

Theo dõi và ước tính khoảng cách đối tượng trong xe tự lái

Bùi Mạnh Hùng

Giáo viên hướng dẫn : Thầy Mai Tiến Dũng

Khoa khoa học máy tính

Trường đại học Công nghệ thông tin

Tp.HCM, ngày 13 tháng 12 năm 2023

- ➊ Giới thiệu
- ➋ Phương pháp
- ➌ Thực nghiệm
- ➍ Kết luận
- ➎ Tham khảo

- ➊ Giới thiệu
- ➋ Phương pháp
- ➌ Thực nghiệm
- ➍ Kết luận
- ➎ Tham khảo

Theo dõi và ước tính khoảng cách của các đối tượng xung quanh là một nhiệm vụ quan trọng đối với xe tự lái

- An toàn: Giúp xe tránh va chạm với các đối tượng xung quanh
- Tăng khả năng lái xe: Giúp xe thực hiện các thao tác lái xe phức tạp, chẳng hạn như chuyển làn, vượt xe,..
- Tăng khả năng tiếp cận: Giúp xe có thể lái trong các điều kiện môi trường phức tạp, chẳng hạn như đường đông đúc, đường tối,..

Định nghĩa bài toán

- Đối tượng: Thực thể có thể được nhận dạng trong môi trường xung quanh, chẳng hạn như xe tải, người đi bộ, người đi xe đạp,..
- Theo dõi: Việc xác định vị trí của đối tượng trong thời gian thực
- Ước tính khoảng cách: Việc xác định khoảng cách giữa camera gắn trên xe tự hành tới đối tượng

Một số thách thức

- Môi trường phức tạp: Môi trường xung quanh xe tự hành thường rất phức tạp, với nhiều đối tượng có kích thước, hình dạng, màu sắc,.. khác nhau
- Thay đổi nhanh chóng: Các đối tượng xung quanh xe tự hành có thể di chuyển nhanh chóng, khiến việc theo dõi và ước tính khoảng cách trở nên khó khăn
- Tác động của môi trường: Các yếu tố môi trường, chẳng hạn như ánh sáng, thời tiết,.. có thể ảnh hưởng đến khả năng nhận dạng và ước tính khoảng cách đến các đối tượng
- Dữ liệu: Có khá ít bộ dữ liệu về khoảng cách giữa xe tự lái đến các đối tượng xung quanh.

- ➊ Giới thiệu
- ➋ Phương pháp
- ➌ Thực nghiệm
- ➍ Kết luận
- ➎ Tham khảo

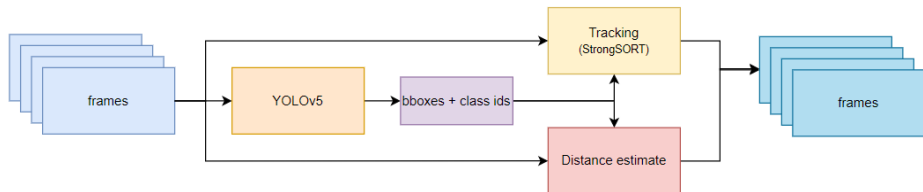


Figure: Tổng quan phương pháp sử dụng

Tổng quan

- Phương pháp: YOLOv5
- Input: Hình ảnh
- Output: Bounding box, Class ID của mỗi đối tượng được phát hiện

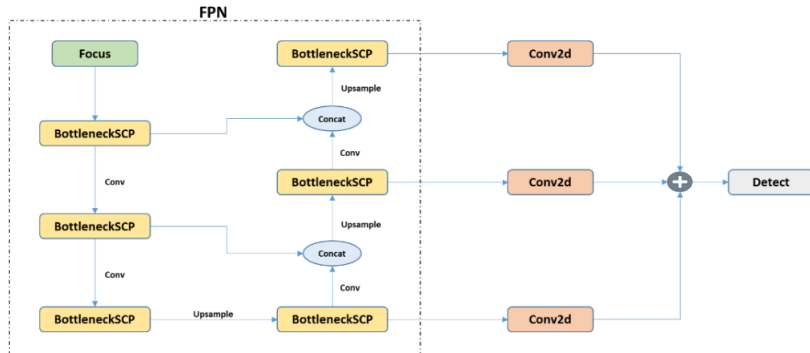


Figure: Kiến trúc YOLOv5

<https://arxiv.org/ftp/arxiv/papers/2111/2111.01715.pdf>

Tổng quan

- Phương pháp: StrongSORT
- Input:
 - Video frames
 - Bouding box được phát hiện từ YOLOv5 qua mỗi frame
- Output:
 - ID và vị trí bounding box của các đối tượng trong từng frame

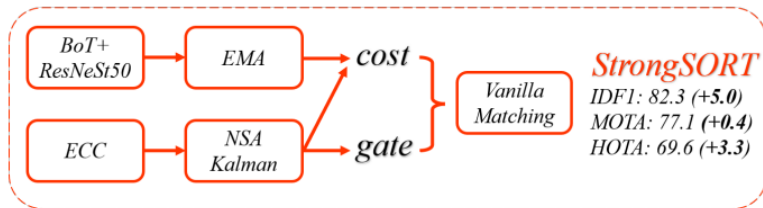


Figure: Mô hình StrongSORT

<https://arxiv.org/pdf/2202.13514.pdf>

Tổng quan

- Phương pháp: GLPDepth + LSTM
- Input: frame + bboxes + class ids
- Output: Khoảng cách từ camera gắn trên xe đến mỗi đối tượng được phát hiện

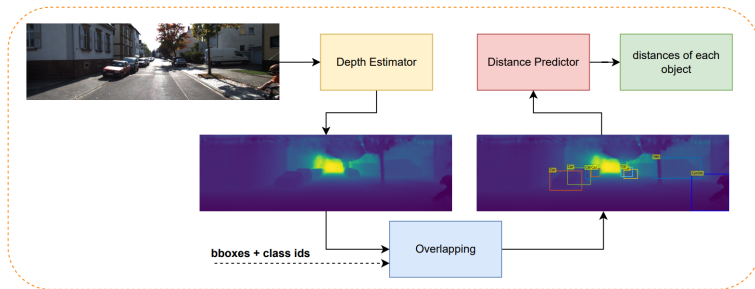


Figure: Tổng quan phương pháp ước tính khoảng cách. Từ hình ảnh đầu vào, ước tính độ sâu để tạo ra bản đồ độ sâu. Tiếp theo, ánh xạ tọa độ và class ids của mỗi bounding box lên bản đồ độ sâu. Điều này giúp trích xuất các đặc trưng cần thiết để đưa vào mô hình dự đoán khoảng cách.

- ➊ Giới thiệu
- ➋ Phương pháp
- ➌ Thực nghiệm
- ➍ Kết luận
- ➎ Tham khảo

Thông tin bộ dataset

- Dataset: KITTI
- Số lượng: 7481 training images and 7518 test images
- Số lớp: 8 (Car, Pedestrian, Van, Cyclist, Truck, Misc, Tram, Person_sitting)
- Nguồn: <https://www.cvlibs.net/datasets/kitti/>

Chuẩn bị dữ liệu cho ước tính khoảng cách

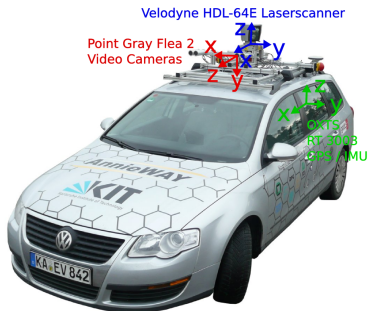


Figure: Thông tin setup sensor của bộ dữ liệu KITTI

Sử dụng công thức sau tạo ra giá trị khoảng cách từ camera đến đối tượng dựa vào giá trị tọa độ x, y, z :

$$\text{distance}_{\text{cam} \rightarrow \text{obj}} = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

Phân phối của dữ liệu khoảng cách

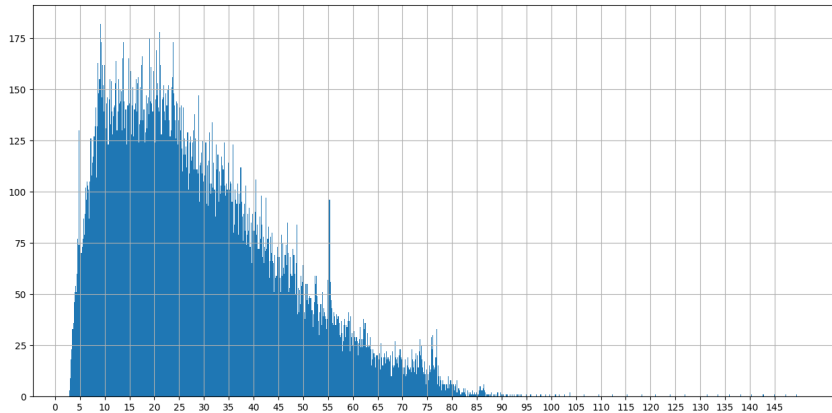


Figure: Phân phối dữ liệu khoảng cách

Cài đặt

- Dữ liệu: 7481 ảnh ; train: 0.8, test: 0.2
- Pretrained: yolov5x.pt
- #epochs: 100

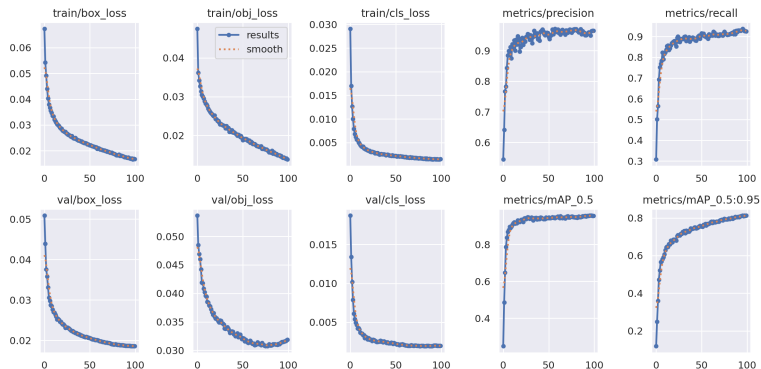


Figure: Kết quả huấn luyện mô hình YOLOv5

| Danh mục | Biến | Miêu tả |
|----------|-----------------|--|
| Input | x_min | Minimum x coordinate of a bounding box |
| | y_min | Minimum y coordinate of a bounding box |
| | x_max | Maximum x coordinate of a bounding box |
| | y_max | Maximum y coordinate of a bounding box |
| | width | Width of a bounding box |
| | height | Height of a bounding box |
| | depth_mean | Mean depth of an object |
| | depth_mean_trim | 20% trimmed mean depth of an object |
| Output | depth_max | Maximum depth of an object |
| | d | Distance of an object |

Table: Các đặc trưng được sử dụng để huấn luyện mô hình

| | | |
|------|---------------------|-------------------|
| LSTM | Input_dim | 10 |
| | Hidden_dim(LSTM) | 612 |
| | Layer_dim(LSTM) | 3 |
| | Hidden_dim(Linear) | 612, 306, 154, 76 |
| | Output_dim(Linear) | 1 |
| | Bidirectional | False |
| | Optimizer | Adam |
| | Activation function | ReLU |
| | Max epoch | 500 |
| | Batch size | 64 |

Table: Các siêu tham số và giá trị tương ứng cho mô hình LSTM

Loss function

$$\text{MAE(Mean Absolute Error)} = \frac{1}{n} \sum_{i=1}^n |y - \hat{y}| \quad (2)$$

Cài đặt

Dữ liệu: 40570 mẫu, train: 80%, valid: 10%, test: 10%

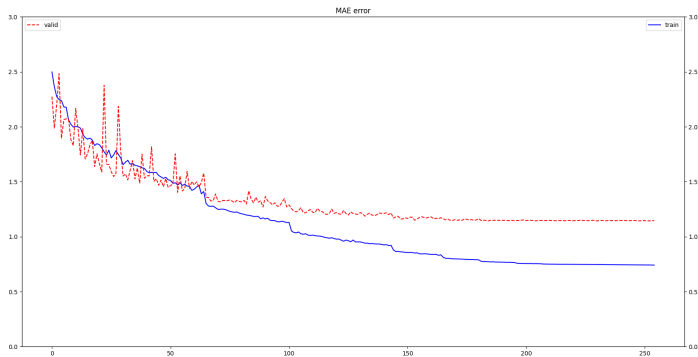


Figure: Độ lỗi MAE qua mỗi epoch

Kết quả đánh giá mô hình ước tính khoảng cách - LSTM

| Model | MAE(m) | | | | | | | | |
|-------|--------|-------|------------|------|---------|----------------|------|------|---------|
| | Car | Truck | Pedestrian | Van | Cyclist | Person sitting | Tram | Misc | Overall |
| LSTM | 1.16 | 1.48 | 0.78 | 2.07 | 1.09 | 1.45 | 1.46 | 1.34 | 1.20 |

Table: MAE cho các lớp đối tượng khác nhau trong tập test

| Model | MAE(m) | | | | | | | |
|-------|--------|---------|---------|---------|---------|---------|---------|---------|
| | 0-9 m | 10-19 m | 20-29 m | 30-39 m | 40-49 m | 50-59 m | 60-69 m | 70-79 m |
| LSTM | 0.91 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 | 0.92 | 0.89 |

Table: MAE cho các khoảng khoảng cách khác nhau trong tập test

DEMO TIME..

- ➊ Giới thiệu
- ➋ Phương pháp
- ➌ Thực nghiệm
- ➍ Kết luận
- ➎ Tham khảo

Kết luận

Đối với mô hình phát hiện đối tượng

- Độ chính xác cao trong việc phát hiện đối tượng xe hơi, người đi bộ,...
- Có thể chạy real-time

Đối với mô hình theo dõi

- Theo dõi chính xác và nhất quán các đối tượng qua các frame
- Ít xảy ra hiện tượng đánh tráo đối tượng

Đối với mô hình ước tính khoảng cách

- Mô hình đào tạo có độ chính xác khá cao
- Tốc độ inference còn chậm do mô hình ước tính độ sâu GLPDepth có thời gian thực thi chậm
- Bộ dữ liệu đào tạo không cân bằng, làm cho mô hình dự đoán sai khi chạy thử

- ➊ Giới thiệu
- ➋ Phương pháp
- ➌ Thực nghiệm
- ➍ Kết luận
- ➎ Tham khảo

- ① <https://github.com/ultralytics/yolov5>
- ② <https://github.com/dyhBUPT/StrongSORT>
- ③ <https://arxiv.org/pdf/2202.13514.pdf>
- ④ <https://arxiv.org/pdf/1905.00953.pdf>
- ⑤ <https://github.com/KaiyangZhou/deep-person-reid>
- ⑥ <https://github.com/vinvino02/GLPDepth>
- ⑦ <https://www.mdpi.com/2073-8994/14/12/2657>

Cảm ơn bạn đã lắng nghe