Báo cáo tiến độ

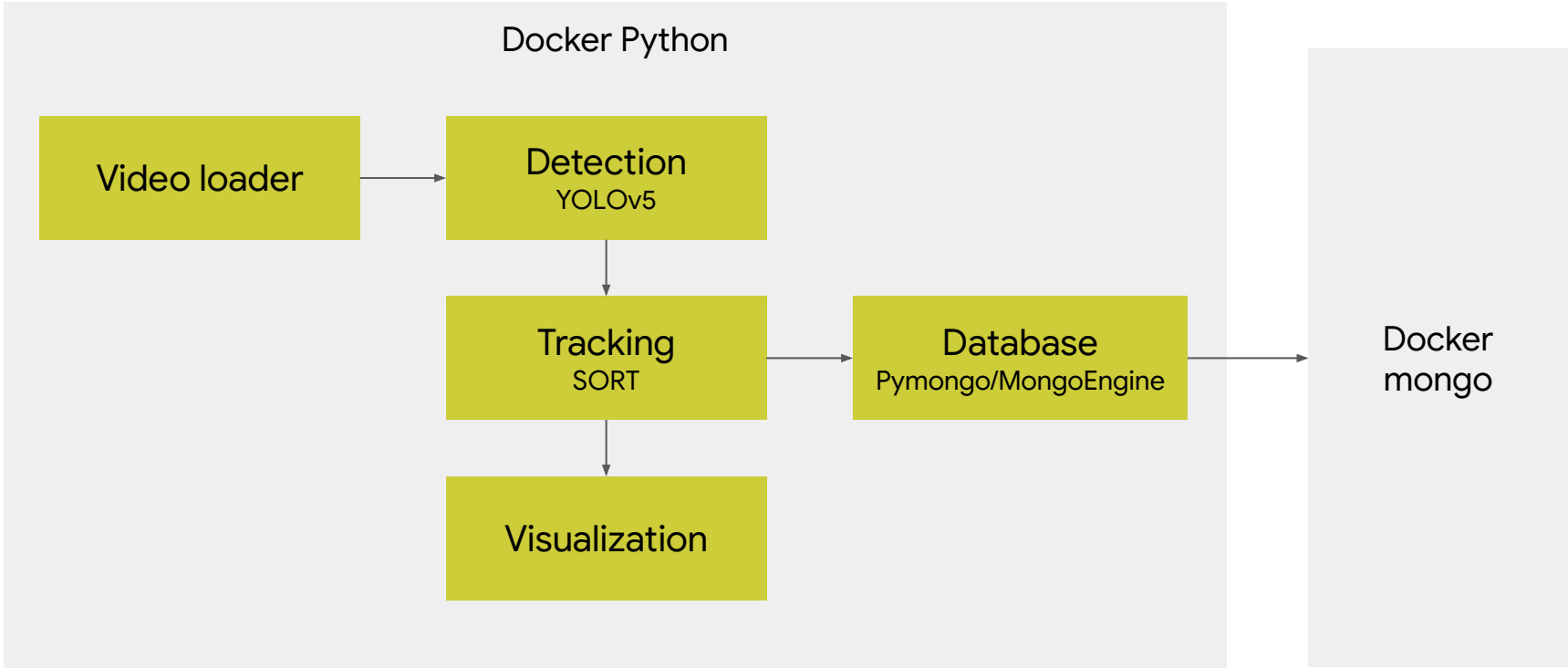
Single Camera Tracking

Lần 1

Tổng quan

1. Nội dung công việc
2. Lý thuyết
3. Kết quả thử nghiệm
4. Demo

1. Nội dung công việc



2. Lý thuyết

2.1. Detection

1. YOLOv5
2. mAP

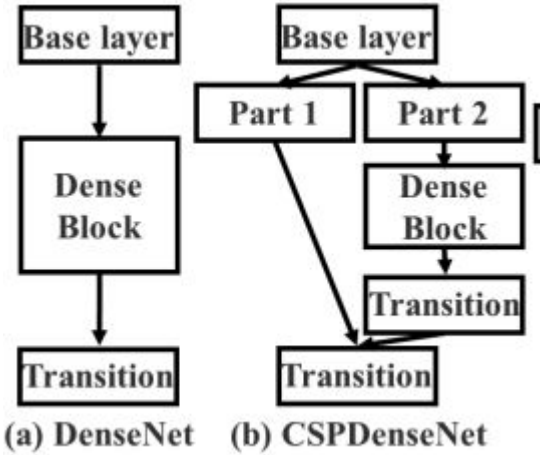
2.1.1. YOLOv5

Backbone: CSPResBlock

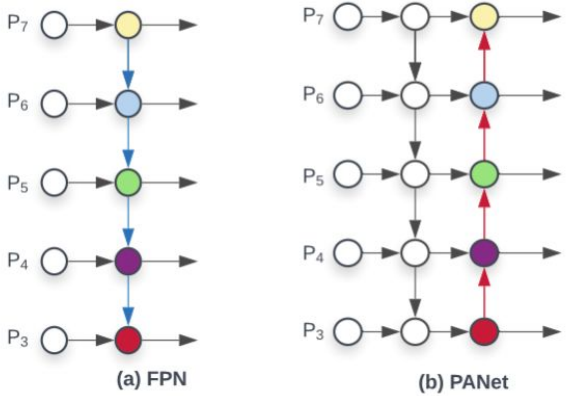
Neck (generate feature pyramids): SPP(F) + PANet

Head (generates final output vectors): giữ nguyên từ v3

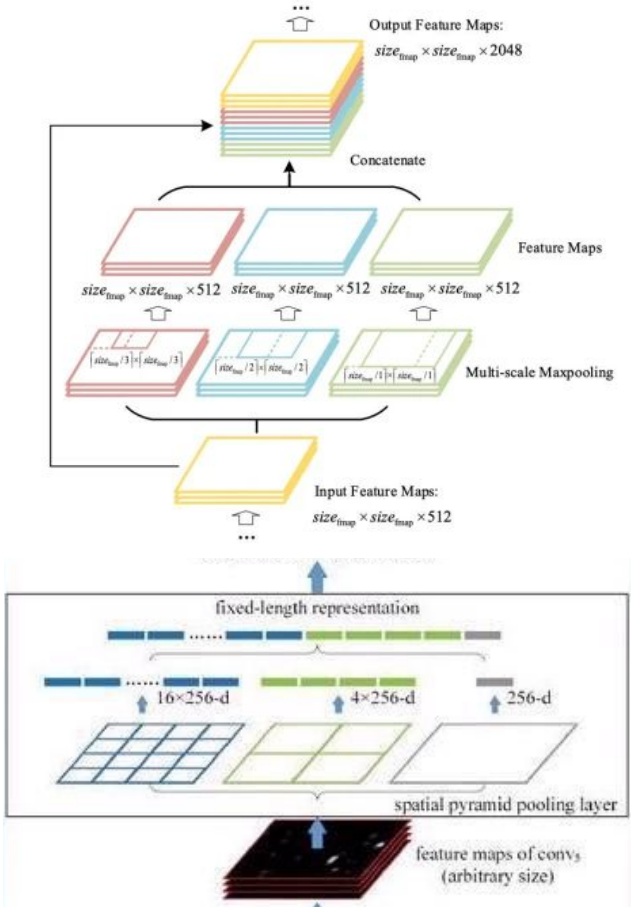
Activation: Leaky RELU for hidden layers, Sigmoid for output layer



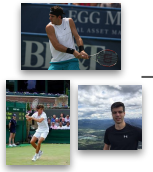
=> Giảm FLOPS, trong khi giàu thông tin gradient hơn

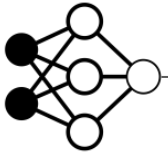


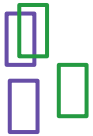
Ở FPN, thông tin từ từ sâu -> nông chỉ qua vài lớp, nhưng nông -> sâu phải qua 100 lớp => PANet tạo short path từ nông -> sâu

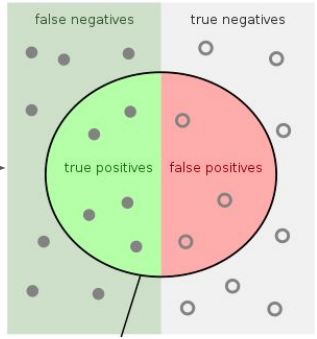


2.1.2. mAP





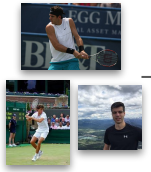


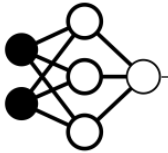


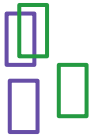
$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

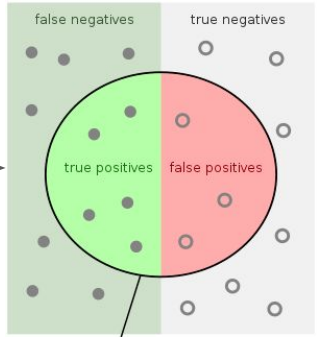
2.1.2. mAP





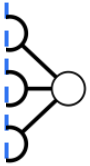


IoU threshold

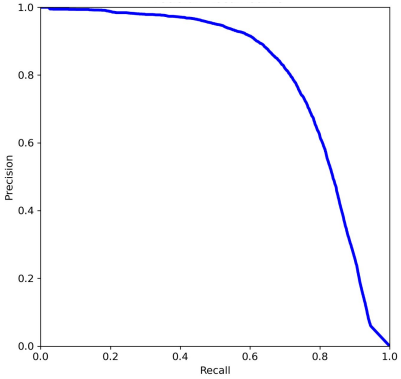


$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$



confidence threshold



AP := area under the PR curve

2.2. Tracking

1. SORT
2. MOTA, MOTP, HOTA

2.2.1. SORT

Đặc điểm:

- 2 bước: Detection & Association
- Detection: Faster R-CNN
- Association: Kalman filter & Hungarian method
    - Chỉ sử dụng thông tin vị trí bounding box, bỏ qua visual features
    - Chỉ dự đoán dựa trên 1 khung hình trước
    - Kalman filter giả thiết vật chuyển động với vectơ vận tốc không đổi
    - Hungarian chỉ dựa trên IoU giữa estimation ở khung hình trước với detection ở khung hình hiện tại
    - Giữ track ID bị bỏ qua để đổi lấy tốc độ realtime
- Không xét yếu tố FPS

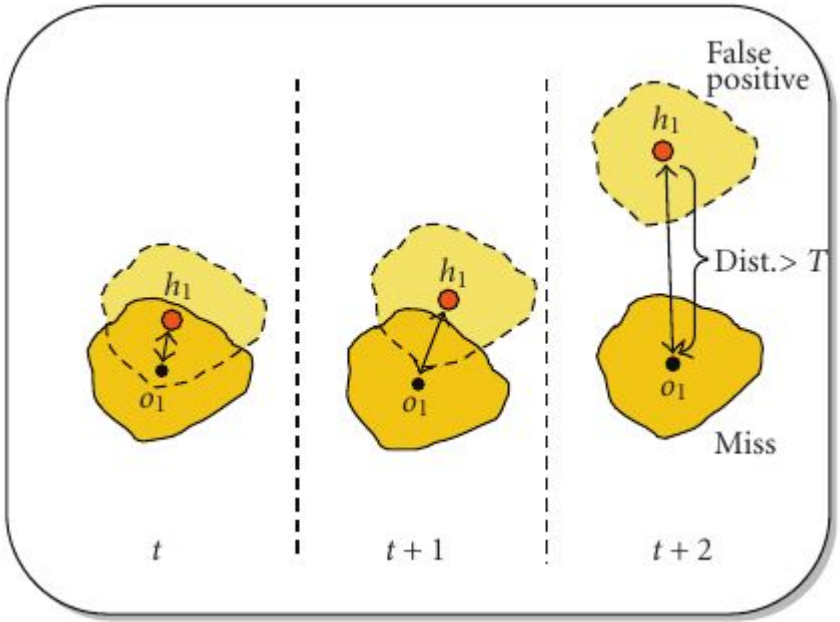
2.2.1. SORT

Tham số:

- Trạng thái: $[u, v, s, r, u', v', s']^T$ (aspect ratio không đổi)
- `iou_threshold=0.3`: loại bỏ 1 cặp gán ở bước Hungarian
- `min_hits=3`: số lần được detect liên tục tối thiểu để hình thành 1 track
- `max_age=1`: số lần không được detect liên tục để xoá 1 track

2.2.2. MOTA, MOTP, HOTA

MOTA, MOTP



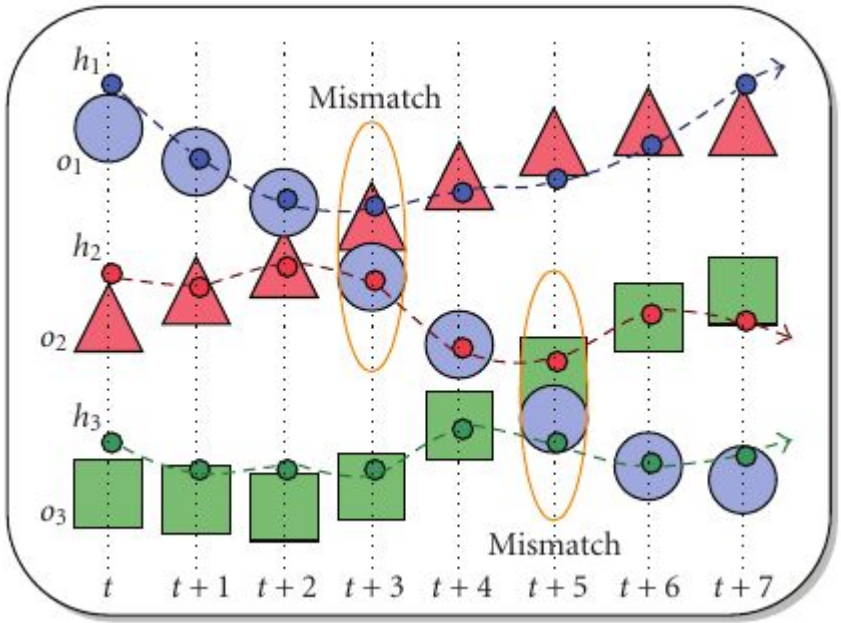
1 Thủ tục mapping

- Giả sử sau frame *t-1* có $M_{t-1}=\{(o_i, h_j)\}$
- Tại frame t, khởi tạo $M_t = \{\}$.
- Với mỗi (o_i, h_j) trong M_{t-1}, kiểm tra xem $dist_{i,j} < T$ không, nếu có thì thêm (o_i, h_j) vào M_t (TP)
- Với những o_i, h_j còn lại, tìm 1 cách match 1-1 sao cho tổng distance error nhỏ nhất, nhưng vẫn thoả mãn $dist_{i,j} < T$.
  - Những o_i không được match: FN
  - Những h_j không được match: FP

c_t: số TP m_t: số FN fp_t: số FP g_t: số object (TP + FN)

2.2.2. MOTA, MOTP, HOTA

MOTA, MOTP



1. Thủ tục mapping

   - với mỗi (o_i, h_j) trong M_t, kiểm tra xem có IDSW so với M_{t-1} không. Nếu có thì tính là 1 lần mismatch

2. Tính MOTA, MOTP

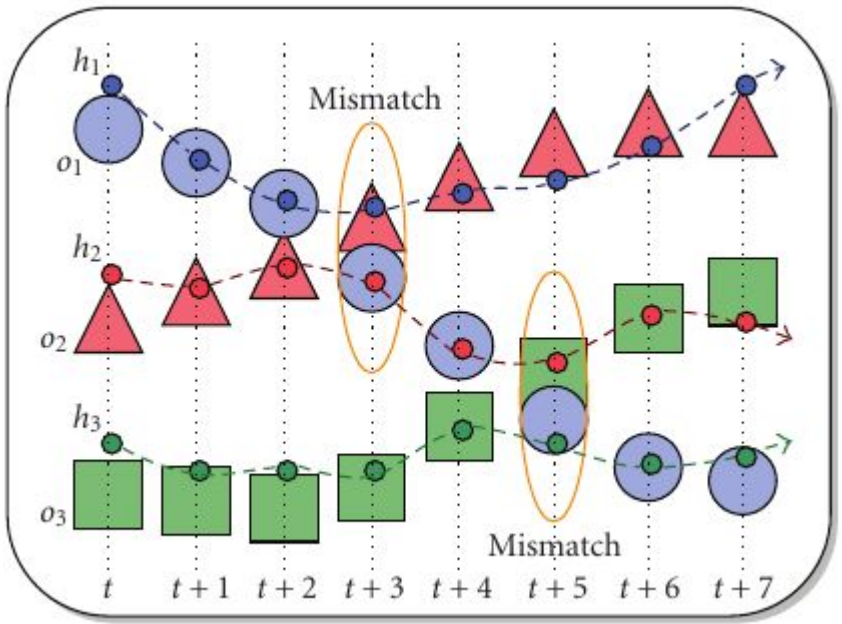
$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \quad \text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}$$

$$\overline{m} = \frac{\sum_t m_t}{\sum_t g_t}, \quad \overline{fp} = \frac{\sum_t fp_t}{\sum_t g_t}, \quad \overline{mme} = \frac{\sum_t mme_t}{\sum_t g_t},$$

c_t: số TP m_t: số FN fp_t: số FP g_t: số object (TP + FN) mme_t: số lần mismatch

2.2.2. MOTA, MOTP, HOTA

MOTA, MOTP



1. Thủ tục mapping

   - với mỗi (o_i, h_j) trong M_t, kiểm tra xem có IDSW so với M_{t-1} không. Nếu có thì tính là 1 lần mismatch

Phạt 2 lần nếu sai rồi sửa

2. Tính MOTA, MOTP

Không thể gộp

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \quad \text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}$$

$$\overline{m} = \frac{\sum_t m_t}{\sum_t g_t}, \quad \overline{fp} = \frac{\sum_t fp_t}{\sum_t g_t}, \quad \overline{mme} = \frac{\sum_t mme_t}{\sum_t g_t},$$

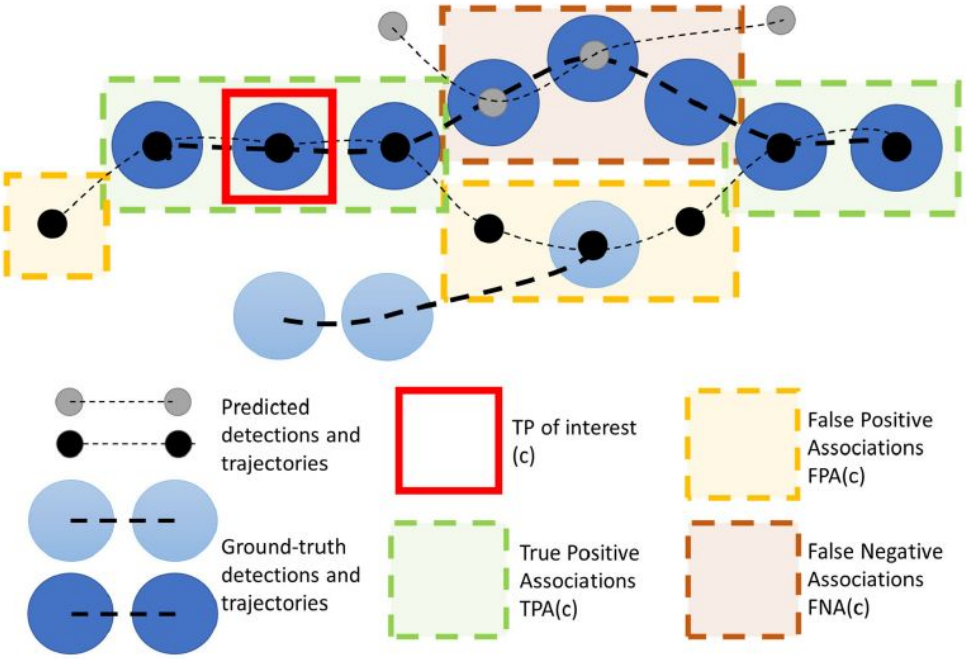
Quá ít so với m_t, fp_t

Phụ thuộc FPS: ^ thì g_t ^ còn mme_t ko

c_t: số TP m_t: số FN fp_t: số FP g_t: số object (TP + FN) mme_t: số lần mismatch

2.2.2. MOTA, MOTP, HOTA

HOTA

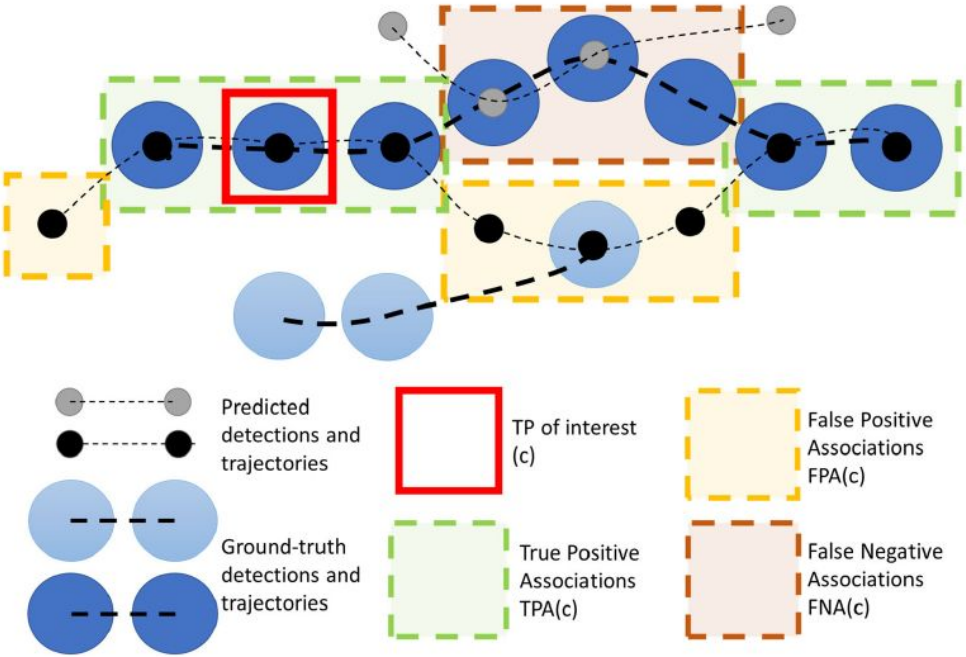


1 Thủ tục mapping

- Ở thời điểm t, dùng thuật toán Hungarian để xác định các cặp match c (TP), sao cho thoả mãn similarity $S < \alpha$.
  - những o_i không được match: FP
  - Những h_j không được match: FN
- Với mỗi TP $c=(o_i, h_j)$ ở frame t, tìm trong tất cả các frame trước và sau:
  - TPA(c): TP có cùng ID với o_i và h_j
  - FPA(c): h cùng ID với h_j nhưng được gán với một o khác ID với o_i hoặc không được gán với o nào.
  - FNA(c): o cùng ID với o_i nhưng được gán với một h khác ID với h_j hoặc không được gán với h nào.

2.2.2. MOTA, MOTP, HOTA

HOTA



2 Tính HOTA_α

$$\text{HOTA}_\alpha = \sqrt{\frac{\sum_{c\in\{\text{TP}\}} \mathcal{A}(c)}{|\text{TP}| + |\text{FN}| + |\text{FP}|}} = \sqrt{\text{DetA}_\alpha \cdot \text{AssA}_\alpha}$$

$$\mathcal{A}(c) = \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FNA}(c)| + |\text{FPA}(c)|}$$

$$\text{DetA}_\alpha = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}| + |\text{FP}|}$$

$$\text{AssA}_\alpha = \frac{1}{|\text{TP}|} \sum_{c\in\{\text{TP}\}} \mathcal{A}(c)$$

⇒ detection và association đóng góp như nhau

2.3. Docker

Sử dụng docker khi:

- Triển khai nhanh một phần mềm ở bất kỳ nền tảng nào
- Cài đặt 1 lần và toàn bộ những dependency cần thiết
- Không làm ảnh hưởng tới các thành phần khác của hệ thống

2.4. MongoDB

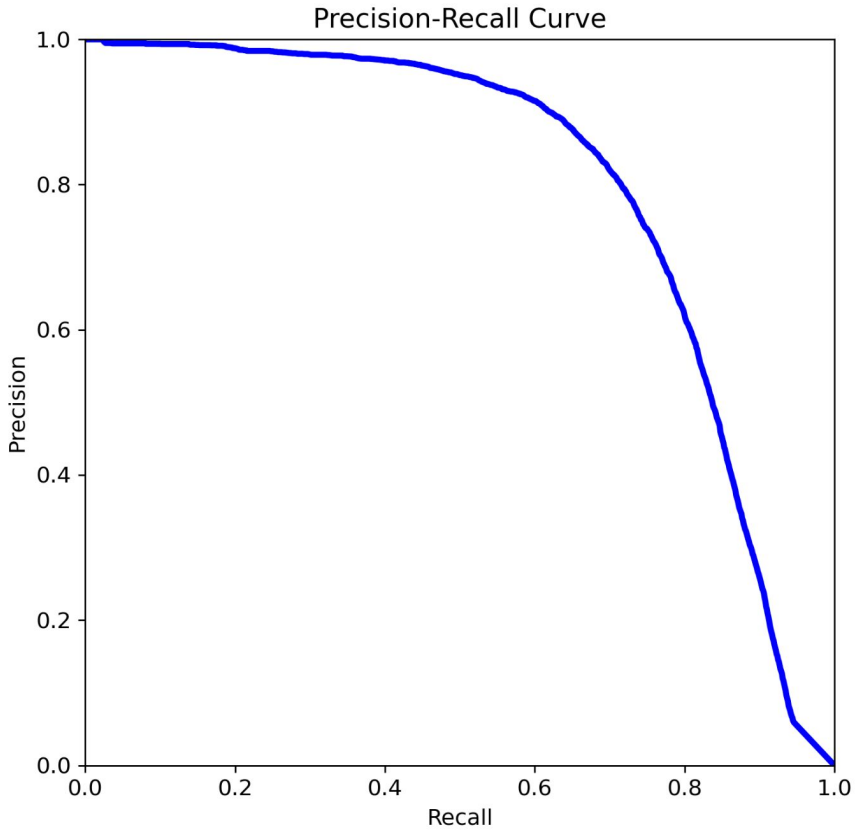
Sử dụng MongoDB khi:

- Lưu trữ các cấu trúc dữ liệu phức tạp (ví dụ như list)
- Thay đổi schema nhanh (xoá/thêm trường)
- Truy xuất thông tin nhanh cho mỗi đối tượng Python

3. Kết quả thử nghiệm

3.1. YOLOv5

1. Dữ liệu
   - Class *Person* từ tập COCO 2017
   - Training set: 64115
   - Validation set: 2693
2. Mô hình
   - YOLOv5s
   - Giữ các tham số mặc định
   - Kích cỡ ảnh: 640
3. Thời gian huấn luyện: 100 epochs
   - AP@*0.5*: 0.79836
   - AP@*0.5:0.95*: 0.54389



3.1. YOLOv5

Chạy thử trên tập MOT17, kích cỡ ảnh 640:

- Tốc độ:
  - Chiếm phần lớn thời gian toàn bộ chương trình.
  - FPS = 9 (máy weak)
- Độ chính xác:
  - Ít FP, nhưng nhiều FN (có thể là do ảnh được đưa về 640, trong khi một số video có kích thước 1920x1080)
  - ở một số video có hiện tượng box chiếm gần trọn màn hình (chưa rõ nguyên nhân)

3.2. SORT

Kết quả đánh giá trên tập MOT17

HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA
31.872	25.28	40.647	26.995	71.427	43.371	82.584	81.938

MOTA	MOTP
25.585	79.799

Dets	GT_Dets	IDs	GT_IDs
127322	336891	2891	1638

3.2. SORT

Vấn đề: giữ track ID

1. Association chỉ dựa vào vị trí của box (IoU), bỏ qua visual feature
2. Chỉ dự đoán dựa trên 1 khung hình trước
3. Giả thiết vectơ vận tốc không đổi
4. Không xét mối liên hệ `max_age`, FPS, tham số Kalman (nhạy cảm)

- (2, 3): nếu nếu FPS lớn + camera không cố định ⟶ hướng tâm box di chuyển sẽ rất nhiễu, làm giảm IoU
- (2, 3): nếu vật dần bị occluded ⟶ Kalman sẽ dự đoán sai hướng vật di chuyển, làm giảm IoU
- (4): nếu `max_age` lớn và vật dần bị occluded ⟶ Kalman có thể làm s giảm rất nhanh, làm giảm IoU
- (4): nếu vật bị occluded hoàn toàn ⟶ cùng 1 `max_age`, FPS lớn hơn sẽ tăng lượng ID
- `max_age=1` là quá bé để giữ track ID, nhưng nếu quá lớn thì với (1) sẽ tạo ra nhiều IDSW nếu đông người

Báo cáo tiến độ

Single Camera Tracking

Lần 1