially outside the image in the training set;
- 3 bounding boxes partially outside the image in the validation set.

Although limited in number, such inconsistencies can negatively affect training stability and localization accuracy.

2.4.1 Cleaning Operations

Based on the detected issues, two corrective actions were applied.

2.4.2 Removal of Empty Annotation Files

All empty label files were permanently removed from the dataset. In total:

- 3 training annotation files were deleted;
- no validation files required removal.

2.4.3 Bounding Box Clipping

Bounding boxes extending outside image boundaries were corrected by clipping their coordinates to remain within the valid $[0, 1]$ range. The bounding box center and dimensions were recomputed accordingly.

This correction affected:

- 517 label files in the training set;
- 80 label files in the validation set.

Clipping was preferred over discarding samples in order to preserve as much training data as possible while ensuring annotation validity.

2.5 Post-cleaning Validation

After applying the cleaning operations, the bounding box boundary checks were repeated. The results show a significant reduction in invalid boxes:

- 12 remaining borderline cases in the training set;
- 2 remaining borderline cases in the validation set.

These residual cases correspond to extremely small numerical deviations and do not compromise the overall dataset quality.

2.6 Final Dataset Status

At the end of the preprocessing phase, the dataset satisfies the following conditions:

- complete image–label consistency;
- no empty annotation files;
- bounding boxes constrained within image boundaries;
- uniform and valid YOLO annotation format.

This cleaning process ensures that the dataset is reliable and well-prepared for training a robust object detection model.

Chapter 3

Training Configuration and Data Augmentation Strategy

This chapter describes the configuration of the training pipeline, with particular focus on dataset specification, model initialization, and data augmentation strategies adopted to improve robustness and generalization.

3.1 Dataset Configuration for YOLO Training

In order to train the YOLOv8 model, a `data.yaml` configuration file was created. This file defines the dataset structure and provides the necessary metadata required by the YOLO training framework.

The configuration specifies:

- the root path of the dataset;
- the relative paths to training and validation image folders;
- the mapping between class identifiers and semantic class names.

The three object classes considered in this project are *player*, *referee*, and *ball*. This explicit configuration ensures full compatibility with the YOLOv8 training pipeline and enables reproducibility of the experiments.

3.2 Model Selection

The YOLOv8n architecture was selected as the base model for this project. This variant represents the lightweight version of the YOLOv8 family and offers a favorable trade-off between computational efficiency and detection performance.

The use of a lightweight model is particularly suitable for:

- limited computational resources;
- faster experimentation and iteration;
- potential deployment in real-time or edge-based scenarios.

Pretrained weights were used as initialization, allowing the model to leverage transfer learning from large-scale object detection datasets.

3.3 Training Setup

The training process was configured with the following main parameters:

- number of epochs: 50;
- input image size: 640×640;
- batch size: 16;
- optimizer: Adam with an initial learning rate of 10^{-3}.

These hyperparameters were chosen to balance convergence stability, training speed, and generalization capability.

3.4 Data Augmentation Strategy

To improve robustness against variations in lighting, scale, viewpoint, and spatial arrangement, an extensive data augmentation strategy was adopted. YOLOv8 applies these augmentations automatically during training.

The following augmentations were explicitly enabled:

- **Horizontal flip** with probability 0.5, to simulate left-right symmetry in football scenes;
- **Random scaling** to improve scale invariance, particularly for objects at different distances from the camera;
- **HSV color jittering**, including hue, saturation, and value perturbations, to account for lighting and color variations across stadiums and broadcast conditions;
- **Random translation** to enhance spatial robustness;
- **Mosaic augmentation**, combining multiple images into a single training sample to improve small object detection and contextual understanding;
- **MixUp augmentation**, blending pairs of images to regularize the training process.

These augmentations are particularly beneficial for detecting small and fast-moving objects such as the football, which are often challenging in real match footage.

3.5 Implicit Augmentations and Training Robustness

Certain augmentations, such as motion blur and random cropping, are not explicitly configured but are implicitly approximated within the YOLOv8 training pipeline through a combination of Mosaic augmentation, scaling, and translation.

This design choice reduces configuration complexity while still providing sufficient variability to improve model generalization.

3.6 Reproducibility Considerations

All augmentation parameters were explicitly documented to ensure experiment reproducibility. The training configuration, including dataset definition, model choice, and augmentation strategy, can be reliably replicated in future experiments or extended for further model improvements.

3.7 Summary

The training configuration integrates a lightweight YOLOv8 model with a carefully designed augmentation strategy tailored to football scenarios. This setup aims to enhance detection performance while maintaining computational efficiency and robustness to real-world variations.

Chapter 4

Model Evaluation and Performance Analysis

This chapter presents the evaluation of the trained YOLOv8 model on the validation set. Both global and class-specific metrics are analyzed, with particular attention to the challenges associated with small object detection in football scenarios.

4.1 Model Loading and Evaluation Protocol

The best-performing model weights obtained during training were loaded for evaluation. Model performance was assessed on the validation set using the same image resolution and batch size adopted during training.

The evaluation follows the standard YOLO protocol and reports metrics computed over the entire validation split, ensuring consistency and fairness in performance assessment.

4.2 Global Detection Performance

The overall performance of the model is summarized using commonly adopted object detection metrics:

- mean Average Precision at IoU 0.5 (mAP@0.5);
- mean Average Precision averaged over IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95);
- precision;
- recall.

The obtained results are:

- mAP@0.5: 0.7049;
- mAP@0.5:0.95: 0.4428;
- precision: 0.8620;
- recall: 0.6670.

These values indicate strong overall detection performance, characterized by high precision and good localization accuracy, with some loss in recall due to challenging detection cases.

4.3 Per-class Performance Analysis

To better understand model behavior, performance was analyzed separately for each object class. The per-class mAP@0.5 scores are reported below:

- **Player**: 0.6563;
- **Referee**: 0.5621;
- **Ball**: 0.1100.

The results show that the model performs well on large and well-defined objects such as players and referees, while ball detection remains significantly more challenging.

4.4 Player vs Ball Detection

A direct comparison between player and ball detection highlights the impact of object scale on model performance. While player detection achieves a satisfactory mAP@0.5 of 0.6563, ball detection reaches only 0.1100.

This performance gap is primarily attributed to:

- the extremely small size of the ball in broadcast images;
- frequent motion blur during fast ball movements;
- partial occlusions caused by players;
- limited number of ball instances compared to player instances.

To complement the quantitative analysis, Figure 4.1 provides a side-by-side comparison of model predictions on selected validation frames. On the left, the ball, players, and referees are all correctly detected, representing a successful case. On the right, the ball is not detected while players remain accurately localized, representing a failure case.

This visual comparison illustrates the impact of object scale and motion on detection performance, confirming the lower mAP observed for the ball class and the challenges associated with small object detection in football scenarios.

4.5 Error Analysis and Confidence Assessment

An error analysis was conducted by inspecting detection confidence scores on the validation set. Out of 1,549 total detections, 104 bounding boxes were associated with confidence scores below 0.4.

This indicates that most detections are produced with high confidence, and low-confidence predictions represent a relatively small fraction of the total outputs.

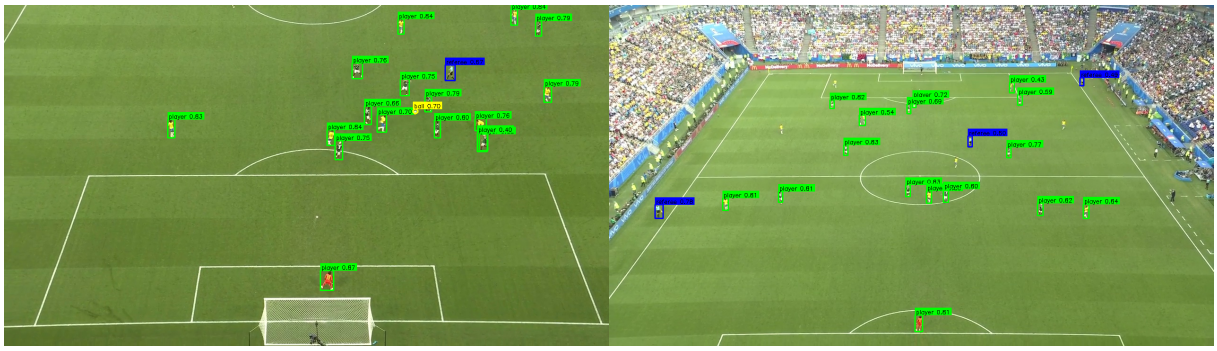

Figure 4.1: Side-by-side comparison of model predictions on validation frames. **Left**: successful detection of ball, players, and referees. **Right**: failure case where the ball is missed, while players are correctly detected. This figure highlights the difficulty of small object detection and provides a qualitative confirmation of the quantitative metrics.

4.6 Small Object Detection Analysis

To further investigate ball detection difficulty, the spatial area of predicted ball bounding boxes was analyzed. The measured bounding box areas show:

- minimum area: 110.69 pixels;
- mean area: 183.33 pixels;
- maximum area: 353.98 pixels.

These values confirm that the ball occupies a very limited number of pixels, making it inherently difficult for convolutional detectors to extract discriminative features, especially at standard input resolutions.

4.7 Discussion

The evaluation results demonstrate that the fine-tuned YOLOv8 model is effective for detecting players and referees in football footage. However, ball detection remains a challenging task due to the small object size and dynamic scene conditions.

While data augmentation strategies significantly improve robustness and generalization, they are not sufficient to fully overcome the limitations associated with small object detection.

4.8 Summary

In summary:

- the model achieves strong global performance with high precision;
- player and referee detection are reliable and accurate;
- ball detection is the main performance bottleneck;
- remaining errors are mainly caused by small object size, motion blur, and occlusions.

These findings motivate future improvements such as higher input resolutions, multi-scale feature enhancement, or class-specific training strategies.

Chapter 5

Video Inference and Qualitative Results

This chapter describes the application of the trained YOLOv8 model to real football video footage. The goal of this phase is to qualitatively assess model performance in realistic conditions and visually analyze detection behavior across different object classes.

5.1 Model Deployment for Video Inference

The best-performing model weights obtained during training were loaded and applied to a previously unseen football video. Frame-by-frame inference was performed using the trained YOLOv8 detector, without any additional fine-tuning.

Inference was executed directly on raw video frames, demonstrating the model's capability to operate in realistic, unstructured environments typical of football broadcast footage.

5.2 Video Processing Pipeline

The video was processed using OpenCV by reading frames sequentially from the input video file. The output video was generated using the same frame rate and spatial resolution as the original input to preserve temporal consistency and visual quality.

For each frame, the model produced a set of detections filtered using a confidence threshold of 0.35, chosen to balance detection sensitivity and false positives.

5.3 Visualization Strategy

To improve interpretability and class distinction, a class-specific visualization strategy was adopted:

- **Players** are displayed using green bounding boxes;
- **Referees** are displayed using blue bounding boxes;
- **Ball** detections are represented using a circular marker, reflecting the roughly spherical shape of the object.

Each detected object is annotated with its class name and confidence score. Labels are rendered on a colored background to ensure readability across different lighting conditions.

5.4 Ball-specific Visualization

Unlike players and referees, the ball is visualized using a circle centered on the bounding box centroid. The circle radius is proportional to the predicted bounding box size, with a minimum radius enforced to ensure visibility.

This design choice improves visual clarity for small objects and highlights ball trajectories more effectively than standard bounding boxes.

5.5 Qualitative Observations

Qualitative inspection of the output video reveals that:

- player detections are generally stable and well-localized;
- referee detections are consistent and rarely confused with players;
- ball detections are correct when the ball is clearly visible but may fail under fast motion, occlusions, or extreme scale reduction.

The visual results confirm the quantitative evaluation findings, emphasizing the difficulty of small object detection in dynamic sports scenes.

Figure 5.1 shows an example frame from the processed football video. Players and referees are detected using green and blue bounding boxes, respectively, while the ball is represented with a yellow circle at the bounding box centroid.

This visualization provides a clear illustration of model performance in real-world conditions. Player and referee detections are generally accurate and well-localized, whereas ball detection remains challenging in cases of fast motion, occlusion, or small object size. The figure reinforces the observations discussed qualitatively in the previous sections.

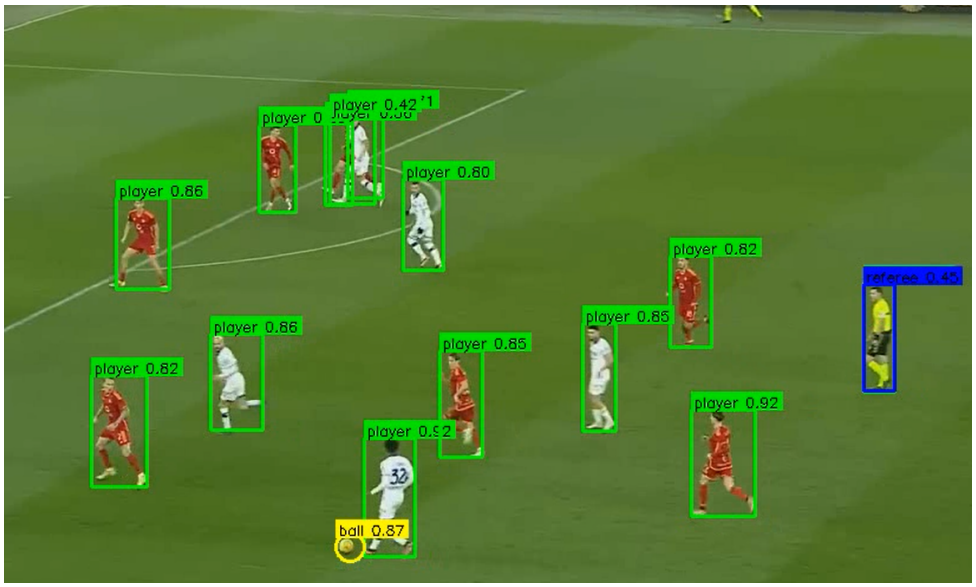

Figure 5.1: Example frame extracted from the video inference output. Green boxes: players; Blue boxes: referees; Yellow circle: ball.

The figure demonstrates the model's capability to detect multiple classes in dynamic, real-world football footage, while highlighting the challenges associated with small object detection.

5.6 Output Generation

All processed frames were written to a clean output video file, producing a continuous visualization of model predictions across the entire video sequence. This output serves both as a qualitative evaluation tool and as a demonstrative artifact of the system's capabilities.

5.7 Summary

This inference experiment demonstrates the practical applicability of the trained YOLOv8 model to real football footage. While player and referee detection performs reliably in dynamic conditions, ball detection remains the most challenging task, reinforcing the need for specialized strategies when dealing with small, fast-moving objects.

Chapter 6

Conclusions

This chapter focused on the design, training, evaluation, and deployment of an object detection system for football scenarios, with the objective of detecting players, referees, and the ball from both images and video footage.

Starting from a structured and well-annotated dataset, a thorough data understanding and cleaning process was carried out to ensure annotation consistency, correctness, and robustness. Particular attention was devoted to validating YOLO-format labels, correcting boundary issues, and removing corrupted samples, resulting in a clean and reliable dataset suitable for training deep learning models.

A lightweight YOLOv8 architecture was selected to balance detection performance and computational efficiency. Transfer learning was leveraged through pretrained weights, while an extensive data augmentation strategy was adopted to improve generalization to real-world football conditions, including variations in lighting, scale, viewpoint, and spatial configuration.

Quantitative evaluation on the validation set demonstrated strong overall performance, characterized by high precision and competitive mAP values. The model achieved reliable detection of large objects such as players and referees, confirming the effectiveness of the chosen architecture and training strategy. However, ball detection proved to be significantly more challenging, mainly due to the small object size, motion blur, frequent occlusions, and strong class imbalance. These limitations were consistently observed across both quantitative metrics and qualitative video-based analysis.

The inference experiments on real football video footage further validated the practical applicability of the system. Visual inspection confirmed stable detections for players and referees in dynamic scenes, while highlighting the intrinsic difficulty of tracking and detecting the ball in complex match situations. The qualitative results closely aligned with the quantitative evaluation, reinforcing the reliability of the experimental findings.

Overall, this project demonstrates that modern one-stage detectors such as YOLOv8 can effectively address football player and referee detection tasks, while small object detection remains an open challenge. The developed pipeline provides a solid foundation for further improvements and extensions.

Future work may include training with higher input resolutions, incorporating multi-scale feature enhancement techniques, increasing the number of ball-specific annotations, or integrating temporal information across frames to improve ball detection and tracking performance.

In conclusion, the proposed system represents a robust and well-engineered solution for football object detection, combining sound data preparation, effective model design,

and comprehensive evaluation within a realistic application scenario.