Deep Learning Based Approach for Ball Tracking : YOLOv5

YOLOv5 is a popular object detection algorithm that is known for its speed and accuracy. The script can perform detection on images, videos, directories, YouTube videos, webcams, and streams.

YOLO (You Only Look Once) is a modern object detection algorithm that offers several advantages over traditional tracking algorithms.

In this implementation, I have used the bounding box with the highest confidence score.

Why YOLOv5 over Classical Computer Vision Algorithms for Ball Tracking?

1. **Real-time performance**: YOLO is designed for real-time object detection, making it suitable for applications that require fast and efficient processing. It processes images in a single pass through a convolutional neural network (CNN), enabling it to achieve high detection speeds.
2. **End-to-end approach**: YOLO performs both object localization and classification in a single step, unlike traditional tracking algorithms that often require separate stages for detection and tracking. This end-to-end approach simplifies the overall pipeline and reduces computational complexity.
3. **High accuracy**: YOLO achieves competitive accuracy on various object detection benchmarks. Its modern architecture, based on deep neural networks, enables it to learn rich and discriminative features for object recognition, resulting in accurate detection results.
4. **Multi-object detection**: YOLO is capable of detecting multiple objects in an image simultaneously. It uses a grid-based approach to divide the image into regions and predicts bounding boxes and class probabilities for each region. This enables YOLO to detect and localize multiple objects of different classes in a single pass.
5. **Flexibility and generalization**: YOLO can be trained on a wide range of object categories and adapt to different domains and datasets. Its deep learning framework allows it to learn complex representations and generalize well to unseen objects or variations in object appearance.
6. **Robustness to occlusion**: YOLO handles occlusion reasonably well due to its grid-based approach and anchor box mechanism. Even if parts of an object are occluded, YOLO can still detect and classify the visible parts accurately, which can be challenging for traditional tracking algorithms.
7. **Ease of implementation**: YOLO is relatively easy to implement and use, thanks to the availability of open-source implementations and pre-trained models. This makes it accessible to a wider range of developers and researchers interested in object detection.

Invoking YOLOv5 for Ball Tracking

```
from email.mime import image
from yolov5.detect import run
import numpy as np
```

```
from imutils.video import VideoStream


if __name__ == "__main__":
    source = "D:/Adithya/GRE/Applications/University of Pennsylvania/Coding Challenge/ball_tracking_video.mp4"
    csv_output = r"D:\\Adithya\\GRE\\Applications\\University of Pennsylvania\\Coding Challenge\\Box_coordinates_Yolo.csv"
    video_output = "D:\\Adithya\\GRE\\Applications\\University of Pennsylvania\\Coding Challenge\\ball_tracking_Yolo.mp4"
    run(weights='./yolov5/yolov5s.pt', source=source, save_txt=True, classes=32, conf_thres=0.5)
```

`Other Approaches Tried : DeepSORT with YOLOv4 & v5 (Implemented, result was Less Robust and class ID ambiguities)`