

Object Detection

Hoàng-Nguyên Vũ

1 Lý Thuyết Cơ Bản

1.1 1. Giới thiệu về YOLOv11

- **YOLOv11** là một phiên bản mới của họ mô hình YOLO (*You Only Look Once*), nổi bật với tốc độ nhanh, độ chính xác cao, và khả năng generalize tốt hơn trên các bài toán phát hiện vật thể (object detection).
- YOLOv11 cải thiện trên các điểm sau:
 - Backbone mới hiệu quả hơn.
 - Neck và Head cải tiến để tối ưu hóa multi-scale feature fusion.
 - Kỹ thuật label assignment và augmentation tiên tiến.

1.2 2. Bài toán phát hiện Người và Mũ Bảo Hiểm

- Mục tiêu: Xác định vị trí (bounding box) và phân loại đối tượng là Người hoặc Mũ Bảo Hiểm trong ảnh hoặc video.
- Ứng dụng thực tế: An toàn giao thông, giám sát công trường, quản lý tuân thủ quy định.

1.3 3. Quy trình cơ bản sử dụng YOLOv11

1. **Chuẩn bị môi trường:** Cài đặt thư viện cần thiết như `ultralytics`, `torch`, `opencv-python`.
2. **Tải mô hình YOLOv11:** Dùng mô hình pretrained hoặc fine-tune thêm với dataset chuyên biệt.
3. **Dự đoán (Inference):** Nạp ảnh/video và dùng mô hình để phát hiện người và mũ bảo hiểm.
4. **Xử lý đầu ra:** Vẽ bounding boxes, gán nhãn đối tượng.

1.4 4. Một số chú ý khi detect Người và Mũ Bảo Hiểm

- Chọn ngưỡng confidence (`confidence threshold`) hợp lý, thường khoảng 0.25 - 0.5.
- Nếu cần phát hiện riêng biệt (ví dụ, người đội mũ vs. người không đội), cần fine-tune thêm mô hình hoặc thêm post-processing logic.

2 Bài Tập Thực Hành

Data tải tại đây

Bài 1: Cài đặt thư viện

Hãy cài đặt các thư viện cần thiết cho bài tập:

```
1 pip install ultralytics opencv-python
```

Gợi ý: Sử dụng Google Colab hoặc local environment.

Bài 2: Tải mô hình YOLOv11 pretrained

Viết đoạn mã tải mô hình YOLOv11 pretrained.

```
1 from ultralytics import YOLO
2
3 model = YOLO('yolov11.pt') # hoặc dùng phiên bản custom
```

Bài 3: Fine-tune mô hình với dữ liệu mũ bảo hiểm riêng

- Thu thập hoặc sử dụng dataset như Helmet Detection Dataset.
- Chuẩn hóa dữ liệu theo định dạng YOLOv11 (.txt label files).
- Fine-tune mô hình với lệnh:

```
1 yolo train model=yolov11.pt data=helmet.yaml epochs=50 imgsz=640
```

Bài 4: Phát hiện Người và Mũ Bảo Hiểm trên ảnh

Viết chương trình đọc một ảnh và phát hiện người và mũ bảo hiểm trong ảnh.

```
1 results = model.predict('input_image.jpg', conf=0.5)
2
3 # Hiển thị ảnh kết quả
4 results[0].show()
```

Yêu cầu: Hiển thị rõ bounding box và nhãn tên đối tượng.

Bài 5: Phát hiện Người và Mũ Bảo Hiểm trên video

Viết chương trình chạy detection trên file video.

```
1 results = model.predict(source='input_video.mp4', stream=True)
2
3 for result in results:
4     frame = result.plot()
5     # Hiển thị từng khung hình hoặc lưu lại
```

Yêu cầu: Hiển thị bounding boxes theo thời gian thực.