|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI



Nguyễn THƯ TRANG

Ứng dụng Học Sâu trong giám sát hành vi tài xế để phát hiện và cảnh báo tình trạng buồn ngủ

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2023

**NGUYỄN THƯ TRANG ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HÀ NỘI, NĂM 2025**

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

NGUYỄN THƯ TRANG

Ứng dụng Học Sâu trong giám sát hành vi tài xế để phát hiện và cảnh báo tình trạng buồn ngủ

|  |  |
| --- | --- |
| Ngành: | Công nghệ thông tin |
| Mã số: | 7480201 |

|  |  |  |
| --- | --- | --- |
| NGƯỜI HƯỚNG DẪN | TS. Trần Mạnh Tuấn | |
|  |  | |
|  | |  | |
|  | |  | |

HÀ NỘI, NĂM 2025

|  |  |
| --- | --- |
|  | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA** **VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc**  ----------★----------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

|  |  |
| --- | --- |
| **Họ tên sinh viên:** NGUYỄN THƯ TRANG  **Lớp:** 62THVA  **Khoa:** Công nghệ thông tin | **Hệ đào tạo:** Đại học chính quy  **Ngành:** Công nghệ thông tin |

1. **TÊN ĐỀ TÀI:**

**Ứng dụng Học Sâu trong giám sát hành vi tài xế để phát hiện và cảnh báo tình trạng buồn ngủ**

1. **CÁC TÀI LIỆU CƠ BẢN:**

|  |  |
| --- | --- |
| [1] | V. Hoàng, »Tìm hiểu về YOLO trong bài toán real-time object detection,« *Viblo,* p. bài viết, 2019. |
| [2] | M. Menegaz, »Understanding YOLO,« *hackernoon,* 2018. |
| [3] | A. Rosebrock, »Intersection over Union (IoU) for object detection,« *pyimagesearch,* 2016. |
| [4] | Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, »Rich feature hierarchies for accurate object detection and semantic segmentation,« *arxiv,* 2014 . |
| [5] | Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, »Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,« *arxiv,* p. Computer Vision and Pattern Recognition, 2016 . |
| [6] | N. V. Hoai, »Faster R-CNN cho bài toán phát hiện đối tượng,« *Viblo,* 2020 . |
| [7] | Tuấn, » Object detection với Faster R-CNN,« *nttuan8.com,* p. Deep Learning, 2019. |
| [8] | P. Potrimba, »What is R-CNN?,« *roboflow,* p. Blog, 2023 . |
| [9] | Lev Craig, David Petersson, »CNN vs. RNN: How are they different?,« *techtarget,* p. AI technologies, 2024. |
| [10] | Jim Holdsworth, Mark Scapicchio, »What is deep learning?,« *ibm,* p. Think, 2024. |

1. **NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:**

|  |  |
| --- | --- |
| **Nội dung các phần** | **Tỷ lệ %** |
| Chương 1: Tổng quan cơ sở lý thuyết | 30% |
| Chương 2: Xây dựng phương pháp dự báo | 40% |
| Chương 3: Xây dựng hệ thống mô phỏng và kết quả thực nghiệm Xây dựng hệ thống  Mô tả hệ thống Các thuật toán, thư viện sử dụng  Kết quả thực nghiệm | 30% |

1. **GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN**

|  |  |
| --- | --- |
| Phần | Họ và Tên giáo viên hướng dẫn |
| Chương 1: Tổng Quan Về Bài Toán Nhận Diện Phát Hiện Tài Xế Buồn Ngủ | TS. Trần Mạnh Tuấn |
| Chương 2: Mô Hình Xử Lý Bài Toán Nhận Diện Phát Hiện Tài Xế Buồn Ngủ | TS. Trần Mạnh Tuấn |
| Chương 3: Xây Dưng Chương Trình Nhận Diện Phát Hiện Tài Xế Buồn Ngủ | TS. Trần Mạnh Tuấn |

1. **NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

|  |  |
| --- | --- |
| Ngày ............ tháng ......... năm 20  Trưởng Bộ môn  (Ký và ghi rõ Họ tên) | Giáo viên hướng dẫn chính  (Ký và ghi rõ Họ tên) |

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

|  |  |
| --- | --- |
|  | Ngày. . . . .tháng. . . . .năm 20....  tịch Hội đồng  (Ký và ghi rõ Họ tên) |

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi

|  |  |
| --- | --- |
|  | Ngày. . . . .tháng. . . . .năm 20....  Sinh viên làm Đồ án tốt nghiệp  (Ký và ghi rõ Họ tên) |

**Nguyễn Thư Trang**

|  |  |
| --- | --- |
|  | **TRƯỜNG ĐẠI HỌC THUỶ LỢI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP** |

**Tên đề tài:** **Ứng dụng Học Sâu trong giám sát hành vi tài xế để phát hiện và cảnh báo tình trạng buồn ngủ**

*Sinh viên thực hiện*: Nguyễn Thư Trang

*Lớp*: 62TH-VA

*Mã sinh viên*: 2051060742

*Số điện thoại*: 0329016173

*Email*: Nguyenthutranghd.202@gmail.com

*Giáo viên hướng dẫn*: TS. Trần Mạnh Tuấn

**Tóm Tắt Đề Tài**

Tình trạng buồn ngủ khi lái xe ngày càng trở thành một vấn đề nghiêm trọng trong xã hội hiện đại, đặc biệt khi sự gia tăng của các phương tiện giao thông và khối lượng công việc khiến người lái xe thường xuyên phải đối mặt với tình trạng mệt mỏi. Buồn ngủ khi lái xe không chỉ ảnh hưởng đến hiệu quả công việc mà còn đặt ra những nguy cơ lớn đối với sự an toàn giao thông. Các nghiên cứu cho thấy rằng tình trạng này có thể làm giảm khả năng phản ứng và tập trung của người lái xe, dẫn đến nguy cơ cao hơn về tai nạn giao thông [1]. Vì vậy việc xác định được chính xác trạng thái của tài xế trong lúc lái xe và đưa ra cảnh báo sớm giúp tài xế giữ trạng thái tỉnh táo để có hành trình an toàn từ đó giảm thiểu nguy cơ tai nạn và đảm bảo an toàn giao thông.

Các nghiên cứu gần đây sử dụng mô hình học sâu mạng tích chập CNN [5] tập trung vào việc phát hiện hiện tượng microsleep-trạng thái ngủ ngắn chỉ vài giây kỳ vọng giúp giảm thiểu và ngăn ngừa tai nạn do microsleep thông qua việc theo dõi nháy mắt của người lái xe, EEG kết hợp TE [3] làm rõ mối quan hệ giữa kết nối não và hiệu suất lái xe trong bối cảnh mệt mỏi, một vấn đề lớn về an toàn giao thông. Xử lý Ngôn ngữ Tự nhiên (NLP) kết hợp học máy và mô hình đám mây [2] tạo cuộc trò chuyện duy trì sự tỉnh táo cho người lái xe thông qua loa thông minh.

Trong đề tài này, để giải quyết bài toán nhận diện phát hiện trạng thái buồn ngủ của lái xe sử dụng mô hình lai kết hợp các kỹ thuật học sâu [6]. Cụ thể, sử dụng YOLO phát hiện nhanh chóng các đặc trưng của tài xế trong mọi góc, và R-CNN để phân loại phát hiện các vùng quan trọng trong hình ảnh xác định rõ về biểu hiện buồn ngủ. Bộ dữ liệu để thử nghiệm mô hình là các hình ảnh, video trích xuất từ cam hành trình ghi lại trạng thái của người lái xe.

**CÁC MỤC TIÊU CHÍNH**

* Mục tiêu 1: Phát triển và tối ưu hóa các mô hình học máy và học sâu để phát hiện trạng thái buồn ngủ của tài xế thông qua phân tích hình ảnh.
* Mục tiêu 2: Phát triển giao diện người dùng thân thiện để tài xế và các nhà quản lý giao thông dễ dàng truy cập thông tin và cảnh báo về trạng thái buồn ngủ.

**NỘI DUNG CHÍNH**

* Thu thập dữ liệu: Thu thập các các hình ảnh video ghi lại ở cam hành trình về trạng thái của người khi lái xe
* Tiền xử lý dữ liệu: Dữ liệu được kiểm tra và điều chỉnh kiểu dữ liệu tương ứng đúng với từng thuộc tính, đồng thời loại bỏ những mẫu dữ liệu không hợp lệ hoặc dữ liệu thiếu.
* Lựa chọn các mô hình phù hợp cho bài toán và dữ liệu.
* Tối ưu hóa mô hình: Tinh chỉnh và tối ưu hóa các tham số của mô hình để cải thiện độ chính xác.
* Đánh giá mô hình để lựa chọn mô hình phù hợp và xây dựng mô hình
* Xây dựng hệ thống tích hợp hệ thống khai thác dữ liệu với mô hình

**KẾT QUẢ DỰ KIẾN**

* Báo cáo tổng kết của đồ án tốt nghiệp
* Chương trình demo mô hình ứng dụng.

# **LỜI CAM ĐOAN**

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

|  |  |
| --- | --- |
|  | Tác giả ĐATN  (Chữ ký)  Nguyễn Thư Trang |

# **LỜI CẢM ƠN**

Trong suốt những năm tháng học tập và rèn luyện tại khoa Công Nghệ Thông Tin, trường Đại Học Thủy Lợi em đã nhận lại được rất nhiều bao gồm cả kiến thức và cả sự quan tâm, hỗ trợ hết mình từ thầy cô. Em được trải nghiệm một môi trường vô cùng nhiệt huyết từ lửa của thầy cô truyền đến và cả của các bạn đang cùng học tập rèn luyện. Qua mỗi bài giảng thứ em nhận được không chỉ là kiến thức có trong giáo trình mà còn cả những câu chuyện trải nghiệm thực tế của thầy cô chia sẻ. Nhà trường và thầy cô đã tạo điều kiện tốt nhất cho em và cũng như tất cả sinh viên được phát triển bản thân qua rất nhiều chương trình dự án: Nghiên cứu khoa học, các cuộc thi Olympic, tìm kiếm tài năng….

Em xin được gửi lời cảm ơn chân thành sâu sắc nhất đến thầy cô trong khoa nói chung và thầy TS. Trần Mạnh Tuấn nói riêng, thầy đã rất kiên nhẫn và hỗ trợ tận tình trong thời gian từ khi bắt đầu đến khi hoàn thành đồ án tốt nghiệp. Những ngày tháng được học tập rèn luyện dưới sự hướng dẫn của các thầy cô trong khoa Công Nghệ Thông Tin là những trải nghiệm quý báu nhất mà em nhận được, giờ đây em đã biết em cần gì và có định hướng rõ ràng trong tương lai. Em xin được gửi lòng biết ơn sâu sắc nhất đến thầy cô!

Đồ án tốt nghiệp, em đã nỗ lực hết mình cẩn thận thực hiện. Tuy nhiên không thể tránh khỏi một số thiếu sót. Mong thầy cô, cũng như các bạn cùng đánh giá và cho em xin những đóng góp quý báu để đồ án được hoàn thiện tốt hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CAM ĐOAN i](#_Toc190347775)

[LỜI CẢM ƠN ii](#_Toc190347776)

[DANH MỤC HÌNH ẢNH iv](#_Toc190347777)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ v](#_Toc190347778)

[MỞ ĐẦU 1](#_Toc190347779)

[CHƯƠNG 1: TỔNG QUAN VỀ BÀI TOÁN NHẬN DIỆN PHÁT HIỆN TÀI XẾ BUỒN NGỦ 3](#_Toc190347780)

[1.1 Mô Tả bài toán 3](#_Toc190347781)

[1.2 Tình hình nghiên cứu quốc tế và trong nước hiện nay 4](#_Toc190347782)

[1.3 Học Sâu (Deep Learning) 5](#_Toc190347783)

[1.3.1 Tổng Quan 5](#_Toc190347784)

[1.3.2 Các loại mô hình học sâu 7](#_Toc190347785)

[1.4. Ngôn ngữ sử dụng 15](#_Toc190347786)

[1.4.1 Tổng Quan về ngôn ngữ lập trình Python 15](#_Toc190347787)

[1.4.2 Ưu và nhược điểm của ngôn ngữ Python 16](#_Toc190347788)

[CHƯƠNG 2: MÔ HÌNH XỬ LÝ BÀI TOÁN NHẬN DIỆN PHÁT HIỆN TÀI XẾ BUỒN NGỦ 18](#_Toc190347789)

[2.1 Tổng quát bài toán nhận diện 18](#_Toc190347790)

[2.2. Chi tiết các bước thực hiện 18](#_Toc190347791)

[2.3. Mô hình YoLo 19](#_Toc190347792)

[2.3.1. Cơ chế hoạt động của YoLo 19](#_Toc190347793)

[2.3.2. Hàm tính IOU 20](#_Toc190347794)

[2.3.3. Loss Function 21](#_Toc190347795)

[2.4. Mô hình YoLov5 23](#_Toc190347796)

[2.5. Mô Hình R-CNN 25](#_Toc190347797)

[2.6. Mô hình Faster R-CNN 27](#_Toc190347798)

[2.7. Nhận Xét và Đánh giá 28](#_Toc190347799)

[CHƯƠNG 3: XÂY DƯNG CHƯƠNG TRÌNH NHẬN DIỆN PHÁT HIỆN TÀI XẾ BUỒN NGỦ 31](#_Toc190347800)

[3.1. Mô tả dữ liệu 31](#_Toc190347801)

[3.1.1 Cấu trúc tập dữ liệu: 31](#_Toc190347802)

[3.1.2 Sử dụng trong phát hiện buồn ngủ 31](#_Toc190347803)

[3.1.3 Phân chia bộ dữ liệu 32](#_Toc190347804)

[3.2. Những thư viện được dử dụng trong bài toán 32](#_Toc190347805)

[3.2.1 Thư viện Torch (PyTorch) 32](#_Toc190347806)

[3.2.2 Thư viện OpenCV-Python 33](#_Toc190347807)

[3.2.3 Thư viện numpy 34](#_Toc190347808)

[3.3. Xây dựng mô hình dự đoán (YoLov5) 35](#_Toc190347809)

[3.3.1 Thu thập và phân tích dữ liệu 35](#_Toc190347810)

[3.3.2 Tiền Xử lý dữ liệu 35](#_Toc190347811)

[3.3.3 Huấn Luyện mô hình 35](#_Toc190347812)

[3.3.4 Kiểm thử mô hình 35](#_Toc190347813)

[3.3.5 Tải mô hình giải quyết nhận diện phát hiện buồn ngủ 36](#_Toc190347814)

[3.3.6 Đánh Gía mô hình bài toán 37](#_Toc190347815)

[3.4. Xây dựng mô hình dự đoán (Faster R-CNN) 38](#_Toc190347816)

[3.4.1 Tiền Xử lý dữ liệu 38](#_Toc190347817)

[3.4.2 Huấn luyện mô hình 39](#_Toc190347818)

[3.4.3 Tải mô hình giải quyết nhận diện phát hiện buồn ngủ 39](#_Toc190347819)

[3.4.4 Đánh Gía mô hình bài toán 41](#_Toc190347820)

[KẾT LUẬN 43](#_Toc190347821)

[TÀI LIỆU THAM KHẢO 45](#_Toc190347822)

# **DANH MỤC HÌNH ẢNH**

[Hình 1.4.1: Biểu tượng ngôn ngữ lập trình Python 16](#_Toc190352921)

[Hình 2.1: Sơ đồ tổng quát 18](#_Toc190352922)

[Hình 2.3: Kiến trúc mạng YoLo 19](#_Toc190352923)

[Hình 2.3.1: Quy trình nhận diện của YoLo 20](#_Toc190352924)

[Hình 2.3.2: Minh họa tính tọa đội IoU 21](#_Toc190352925)

[Hình 2.4.4: Khả năng nhận diện của YoLov5 24](#_Toc190352926)

[Hình 2.5: Mô Hình R-CNN 26](#_Toc190352927)

[Hình 2.6: Mô hình faster R-CNN 27](#_Toc190352928)

[Hình 3.1a: gộp dữ liệu 31](#_Toc190352929)

[Hình 3.1b chia dữ liệu tập train và test 32](#_Toc190352930)

[Hình 3.3.5a: Lấy dữ liệu vào 38](#_Toc190352931)

[Hình 3.3.5b: Vẽ bounding box 38](#_Toc190352932)

[Hình 3.4.3a: Xử lý dữ liệu chuyển đổi định dạng nhãn 41](#_Toc190352933)

[Hình 3.4.3b: Dự đoán và vẽ bounding box 42](#_Toc190352934)

# **DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ**

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Thuật Ngữ | Giải thích |
| CNN | Convolutional Neural Network | Mạng nơ-ron tích chập |
| R-CNN | Region-based Convolutional Neural Network | là một phương pháp phát hiện đối tượng (object detection) trong lĩnh vực học sâu (deep learning) |
| YOLO | You Only Look Once | Một mô hình học sâu dùng để nhận diện đối tượng trong ảnh |
| LSTM | Long Short-Term Memory | Mạng nơ-ron hồi tiếp dài hạn |
| GPU | Graphics Processing Unit | Đơn vị xử lý đồ họa |
| CPU | Central Processing Unit | Bộ xử lý trung tâm |
| IoU | Intersection over Union | là một chỉ số đánh giá độ chính xác của các mô hình phân tích ảnh |
| mAP | mean Average Precision | chỉ số trung bình độ chính xác của mô hình |
| P | Precision | đo lường tỷ lệ dự đoán đúng trong tổng số dự đoán được tạo ra. |
| R | Recall | đo lường tỷ lệ số lượng đối tượng thực tế được phát hiện đúng. |
| GRU | Gated Recurrent Unit | mạng nơ-ron hồi tiếp |
| NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
| COCO | Common Objects in Context | Một tập dữ liệu lớn |
| RPN | Region Proposal Network | các vùng khả nghi chứa đối tượng trong mạng |

# **MỞ ĐẦU**

Hiện nay tai nạn giao thông đang diễn ra vô cùng nghiêm trọng, khó giải quyết. Nó diễn ra hàng ngày gây thiệt hại vô cùng to lớn về người và của. Và một trong những nguyên nhân trực tiếp gây ra tai nạn chính là chủ cầm lái lâm vào trạng thái buồn ngủ không tỉnh táo.

Hiện tượng buồn ngủ khi đang điều khiển phương tiện nguy hiểm hơn là không thể xác định được khi nào nó đến. Có thể trước đó đang trong trạng thái hoàn toàn tỉnh táo, nhưng do sự cộng hưởng của chuyển động xe, không gian, âm thanh,… Cơn buồn ngủ có thể bất chợt xảy ra ngay lập tức. Dù có thể đã cố gắng tỉnh táo hết sức nhưng cơn buồn ngủ vẫn gây ảnh hưởng đến phản ứng vô điều kiện của tài xế khi cần phanh, đánh lái…

Giáo sư Telfilo Lee Chiong đã từng có báo cáo tại hội nghị khoa học thường niên Việt Nam năm 2015 về việc tai nạn giao thông với triệu chứng rối loạn giấc ngủ. Ông cho biết giấc ngủ không điều độ là một trong số nguyên nhân chính dẫn đến hậu quả thương tâm trong tai nạn giao thông thế giới. Số trường hợp tai nạn xe được ước tính lên đến 15% trong tổng số trường hợp. Đã có nghiên cứu tổng rà soát ở các nước Châu Âu (`19 quốc gia) cho thấy trung bình lên đến 17% tài xế ở trạng thái không tỉnh táo do buồn ngủ trong quá trình lái xe. Trong đó có tới 7% trường hợp đã từng trong trạng thái này mà gây tai nạn, còn lại là tỷ lệ số trường hợp không tỉnh táo trong quá trình lái xe ít nhất một lần trong tháng tới 10,8 và suýt xảy ra tai nạn với khoảng 18% [11].

Với tình hình như vậy, việc xây dựng hệ thống nhận diện phát hiện tài xế buồn ngủ đồng thời đưa ra tín hiệu cảnh báo là rất quan trọng trong công cuộc giảm thiểu tai nạn giao thông . Đồ án này đề xuất sử dụng hai mô hình học sâu để chọn ra mô hình có kết quả tối ưu nhất. Mô hình sẽ được kiểm nghiệm trên bộ dư liệu ảnh khuôn mặt với trọng điểm là mắt và video được cắt từ camera hành trình.

Trong đồ án này đề xuất hai mô hình YoLov5 và R-CNN. Tận dụng ưu điểm của cả hai mô hình để nhận diện nhanh và chính xác nhất. Đánh giá mô hình dựa trên các chỉ số P, R, F1 score và mAP để đảm bảo tính chính xác đạt hiệu suất tốt nhất trong việc nhận diện lái xe buồn ngủ. Trong đồ án này trình bày các bước giải thuật và xây dựng chương trình minh họa.

**Bố cục đồ án gồm có 3 chương:**

Chương 1: Tổng quan về bài toán nhận diện tài xế buồn ngủ

Chương 2: Giải pháp đề xuất

Chương 3: Thực nghiệm và đánh giá kết quả

# **CHƯƠNG 1: TỔNG QUAN VỀ BÀI TOÁN NHẬN DIỆN PHÁT HIỆN TÀI XẾ BUỒN NGỦ**

## **Mô Tả bài toán**

Buồn ngủ là trạng thái tự nhiên của con người trong một khoảng thời gian không nhất quán dựa vào giấc ngủ có đủ hay không. Trong khoảng thời điểm này người ta thường cảm thấy mệt mỏi dẫn đến thiếu tỉnh táo có thể nhắm mắt tạm thời để giảm thiểu cơn buồn ngủ hoặc thậm chí có thể mất ý thức một phần do đó khả năng phản ứng với tác động đột ngột sẽ ảnh hưởng. Giấc ngủ trắng là tình trạng mà con người rơi vào trạng thái mất ý thức tạm thời trong một khoảng thời gian rất ngắn và không có ý thức được các tác động xảy ra xung quanh tương tự với "Microsleep".

Tai nạn giao thông do chủ thể điều khiển xe buồn ngủ mất ý thức trong thời gian ngắn xảy ra ở mức độ từ nghiêm trọng đến đặc biệt nghiêm trọng, dẫn đến những hậu quả khủng khiếp là những vụ tai nạn thương tâm [12]. Nhận diện phát hiện tài xế buồn ngủ và đưa ra tín hiệu cảnh báo kịp thời sẽ là đóng góp quan trọng trong công cuộc giảm thiểu tai nạn giao thông.

Bài toán đặt ra là phát hiện buồn ngủ của tài xế khi đang lái xe qua quan sát nhận diện mắt. Dựa trên bộ dữ liệu được thu thập từ những hình ảnh trạng thái của mắt và video được cắt ra từ camera hành trình. Để thực hiện giải quyết vấn đề này, đồ án đề xuất sử dụng lai hai mô hình học sâu yolov5 và faster R-CNN.

Mục tiêu của đồ án là xây dựng một mô hình nhận diện chính xác, với tốc độ ngắn bảo đảm đưa ra dự đoán cảnh báo nhanh nhất cho tài xế khi có dấu hiệu thiếu tỉnh táo, tạo một chuyến hành trình an toàn.

Đồ án đề xuất chọn mô hình yolov5 và faster R-CNN do những ưu điểm nổi bật của hai mô hình và mong muốn khả năng sau nhận diện tối ưu nhất. Mô hình sẽ được huấn luyện và kiểm thử trên bộ dữ liệu cụ thể gồm hình ảnh và video, đánh giá sự hiệu quả qua các chỉ số F1 score, R, P và mAP. Mục tiêu cuối cùng là tạo ra công cụ cảnh báo hỗ trợ tài xế lái xe để có hành trình an toàn.

## **Tình hình nghiên cứu quốc tế và trong nước hiện nay**

Nghiên cứu về phát hiện buồn ngủ khi lái xe là lĩnh vực nhận được rất được quan tâm, và đã không ít dự án nghiên cứu đạt kết quả khả quan. Việc nhận diện phát hiện buồn ngủ khi lái xe đóng vai trò quan trọng trong giảm thiểu tai nạn giao thông tránh hậu quả thương tâm.

Tình trạng tai nạn giao thông do tài xế buồn ngủ, không tỉnh táo khi điều khiển xe trong những năm gần đây là vấn đề đáng lo ngại. Các dự án nghiên cứu ở nước ngoài gần đây đã phát triển các mô hình nhận diện chính xác trạng thái tài xế. Một số dự án điển hình như sử dụng mô hình học sâu mạng tích chập CNN [4] tập trung vào việc phát hiện hiện tượng microsleep-trạng thái ngủ ngắn chỉ vài giây kỳ vọng giúp giảm thiểu và ngăn ngừa tai nạn do microsleep thông qua việc theo dõi nháy mắt của người lái xe, EEG kết hợp TE [5] làm rõ mối quan hệ giữa kết nối não và hiệu suất lái xe trong bối cảnh mệt mỏi, một vấn đề lớn về an toàn giao thông,…

Tại Việt Nam, cũng có rất nhiều nghiên cứu sử dụng trí tuệ nhân tạo về phát hiện cảnh báo tình trạng buồn ngủ khi lái xe. Tại sự kiện “Ngày Trí tuệ nhân tạo (AI Day 2022)” với chủ đề "Kiến tạo tương lai", diễn ra trong hai ngày 26 và 27-8 tại Hà Nội,  Công ty Nghiên cứu và ứng dụng trí tuệ nhân tạo VinAI giới thiệu “Hệ thống giám sát người lái”[3]; nhóm nghiên cứu  thầy và trò Trường THPT Chi Lăng, huyện Chi Lăng với mục đích hạn chế tai nạn giao thông sử dụng kỹ thuật học máy [6],…

Các nghiên cứu trên cho thấy sức mạnh xử lý của các mô hình học sâu, học máy. Trong đề tài này sử dụng lai kết hợp hai mô hình có khả năng xử lý dữ liệu nhanh chính xác. Bộ dữ liệu thử nghiệm gồm tập hình ảnh trạng về mắt, gương mặt và video trích xuất từ camera hành trình.

Đề tài "**G**iám sát hành vi tài xế để phát hiện và cảnh báo tình trạng buồn ngủ" đưa ra một cách giải quyết theo ứng dụng học sâu tại Việt Nam. Việc đề xuất lai kết hợp mô hình Yolo và faster R-CNN cho thấy tiềm năng trong việc cải thiện độ chính xác của giám sát hành vi. Điều này có ý nghĩa trong việc giảm thiểu tai nạn giao thông.

Mong rằng nghiên cứu của đồ án này sẽ đóng góp vào việc giảm thiểu tai nạn giao thông một cách hiệu quả.

## **1.3 Học Sâu (Deep Learning)**

### **Tổng Quan**

Deep Learning (học sâu) là một mạng nơ-ron nhiều lớp với số lớp tối thiểu là ba, thuộc tập con của Machine Learning (học máy). Deep learning được huấn luyện , “ học” qua bộ dữ liệu với số lượng lớn, nhờ những nơ-ron thần kinh mô phỏng bộ não con người.

Deep Learning (học sâu) chiếm tỷ lệ cao trong công cuộc phát triển ứng dụng và dịch vụ thuộc lĩnh vực trí tuệ nhân tạo (AI) với mục đích cải thiện tự động hóa, giảm thiểu công việc cho con người trong việc phân tích và cần về mặt vật lý.

Được ứng dụng phổ biến trong hầu hết các sản phẩm, dịch vụ hàng ngày: chụp X-Quang trong y khoa, phiên dịch (google translate), tìm kiếm bài hát, đề xuất mục yêu thích trong các ứng dụng mạng xã hội,…

Mạng nơ-ron trong Deep Learning (học sâu) mô phỏng cách tư duy hoạt động của não người với hướng hoạt động:

* Với mỗi mạng nơ-ron gồm nhiều lớp không giống nhau, số lớp càng nhiều thì mạng sẽ càng tối ưu. Các lớp được liên kết với các lớp khác liền kề và trong mỗi lớp đều có các nút mạng (node).
* Mức độ ảnh hưởng của liên kết giữa các node đến mạng nơ-ron tùy thuộc vào trọng số tương ứng của mối nối, trọng số càng cao tương ứng mức độ ảnh hưởng càng lớn.
* Mỗi nơ-ron đều có chức năng kích hoạt để chuẩn hóa đầu ra.
* Users nhập dữ liệu vào mạng nơ-ron, di chuyển lần lượt tất cả các lớp và trả kết quả ở lớp cuối cùng, hay còn gọi là lớp đầu ra.
* Các trọng số sẽ thay đổi trong suốt quá trình mô hình mạng được huấn luyện, việc cần làm của mô hình là tìm ra tập giá trị các trọng số với phán đoán đúng nhất.

Deep Learning (học sâu) phát triển nhiều ứng dụng trong lĩnh vực AI cho các sản phẩm được sử dụng hàng ngày: hệ thống face ID, đề xuất yêu thích trong các mạng xã hội và các sàn thương mại điện tử,… Và là lựa chọn hàng đầu cho doanh nghiệp trong việc phân tích dữ liệu và nhiều ứng dụng cần thiết khác.

1. Ưu điểm của Học Sâu (Deep Learning)

* **Xử lý dữ liệu phức tạp**: Với khả năng rút ra các đặc trưng tự động từ dữ liệu của mạng nơ-ron, deep learng cho thấy khả năng xử lý mạnh mẽ ngay cả những tập dữ liệu phức tạp như âm thanh, hình ảnh,…
* **Hiệu suất cao trong nhiều ứng dụng**: Học sâu đã chứng minh hiệu quả vượt trội trong các bài toán như nhận dạng hình ảnh, nhận diện giọng nói, dịch ngôn ngữ tự động, xe tự lái, và các ứng dụng y tế như phát hiện bệnh từ hình ảnh y tế.
* **Khả năng học từ lượng dữ liệu lớn**: Mạng nơ-ron sâu có thể học và cải thiện hiệu suất khi có một lượng dữ liệu lớn, giúp nâng cao độ chính xác của mô hình, đặc biệt trong các ứng dụng lớn như nhận diện hình ảnh.
* **Học tự động các đặc trưng**: Với khả năng tự động trích xuất các đặc trưng của mạng nơ ron thì quá trình chuẩn bị dữ liệu được tối giản khi không cần quá phụ thuộc vào kiến thức chuyên sâu.
* **Khả năng tổng quát tốt**: Một khi đã được huấn luyện tốt, các mô hình học sâu có thể tổng quát rất tốt, tức là có thể áp dụng hiệu quả vào những dữ liệu chưa từng gặp trong quá trình huấn luyện.

1. Nhược điểm của Học Sâu (Deep Learning)

* **Yêu cầu tài nguyên tính toán lớn**: Học sâu đòi hỏi phần cứng mạnh mẽ (như GPU) và bộ nhớ lớn để huấn luyện mô hình, đặc biệt khi làm việc với bộ dữ liệu khổng lồ. Điều này có thể tốn kém về chi phí và thời gian.
* **Khó giải thích (Black-box)**: Mô hình học sâu thường được coi là "hộp đen", tức là khó giải thích vì sao mô hình đưa ra một dự đoán nhất định.
* **Độ lớn và đa dạng dữ liệu huấn luyện**: Nếu dữ liệu không đầy đủ hoặc không đại diện cho tất cả các tình huống có thể xảy ra, mô hình có thể không hoạt động tốt hoặc bị sai lệch.
* **Thời gian huấn luyện dài**: Việc huấn luyện các mô hình học sâu có thể mất rất nhiều thời gian, đôi khi hàng tuần hoặc thậm chí hàng tháng, đặc biệt với các mạng nơ-ron phức tạp và dữ liệu khổng lồ.
* **Quá tải dữ liệu (Overfitting)**: Nếu không được kiểm soát tốt, các mô hình học sâu có thể dễ bị quá tải dữ liệu, nghĩa là chúng học quá sâu vào các chi tiết của dữ liệu huấn luyện và mất khả năng tổng quát tốt trên dữ liệu chưa thấy.
* **Tối ưu hóa và siêu tham số phức tạp**: Cần phải điều chỉnh nhiều siêu tham số (hyperparameters) của mô hình, như số lượng lớp, số lượng nơ-ron mỗi lớp, tốc độ học (learning rate), v.v., điều này có thể rất tốn thời gian và khó khăn.

1. **Các loại mô hình học sâu**

Trong học sâu có nhiều mô hình khác nhau được phát triển để giải quyết các bài toán, một số mô hình phổ biến được sử dụng như sau: CNN, RNN, LSTM,...

1. Mạng Nơ-ron tích chập (CNN)

CNN là một loại mạng nơ-ron nhân tạo đặc biệt hiệu quả trong việc xử lý và phân tích dữ liệu dạng hình ảnh hoặc dữ liệu có cấu trúc lưới (như âm thanh, video, hay tín hiệu thời gian). CNN được thiết kế để mô phỏng cách mà hệ thống thị giác của con người xử lý hình ảnh, qua đó phát hiện các đặc trưng của hình ảnh như cạnh, góc, kết cấu và các yếu tố hình ảnh phức tạp khác.

1.3.2.1.1. Các thành phần chính của Mạng Nơ-ron Tích Chập (CNN)

CNN cấu tạo bởi mạng lưới lớp khác nhau với chức năng thực hiện cụ thể khác nhau đưu ra các các đặc trưng của dữ liệu. Các thành phần chính của CNN bao gồm:

1. **Lớp Tích Chập (Convolutional Layer)**:

* Đây là thành phần quan trọng nhất của CNN, nơi thực hiện phép toán **tích chập** (convolution) giữa một **hạt nhân (kernel)** hoặc **filter** và ảnh đầu vào.
* Mỗi hạt nhân có kích thước nhỏ (ví dụ: 3x3, 5x5) và được di chuyển trên toàn bộ ảnh đầu vào, thực hiện phép tính tích chập để tạo ra các bản đồ đặc trưng (feature maps).

1. **Lớp Kích Thước Giảm (Pooling Layer)**:

* Bản đồ đặc trưng sẽ được giảm kích thước ở lớp này sau khi được xử lý ở lớp tích chập. Từ đó giảm độ phức tạp tính toán và hiện tượng overfitting.
* Có hai loại pooling phổ biến:
* Max pooling: Chọn giá trị lớn nhất trong mỗi vùng con của bản đồ đặc trưng.
* Average pooling: Tính giá trị trung bình trong mỗi vùng con của bản đồ đặc trưng.

1. **Lớp Kết Nối Đầy Đủ (Fully Connected Layer - FC)**:

* Sau khi thông tin từ các lớp tích chập và pooling được trích xuất, nó sẽ được đưa vào một hoặc nhiều lớp kết nối đầy đủ (fully connected layers) để phân loại hoặc thực hiện các dự đoán.

1. **Lớp Softmax**:

* Đây là lớp được sử dụng để phân loại trong các bài toán phân loại đối tượng. Lớp này sử dụng hàm **Softmax** để chuyển đổi đầu ra thành xác suất, giúp xác định lớp của đối tượng trong ảnh.
* **Softmax** giúp xác định đối tượng nào trong các lớp phân loại có xác suất cao nhất.

1.3.2.1.2. Ưu điểm của CNN:

* **Tự học đặc trưng**: CNN có khả năng học từ dự liệu các đặc trưng.
* **Giảm số lượng tham số**: Các lớp tích chập sử dụng bộ lọc nhỏ để thực hiện.
* **Khả năng kháng biến dạng**: Nhờ vào pooling và các phương pháp khác, CNN có khả năng chống lại sự thay đổi về vị trí, kích thước, góc độ của đối tượng trong hình ảnh.

1.3.2.1.3. Ứng dụng của CNN

* **Nhận dạng hình ảnh và phân loại ảnh**: Được sử dụng phổ biến với mục đích phân loại trong các bài toán
* **Phát hiện đối tượng**: Một trong số các mô hình ưu tiên sử dụng trong việc nhận diện.
* **Nhận dạng chữ viết tay**: Các hệ thống như MNIST, dùng CNN để nhận diện chữ viết tay.
* **Xử lý video**: CNN có thể được sử dụng trong các bài toán phân tích video, như nhận diện hành động.

CNN đã trở thành công cụ cực kỳ mạnh mẽ và phổ biến trong các bài toán thị giác máy tính, đặc biệt khi kết hợp với các kỹ thuật huấn luyện mạng nơ ron sâu (Deep Learning).

1. Mạng Nơ-ron hồi tiếp (RNN)

**Mạng Nơ Ron Hồi Tiếp (RNN) được xây dựng với mục đích xử lý dữ liệu tuần tự, khi mà thông tin ở các bước có thể phụ thuộc vào nhau**. Các mạng này có khả năng "nhớ" thông tin từ quá khứ thông qua các kết nối hồi tiếp giữa các đơn vị nơ ron, cho phép mô hình học và xử lý các chuỗi thời gian hoặc dữ liệu tuần tự như văn bản, âm thanh, video, và các tín hiệu thời gian khác.

**1.3.2.2.1.** Cấu trúc cơ bản của RNN

RNN khác biệt so với các mạng nơ ron truyền thống ở chỗ nó có các **kết nối hồi tiếp** giữa các nơ ron trong cùng một lớp, cho phép nó duy trì trạng thái và truyền thông tin qua các bước thời gian (timesteps). Mỗi bước trong chuỗi đầu vào không chỉ được xử lý dựa trên chính nó, mà còn có sự ảnh hưởng của các bước trước đó nhờ vào các kết nối hồi tiếp.

Quá trình tính toán của RNN tại mỗi thời điểm t là:

Trong đó:

* Wy , Wh  , Wx : là các ma trận trọng số
* by, bh: là các độ dịch
* σ: là hàm kích hoạt thường là tanh hoặc ReLU
* h(t): là trạng thái ẩn tại thời điểm t, chứa thông tin được "nhớ" từ các bước trước.
* X(t): là đầu vào tại điểm t
* Y(t): đầu ra tại điểm t

**1.3.2.2.1.** Ưu và nhược điểm của RNN

Ưu điểm:

* **Khả năng học chuỗi**: RNN đã tận dụng rất tốt ưu điểm của mình, khi đây là mục đích mô hình này được xây dựng.
* **Xử lý dữ liệu có phụ thuộc thời gian**: RNN giúp học các mối quan hệ trong chuỗi dữ liệu, ví dụ, trong văn bản, một từ có thể phụ thuộc vào từ trước đó.

Nhược điểm:

* **Vấn đề vanishing/exploding gradient**: Khi huấn luyện RNN, đặc biệt là trong các chuỗi dài, vấn đề vanishing gradient (gradients quá nhỏ) và exploding gradient (gradients quá lớn) có thể xảy ra, làm cho quá trình huấn luyện trở nên khó khăn.
* **Khó khăn trong việc lưu trữ thông tin dài hạn**: Mặc dù RNN có thể "nhớ" thông tin từ các bước trước, nhưng với chuỗi dài, khả năng giữ thông tin lâu dài (long-term dependencies) của nó không mạnh mẽ, dẫn đến việc khó khăn trong việc học các phụ thuộc dài hạn.

**1.3.2.2.2. Các biến thể của RNN**

Để khắc phục các nhược điểm của RNN cơ bản, một số biến thể của RNN đã được phát triển, bao gồm:

1. **Long Short-Term Memory (LSTM)**:

* LSTM là một dạng cải tiến của RNN, được thiết kế để giải quyết vấn đề vanishing gradient và giữ được thông tin trong khoảng thời gian dài hơn.
* **Ứng dụng**: Dự báo chuỗi thời gian, dịch máy, nhận dạng giọng nói.

1. **Gated Recurrent Unit (GRU)**:

* GRU là một biến thể khác của RNN và LSTM, với cấu trúc đơn giản hơn nhưng vẫn duy trì hiệu suất mạnh mẽ trong việc học các chuỗi dài.
* **Ứng dụng**: phiên dịch, ....

**1.3.2.2.3. Ứng dụng của RNN**

RNN và các biến thể của nó (LSTM, GRU) được sử dụng rộng rãi trong các ứng dụng xử lý chuỗi và thời gian:

1. **Xử lý ngôn ngữ tự nhiên (NLP):**

* **Phiên Dịch**: Phiên dịch ngôn ngữ văn bản..
* **Phân tích cảm xúc**: Xác định cảm xúc (tích cực, tiêu cực) trong văn bản.
* **Tóm tắt văn bản**: Tóm tắt các văn bản dài thành các đoạn văn ngắn hơn.
* **Nhận diện thực thể tên (NER)**: Phát hiện các thực thể như tên, địa điểm trong văn bản.

1. **Nhận diện giọng nói:**
2. **Dự báo chuỗi thời gian:**

* **Dự báo giá trị thị trường chứng khoán**
* **Dự báo khí tượng**: Dự đoán thời tiết dựa trên dữ liệu thời gian.

1. **Video và hình ảnh:**

RNN cũng có thể được sử dụng trong phân tích video, nơi thông tin về các khung hình (frame) cần được xử lý theo tuần tự.

1.3.2.3. LSTM

LSTM là một biến thể của mạng nơ ron hồi tiếp (Recurrent Neural Network - RNN) được thiết kế để giải quyết các vấn đề mà RNN truyền thống gặp phải, đặc biệt là **vấn đề vanishing gradient** và **exploding gradient** khi học từ các chuỗi dữ liệu dài.

* + - * 1. **Cấu trúc của LSTM**

LSTM có một cấu trúc phức tạp hơn RNN thông thường, với các cổng (gates) đặc biệt giúp điều khiển dòng chảy của thông tin qua mạng nơ ron. Cấu trúc chính của một đơn vị LSTM gồm ba cổng chính:

1. **Cổng quên (Forget Gate)**:

* Cổng này quyết định phần nào của trạng thái ẩn (hidden state) trước đó sẽ bị loại bỏ.
* Input: Trạng thái ẩn trước đó và đầu vào hiện tại
* Output: Từng phần của trạng thái ẩn được gán giá trị trong khoảng [0,1].
* **Công thức**:

Trong đó:

* **f(t)** là đầu ra của cổng quên
* **σ** là hàm kích hoạt sigmoid.
* là trọng số, là độ dịch.

1. **Cổng vào (Input Gate)**:

* Cổng này bao gồm một hàm sigmoid (giống như cổng quên) để quyết định "lượng" thông tin cần thêm vào trạng thái, và một hàm tanh để tạo ra một giá trị ứng với thông tin mới.
* **Công thức**:

**Trong đó:**

* **i(t)** là cổng vào (input gate).
* (t) là thông tin tiềm năng mới sẽ được thêm vào trạng thái.

1. **Cổng đầu ra (Output Gate)**:

* Cổng này quyết định phần nào của trạng thái ẩn sẽ được xuất ra làm đầu ra của LSTM.
* **Công thức**:

Trong đó:

* **o(t)** là cổng đầu ra (output gate).
* **C(t)** là trạng thái nhớ (cell state) tại thời điểm t.
* **h(t)** là trạng thái ẩn (hidden state) tại thời điểm t, được xuất ra.

1. **Cập nhật trạng thái nhớ (Cell State Update)**:

* Trạng thái nhớ (cell state) chứa thông tin dài hạn và quyết định thông tin nào sẽ được lưu lại hoặc quên đi qua thời gian.
* Trạng thái nhớ được cập nhật như sau:

Trong đó:

* **C(t-1)** là trạng thái nhớ của thời điểm trước.
* **C(t)** là trạng thái nhớ của thời điểm hiện tại.

**1.3.2.3.2. Quá trình tính toán của LSTM**

Mỗi bước thời gian trong LSTM đều sử dụng ba cổng trên để điều khiển và cập nhật trạng thái nhớ, đồng thời tạo ra trạng thái ẩn và đầu ra. Cụ thể:

* Cổng quên sẽ quyết định thông tin nào từ trạng thái nhớ cũ cần bị quên.
* Cổng vào sẽ quyết định thông tin nào sẽ được lưu vào trạng thái nhớ.
* Cổng đầu ra sẽ quyết định trạng thái ẩn nào sẽ được xuất ra.

Nhờ vào các cổng này, LSTM có thể "ghi nhớ" thông tin trong thời gian dài mà không gặp phải vấn đề vanishing gradient như RNN.

**1.3.2.3.3. Ưu điểm của LSTM**

* **Giải quyết vấn đề vanishing gradient**: Các cổng trong LSTM giúp duy trì thông tin qua các bước thời gian, cho phép nó học được các phụ thuộc dài hạn mà không bị mất đi thông tin quan trọng trong quá trình huấn luyện.
* **Khả năng học các phụ thuộc dài hạn**: LSTM đặc biệt hiệu quả trong việc xử lý dữ liệu có sự phụ thuộc lâu dài giữa các phần tử (ví dụ, trong các văn bản dài hay chuỗi thời gian dài).
* **Ứng dụng rộng rãi**: LSTM là lựa chọn phổ biến trong các bài toán như phân tích chuỗi thời gian, dịch máy, nhận diện giọng nói, và các tác vụ xử lý ngôn ngữ tự nhiên.
  + - * 1. **Ứng dụng của LSTM**

1. **Xử lý ngôn ngữ tự nhiên (NLP):**

* **Phiên Dịch**: Phiên dịch ngôn ngữ văn bản.
* **Phân tích cảm xúc**: Xác định cảm xúc của văn bản (tích cực, tiêu cực).
* **Tóm tắt văn bản**: Tạo ra bản tóm tắt ngắn gọn từ văn bản dài.
* **Phân tích cú pháp ngữ pháp**: Phân tích cấu trúc của câu.

1. **Dự báo chuỗi thời gian:**

* **Dự đoán giá thị trường chứng khoán**
* **Dự báo thời tiết**: Dự đoán thời tiết trong tương lai từ dữ liệu thời gian.

1. **Nhận diện giọng nói:**

LSTM có thể chuyển đổi từ âm thanh thành văn bản

1. **Nhận diện hình ảnh:**

LSTM cũng có thể được kết hợp với các mạng nơ ron tích chập (CNN) để xử lý dữ liệu hình ảnh theo chuỗi (ví dụ, nhận dạng đối tượng trong video).

1. **Phân tích video:**

Xử lý video trong các tác vụ nhận diện hành động hoặc phân loại video.

## **1.4. Ngôn ngữ sử dụng**

1. **Tổng Quan về ngôn ngữ lập trình Python**

[Python](https://csc.edu.vn/data-science-machine-learning/Fundamentals-of-Python-Lap-trinh-Python-co-ban_189) được biết đến lần đầu tiên vào năm 1991 với cha đẻ là Guido van Rossum. Khác với nhiều ngôn ngữ trước đó Python thực thi theo dòng lệnh, tính từ trên xuống. Hay còn được gọi là ngôn ngữ lập trình thông dịch. Python được sự dụng phổ biến bởi sự đa dạng trong hỗ trợ code (OOP,..).



Hình 1.4.1: Biểu tượng ngôn ngữ lập trình Python

Python trở nên phổ biến khi mà cách hoạt động và thực thi thuận tiện dễ dàng tiếp cận. Và có thể sử dụng cho nhiều mục đích ứng dụng khác nhau.

Python thuận tiện hơn khi có sẵn cung cấp khối thư viện và framework khổng lồ.

Phù hợp với mọi đối tượng và lĩnh vực, Python là một trong những lựa chọn hàng đầu trong những ngôn ngữ lập trình

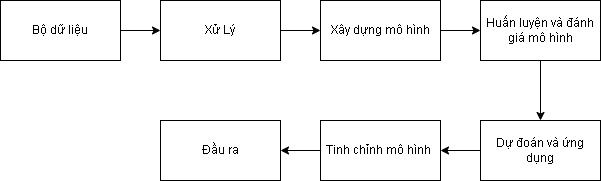
1. **Ưu và nhược điểm của ngôn ngữ Python**

* Ưu Điểm:
* **Dễ dàng trong học tập**
* **Giảm chi phí bảo trì**
* **Tránh tác hại lỗi phần mềm**
* **Khả năng ứng dụng**
* **Quản lý bộ nhớ**
* **Đơn giản, nhanh chóng**
* **Mã hóa không đồng bộ:** Mã hóa không đồng bộ sử dụng một vòng lặp sự kiện duy nhất để hoàn thành công việc trong những khoảng thời gian nhỏ.
* **Tích hợp các ngôn ngữ khác**
* **Tích hợp ứng dụng doanh nghiệp**
* Nhược Điểm:
* Tốc độ thực thi chậm
* Tiêu thụ lượng lớn bộ nhớ.
* Không thích hợp cho phát triển game và mobile device
* Hạn chế khả năng của developer
* Phát hiện lỗi trong mã: Vì Python được thực thi thông qua trình thông dịch thay vì trình biên dịch, nên không thể phát hiện lỗi trong quá trình biên dịch và điều đó không tốt cho các nhà phát triển.
* Quyền truy cập cơ sở dữ liệu
* Hạn chế thiết kế: Một trong những vấn đề quan trọng của Python là các hạn chế về thiết kế của nó.
* Khó kiểm tra.

# **CHƯƠNG 2: MÔ HÌNH XỬ LÝ BÀI TOÁN NHẬN DIỆN PHÁT HIỆN TÀI XẾ BUỒN NGỦ**

## **Tổng quát bài toán nhận diện**

* Đầu vào: hình ảnh, video cắt ra từ camera hành trình.
* Đầu ra: Kết quả nhận diện tình trạng của tài xế khi đang lái xe

****

Hình 2.1: Sơ đồ tổng quát

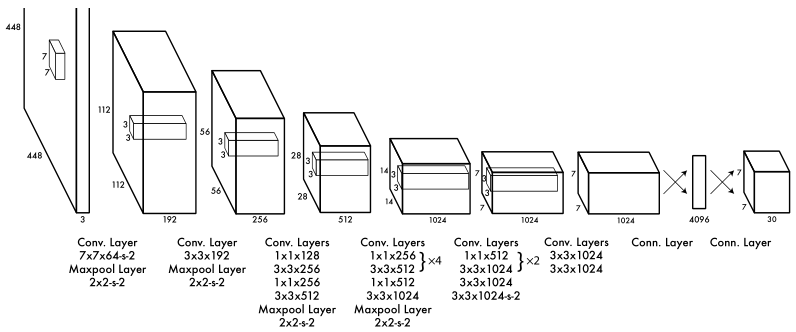
Nhận diện phát hiện tình trạng trạng thái của tài xế khi đang lái xe được xây dựng bằng cách sử dụng thuật toán học sâu. Mô hình nhận diện diện phát hiện tình trạng trạng thái tài xế dựa trên hình ảnh và video cắt từ camera hành trình.

## **2.2. Chi tiết các bước thực hiện**

* Bộ dữ liệu ban đầu: Thu thập dữ liệu đa dạng sắc thái, góc độ
* Xử lý: chỉnh sửa định dạng, size ảnh đồng bộ, dán nhãn dữ liệu.
* Xây dựng mô hình: Có thể sử dụng mô hình mạng nơ-ron tích chập (CNN), 3D CNN, YOLO, R-CNN…. Mô hình này sẽ học từ các hình ảnh trong dữ liệu huấn luyện.
* Huấn luyện và đánh giá: Dữ liệu sau khi huấn luyện được dùng để tinh chỉnh tham số và lựa chọn mô hình tối ưu. Dựa vào các chỉ số đánh giá mô hình, đảm bảo tính khả quan đáng tin cậy.
* Tinh chỉnh mô hình: Sau khi huấn luyện muốn hiệu suất tốt hơn có thể tinh chỉnh mô hình, bao gồm: thử nghiệm mô hình với nhiều phiên bản khác nhau, thay đổi các lớp cuối,...
* Đầu ra: Đầu ra của mô hình trạng thái của tài xế khi đang lái xe được nhận diện.

## **2.3. Mô hình YoLo**

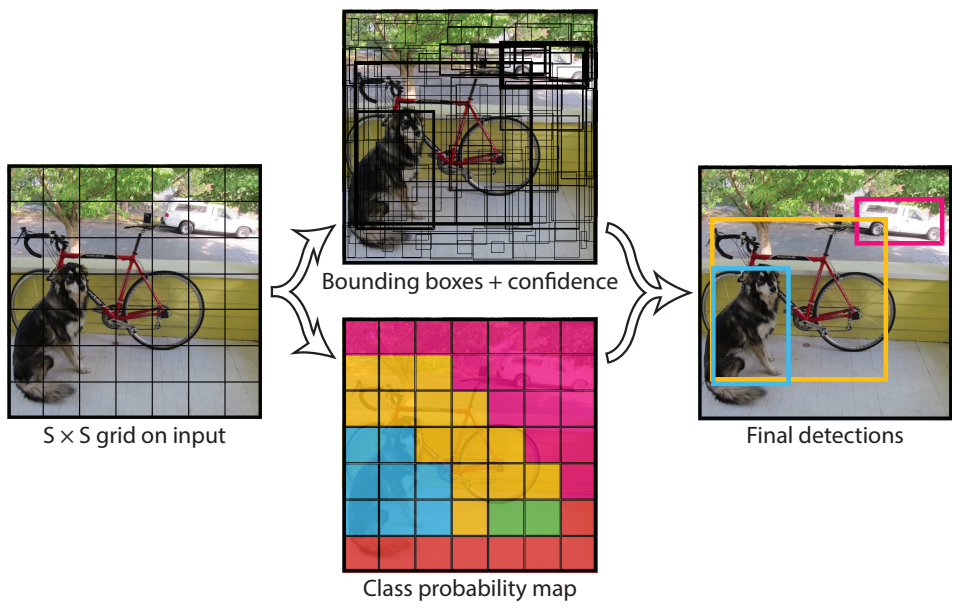
Yolo là mô hình được phát triển từ cấu trúc mạng CNN với mức hiệu quả hơn trong việc cho việc phát giác, nhận diện và phân loại đối tượng. Yolo được xây dựng qua quá trình kết hợp của lớp tích chập và lớp kết nối đầy đủ. Với lớp tích chập sẽ trích xuất tính năng ảnh, lớp liên kết đầy đủ sẽ dự đoán tọa độ của đối tượng và xác suất.



Hình 2.3: Kiến trúc mạng YoLo

### **2.3.1. Cơ chế hoạt động của YoLo**

Đầu vào của mô hình là một ảnh, bắt đầu quá trình rà soát phân tích ảnh, nếu có phát hiện đối tượng sẽ đến bước tiếp theo là xác định tọa độ đối tượng. Ảnh đầu vào được chia thành thành S×*S* ô thường thì sẽ là 3×3,  7×7,  9×9...



Hình 2.3.1: Quy trình nhận diện của YoLo

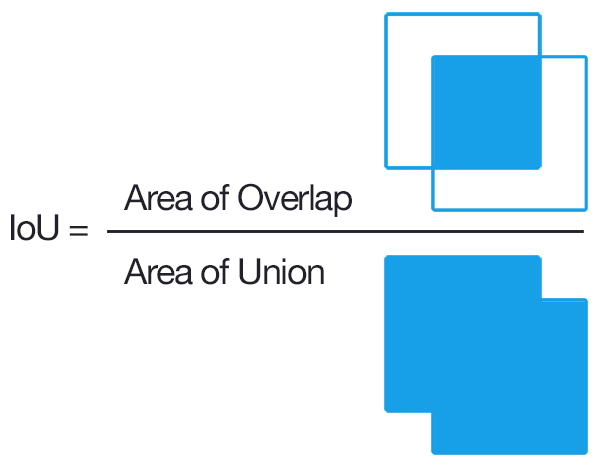
Input là một ảnh thì đầu ra là ma trận ba chiều có kích thước S × S × (5×N×M) với mỗi ô có lượng tham số bằng (5×M×N) với N là Box và M là Class, số cần dự đoán của mỗi ô.

Mỗi bounding box gồm 5 thành phần : (x, y, w, h, prediction), (x, y ) là tọa độ tâm, (w, h) với w là chiều rộng và h là chiều cao, công thức prediction:

Pr(Object) \* IOU(pred,truth)

### **2.3.2. Hàm tính IOU**

Prediction được định nghĩa bằng Pr(Object) \* IOU(pred,truth). IOU là hàm đánh giá độ chính xác của máy dò đối tượng trên bộ dữ liệu cụ thể. IOU được tính bằng:



Hình 2.3.2: Minh họa tính tọa đội IoU

Trong đó:

* Area of Overlap: diện tích của phần giao nhau giữa hộp giới hạn dự đoán với hộp giới hạn thực tế.
* Area of Union: diện tích của phần hợp giữa hộp giới hạn dự đoán với hộp giới hạn thực tế

### **2.3.3. Loss Function**

Loss Function YOLO được tính trên tập dự đoán và nhãn. Cụ thể Loss Function là tổng 3 độ lỗi sau :

* Classification loss : Độ lỗi dự đoán nhãn đối tượng
* Localization loss : Độ lỗi dự đoán kích thước, tọa độ tâm của hộp giới hạn
* Confidence loss: Độ lỗi dự đoán hộp giới hạn chứ đối tượng so với nhãn thực tế

2.3.3.1. **Classification loss**

Độ lỗi dự đoán loại nhãn đối tượng, hàm này chỉ tính trên những ô vuông có xuất hiện đối tượng. Công thức tính:

(c))2

2.3.3.2. **Localization loss**

Hàm này tính giá trị lỗi hộp giới hạn được dự đoán kích thước và tọa độ tâm so với thực tế từ bộ huấn luyện. Lưu ý chúng ta nên chuẩn hóa về [0, 1] so với tâm của hộp giới hạn trước khi tính giá trị hàm. Chuẩn hóa kích thước giúp mô hình dự đoán nhanh và chính xác hơn.

Giá trị hàm được tính bằng tổng giá trị lỗi dự đoán kích thước và tọa độ tâm hộp giới hạn dự đoán với hộp giới hạn thực tế. Tại mỗi ô chứa đối tượng, chọn hộp giới hạn có IOU tốt nhất, rồi tính độ lỗi theo các hộp giới hạn này.

Công thức tính giá trị hàm lỗi dự đoán tọa độ tâm hộp giới hạn dự đoán và tọa độ tâm hộp giới hạn thực tế:

Công thức tính giá trị hàm lỗi dự đoán kích thước của hộp giới hạn dự đoán so với hộp giới hạn thực tế :

2.3.3.3. **Confidence loss**

Độ lỗi giữa dự đoán hộp giới hạn chứa đối tượng so với nhãn thực tế tại ô vuông đó. Độ lỗi này tính trên tất cả các ô vuông.

2.3.3.4. **Total loss**

Ta có công thức tính hàm lỗi như sau:

Mô hình YOLO được xây dựng phát triển từ cấu trúc mạng CNN gồm các lớp tích chập kết hợp với lớp giảm phân giải tối đa và cuối cùng là hai lớp kết nối đầy đủ, với hàm kích hoạt tuyến tính cho lớp cuối cùng và tất cả những lớp khác sử dụng leaky RELU.

## **2.4. Mô hình YoLov5**

**Năm phát hành:** 2020  
YOLOv5 là một phiên bản quan trọng trong dòng mô hình YOLO (You Only Look Once), nổi bật với các tính năng tối ưu hóa hiệu suất, khả năng ứng dụng linh hoạt và khả năng triển khai dễ dàng.

**Đặc điểm chính của YOLOv5**

1. **Hiệu suất và Độ chính xác:**
   * YOLOv5 được thiết kế để đạt độ chính xác cao hơn trong phát hiện đối tượng, đồng thời duy trì tốc độ xử lý nhanh.
   * Sử dụng các phương pháp tối ưu hóa như **Mosaic Data Augmentation**, **AutoAnchor** và **Label Smoothing**, giúp cải thiện độ chính xác và khả năng khái quát hóa trên dữ liệu đa dạng.
2. **Triển khai linh hoạt:**
   * Được viết hoàn toàn bằng **PyTorch**, dễ dàng tích hợp với các hệ thống AI khác.
   * Hỗ trợ nhiều nền tảng: Google Colab, Docker, MXNet,…
3. **Cấu trúc mô hình:**

* **Backbone:** Từ hình ảnh đầu vào dùng CSPN để trích xuất các đặc trứng.
* **Neck:** Sử dụng kiến trúc Feature Pyramid Networks (FPN) và Path Aggregation Network (PAN) để kết hợp thông tin từ nhiều cấp độ.
* **Head:** Đầu ra các dự đoán bao gồm bounding boxes, confidence score, và class probabilities.

1. **Khả năng phát hiện đối tượng đa dạng:**

* Hỗ trợ phát hiện đối tượng trong các tình huống khác nhau, từ vật thể lớn đến vật thể nhỏ, từ đơn giản đến phức tạp.
* Độ chính xác cao trong việc phát hiện các đối tượng bị che khuất một phần hoặc có độ tương phản thấp.



Hình 2.4.4: Khả năng nhận diện của YoLov5

1. **Hiệu quả tính toán:**

Tối ưu tốc độ hiệu suất, cho phép sử dụng trên các thiết bị có tài nguyên hạn chế như **Raspberry Pi** hoặc các hệ thống nhúng.

**Ưu điểm của YOLOv5**

* **Dễ sử dụng:** Tài liệu chi tiết và khả năng triển khai nhanh chóng trên nhiều nền tảng.
* **Xử lý với tốc độ nhanh:** Thích hợp với các ứng dụng cần thời gian thực như phân tích cảm xúc, Face ID,....
* **Tích hợp các kỹ thuật mới:** Hỗ trợ augmentation tự động và learning rate scheduler để cải thiện quá trình huấn luyện.
* **Tính tương thích cao:** Hoạt động tốt với các framework phổ biến như PyTorch và TensorFlow.

**Nhược điểm của YOLOv5**

* **Phát hiện đối tượng nhỏ:** Mặc dù hiệu quả, YOLOv5 vẫn có nhược điểm khó nhận diện đối tượng ở xa hoặc rất nhỏ.
* **Không chính thức từ Darknet:** Không phải phiên bản chính thức từ nhà phát triển gốc của YOLO (Darknet), dẫn đến tranh cãi về tính "chính thống."
* **Hiệu năng yêu cầu tài nguyên:** Mặc dù tối ưu, việc triển khai trên các thiết bị có phần cứng hạn chế vẫn gặp khó khăn.

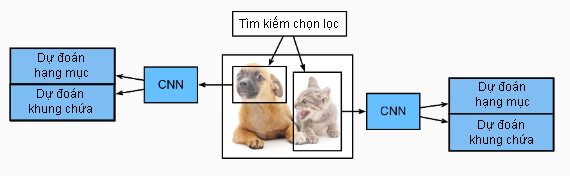
**Ứng dụng của YOLOv5**

* **An ninh và giám sát:** Nhận diện và theo dõi con người, phương tiện, hoặc vật thể nguy hiểm.
* **Xe tự lái:** Phát hiện biển báo, người đi bộ, phương tiện giao thông.
* **Y tế:** Nhận diện bất thường trong ảnh y khoa (CT, X-Ray).
* **Thương mại điện tử:** Nhận diện và phân loại sản phẩm.

## **2.5. Mô Hình R-CNN**

Mạng nơ-ron tích chập theo vùng với đặc trưng CNN (R-CNN) là một hướng tiếp cận tiên phong trong ứng dụng mô hình học sâu vào bài toán phát hiện nhận diện đối tượng với các mô hình được cải tiến sau đó: Fast R-CNN, Faster R-CNN, và Mask R-CNN.

Đầu tiên, mô hình R-CNN sẽ chọn một số vùng từ ảnh tiếp theo gán nhãn hạng mục và khung chứa. Sau đó, mô hình sử dụng CNN để truyền xuôi nhằm trích xuất đặc trưng từ từng vùng đề xuất. Cuối cùng, sử dụng các đặc trưng từng vùng dự đoán hạng mục và khung chứa .



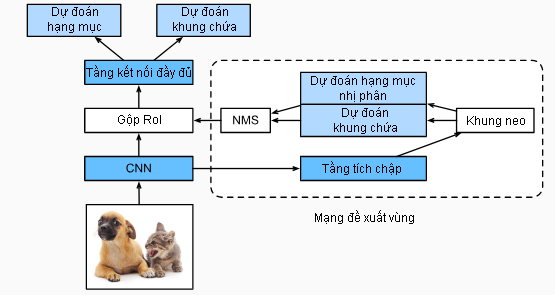
Hình 2.5: Mô Hình R-CNN

Cụ thể, R-CNN có bốn phần chính sau:

* Tìm kiếm và lọc từ ảnh đầu vào lựa chọn các vùng đề xuất tiềm năng. Thông thường các vùng đề xuất có nhiều hình dáng, tỉ lệ, kích thước khác nhau. Hạng mục, khung có nhãn gốc được gán cho từng vùng đề xuất.
* Ở trước đầu ra đặt một mạng CNN đã được huấn luyện và được rút gọn. Các vùng đề xuất được mạng này biến đổi lần lượt thành các đầu vào với chiều tương thích mạng, tiếp thực hiện các lượt truyền xuôi trích xuất đặc trưng từ các vùng tương ứng được đề xuất.
* Các máy vector hỗ trợ được mẫu kết hợp bởi nhãn hạng mục với các đặc trưng từng vùng đề xuất huấn luyện thành công cụ để phân loại đối tượng. Công cụ này được sử dụng nhằm xác định mẫu đó có thuộc một hạng mục nào không.
* Để dự đoán khung có nhãn gốc, mô hình hồi quy tuyến tính được huấn luyện từ một mẫu là sự kết hợp của các đặc trưng và khung chứa đã được gán nhãn mỗi vùng đề xuất.
* Tốc độ chậm chính là nhược điểm của các mô hình R-CNN, dù cho nó rất hiệu quả trong việc trích xuất đặc trưng nhờ các mạng CNN đã được huấn luyện. Chính vì vậy trong các ứng dụng thực tế các mô hình R-CNN không được áp dụng phổ biến. Với khối vùng đề xuất lớn từ ảnh thì tương ứng cần khối phép tính truyền xuôi từ CNN để phát hiện đối tượng. Với khối lượng tính toán khổng lồ như vậy khiến mô hình chậm xử lý so với các mô hình khác.

## **2.6. Mô hình Faster R-CNN**

Mô hình yêu cầu cần nhiều vùng đề xuất được tạo khi chọn lọc trong tìm kiếm nhằm đạt kết quả nhận diện đối tượng chính xác.Tìm kiếm chọn lọc được thay bằng mạng đề xuất vùng. Bảo đảm nhận diện đối tượng chính xác khi các vùng đề xuất được giảm do mnagj này.



Hình 2.6: Mô hình faster R-CNN

Thay vì tìm kiếm chọn lọc thì Faster R-CNN chuyển sang mạng đề xuất vùng để sản sinh các vùng đề xuất, nếu so vs fast R-CNN thì chỉ thay đổi phương pháp này. Những phần còn lại trong mô hình không đổi. Quá trình tính toán mạng đề xuất vùng được mô tả chi tiết:

1. Đầu ra của CNN được biến đổi nhờ một tầng tích chập 3×3, đệm bằng 1 và đặt số kênh đầu ra bằng cc. Đặc trưng mới có độ dài bằng cc là phần tử trong ánh xạ đặc trưng đã được CNN trích xuất từ bức ảnh nhờ cách này.
2. Lấy mỗi phần tử trong ánh xạ đặc trưng làm tâm để tạo ra nhiều khung neo có kích thước và tỷ lệ khác nhau, sau đó gán nhãn cho chúng.
3. Những đặc trưng của các phần tử có độ dài bằng cc ở tâm khung neo được dùng để phân loại nhị phân (là đối tượng hay là nền) và dự đoán khung chứa tương ứng.
4. Tiếp theo, sử dụng triệt phi cực đại để loại bỏ các khung chứa có kết quả giống nhau của hạng mục đối tượng. Các khung chứa dự đoán cuối cùng được xuất ra là các vùng đề xuất và được đưa vào tầng gộp RoI.

Lưu ý trong đối tượng Faster R-CNN còn chứa các hàm dự đoán hạng mục và khung chứa trong bài toán phát hiện đối tượng, các hàm dự đoán hạng mục nhị phân và khung chứa cho các khung neo trong mạng đề xuất vùng. Mạng đề xuất vùng có thể học được cách sinh ra những vùng đề xuất có chất lượng cao, giảm đi số lượng vùng đề xuất trong khi vẫn giữ được độ chính xác khi phát hiện vật thể. Ngoài ra,vì là một phần của mô hình Faster R-CNN, nên mạng đề xuất vùng được huấn luyện cùng với phần còn lại trong mô hình.

## **2.7. Nhận Xét và Đánh giá**

YoLov5:

* Ưu điểm:
* Tốc độ nhanh: YOLOv5 được tối ưu hóa để hoạt động với tốc độ rất nhanh, đặc biệt trong các ứng dụng yêu cầu nhận diện thời gian thực (real-time), như giám sát an ninh, xe tự lái, hoặc robot.
* Dễ triển khai: YOLOv5 được xây dựng trên nền tảng PyTorch, một framework học sâu phổ biến, giúp người dùng dễ dàng triển khai và điều chỉnh mô hình. Ngoài ra, YOLOv5 hỗ trợ nhiều công cụ triển khai như TensorRT, ONNX, và CoreML.
* Mô hình nhẹ và linh hoạt: YOLOv5 có các phiên bản mô hình nhỏ (YOLOv5s, YOLOv5m) và lớn (YOLOv5l, YOLOv5x), phù hợp với nhiều yêu cầu về hiệu suất và tài nguyên.
* Độ chính xác cao: Mặc dù ưu tiên về tốc độ, YOLOv5 vẫn duy trì được độ chính xác rất tốt và có thể áp dụng vào các bài toán nhận diện đối tượng phức tạp.
* Tối ưu cho môi trường có tài nguyên hạn chế: Với khả năng tối ưu hóa và giảm tải tài nguyên, YOLOv5 có thể chạy trên các thiết bị di động hoặc các hệ thống có tài nguyên tính toán thấp.
* Nhược điểm:
* Khả năng phát hiện đối tượng nhỏ: Dù YOLOv5 rất mạnh mẽ trong việc phát hiện đối tượng lớn, nhưng hiệu suất trong việc nhận diện các đối tượng nhỏ hoặc phức tạp (ví dụ: các chi tiết mờ trong ảnh) có thể thấp hơn so với một số mô hình khác như Faster R-CNN.
* Thiếu tính chính xác cao trong các trường hợp phức tạp: Trong một số tình huống đặc biệt, YOLOv5 có thể không đạt được độ chính xác cao như Faster R-CNN, đặc biệt khi dữ liệu đầu vào có tính chất phức tạp hoặc có nhiều đối tượng chồng lấn.
* Xử lý các đối tượng không đều: YOLOv5 có thể gặp khó khăn khi phát hiện đối tượng có kích thước khác biệt trong cùng một ảnh.

Faster R-CNN

* Ưu Điểm:
* Độ chính xác cao: Với khối lượng tính toán khổng lồ của Faster R-CNN thì phát hiện đối tượng chính xác là ưu thế so với YoLo, nhất là đối với các đối tượng có kích thước nhỏ hoặc phức tạp. Bước Region Proposal Network (RPN) giúp phát hiện và tinh chỉnh các bounding box chính xác hơn.
* Khả năng phát hiện đối tượng nhỏ tốt hơn: Do kiến trúc two-stage, Faster R-CNN có thể xử lý tốt hơn các đối tượng nhỏ trong ảnh, điều mà YOLOv5 gặp khó khăn hơn.
* Khả năng phân loại mạnh mẽ: Mô hình có khả năng phân loại đối tượng một cách chi tiết và chính xác hơn trong các tình huống phức tạp.
* Linh hoạt trong các bài toán phức tạp: Faster R-CNN hoạt động tốt trong các môi trường có nhiều đối tượng chồng lấn hoặc các đối tượng có kích thước khác biệt lớn.
* Nhược điểm:
* Tốc độ chậm: So với YOLOv5, Faster R-CNN có tốc độ suy luận chậm hơn nhiều, đặc biệt là trong các ứng dụng thời gian thực. Do được thiết kế theo mô hình two-stage, nó cần thực hiện nhiều bước để hoàn thành nhận diện đối tượng, dẫn đến thời gian xử lý lâu hơn.
* Đòi hỏi tài nguyên tính toán lớn: Faster R-CNN yêu cầu nhiều tài nguyên tính toán và bộ nhớ hơn, điều này khiến nó khó triển khai trên các thiết bị với tài nguyên hạn chế (như điện thoại di động hoặc các hệ thống có GPU yếu).
* Khó triển khai và tinh chỉnh: Đối với Faster R-CNN có thể phức tạp hơn, nhất là đối với người mới. Các bước huấn luyện và tinh chỉnh có thể đòi hỏi thời gian và công sức lớn hơn so với YOLOv5.

YOLOv5 là lựa chọn tuyệt vời nếu bạn cần tốc độ nhanh và dễ triển khai, đặc biệt cho các ứng dụng thời gian thực.

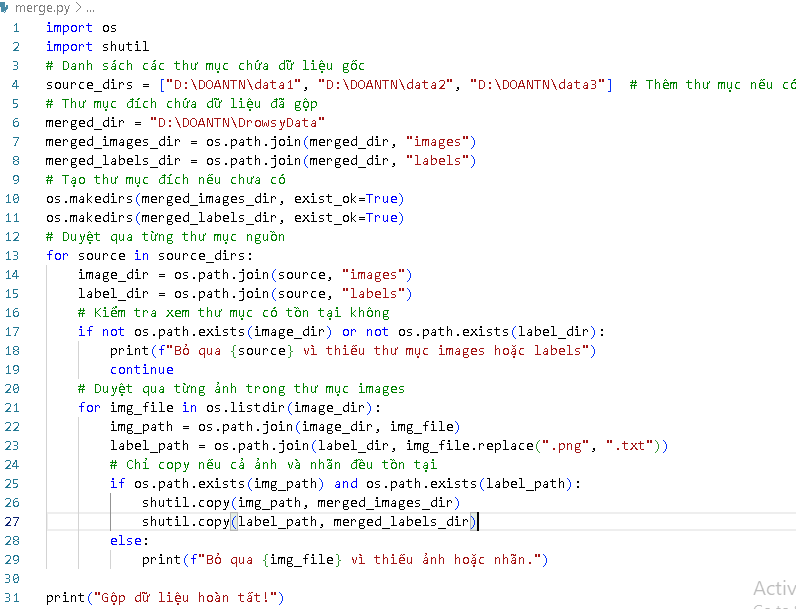
Faster R-CNN có độ chính xác cao hơn và hoạt động tốt hơn trong việc phát hiện đối tượng nhỏ và đối tượng phức tạp, nhưng lại có nhược điểm lớn về tốc độ và yêu cầu tài nguyên tính toán cao hơn, vì vậy nó phù hợp hơn với các ứng dụng không yêu cầu real-time.

**CHƯƠNG 3: XÂY DƯNG CHƯƠNG TRÌNH NHẬN DIỆN PHÁT HIỆN TÀI XẾ BUỒN NGỦ**

**3.1. Mô tả dữ liệu**

Bộ dữ liệu bao gồm hơn 6700 hình ảnh tổng hợp các trạng thái của mắt, biểu cảm khuôn mặt được thu thập từ nhiều nguồn bao gồm roboflow..., đã được xử lý dán nhãn để huấn luyện và đánh giá mô hình.

Sau khi thu thập dữ liệu từ nhiều nguồn tiến hành gộp dữ liệu bằng code python:



Hình 3.1a: gộp dữ liệu

Sau khi quá trình gộp hoàn tất tiến hành chia tập dữ liệu thành tập test và train theo tỉ lệ 80:20



Hình 3.1b chia dữ liệu tập train và test

Sau khi chia thì bộ dữ liệu bao gồm:

* Tập train: 5334 hình ảnh và tập txt (bao gồm nhãn và bouding box) tương ứng
* Tập test:
* 1368 hình ảnh và tập txt (bao gồm nhãn và bouding box) tương ứng
* 20 video cắt ra từ các camera hành trình.

Bộ dữ liệu cung cấp chú thích chi tiết và hộp giới hạn ( bounding boxes) bao quanh đối tượng. Chú thích gồm:

* Vị trí (Bounding Box): một hình chữ nhật được vẽ xung quanh đối tượng (mắt) trong hình ảnh, xác định bởi toạ độ của góc trên bên trái (x1, y1) và toạ độ của góc dưới bên phải (x2, y2)
* Lớp: Nhãn của đối tượng (mắt) trong hộp giới hạn, drowsy hoặc awake

1. **Cấu trúc tập dữ liệu:**

Hình ảnh: Tập dữ liệu bao gồm hơn 6700 hình ảnh thu thập từ nhiều nguồn bao gồm roboflow,.. đã được xử lý dán nhãn, bounding box.

Video: Được thu thập cắt trên các camera hành trình và chuyển định dạng mp4.

Chú thích: Được cung cấp dưới dạng yolov5, bao gồm thông tin về hộp giới hạn đối tượng, nhãn.

Lớp: Tập dữ liệu bao gồm 2 lớp drowsy và awake

1. **Sử dụng trong phát hiện buồn ngủ**

Trong bối cảnh phát hiện buồn ngủ, điểm chủ chốt là mắt. Việc xác định chính xác điểm này cho phép mô hình ước lượng trạng thái của một người trong một hình ảnh hoặc video.

**3.1.2.1. Quy trình phát hiện điểm chủ chốt:**

* **Bước 1: Ảnh hoặc khung hình từ video được đưa vào mô hình YOLOv5.**
* **Bước 2: Mô hình dự đoán hộp giới hạn (bounding boxes) và xác định các điểm chủ chốt (mắt).**

**3.1.2.2. Dữ Liệu Huấn Luyện**

**Mô hình YOLOv5 được huấn luyện trên dữ liệu huấn luyện do các chú thích toàn diện và đa dạng của nó.**

**3.1.2.2. Dữ Liệu Huấn Luyện**

**Mô hình YOLOv5 được huấn luyện trên dữ liệu huấn luyện do các chú thích toàn diện và đa dạng của nó.**

**3.1.2.3. Quy trình Huấn Luyện**

* **Dữ liệu được sử dụng để huấn luyện mô hình YOLOv5. Các hình ảnh và chú thích về điểm chủ chốt (mắt) được đưa vào mô hình.**
* **Mô hình học cách phát hiện các hộp giới hạn và điểm chủ chốt (mắt)**
* **Mô hình được tối ưu hóa để cải thiện độ chính xác và tốc độ phát hiện.**
* **Mô hình được kiểm tra và đánh giá trên một tập dữ liệu thử nghiệm để đảm bảo hiệu suất tốt.**

**3.1.2.4. Quy trình phát hiện buồn ngủ**

* Video hoặc hình ảnh đầu vào được đưa vào mô hình YOLOv5
* Mô hình phát hiện đối tượng và điểm chủ chốt (mắt) trên cơ thể người.
* Điểm chủ chốt (mắt) được phân tích để xác định trạng thái tài xế
* Các điều kiện phát hiện buồn ngủ được kiểm tra.

1. **Phân chia bộ dữ liệu**

* **Bộ huấn luyện (Train)**: Chứa 5334 hình ảnh.
* **Bộ kiểm thử (Test)**: Chứa 1368 hình ảnh.

**3.2. Những thư viện được dử dụng trong bài toán**

1. **Thư viện Torch (PyTorch)**

PyTorch là một thư viện mạnh mẽ và linh hoạt cho học sâu, giúp các nhà nghiên cứu và lập trình viên dễ sử dụng để xây dựng, huấn luyện và phát triển các mô hình học máy phức tạp.

Một số đặc điểm nổi bật của pytorch:

1. Tensor
2. Autograd (Tự động đạo hàm)
3. Torch.nn (Mạng nơ-ron)
4. Tối ưu hóa
5. GPU và CUDA
6. Ứng dụng
7. Thư viện liên kết
8. Khả năng tương thích với các mô hình tiên tiến
9. **Thư viện OpenCV-Python**

OpenCV là một thư viện rất mạnh và linh hoạt, cho phép xử lý và phân tích hình ảnh video trong các ứng dụng Python. Với nhiều tính năng và công cụ hỗ trợ, OpenCV được sử dụng trong rất nhiều lĩnh vực như nhận diện khuôn mặt, tự động hóa, điều khiển robot, và các bài toán thị giác máy tính khác.

Các đặc điểm chính của **opencv-python**:

* Xử lý và phân tích ảnh
* Xử lý video
* Xử lý hình ảnh nâng cao
* Phát hiện đối tượng và nhận diện
* Phát hiện đối tượng và nhận diện
* Các thao tác hình học

1. **Thư viện numpy**

**NumPy** là một thư viện cực kỳ mạnh mẽ và cần thiết cho các tác vụ tính toán khoa học, đặc biệt khi làm việc với dữ liệu số, mảng và ma trận. Nó giúp tối ưu hóa các phép toán số học và đại số tuyến tính, đồng thời cung cấp một cú pháp dễ dàng và nhanh chóng để xử lý và phân tích dữ liệu trong Python. NumPy cũng có vai trò quan trọng hỗ trợ các thư viện khác như **SciPy, scikit-learn,** pandasvà **TensorFlow**.

Các đặc điểm chính của **NumPy**:

* **Mảng NumPy (ndarray)**
* Các phép toán mảng (Array operations)
* Các phép toán đại số tuyến tính
* Lập chỉ mục và cắt mảng (Indexing and slicing)
* **Tạo mảng đặc biệt**
* **Các phép toán thống kê**
* **Quản lý dữ liệu thiếu (NaN)**
* **Các phép toán điều kiện (Conditionals)**
* Các hàm hoạt động trên ma trận
* Hỗ trợ học máy (Machine Learning)
* Hỗ trợ CUDA (GPU)

**3.3. Xây dựng mô hình dự đoán (YoLov5)**

1. **Thu thập và phân tích dữ liệu**

Bộ dữ liệu thu thập từ nhiều nguồn khác nhau, dùng code python để chia dữ liệu thành tệp Huấn Luyện và Kiểm thử theo tỷ lệ mong muốn

1. Tiền Xử lý dữ liệu

Trước khi đưa vào huấn luyện, ảnh và video từ tập dữ liệu phải được tiền xử lý đảm bảo định dạng và chất lượng tương đồng. Ví dụ thay size thước ảnh, chuẩn hóa giá trị pixel, và chuyển đổi định dạng ảnh nếu cần thiết.

1. Huấn Luyện mô hình

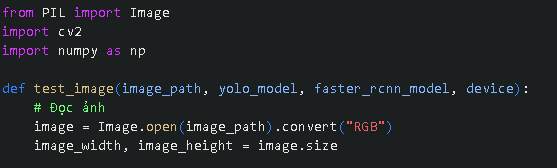
* Kiến trúc mô hình: YOLOv5 sử dụng kiến trúc mạng YOLO (You Only Look Once), một mạng nơ-ron tích chập (CNN) được thiết kế cho việc nhận diện vật thể real-time.
* Dữ liệu đầu vào: Ảnh đã được gán nhãn hoặc video đã xử lý frame được đưa vào mô hình bắt đầu quá trình huấn luyện.
* Tiến trình huấn luyện: Mô hình YOLOv5 được huấn luyện với các dữ liệu này để học cách dự đoán.
* Hàm mất mát (Loss function): Đo lường chỉ số sai lệch giữa nhãn thực tế và dự đoán của mô hình, hàm mất mát như Mean Squared Error (MSE) hoặc các biến thể như YOLO loss được dùng để huấn luyện mô hình.

1. Kiểm thử mô hình

* Đánh giá hiệu suất: Mô hình YOLOv5 được đánh giá trên các tập dữ liệu kiểm thử nhằm đo lường hiệu suất của mô hình . Các phép đo hiệu suất có thể bao gồm độ chính xác P, độ phủ (recall), độ đo F1 (F1 score), và thời gian thực thi của mô hình.
* Cải tiến mô hình**:** Dựa trên kết quả đánh giá, mô hình có thể được cải tiến bằng cách điều chỉnh tham số huấn luyện, tăng cường dữ liệu huấn luyện, sử dụng các kỹ thuật tối ưu hóa mô hình như regularization, dropout, hay thay đổi kiến trúc mạng.

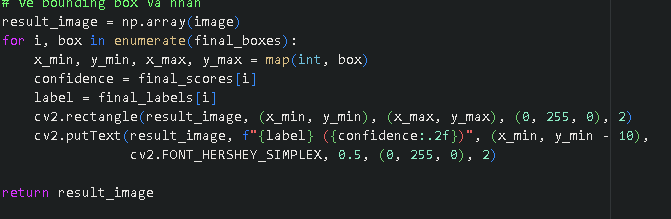
1. Tải mô hình giải quyết nhận diện phát hiện buồn ngủ

* Lấy dữ liệu bài toán vào:



Hình 3.3.5a: Lấy dữ liệu vào

* Hàm khoanh vùng trạng thái (mắt)



Hình 3.3.5b: Vẽ bounding box

* Kết quả:



1. Đánh Gía mô hình bài toán

3.3.6.1. Độ chính xác

Dựa vào dữ liệu đánh giá sau khi huấn luyện mô hình thì kết quả ban đầu cho ra các chỉ số như sau: Precision: 78.93%, Recall: 86.2%, F1 score: 82.4%, mAP: 89.63%

mAP (Độ chính xác trung bình): Đạt 89.63%, đây là đo độ chính xác tổng hợp của mô hình qua tất cả các lớp và ở nhiều mức độ recall khác nhau.. Mức độ này cho thấy mô hình có khả năng phat hiện đối tượng một cách chính xác trong hầu hết các trường hợp.

Precision (Độ chính xác): Đạt 78.93 %, đo lường tỉ lệ các dự đoán đúng (True Positives) trên tổng số các dự đoán. Với số đo kết quả cho thấy tỷ lệ các dự đoán đúng là khá cao.

**Recall (Độ nhạy):** Đạt 86.2%, đo lường tỉ lệ các dự đoán đúng (True Positives) trên tổng số các đối tượng thực sự đúng. Con số này cho thấy mô hình nhạy bén khá tốt trong việc phát hiện và bao quát đối tượng.

**F1 score:** Đạt 82.4%, là một chỉ số tổng hợp dùng để đánh giá sự cân bằng giữa Precision và Recall, dùng để đánh giá tổng quát hiệu quả của mô hình. Như ta thấy F1 score cao tương đương với mô hình đã đạt được sự cân bằng tốt giữa độ chính xác và độ nhạy.

3.3.6.2. Tốc độ xử lý

Mô hình YOLOv5 được tối ưu hóa để sử dụng GPU, vì vậy có khả năng xử lý các video với tốc độ nhanh hơn đáng kể so với các phương pháp xử lý truyền thống. Điều này đảm bảo rằng không có độ trễ đáng kể khi phát hiện tài xế có dấu hiệu buồn ngủ, đặc biệt là trong các bối cảnh yêu cầu phản ứng và xử lý dữ liệu thời gian thực

3.3.6.3 Khả năng ứng dụng

YOLOv5 rất linh hoạt và đã được áp dụng cho nhiều lĩnh vực như: y khoa, an ninh, các khối ngành kinh tế, …. Sự kết hợp giữa tốc độ và độ chính xác của YOLOv5 giúp nó trở thành công cụ lý tưởng cho các ứng dụng cần phát hiện đối tượng trong thời gian thực.

Ứng dụng trong Robot:

Ứng dụng chính: YOLOv5 có thể giúp dạy robot nhận diện đối tượng và điều hướng trong môi trường khác nhau.

Giám sát đám đông (Crowd Monitoring)

Ứng dụng chính: YOLOv5 có thể phát hiện và phân tích đám đông trong các sự kiện đông người.

**3.4. Xây dựng mô hình dự đoán (Faster R-CNN)**

1. Tiền Xử lý dữ liệu

Trước khi đưa vào huấn luyện, ảnh và video từ tập dữ liệu phải được tiền xử lý đảm bảo định dạng và chất lượng tương đồng. Thay đổi size ảnh, chuẩn hóa giá trị pixel, và chuyển đổi định dạng ảnh (định dạng dữ liệu COCO).

1. **Huấn luyện mô hình**

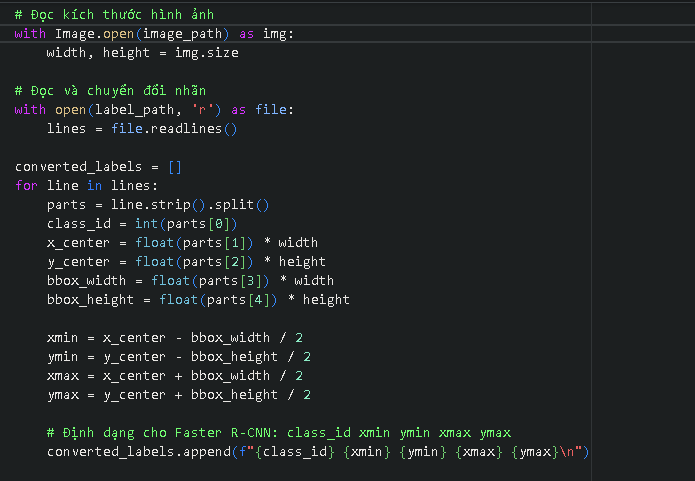
* Kiến trúc mô hình: **Faster R-CNN** là một mô hình phát hiện đối tượng dựa trên kiến trúc mạng nơ-ron tích chập (CNN), gồm 3 thành phần chính: Backbone, RPN, RoI Pooling.
* Dữ liệu đầu vào: là các ảnh có nhãn với bounding boxes (tọa độ hộp giới hạn các đối tượng) và các lớp tương ứng. Dữ liệu này được gán nhãn thủ công hoặc tự động, và các nhãn này sẽ được sử dụng trong quá trình huấn luyện để tính toán độ chính xác và độ mất mát.
* Tiến trình huấn luyện: Dữ liệu được đưa vào xử lý trích xuất đặc trưng, tạo ra các đề xuất vùng, cuối cùng phân thành các lớp và dự đoán bounding box chính xác hơn. Sau đó tính toán hàm mất mát, để giảm thiếu mất mát. Lặp lại quá trình nhiều lần ến khi mô hình hội tụ và có khả năng dự đoán chính xác các bounding boxes và nhãn lớp.
* Hàm mất mát (Loss function):

Classification loss: Đo lường độ sai lệch trong việc phân loại đối tượng vào các lớp khác nhau. Thường sử dụng cross-entropy loss cho bài toán phân loại.

Regression loss: Đo lường sự khác biệt giữa bounding boxes dự đoán và các bounding boxes thực tế. Phần này thường sử dụng Smooth L1 Loss để tính toán độ lệch giữa các tọa độ của bounding box.

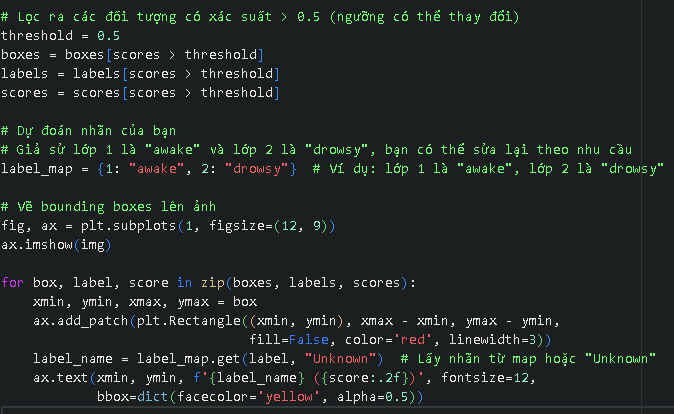
1. Tải mô hình giải quyết nhận diện phát hiện buồn ngủ

* Tiền xử lý dữ liệu



Hình 3.4.3a: Xử lý dữ liệu chuyển đổi định dạng nhãn

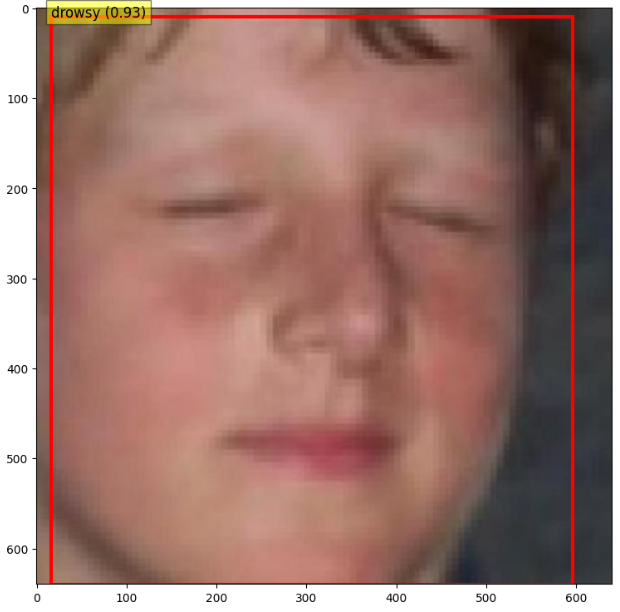
* Tạo bộ dữ liệu mới gồm hình ảnh và nhãn đã đổi định dạng (json)
* Huấn luyện mô hình
* Kiểm thử mô hình: dự đoán và vẽ bounding box



Hình 3.4.3b: Dự đoán và vẽ bounding box

* Kết quả:

|  |  |  |
| --- | --- | --- |
|  |  |  |



1. Đánh Gía mô hình bài toán

3.4.4.1. Độ chính xác

Dựa vào dữ liệu đánh giá sau khi huấn luyện mô hình thì kết quả ban đầu cho ra các chỉ số như sau: Precision: 85%, Recall: 95%, F1 score: 92,43%, mAP: 90%

mAP (Độ chính xác trung bình): Đạt 90%, đây là đo độ chính xác tổng hợp của mô hình qua tất cả các lớp và ở nhiều mức độ recall khác nhau.. Mức độ này cho thấy mô hình có khả năng phat hiện đối tượng một cách chính xác trong hầu hết các trường hợp.

Precision (Độ chính xác): Đạt 85 %, đo lường tỉ lệ các dự đoán đúng (True Positives) trên tổng số các dự đoán. Với số đo kết quả cho thấy tỷ lệ các dự đoán đúng khá cao.

**Recall (Độ nhạy):** Đạt 95%, đo lường tỉ lệ các dự đoán đúng (True Positives) trên tổng số các đối tượng thực sự đúng. Con số này cho thấy mô hình nhạy bén tốt trong việc phát hiện và bao quát đối tượng.

**F1 score:** Đạt 92.43%, là một chỉ số tổng hợp dùng để đánh giá sự cân bằng giữa Precision và Recall, dùng để đánh giá tổng quát hiệu quả của mô hình. Như ta thấy F1 score cao tương đương với mô hình đã đạt được sự cân bằng tốt giữa độ chính xác và độ nhạy.

3.4.4.2. Tốc độ xử lý

Vì tính năng yêu cầu đạt độ chính xác cao nên Faster R-CNN chiếm thời gian xử lý nhiều hơn các mô hình khác như YoLo… tuy ưu điểm là độ chính xác cao nhưng với bối cảnh yêu cầu phản ứng anh và thời gian thực thì đây là một điểm trừ lớn

3.4.4.3. Khả năng ứng dụng

Faster R-CNN là mô hình phát hiện đối tượng mạnh mẽ phổ biến trong lĩnh vực Computer Vision, nhất là khi cần độ chính xác cao. aster R-CNN vẫn có rất nhiều ứng dụng quan trọng nhờ vào khả năng phát hiện đối tượng chính xác trong các điều kiện khác nhau. Dưới đây là một số khả năng ứng dụng của Faster R-CNN:

* Nhận diện đối tượng trong video (Video Object Detection)
* Ứng dụng trong xe tự lái (Autonomous Vehicles)
* Sử dụng trong hệ thống giám sát an ninh
* Phát hiện các đối tượng trong y tế (Medical Imaging)

KẾT LUẬN

Kết luận và hướng phát triển

Sau quá trình nghiên cứu và dưới sự hướng dẫn tận tình của thầy Trần Mạnh Tuấn, em đã nắm vững kiến thức cơ bản và nâng cao về các phương pháp học sâu, đặc biệt là trong việc áp dụng các mô hình học sâu vào nhận diện và phát hiện lái xe buồn ngủ. Qua quá trình thực hiện, em đã hiểu rõ cách sử dụng các mô hình học sâu, trong đó có YOLOv5, Faster R-CNN, để phân tích và nhận diện các đặc điểm liên quan đến trạng thái buồn ngủ của tài xế từ hình ảnh, đồng thời triển khai được hệ thống thử nghiệm thực tế.

Mô hình YOLOv5 được lựa chọn trong nghiên cứu này đã chứng minh khả năng hiệu quả trong việc nhận diện và phát hiện các dấu hiệu của lái xe buồn ngủ, đáp ứng yêu cầu của bài toán thực tế và có thể ứng dụng vào các hệ thống giám sát giao thông, giúp nâng cao sự an toàn cho người tham gia giao thông. Việc phát hiện sớm tình trạng buồn ngủ của tài xế có thể giúp giảm thiểu các tai nạn giao thông, góp phần vào việc bảo vệ sức khỏe và sự an toàn của cộng đồng. Tuy nhiên, để ứng dụng vào thực tế, độ chính xác của mô hình cần tiếp tục được cải thiện và thích ứng với các điều kiện môi trường và dữ liệu mới.

Mô hình Faster R-CNN đạt như kỳ vọng ở mức độ phát hiện nhận diện chính xác tình trạng trạng thái của tài xế, tuy nhiên với nghiên cứu này đòi hỏi tốc độ, thời gian thực để đảm bảo an toàn cho tài xế cũng như ngăn chặn tai nạn giao thông thì mô hình Faster R-CNN không làm được.

Nếu như kết hợp được ưu điểm lợi thế của hai mô hình thì kết quả đem lại sẽ rất khả thi. Trong đồ án này em chưa hoàn thành được và sẽ cố gắng tiếp tục cải thiện trong tương lai.

Kinh nghiệm thu được

Nắm bắt được kiến thức về học sâu và cách áp dụng các mô hình học sâu vào các bài toán nhận diện hình ảnh.

Hiểu và có thêm kinh nghiệm trong việc sử dụng YOLOv5 cho các bài toán nhận diện đối tượng, đặc biệt là trong việc phát hiện trạng thái buồn ngủ của tài xế.

Học cách xây dựng và triển khai một hệ thống nhận diện thực tế từ lý thuyết học máy và học sâu.

Củng cố kiến thức về thị giác máy tính, xử lý ảnh và nhận diện đối tượng.

Rèn luyện kỹ năng giao tiếp, thuyết trình và báo cáo kết quả nghiên cứu.

Có cơ hội áp dụng kiến thức lý thuyết vào các bài toán thực tế, đặc biệt trong lĩnh vực giao thông và an toàn.

Hướng phát triển

Tiếp tục nghiên cứu và cải tiến thuật toán YOLOv5 và Faster R-CNN hiện tại để nâng cao độ chính xác và khả năng nhận diện trong các điều kiện thực tế đa dạng hơn.

Nghiên cứu và tích hợp thêm các thuật toán mới, như kết hợp với các mô hình học sâu khác để cải thiện hiệu suất của hệ thống phát hiện lái xe buồn ngủ.

Mở rộng ứng dụng của mô hình, không chỉ trong việc phát hiện lái xe buồn ngủ mà còn trong việc nhận diện các trạng thái nguy hiểm khác của tài xế như mệt mỏi, mất tập trung.

Tăng cường việc thu thập và xử lý dữ liệu thực tế từ các video giám sát giao thông để cải thiện mô hình và đảm bảo tính khả thi của hệ thống trong môi trường thực.

Cải tiến và ứng dụng hệ thống này sẽ có ý nghĩa to lớn trong việc hỗ trợ người lái xe, giúp nâng cao an toàn giao thông và giảm thiểu các vụ tai nạn do nguyên nhân buồn ngủ hoặc mệt mỏi của tài xế.

# **TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | V. Hoàng, "Tìm hiểu về YOLO trong bài toán real-time object detection," *Viblo,* p. bài viết, 2019. |
| [2] | M. Menegaz, "Understanding YOLO," *hackernoon,* 2018. |
| [3] | A. Rosebrock, "Intersection over Union (IoU) for object detection," *pyimagesearch,* 2016. |
| [4] | Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arxiv,* 2014 . |
| [5] | Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *arxiv,* p. Computer Vision and Pattern Recognition, 2016 . |
| [6] | N. V. Hoai, "Faster R-CNN cho bài toán phát hiện đối tượng," *Viblo,* 2020 . |
| [7] | Tuấn, " Object detection với Faster R-CNN," *nttuan8.com,* p. Deep Learning, 2019. |
| [8] | P. Potrimba, "What is R-CNN?," *roboflow,* p. Blog, 2023 . |
| [9] | Lev Craig, David Petersson, "CNN vs. RNN: How are they different?," *techtarget,* p. AI technologies, 2024. |
| [10] | Jim Holdsworth, Mark Scapicchio, "What is deep learning?," *ibm,* p. Think, 2024. |
| [11] | K. giả, "Giấc ngủ liên quan đến tai nạn giao thông như thế nào?," *vatgia,* pp. Ôtô & Xe máy - Khác, 2015. |
| [12] | Y. Chi, "Liên tiếp xảy ra tai nạn do lái xe mệt mỏi, ngủ gật, ngăn cách nào?," *atgt,* p. Giao thông 24h, 2024. |