YOLO Models for Football Analysis System

Your Name

June 2, 2025

Abstract

This document provides an overview of YOLO (You Only Look Once) models, focusing on their application in a football analysis system for detecting and tracking players, referees, and footballs in video footage. We describe the YOLO architecture, training process, and inference, with specific reference to a fine-tuned YOLOv5x model (`best.pt`) and a pretrained YOLOv8x model. The system leverages object detection to enable team assignment (via KMeans clustering) and movement analysis (via optical flow and perspective transformation).

1 Introduction

YOLO (You Only Look Once) is a family of real-time object detection models that predict bounding boxes and class probabilities in a single forward pass. This efficiency makes YOLO ideal for applications like the football analysis system, which detects players, referees, and footballs in video frames for downstream tasks such as tracking and movement analysis. This report explains the YOLO architecture, training, and inference, with details from a project using a fine-tuned YOLOv5x model.

2 YOLO Architecture

YOLO models consist of three components:

- **Backbone**: Extracts features using convolutional layers (e.g., `Conv [3, 80, 6, 2, 2]`) and C3 modules (cross-stage partial connections). In YOLOv5x, the backbone processes 640x640 images.
- **Neck**: Aggregates multi-scale features using SPPF (Spatial Pyramid Pooling - Fast) and FPN (Feature Pyramid Network) to detect objects of varying sizes (e.g., footballs vs. players).
- **Head**: Predicts bounding boxes, class probabilities, and confidence scores for 4 classes in the project (e.g., player, referee, football).

The YOLOv5x model used has 285 layers and approximately 97.2 million parameters, with a computational cost of 246.9 GFLOPs.

3 Training Process

The YOLOv5x model was fine-tuned on the `football-players-detection-2` dataset (612 training images, 38 validation images) for 100 epochs, with a batch size of 16 and image size of 640x640. Key training details include:

- **Losses**:

  - `box_loss`: Error in bounding box coordinates (1.511 to 0.9904 over 18 epochs).
  - `cls_loss`: Error in class predictions (2.015 to 0.5373).
  - `dfl_loss`: Distribution Focal Loss for bounding box refinement (0.9084 to 0.8195).

- **Optimizer**: AdamW with a learning rate of 0.00125 and momentum of 0.9.

- **Metrics**: Mean Average Precision (mAP50) improved from 0.231 to 0.757, with precision from 0.499 to 0.887 and recall from 0.325 to 0.646.

Data augmentation (e.g., mosaic, HSV adjustments) enhanced generalization.

4 Inference

Inference was performed on a video (`08fd33_4.mp4`) using two models:

- **Fine-tuned YOLOv5x** (`best.pt`): Detects custom classes (player, referee, football) with high accuracy due to fine-tuning.

- **Pretrained YOLOv8x**: Detects generic COCO classes (e.g., person, sports ball), less suitable for the project.

The code for inference is:

```
from ultralytics import YOLO
model = YOLO('models/best.pt')  % or 'yolov8x'
results = model.predict('input_videos/08fd33_4.mp4', save=True)
for box in results[0].boxes:
    print(f"Class: {box.cls}, Confidence: {box.conf}, Coordinates: {box.xyxy}")
```

The fine-tuned model is preferred for its alignment with project goals.

5 Project Relevance

The football analysis system uses YOLOv5x for object detection, enabling:

- **Tracking**: BoT-SORT tracks objects across frames.

- **Team Assignment**: KMeans clusters t-shirt colors.

- **Movement Analysis**: Optical flow and perspective transformation calculate player speed and distance.

The fine-tuned model (`best.pt`, mAP50=0.757) ensures accurate detection for these tasks.

6 Conclusion

YOLO models provide a robust framework for real-time object detection, with YOLOv5x being highly effective for the football analysis system after fine-tuning. The pretrained YOLOv8x model, while powerful, is less suitable due to its generic classes. Future work includes integrating detection results with tracking and analysis modules.