

# **BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**Lớp: CS114.(K21+K21.KHTN)**

**Môn: MÁY HỌC**

**GV: PGS.TS Lê Đình Duy - THS. Phạm Nguyễn Trường An**  
**Trường ĐH Công Nghệ Thông Tin, ĐHQG-HCM**

# TÊN ĐỀ TÀI - Phát hiện vượt đèn đỏ thông qua camera giám sát

Trần Doãn Thuyền- MSSV 18521479- CS114.K21

Link Github:

<https://github.com/doanthuyennt/CS114.K21>

# Bài toán

Hiện nay, tình hình tai nạn giao thông vẫn là vấn đề nhức nhối của xã hội, bởi vì không những các tai nạn giao thông gây nên cho ta các thiệt hại về của cải mà cả về tính mạng con người. Vậy khi đi xét về nguyên nhân dẫn tới các vụ tai nạn này - có thể là do yếu tố thời tiết; hay do tác động của hệ tầng giao thông và ý thức của người tham gia giao thông. Nhưng xét cho cùng, các vụ tai nạn nghiêm trọng xuất phát từ ý thức của người tham gia giao thông như việc vi phạm vượt đèn đỏ chiếm một số lượng không nhỏ.

# Bài toán

Theo đó, trong đề tài này sẽ hướng tới việc xây dựng hệ thống phát hiện vi phạm vượt đèn đỏ bằng việc áp dụng các kĩ thuật Deep Learning trên dữ liệu thu được từ qua camera giám sát nhằm mục đích giảm một phần nào số lượng công việc cho các nhân viên giám sát và lực lượng chức năng ; và cũng như góp phần nhanh chóng phát hiện vi phạm và xử lý vi phạm triệt để và nhanh chóng nhằm góp phần nâng cao ý thức chung của người dân khi tham gia giao thông.

# Cách giải quyết

Như vậy, bài toán phát hiện hành vi phạm vượt đèn đỏ là bài toán phức tạp bao gồm nhiều bài toán con như bài toán phát hiện (vehicle detection), bài toán phân loại tín hiệu (classification) và bài toán truy vết đối tượng (tracking) (hình 2 minh họa một số bài toán chọn trong hệ thống). Đây đều là các bài toán phức tạp trong lĩnh vực thị giác máy.

# Cách giải quyết

Input: Một video đầu vào từ camera.

Output: Video ra chứa các đối tượng vi phạm vượt đèn đỏ.

# Mô tả dữ liệu

Cào dữ liệu video trên mạng và cắt các khung hình chứa các đối tượng ra, trong đó dùng công cụ labellmg để tạo bounding box

Kích thước dữ liệu:

- Đi bộ: 372 đối tượng;
- Xe đạp: 15 đối tượng;
- Xe máy: 3511 đối tượng;
- Xe hàng rong: 3 đối tượng;
- Xe ba gác: 52 đối tượng;
- Xe taxi: 76 đối tượng;
- Xe hơi: 1026 đối tượng;
- Xe bán tải: 33 đối tượng;
- Xe cứu thương: 11 đối tượng;
- Xe khách: 141 đối tượng;
- Xe buýt: 194 đối tượng;
- Xe tải: 836 đối tượng;
- Xe container: 154 đối tượng;
- Xe cứu hỏa: 2 đối tượng;

# Mô tả dữ liệu

Nhận xét:

Bộ dữ liệu bị skewed rất nhiều về đối tượng **xe máy** và **xe hơi** ; và rất ít đối với **xe cứu hỏa** và **xe hàng rong**



# Kết quả đạt được (YOLO)

552

```
detections_count = 8232, unique_truth_count = 6227  
class_id = 0, name = di_bo , ap = 38.61% (TP = 195, FP = 177)  
class_id = 1, name = xe_dap , ap = 58.00% (TP = 11, FP = 4)  
class_id = 2, name = xe_may , ap = 66.84% (TP = 2473, FP = 1038)  
class_id = 3, name = xe_hang_rong , ap = 100.00% (TP = 3, FP = 0)  
class_id = 4, name = xe_ba_gac , ap = 82.69% (TP = 44, FP = 8)  
class_id = 5, name = xe_taxi , ap = 82.23% (TP = 60, FP = 16)  
class_id = 6, name = xe_hoi , ap = 80.50% (TP = 846, FP = 180)  
class_id = 7, name = xe_ban_tai , ap = 96.47% (TP = 30, FP = 3)  
class_id = 8, name = xe_cuu_thuong , ap = 65.57% (TP = 7, FP = 4)  
class_id = 9, name = xe_khach , ap = 80.43% (TP = 116, FP = 25)  
class_id = 10, name = xe_buyt , ap = 94.05% (TP = 185, FP = 9)  
class_id = 11, name = xe_tai , ap = 92.44% (TP = 762, FP = 74)  
class_id = 12, name = xe_container , ap = 97.95% (TP = 146, FP = 8)  
class_id = 13, name = xe_cuu_hoa , ap = 100.00% (TP = 2, FP = 0)
```

IoU threshold = 75 %, used Area-Under-Curve for each unique Recall  
mean average precision (mAP@0.75) = 0.811269, or 81.13 %

Với việc lấy ngưỡng chính xác là 0.25:

precision = 0.76, recall = 0.78, F1-score = 0.77 và  
TP = 4880, FP = 1546 và average IoU = 64.57 %

Với việc lấy ngưỡng IOU là 0.75 theo phương pháp đánh giá COCO:

Used Area-Under-Curve mean average precision sẽ là (mAP@0.75) = 81.13 %

Tất cả đều sử dụng dataset VLSP SA 2016

# Kết quả đạt được (FASTER-RCNN)

```
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.037
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.075
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.031
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.016
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.051
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.068
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.277
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.288
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.001
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.147
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.354
```

Các chỉ số average precision và average recall đều thấp hơn của YOLO

Tất cả đều sử dụng dataset VLSP SA 2016

# Kết quả demo



**Xe thứ nhất vi phạm bị đưa vào khung hình bên cạnh.**

# Kết quả demo



**Xe thứ nhất vi phạm bị đưa vào khung hình bên cạnh.**

# Kết quả demo



**Xe thứ hai vi phạm bị đưa vào khung hình bên cạnh đưa lên trên hình xe thứ nhất vi phạm.**

# Hướng phát triển

Tìm hiểu thêm các deep model để phát triển thêm:

- Thu thập thêm dữ liệu;
- Đánh giá lại các kiến trúc mạng nhiều hơn.



# Thanks!

**Contact us:**

Big Data & Deep  
Learning Lab  
Ho Chi Minh City,  
Vietnam

[b2dl.uit@gmail.com](mailto:b2dl.uit@gmail.com)

