

ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN KHOA HỌC MÁY TÍNH

Báo cáo bài thực hành 4

Đề tài: THỰC HÀNH VỚI YOLO

Môn học: Xử lý ảnh và video số

Sinh viên thực hiện:

Lê Châu Hữu Thọ (22120350)

Giáo viên hướng dẫn:

Thầy Lý Quốc Ngọc

Thầy Phạm Minh Hoàng

Thầy Nguyễn Mạnh Hùng

Thầy Phạm Thanh Tùng

Ngày 27 tháng 12 năm 2024



Mục lục

1	Tự đánh giá	1
1.1	Bảng đánh giá	1
1.2	Các nội dung chưa hoàn thành	1
2	Giới thiệu	1
2.1	Giới thiệu mô hình	1
2.2	Dánh giá hiệu suất mô hình	2
3	Bộ dữ liệu	2
3.1	Giới thiệu	2
3.2	Mô tả dữ liệu	2
4	Hướng dẫn cài đặt và huấn luyện mô hình bằng Google Colab	3
4.1	Kiểm tra môi trường	3
4.2	Cấp quyền cho Drive	3
4.3	Tải tập tin pretrain	4
4.4	Cài đặt bộ dữ liệu	4
4.4.1	Định dạng cây thư mục dữ liệu	4
4.4.2	Cấu trúc tập tin .yaml	5
4.4.3	Cấu trúc tập tin nhãn dán	5
4.5	Huấn luyện mô hình	6
5	Đánh giá mô hình	7
5.1	Thông số huấn luyện	7
5.2	Kết quả huấn luyện	7
5.2.1	Thông tin chung	7
5.2.2	Kết quả đánh giá trên tập kiểm tra	8
5.2.3	Thời gian xử lý	8
5.2.4	Nhận xét	9
5.3	Kết quả dự đoán của mô hình	9

6 Mã nguồn và tài liệu tham khảo

10

Danh sách bảng

1	Tự đánh giá công việc	1
2	Hiệu suất của mô hình trên tập kiểm tra với các lớp khác nhau	8

Danh sách hình vẽ

1	Dánh giá hiệu suất các mô hình theo độ trễ (Latency) và số tham số (Parameters)	2
2	Chuyển sang chế độ GPU	3
3	Kết quả sau khi huấn luyện	7
4	Kết quả dự đoán của mô hình sau khi huấn luyện	9

1 Tự đánh giá

1.1 Bảng đánh giá

Viết mã nguồn đầy đủ	100%
Viết báo cáo đầy đủ	100%
Thu thập và xử lý dữ liệu	100%
Huấn luyện mô hình	100%
Xuất ảnh minh họa dự đoán sau huấn luyện	100%

Bảng 1: Tự đánh giá công việc

1.2 Các nội dung chưa hoàn thành

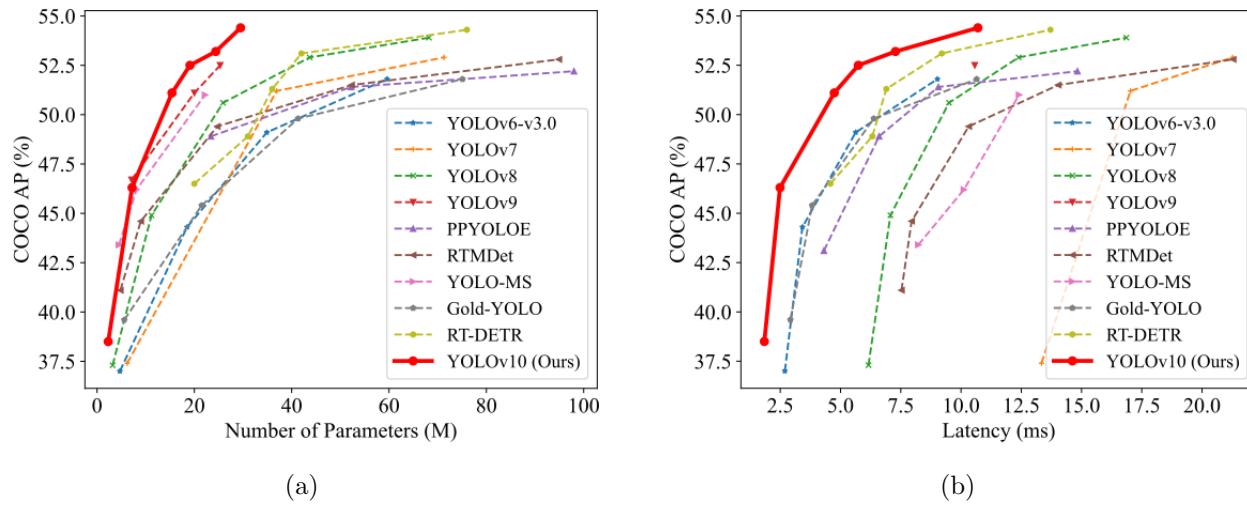
- Bài tập điểm cộng

2 Giới thiệu

2.1 Giới thiệu mô hình

Mô hình được sử dụng trong bài thực hành là **YOLOv10**, đây là phiên bản **YOLO** được phát hành vào tháng 5/2024. Lý do sử dụng mô hình **YOLOv10** thay cho **YOLOv4** là do sự tương thích của **YOLOv10** với phiên bản hiện tại của Colab là 3.10.12 (so với **YOLOv4** là 3.6/3.7). Đồng thời tránh việc xung đột giữa các thư viện được yêu cầu cài đặt so với các thư viện hiện có trong Colab.

2.2 Đánh giá hiệu suất mô hình



Hình 1: Đánh giá hiệu suất các mô hình theo độ trễ (Latency) và số tham số (Parameters)

Các mô hình được đánh giá trên tập dữ liệu COCO. Về số lượng tham số, YOLOv10 cho thấy độ chính xác cao nhất ở mức tham số nhỏ và vừa. Đồng thời ở giá trị độ trễ, YOLOv10 cũng cho độ chính xác tốt nhất ở các mức độ trễ thấp nhất. Điều này cho thấy YOLOv10 đã cải thiện đáng kể về cả độ chính xác, hiệu năng và tốc độ xử lí so với các mô hình tiền nhiệm.

3 Bộ dữ liệu

3.1 Giới thiệu

Bộ dữ liệu được sử dụng trong bài thực hành là bộ dữ liệu về đồ nội thất. Bộ dữ liệu được sử dụng để huấn luyện các mô hình xác định một số loại vật liệu nội thất phổ biến từ trong ảnh.

3.2 Mô tả dữ liệu

- **Tác giả:** Minoj Selvaraj
- **Chứng chỉ:** CC BY 4.0
- **Tổng số dữ liệu:** 689
- **Phân bổ dữ liệu theo tập train/valid/test:**

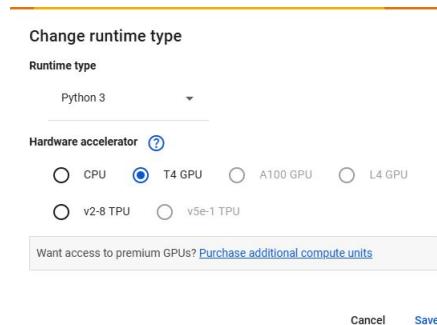
- Train: 454
- Valid: 161
- Test: 74

- Số lớp nhãn dán: 3: Chair, Sofa, Table
- Định dạng: tương thích với mô hình YOLOv7 trở lên
- Dữ liệu không có nhãn: 0

4 Hướng dẫn cài đặt và huấn luyện mô hình bằng Google Colab

4.1 Kiểm tra môi trường

Trong giao diện chính của Colab, chọn mục **Runtime → Change runtime type → T4 GPU** để chuyển sang chế độ GPU



Hình 2: Chuyển sang chế độ GPU

Có thể kiểm tra lại thông tin GPU bằng câu lệnh

```
!nvidia-smi
```

4.2 Cấp quyền cho Drive

Cấp quyền để Colab truy cập vào thư mục Drive của người dùng thông qua đoạn mã sau:

```

1 from google.colab import drive
2 drive.mount('/content/drive')

```

4.3 Tải tập tin pretrain

Tập tin pretrain dùng trong mô hình huấn luyện. Người dùng chỉ cần chọn tập tin pretrain thích hợp và huấn luyện tiếp bộ trọng số có trong tập tin đó để tiết kiệm thời gian thay vì phải huấn luyện mô hình từ con số 0. YOLOv10 cung cấp nhiều tập tin pretrain khác nhau tùy theo mục đích người sử dụng. Dưới đây là đoạn mã ví dụ tải tập tin pretrain YOLOv10 - N:

```

1 %cd '/content/drive/MyDrive'
2 !mkdir YOLOv10
3 %cd '/content/drive/MyDrive/YOLOv10'
4 !mkdir pretrain
5 %cd pretrain
6 !wget -P /content https://github.com/jameslahm/yolov10/releases/download/v1.0/
      yolov10n.pt

```

4.4 Cài đặt bộ dữ liệu

4.4.1 Định dạng cây thư mục dữ liệu

Đối với các thư mục dữ liệu được sử dụng cho mô hình YOLOv7 trở lên, cây thư mục cần phải có định dạng như sau:

```

1 /train_data
2   /train
3     /images
4       img1.jpg
5       img2.jpg
6       ...
7     /labels
8       img1.txt
9       img2.txt
10      ...
11   /val
12   ...

```

```

13 /test
14 ...
15 /data.yaml

```

4.4.2 Cấu trúc tập tin .yaml

Trong các mô hình YOLOv7 trở lại đây, tập tin .yaml là một tập tin có chức năng hướng dẫn mô hình cần phải trỏ tới thư mục nào để lấy dữ liệu dùng để huấn luyện, đồng thời xác định các lớp nhãn dán có trong mô hình. Cấu trúc một tập tin .yaml gồm có các thành phần như sau:

- **train:** Đường dẫn đến thư mục chứa ảnh trong tập train. Ví dụ: /train/images
- **valid:** Đường dẫn đến thư mục chứa ảnh trong tập valid. Ví dụ: /valid/images
- **test:** Đường dẫn đến thư mục chứa ảnh trong tập test. Ví dụ: /test/images
- **nc:** số lớp nhãn dán
- **names:** danh sách tên các lớp nhãn dán.

4.4.3 Cấu trúc tập tin nhãn dán

Nội dung của một file nhãn dán trong bộ dữ liệu có thể gồm một số dòng nhất định, trong đó mỗi dòng tương ứng cho nhãn dán của một đối tượng. Nội dung nhãn dán của một đối tượng có cấu trúc như sau

```

1 <class_id> <x_center> <y_center> <width> <height>

```

Trong đó:

- **class_id:** ID của lớp đối tượng, là giá trị nguyên bắt đầu từ số 0, phải khớp với thứ tự trong danh sách **names** trong file .yaml
- **x_center:** Tọa độ tâm của đối tượng theo chiều ngang.
- **y_center:** Tọa độ tâm của đối tượng theo chiều dọc.
- **width:** Độ rộng của bounding box.
- **height:** Độ cao của bounding box.

Lưu ý: Tên tập tin nhãn dán phải khớp với tên tập tin ảnh. Ngoại trừ **class_id**, các giá trị khác của đối tượng phải được chuẩn hóa về khoảng (0, 1) theo tỉ lệ độ dài/rộng của ảnh.

Ví dụ: Một ảnh có kích thước rộng/dài là 800x600 và có thông tin đối tượng như sau:

- ID: 0
- Tọa độ chưa được chuẩn hóa: (400, 300, 200, 150)
- Chuẩn hóa tọa độ tâm: $x_center = 400/800 = 0.5$, $y_center = 300/600 = 0.5$
- Chuẩn hóa kích thước hộp: $width = 200/800 = 0.25$, $height = 150/600 = 0.25$
- Nhãn dán đã chuẩn hóa: (0, 0.5, 0.5, 0.25, 0.25)

4.5 Huấn luyện mô hình

Để huấn luyện mô hình, ta chỉ cần gọi thư viện YOLOv10 đã tải về trước đó, sau đó gọi lệnh **!yolo** kèm với đường dẫn chứa tập tin pretrain, tập tin **.yaml** và một số thông số khác. Dưới đây là ví dụ về đoạn mã huấn luyện mô hình:

```
1 import os
2 from ultralytics import YOLOv10
3 os.environ["WANDB_MODE"] = "disabled"
4 %cd '/content/drive/MyDrive/YOLOv10'
5 !yolo task=detect mode=train epochs=50 batch=32 plots=True model='./pretrain/
    yolov10n.pt' data='/content/drive/MyDrive/YOLOv10/train_data/data.yaml'
```

Nếu mô hình được cài đặt và huấn luyện thành công, kết quả trả về sẽ như trong hình minh họa dưới đây.

Báo cáo bài thực hành 4

```

Epoch GPU_mem box_om cls_om dfl_om box_oo cls_oo dfl_oo Instances Size
47/50 5.63G 0.2561 0.2623 0.9007 0.2727 0.9146 0.9173 6 640: 100% 15/15 [00:08<00:00, 1.68it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% 3/3 [00:02<00:00, 1.02it/s]
all 161 161 0.865 0.852 0.947 0.849

Epoch GPU_mem box_om cls_om dfl_om box_oo cls_oo dfl_oo Instances Size
48/50 5.64G 0.2771 0.2663 0.9036 0.2934 0.9173 0.9173 6 640: 100% 15/15 [00:10<00:00, 1.40it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% 3/3 [00:02<00:00, 1.26it/s]
all 161 161 0.875 0.895 0.958 0.853

Epoch GPU_mem box_om cls_om dfl_om box_oo cls_oo dfl_oo Instances Size
49/50 5.63G 0.2353 0.2473 0.8701 0.2417 0.924 0.8786 6 640: 100% 15/15 [00:12<00:00, 1.23it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% 3/3 [00:02<00:00, 1.03it/s]
all 161 161 0.937 0.928 0.981 0.875

Epoch GPU_mem box_om cls_om dfl_om box_oo cls_oo dfl_oo Instances Size
50/50 5.64G 0.2321 0.241 0.8625 0.2397 0.2942 0.8718 6 640: 100% 15/15 [00:11<00:00, 1.26it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% 3/3 [00:02<00:00, 1.16it/s]
all 161 161 0.946 0.935 0.982 0.876

50 epochs completed in 0.237 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 5.8MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 5.8MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.1.34 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv10n summary (fused): 285 layers, 2695586 parameters, 0 gradients, 8.2 GFLOPs
    Class Images Instances Box(P R mAP50 mAP50-95): 100% 3/3 [00:02<00:00, 1.01it/s]
    all 161 161 0.926 0.954 0.982 0.877
    Sofa 161 23 0.882 0.975 0.977 0.893
    Table 161 138 0.97 0.933 0.987 0.86

Speed: 0.3ms preprocess, 5.5ms inference, 0.0ms loss, 0.1ms postprocess per image
Results saved to runs/detect/train2
💡 Learn more at https://docs.ultralytics.com/modes/train

```

Hình 3: Kết quả sau khi huấn luyện

5 Đánh giá mô hình

5.1 Thông số huấn luyện

Mô hình được huấn luyện với các thông số như sau:

- Tập tin pretrain: YOLOv10 - N
- Số batch: 32
- Số epoch: 50

5.2 Kết quả huấn luyện

5.2.1 Thông tin chung

- Thời gian huấn luyện: 14.22 phút
- Kích thước file pretrain sau huấn luyện: 5.8 MB
- Số lớp: 258
- Số tham số: 2,695,586

5.2.2 Kết quả đánh giá trên tập kiểm tra

Name	No. Img	No. Ins	Precision (P)	Recall (R)	mAP@50	mAP@50-95
All	161	161	0.926	0.954	0.982	0.877
Sofa	161	23	0.882	0.975	0.977	0.893
Table	161	138	0.970	0.933	0.987	0.860

Bảng 2: Hiệu suất của mô hình trên tập kiểm tra với các lớp khác nhau

Giải thích ý nghĩa các ô trong bảng:

- **Name:** Tên lớp
- **No. Img:** Số ảnh được kiểm tra
- **No. Ins:** Số đối tượng (instance) được kiểm tra
- **Precision (P):** khả năng phân loại đúng khi mô hình dự đoán.
- **Recall (R):** khả năng phát hiện các đối tượng thực tế.
- **mAP@50:** độ chính xác trung bình tại ngưỡng IoU 50%.
- **mAP@50-95:** độ chính xác trung bình trên các ngưỡng IoU từ 50% đến 95%

5.2.3 Thời gian xử lý

Kết quả đo thời gian xử lí trung bình trên mỗi ảnh:

- **Preprocess** - tiền xử lí ảnh trước khi vào mô hình: 0.3 ms.
- **Inference** - thực hiện dự đoán trên dữ liệu đầu vào: 5.5 ms.
- **Loss** - đo lường sự khác biệt giữa dự đoán của mô hình và giá trị thực tế: 0.0 ms.
- **Postprocess** - xử lý và xuất ảnh đã dự đoán: 0.1 ms.

Tổng thời gian xử lý: 5.9 ms/ảnh

5.2.4 Nhận xét

Ưu điểm:

- Độ chính xác cao với mAP@50 đạt 98.2%.
- Tốc độ xử lý nhanh (5.9 ms/ảnh), phù hợp với ứng dụng thời gian thực.
- Kích thước mô hình nhỏ gọn, dễ triển khai trên thiết bị hạn chế tài nguyên.

Hạn chế:

- Thiếu tập ảnh đối tượng **chair** trong tập test, không thể đánh giá hiệu suất của lớp này.
- mAP@50-95 thấp hơn đáng kể (chỉ đạt 87.7% so với 98.2% ở mAP@50).
- Precision của lớp Sofa thấp hơn Recall, cần giảm số lượng dự đoán thừa (false positives).

5.3 Kết quả dự đoán của mô hình

Dưới đây là ảnh đã được mô hình dự đoán sau khi huấn luyện



Hình 4: Kết quả dự đoán của mô hình sau khi huấn luyện

6 Mã nguồn và tài liệu tham khảo

Mã nguồn Colab

Link dataset chính thống

Link dataset đã định dạng theo YOLOv7

Github YOLOv10

Paper YOLOv10