CS230

Implementing Multi-Class Object Detection in Soccer Matches Through YOLOv5

George R. Dimopoulos
Stanford University
gdimop@stanford.edu

Abstract

The abstract should consist of 1 paragraph describing the motivation for your paper and a high-level explanation of the methodology you used/results obtained. Due to the existing capabilities of AI and Deep Learning, the sport of soccer is becoming more open to using such technology to enhance the quality of the analyzation of the sport. In particular, many improvements can be made to enhance the video quality of televised soccer matches. This paper compares two YOLOv5 models–YOLOv5s and YOLOv5m– which were trained on the SoccerNet dataset to classify objects on a football pitch and draw boundary boxes around them. Although the YOLOv5s model outperformed the YOLOv5m model in precision, the reverse is true for recall; as such, further studies are needed to determine which model is truly better at multi-class object detection of soccer videos.

1 Introduction

Soccer, or football, is a fast-paced game at all levels of the sport. As such, it is difficult for spectators to possess a firm grasp of every element in a game: Goettker and Gegenfurtner [2021] find that because of these moving variables, spectators have been forced to use contextual cues to determine where the ball is at any given moment. While these predictive eye movements are highly valuable when watching a game, the ability for a spectator to gather information is oftentimes hindered by players' obscured faces and numbers on their kits.

However, the ever-increasing integration of AI into the sport on an international level Price [2022] provides Deep Learning a window of opportunity to assist a viewer's experience watching televised matches. For example, Deep Learning may mitigate the viewer's difficulty to identify players on the pitch by displaying the name of each player above their heads in real-time, which is similar to what is seen in the gameplay of Electronic Arts's FIFA 22:

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Figure 1: A snippet of my own FIFA 22 gameplay, with a yellow oval highlighting Bruno Fernandes's name above his character in-game.

This may be done through implementing multi-class object detection, where a model identifies soccer players, referees, and the ball in-game, and creates boundary boxes around them in accordance to their respective designation while they are on the pitch. In particular, this paper explores the intersection between computer vision and sports by implementing two models to accomplish the goal of multi-class object detection. The two models–YOLOv5s and YOLOv5m–are then compared against each other to determine which most effectively conducts multi-class object detection.

2 Dataset and Features

I used the SoccerNet multi-object tracking (MOT) dataset derived from Cioppa et al. [2022] to classify objects on the pitch. The SoccerNet MOT dataset consists of 100 high definition (1080p) video clips of thirty seconds each showcasing soccer footage from the main broadcasting camera angle. In total, each video consists of 750 images, complete with ground-truth labelings and tracklets of eight classes: player team left, player team right, goalkeeper team left, goalkeeper team right, main referee, side referee, staff, ball.

Because this dataset is designed to be used by MOT models (such as SORT, DeepSORT, Tracktor, FairMOT, and ByteTrack, as analyzed by Cioppa et al. [2022]), it was necessary to first preprocess this dataset to fit the requirements of a YOLOv5 model. In particular, I extracted the tracklet information and generated .txt files corresponding to each image which contained the boundary boxes and classes of all objects in that image. Running this newly-formed dataset on models' pretrained weights produced these images:

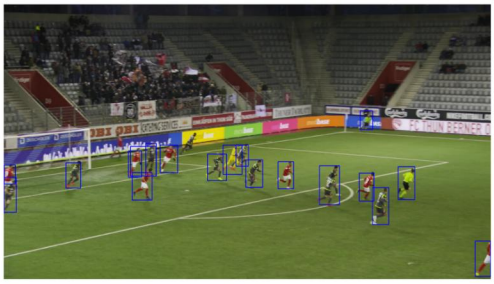

Figure 1: YOLOv5s pretrained classification

Figure2: YOLOv5m pretrained classification

3 Model Architectures and Benchmark

3.1 The YOLOv5 Model

YOLO, or "You Only Look Once", is a model used to for multi-class object detection within images. It splits the images into n cells on a grid, and determines whether a specific cell contains the center coordinates of a classifiable object by returning the probability that it indeed does contain the coordinate (along with the height and width of the bounding box). The model then applies non-max suppression to the bounding box coordinates to deduce which coordinate has the highest probability of being the true center of the object.

The YOLOv5 architecture consists of four main sections: input, backbone, neck, and output Li et al. [2022]. The input terminal contains the preprocessing of data, including adaptive image filling and mosaic data augmentation Wu et al. [2017]. The backbone utilizes multiple convolution and pooling to extract feature maps of different sizes from the input image Li et al. [2022]. The neck uses FPN and PAN pyramid structures, which improve the detection capability of the model Li et al. [2022]. Finally, the output "predicts targets of different sizes on feature maps" Li et al. [2022].

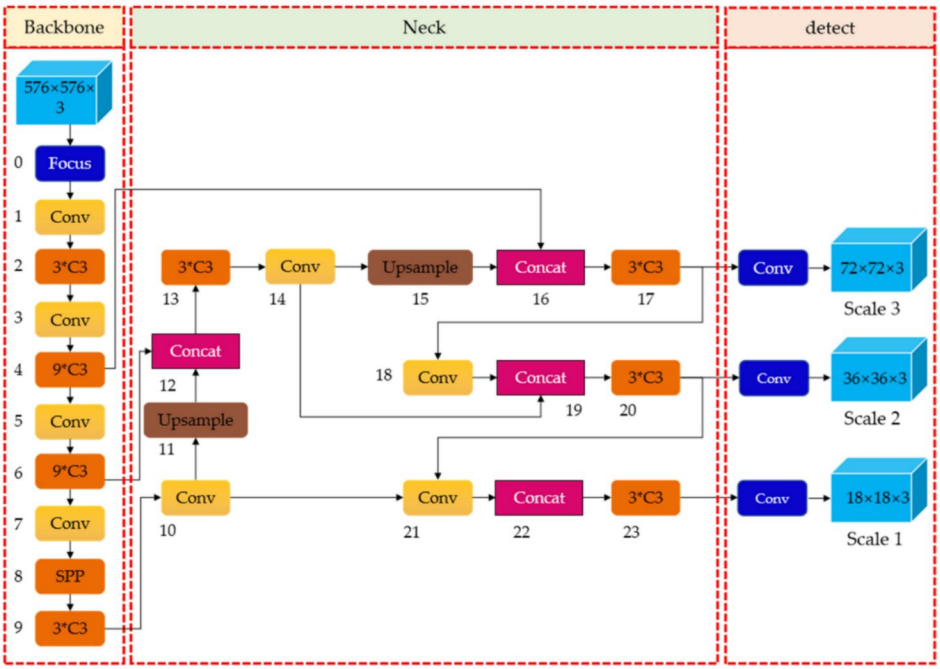

Figure 3: The architecture of YOLOv5 Li et al. [2022].

The main difference between the two models I am using are the amount of layers present within each model. YOLOv5s contains 270 layers, 7041205 parameters, 7041205 gradient, and 159 GFLOPs. On the other hand, YOLOv5m is a larger model, containing 369 layers, 20899605 parameters, 20899605 gradients, and 48.1 GFLOPs.

These two models are pretrained using the COCO dataset Lin et al. [2014], which contains over 328,000 images, 2,500,000 labeled object instances, and 91 objects types. However, these classes do not fit those of the SoccerNet dataset; as such, I applied transfer learning to the last layer of each model to only allow the models to identify one of the eight classes available in the SoccerNet dataset.

3.2 Baseline

The baseline of the transfer-learned YOLOv5 models will be taken from the YOLOv5 pretrained precision (P) and recall (R) benchmarks after 150 epochs of training on the COCO dataset. This is to compare my models' abilities with to minimize the number of false positives and false negatives with respect to the abilities of the pretrained model. Through examining the YOLOv5 official GitHub repository, I found the benchmarks to be P = .995, and R = 1.

4 Limitations

Before discussing results, it is important to first outline the limitations of this paper and my attempts to mitigate their effects. One large limitation I encountered was my inability to use AWS due to extenuating circumstances. Instead, I used Google Colab, which imposes an inactivity timeout to discourage users from performing long-running training tasks.
In an effort to speed up training (and thus prevent timeouts), I trained the models on a subset of the SoccerNet MOT dataset, consisting of 675 frames of one thirty-second video clip. Although the models would experience difficulty testing on different video clips due to a lack of a diversified dataset, I believe that training on the same data subset (the remaining seventy-five frames to produce a 90/10 train/test split) will provide a result that is generalizable to any dataset given proper training.

Google Colab's forced timeout is also the reason why I was unable to completely train the YOLOv5m model. As stated in the prior section, the YOLOv5m model is much larger than the YOLOv5m model; as such, it requires much more time to train. However, I will attempt to stratify the over six hours worth of training and testing data gathered before timeout to draw comparisons between this model and YOLOv5s.

5 Results and Discussion

Here are the results of the experimentation on the SoccerNet datset, which includes training for 10 epochs with a batch size of 16. The graphs in red represent the YOLOv5s model, and the graphs in blue represent the YOLOv5m model.

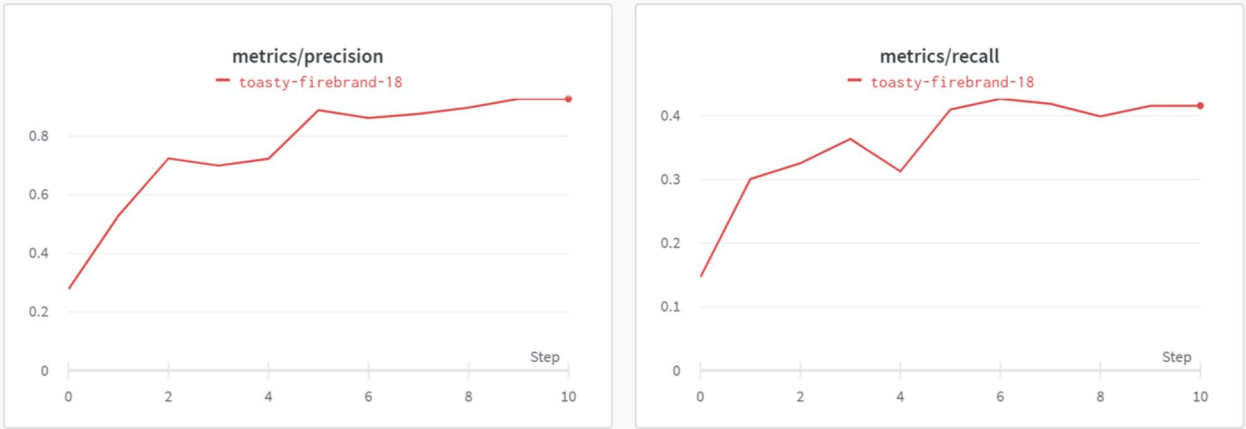

Figure 4: Precision and Recall graphs of the YOLOv5s model per epoch.

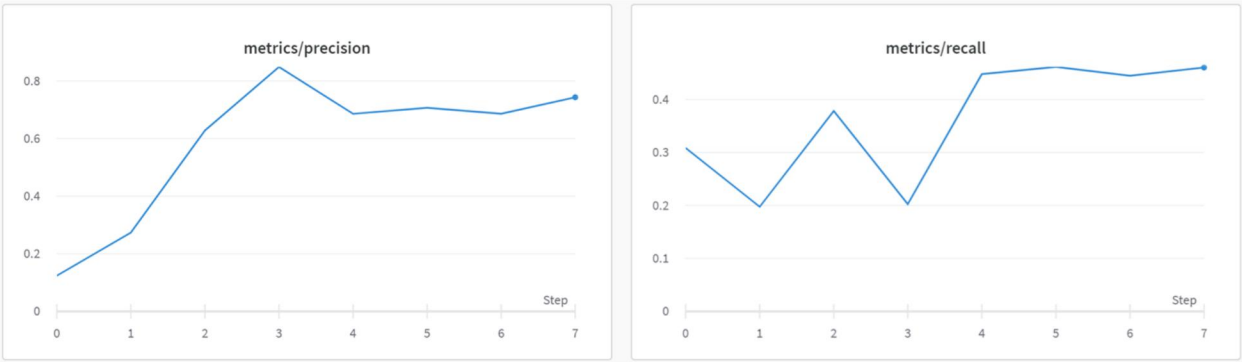

Figure 5: Precision and Recall graphs of the YOLOv5m model per epoch.

By the tenth epoch, the YOLOv5s model reached a precision of .9275. By the seventh epoch, the YOLOv5m model reached a precision of .7433. The YOLOv5s model's precision in comparison to

the benchmark of the pretrained model (.995) is especially astonishing due to the difference in the amount of epochs used to train each model (150 epochs in the benchmark model as opposed to 10 in my model). This could be because the training and test sets were too similar, and do not reflect real-world situations (which would have a multitude of teams competing against each other instead of the two same teams always competing).

By the tenth epoch, the YOLOv5s model reached a recall of .4157. By the seventh epoch, the YOLOv5m model reached a precision of .4603. These are far lower than the benchmark of 1 provided by the pretrained model; perhaps the lack of a diversified training set led to the lackluster recall results in both models.

It is worth noting the differences in the successes of each model. Although the YOLOv5s model achieved a higher precision after ten epochs than the YOLOv5m model achieved after seven epochs (which is to be expected), the YOLOv5m model achieved a higher recall after lesser epochs. This indicates that with perhaps a substantially larger number of epochs used to train both models, we may see the YOLOv5m model achieve both a higher precision and recall than the YOLOv5s model and therefore out-compete it.

6 Future Work

There are many venues for expansion of this project in the future. Although we arrived at the conclusion that the YOLOv5m model may approach the benchmark with greater precision and recall than the YOLOv5s model given enough epochs, this is merely a speculation given the current information; training with a much larger number of epochs can truly tell whether our assumptions are correct. In addition, although our assumptions may be correct that the YOLOv5m model approaches closer to the benchmark in comparison to the YOLOv5s model, the recall is still lackluster; future work could include using a more diversified training and test set (with different teams, different amount of objects, etc.) to ensure that the models are not overfitting to one or two specific teams within the current dataset. Furthermore, this paper only examined the power of two YOLOv5 models; another future area of research would be to determine how effective the other versions of the YOLOv5 model (including YOLOv5m, YOLOv5l, and YOLOv5x) would be in accomplishing the goal outlined in this essay.

References

Alexander Goettker and Karl R. Gegenfurtner. A change in perspective: The interaction of saccadic and pursuit eye movements in oculomotor control and perception. *Vision Research*, 188:283–296, 2021. doi: 10.1016/j.visres.2021.08.004.

Steve Price. Artificial intelligence is changing soccer and could decide the 2022 world cup. *Forbes*, 2022. URL `https://www.forbes.com/sites/steveprice/2022/04/11/artificial-intelligence-is-changing-soccer-and-could-decide-the-2022-world-cup/?sh=567f50c52d3e`.

Anthony Cioppa, Silvio Giancola, Adrien Deliege, Le Kang, Xin Zhou, Zhiyu Cheng, Bernard Ghanem, and Mark Van Droogenbroeck. Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. 2022.

Zhuang Li, Xincheng Tian, Xin Liu, Yan Liu, and Xiaorui Shi. A two-stage industrial defect detection framework based on improved-yolov5 and optimized-inception-resnetv2 models. *Applied Sciences*, 12(2):834, Jan 2022. ISSN 2076-3417. doi: 10.3390/app12020834. URL `http://dx.doi.org/10.3390/app12020834`.

Chunpeng Wu, Wei Wen, Tariq Afzal, Yongmei Zhang, Yiran Chen, and Hai Li. A compact dnn: Approaching googlenet-level accuracy of classification and domain adaptation. 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. URL `https://arxiv.org/abs/1405.0312`.