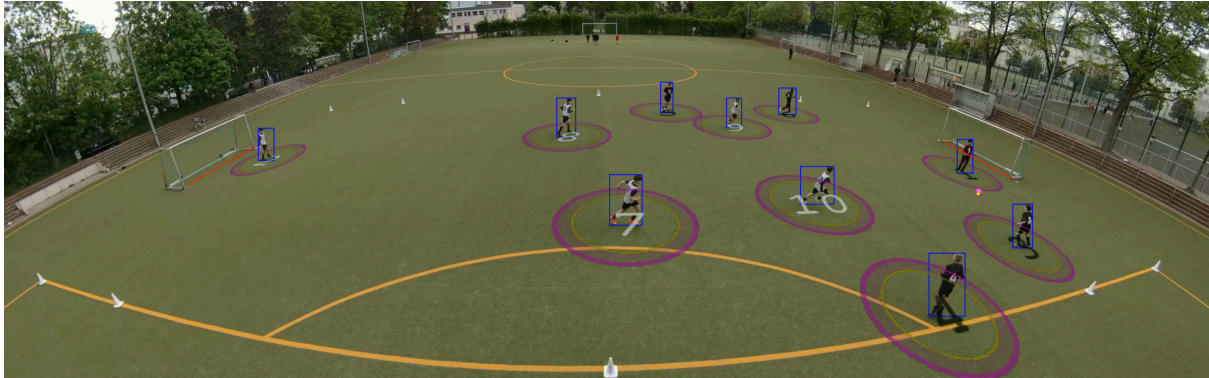
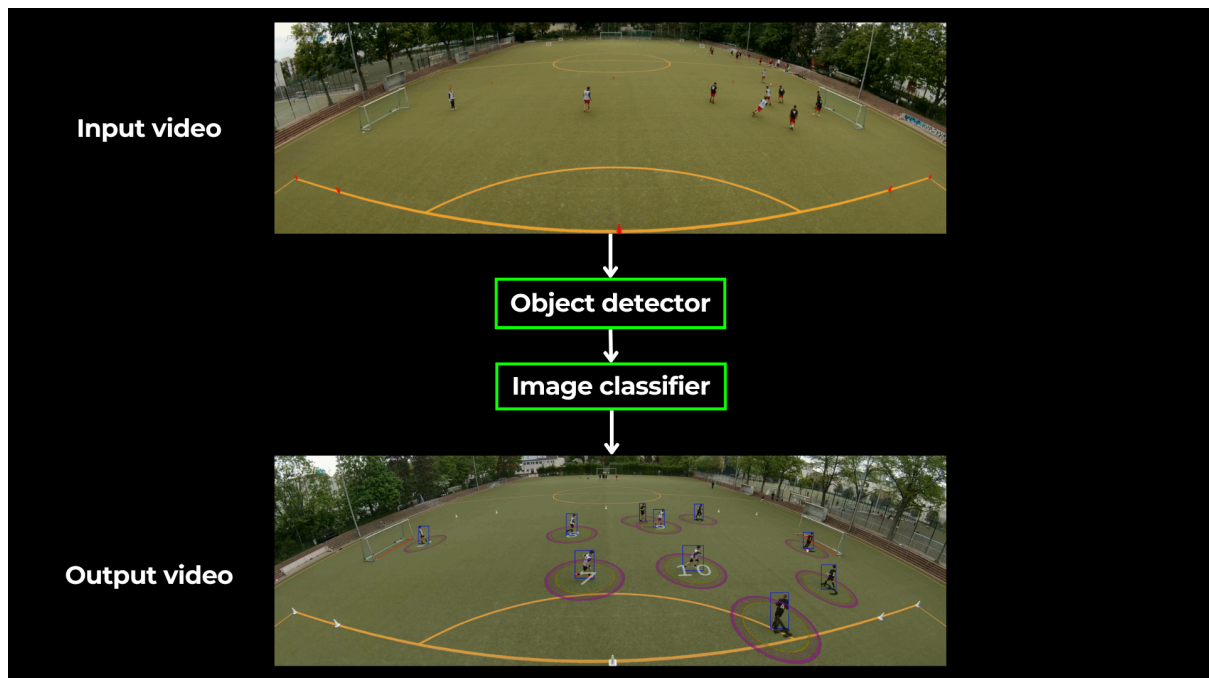


Goal: Xây dựng 1 AI pipeline nhận diện và phân loại số áo cầu thủ. Pipeline bao gồm 2 models: Object Detection và Image Classification.



Minh họa output của Pipeline

Mô tả: Hình phía dưới mô tả pipeline các bạn cần xây dựng



Minh họa Pipeline

Dưới đây là yêu cầu và hướng dẫn cho từng thành phần trong pipeline

1.Object detection

Đây là model đầu tiên trong pipeline. Model này nhận từng frame từ video như là input đầu vào. Với mỗi frame mô hình sẽ phát hiện vị trí của các cầu thủ (và quả bóng). Các bạn có 2 sự lựa chọn:

- Model chỉ phát hiện cầu thủ (1 class)
- Model phát hiện cầu thủ và quả bóng (2 classes)

Có nhiều cách để có thể thực hiện model này. Một trong số những cách phổ biến nhất là sử dụng repository Yolov5/Yolov8/Yolov11 của Ultralytics

Link to repo: <https://github.com/ultralytics/ultralytics> (Yolov11)

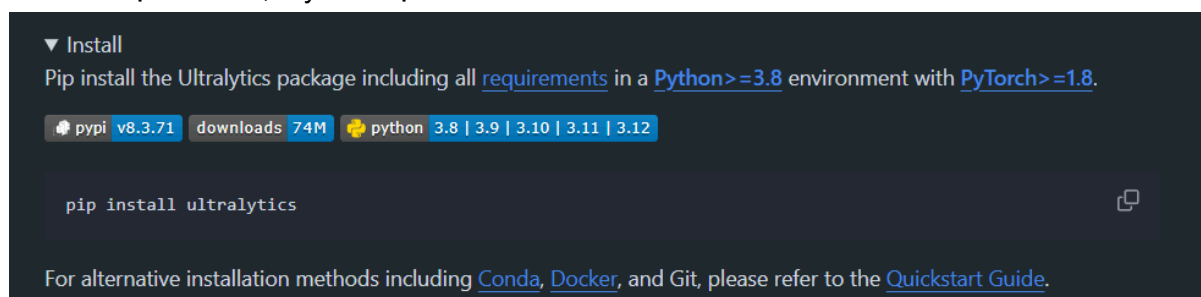
Bước 1: Các bạn hãy đọc hướng dẫn cách sử dụng repo dành cho bài toán Object Detection ở <https://docs.ultralytics.com/tasks/detect/>

Nếu đây là lần đầu các bạn nhìn thấy repository này, hãy đọc thật kỹ, vì đây là 1 repository vô cùng nổi tiếng và được sử dụng rất nhiều trong thực tế trong Computer Vision

Bước 2: Để có thể sử dụng được repository này, phần mất nhiều thời gian nhất là chuẩn bị Dataset theo format mà các model Yolo của Ultralytics yêu cầu. Hướng dẫn về cách thức chuẩn bị dữ liệu các bạn có thể đọc ở đây: <https://docs.ultralytics.com/datasets/detect/>

Bước 3: Sau khi đã hiểu về cách thức chuẩn bị dữ liệu, các bạn hãy viết 1 script Python để có thể chuyển đổi dữ liệu về Football Dataset thành dạng mà Yolo yêu cầu. Gợi ý: Các bạn sẽ cần tách từng frame từ các video ra, và với mỗi frame các bạn cũng sẽ cần trích xuất thông tin tương ứng từ file annotation .json. Sau khi tách xong hãy kiểm tra kỹ để đảm bảo rằng các bạn không gán nhầm annotation của 1 frame với ảnh của 1 frame khác

Bước 4: Sau khi đã chuẩn có data đúng theo format mà Yolo yêu cầu, các bạn có thể bắt đầu quá trình huấn luyện. Có 1 vài cách để các bạn có thể huấn luyện mô hình. Nhưng dù là với các nào thì các bạn cũng sẽ cần cài đặt các thư viện cần thiết. Hướng dẫn cài đặt có ở trang chủ. Đơn giản nhất thì các bạn có thể cài qua pip. Các bạn cũng có thể cài thông qua Conda hoặc Docker, tùy các bạn



Hướng dẫn cài đặt từ README của Yolo

Bước 5: Để huấn luyện mô hình, các bạn có 3 cách

1. Qua Python

Train

Train YOLO11n on the COCO8 dataset for 100 [epochs](#) at image size 640. For a full list of available arguments see the [Configuration](#) page.

Example

Python

CLI

```
from ultralytics import YOLO

# Load a model
model = YOLO("yolo11n.yaml") # build a new model from YAML
model = YOLO("yolo11n.pt") # load a pretrained model (recommended for training)
model = YOLO("yolo11n.yaml").load("yolo11n.pt") # build from YAML and transfer weights

# Train the model
results = model.train(data="coco8.yaml", epochs=100, imgsz=640)
```

2. Qua CLI

Train

Train YOLO11n on the COCO8 dataset for 100 [epochs](#) at image size 640. For a full list of available arguments see the [Configuration](#) page.

Example

Python

CLI

```
# Build a new model from YAML and start training from scratch
yolo detect train data=coco8.yaml model=yolo11n.yaml epochs=100 imgsz=640

# Start training from a pretrained *.pt model
yolo detect train data=coco8.yaml model=yolo11n.pt epochs=100 imgsz=640

# Build a new model from YAML, transfer pretrained weights to it and start training
yolo detect train data=coco8.yaml model=yolo11n.yaml pretrained=yolo11n.pt epochs=100 imgsz=640
```

3. Qua Ultralytics Hub

<https://hub.ultralytics.com/home>

Chú ý, đừng sử dụng các parameter mặc định (epoch=100, imgsz=640). Hãy xem danh sách các parameter ở <https://docs.ultralytics.com/usage/cfg/#tasks> và thử nghĩ xem, nên điều chỉnh giá trị các parameter như thế nào

Sau khi quá trình huấn luyện hoàn tất, các bạn sẽ có 1 model có khả năng phát hiện và định vị vị trí của các cầu thủ (và quả bóng).

2. Image Classification

Sau khi từng frame đi qua model đầu tiên - Object detection, ta đã biết vị trí của từng cầu thủ (~10 cầu thủ trong mỗi khung hình). Model thứ hai sẽ nhận input là hình ảnh của từng cầu thủ, và đưa ra dự đoán về số áo của cầu thủ đó. Đây là 1 model Image Classification. Model này tương tự những gì chúng ta đã thực hành trên lớp với bộ Animal. Các bạn có thể dùng lại training script chúng ta đã viết cho bộ Animal. Class Dataset dành cho bộ Football

các bạn đã làm ở tuần 14 rồi. Việc còn lại chỉ là kết nối training script với Dataset mới là xong

Các bạn hãy chú ý, với model nhận diện số áo này, mình không bắt buộc các bạn phải xây dựng được 1 model phân biệt được mọi số áo khác nhau. Trong data của chúng ta, có nhiều sample của các số từ 1 đến 10 hơn các số còn lại. Các bạn có thể định nghĩa các class như sau:

- Class 0: Dành cho những cầu thủ mà ta không nhìn rõ được số áo
- Class 1-10: Dành cho các số từ 1 đến 10
- Class 11: Dành cho các số từ 11 trở lên

Sau khi xem và phân tích dữ liệu (e.g. thống kê số lượng sample của từng số), nếu các bạn có cách phân loại class nào khác, thì cũng hoàn toàn ok

Bonus task: Nếu các bạn đã xây dựng được 1 model Image Classification để phân loại số áo của cầu thủ, hãy thử nghĩ xem, làm sao để có thể xây dựng được 1 mô hình Image Classification để vừa phân loại được số áo, vừa phân loại được màu áo. Chú ý về màu áo chúng ta chỉ có 2 loại là màu đen và màu trắng. Nếu dùng 2 model Image Classification riêng biệt cho 2 task thì rất đơn giản. Nhưng nếu muốn chỉ dùng duy nhất 1 model mà hoàn thành được 2 task thì phải làm sao ? Các bạn có thể chỉ cần nêu ý tưởng, không nhất thiết phải code hoàn chỉnh