

MÔ HÌNH NHẬN DIỆN BIỂN SỐ XE VÀ GIÁM SÁT LƯU LƯỢNG XE LƯU THÔNG

Trần Việt Hoàng - 240101046

Tóm tắt

- Lớp: CS2205.FEB2025
- Link Github của nhóm:
<https://github.com/hoangtv19/CS2205.FEB2025>
- Link YouTube video: <https://youtu.be/GE2jFy6pUJ0>
- Trần Việt Hoàng - 240101046



Giới thiệu

- Sự gia tăng nhanh chóng của mật độ giao thông tại các đô thị lớn nhưng các giải pháp truyền thống giám sát giao thông hiện có thường gặp hạn chế về độ chính xác, đặc biệt trong điều kiện ánh sáng yếu, thời tiết xấu, hoặc khi xe di chuyển nhanh [1] [2].
- Nghiên cứu này tập trung vào việc phát triển một hệ thống prototype, sử dụng mô hình YOLOv5 để phát hiện phương tiện và biển số, kết hợp DeepSORT để theo dõi và đếm lưu lượng xe.
- **Input:** Video hoặc ảnh từ camera giao thông.
- **Output:** Danh sách biển số xe được nhận diện và số lượng xe được đếm trong thời gian thực.

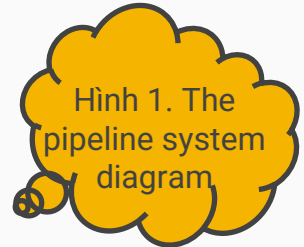
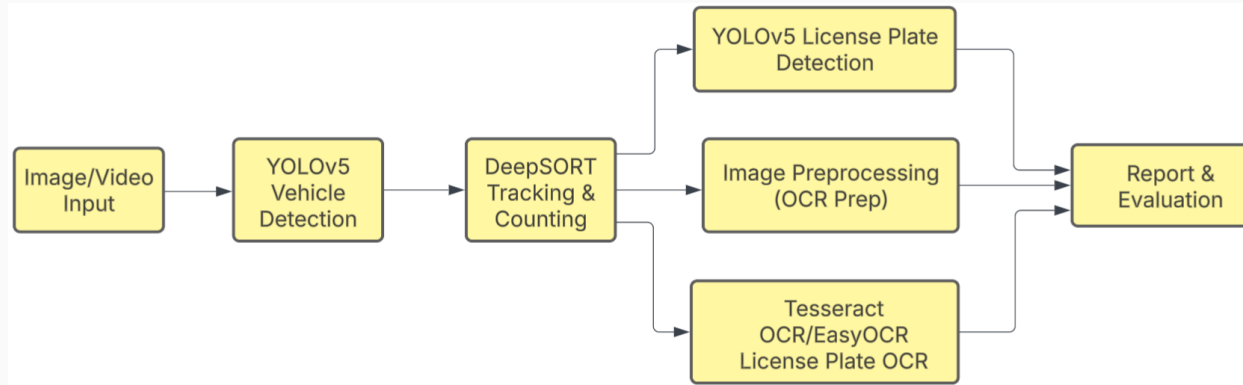
Mục tiêu

- Xây dựng hệ thống pipeline phát hiện phương tiện và gán ID duy nhất cho mỗi xe bằng việc sử dụng YOLOv5 và DeepSORT với độ chính xác (Precision) trên 90% và Recall trên 85% trong các điều kiện môi trường/thời tiết khác nhau.
- Đếm và phân loại lưu lượng xe (ô tô, xe máy) qua khu vực quan tâm (ROI) trong video giám sát, xuất báo cáo theo khung giờ.
- Tốc độ xử lý đạt 15~20 FPS cho việc phát hiện và đếm xe, 0.3~0.5 giây/lần crop biển số và Optical Character Recognition (OCR).

Nội dung và Phương pháp

● Nội dung

- Chuẩn bị tập dữ liệu thử nghiệm gồm video giao thông thực tế và ảnh biển số xe để huấn luyện, kiểm định và đánh giá mô hình.
- Huấn luyện mô hình YOLOv5 tích hợp DeepSORT vào một pipeline thống nhất để nhận diện và đếm xe.
- Đánh giá hiệu suất hệ thống thông qua các chỉ số Precision, Recall (cho nhận diện biển số) và sai số đếm (cho giám sát lưu lượng giao thông).



Nội dung và Phương pháp

● Phương pháp

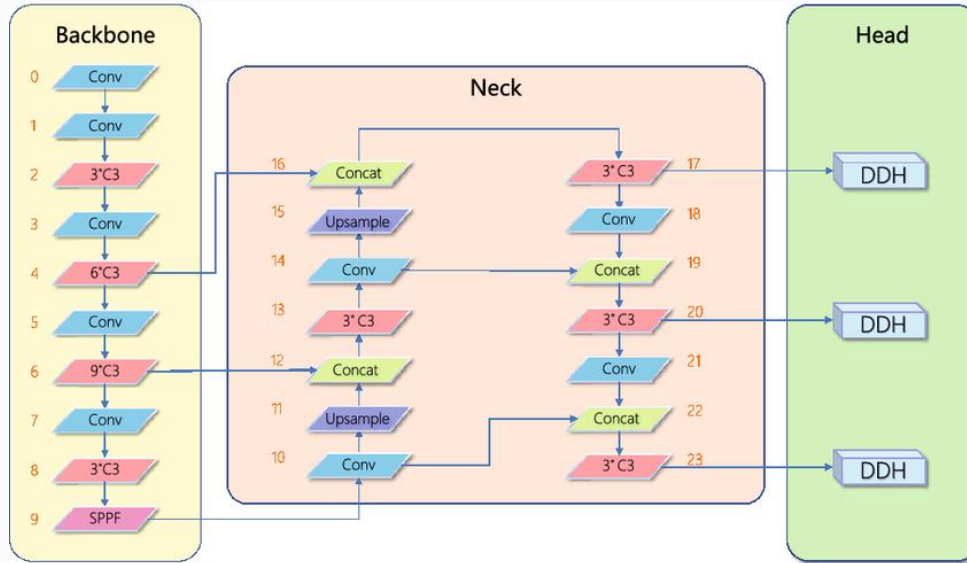
○ Giai đoạn 1 – Thu thập dữ liệu

- ✓ Tải ảnh/video giao thông (độ phân giải $\geq 720p$, 25–30 FPS) trong điều kiện môi trường/ ánh sáng khác nhau. (Ví dụ dữ liệu từ UA-DETRAC)
- ✓ Gắn nhãn bounding box phương tiện và nội dung biển số bằng Labellmg hoặc Roboflow, tạo dataset train/validation/test.

○ Giai đoạn 2 – Huấn luyện mô hình

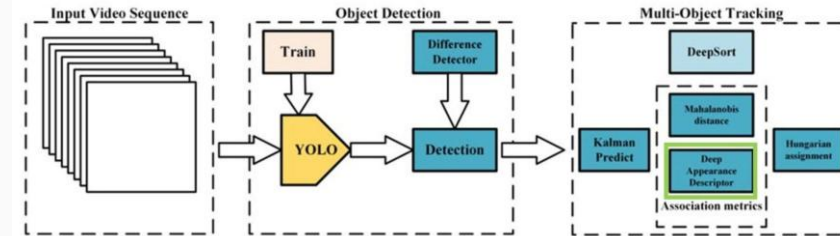
- ✓ Sử dụng YOLOv5 với backbone CSPDarknet53, huấn luyện trên tập dữ liệu biển số xe bằng PyTorch.
- ✓ Áp dụng hàm mất mát kết hợp (classification loss, localization loss) để tối ưu hóa nhận diện biển số.
- ✓ Tích hợp DeepSORT với đặc trưng sâu từ YOLOv5, sử dụng bộ lọc Kalman để theo dõi và gán ID cho từng xe xuyên suốt các frame.

Nội dung và Phương pháp



Hình 2. YOLOv5 Model with CSPDarknet53 Backbone Architecture

Hình 3. DeepSORT Algorithm



Nội dung và Phương pháp

- Phương pháp
 - Giai đoạn 3 – Xử lý và nhận diện
 - ✓ Tiền xử lý ảnh biến số: chuyển grayscale, cân bằng sáng, threshold và loại bỏ nhiễu.
 - ✓ Từ video đầu vào, YOLOv5 phát hiện vùng chứa biển số xe, sau đó sử dụng Tesseract OCR hoặc EasyOCR để nhận dạng ký tự, tiếp theo áp dụng regex để chuẩn hóa định dạng biển số.
 - ✓ DeepSORT theo dõi các xe qua các khung hình, đếm số lượng xe bằng cách đặt một đường ảo (virtual line) trong khung hình.
 - ✓ Kết quả được lưu vào cơ sở dữ liệu MySQL (biển số, thời gian, ID xe).
 - Giai đoạn 4 – Báo cáo và đánh giá
 - ✓ Tính toán báo cáo lưu lượng xe theo khung giờ, biểu diễn bằng biểu đồ matplotlib.
 - ✓ Đánh giá nhận diện biển số bằng Precision, Recall, và F1-score, so sánh với ground-truth.
 - ✓ Đánh giá đếm xe bằng sai số tuyệt đối trung bình (Mean Absolute Error - MAE).

Kết quả dự kiến

- Xây dựng được một hệ thống tích hợp YOLOv5 và DeepSORT, nhận diện biển số xe với độ chính xác (Precision) trên 90% và Recall trên 85% trong các điều kiện thực tế.
- Giám sát lưu lượng xe với sai số đếm dưới 5%, hoạt động ổn định trong môi trường đô thị đông đúc với các điều kiện môi trường khác nhau.
- Triển khai ứng dụng web bằng Streamlit, cho phép người dùng tải video/ảnh đầu vào, hiển thị kết quả nhận diện biển số và số lượng xe, cùng với các chỉ số đánh giá (Precision, Recall, Mean Absolute Error - MAE).
- Tốc độ xử lý đạt 15~20 FPS cho việc phát hiện và đếm xe, 0.3~0.5 giây/lần crop biển số và OCR.
- Đánh giá thực nghiệm trên tập dữ liệu dự kiến sẽ cho thấy hệ thống vượt trội hơn các phương pháp truyền thống (như Haar Cascade)[5] về độ chính xác và tốc độ xử lý.

Tài liệu tham khảo

- [1] Ultralytics. (2021). YOLOv5 Documentation. <https://docs.ultralytics.com/>.
- [2] Wojke, N., Bewley, A., & Paulus, D. (2017). Simple Online and Realtime Tracking with a Deep Association Metric. arXiv:1703.07402
- [3] Zhang, S., et al. (2020). A Survey on Deep Learning-based Object Detection. arXiv:2001.08792
- [4] Jocher, G., et al. (2021). YOLOv5: A Family of Object Detection Architectures. <https://github.com/ultralytics/yolov5>.
- [5] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv:1804.02767.
- [6] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934.
- [7] Silabarasan, K. D., & Vijayarani, S. (2021). Automatic License Plate Recognition Using Deep Learning. International Journal of Computer Applications, 175(40), 1–5.
- [8] Rodriguez, M., & Gonzalez, J. (2019). Vehicle Detection and Tracking using Deep SORT and YOLOv3. Proceedings of the 16th International Conference on Computer Vision.
- [9] Silva, R. C., & Souza, J. M. (2021). A Comparative Study of OCR Techniques for License Plate Recognition. IEEE Transactions on Intelligent Transportation Systems, 22(3), 2000–2010.
- [10] Tran, V. H., & Nguyen, T. L. (2023). Ứng dụng YOLOv5 và DeepSORT cho giám sát lưu lượng xe tại TP.HCM. Hội thảo Khoa học Trường ĐH Bách Khoa TP.HCM.
- [11] Lien, H. T., & Pham, Q. H. (2022). Nhận diện biển số xe Thành phố bằng PyTesseract và Haar Cascade. Tạp chí Công nghệ Thông tin & Truyền thông.
- [12] HuiWang, YangJin, Hongchang Ke, Xinping Zhang (2022). DDH-YOLOv5: improved YOLOv5 based on Double IoU-aware Decoupled Head for object detection. https://www.researchgate.net/figure/The-architecture-of-the-DDH-YOLOv5-a-Use-the-CSPDarknet53-as-the-backbone-of-the-model_fig3_362680027.