

MỤC LỤC

I. Câu hỏi hay vấn đề đặt ra	3
1. Vấn đề tình trạng giao thông.....	3
2. Vấn đề về công nghệ	4
3. Nghiên cứu bắt đầu dự án “ADVANCED-DRIVER-ASSISTANCE- SYSTEM”	6
II. Mục tiêu kết quả mong đợi	6
1. Hệ thống ADVANCED-DRIVER-ASSISTANCE-SYSTEM mô phỏng trên máy tính	6
2. Hệ thống ADVANCED-DRIVER-ASSISTANCE-SYSTEM triển khai vào thực tiễn	7
3. Tổng thể về hệ thống ADVANCED-DRIVER-ASSISTANCE-SYSTEM- ADAS	7
III. Giả thuyết khoa học và những điều ADVANCED-DRIVER- ASSISTANCESYSTEM-ADAS đã làm được	9
1. Giả thuyết khoa học	9
2. Những điều ADVANCED-DRIVER-ASSISTANCE-SYSTEM đã làm được 9	
a) Tổng thể ADVANCED-DRIVER-ASSISTANCE-SYSTEM	9
b) Hệ thống mô phỏng trên máy tính	12
c) Hệ thống thiết bị áp dụng thực tiễn	14
IV. Phạm vi nghiên cứu	16
1. Tổng thể hệ thống mô phỏng trên máy tính	16
2. Tổng thể hệ thống thiết bị áp dụng vào thực tiễn	17
V. Kế hoạch nghiên cứu và chuẩn bị thực hiện	17
1. Kế hoạch	17
a) Nền tảng mô phỏng trên máy tính	17
b) Nền tảng hệ thống thiết bị thực tiễn đã tiến hành trên mô phỏng trên máy tính	22

2. Chuẩn bị	23
3. Thực hiện	23
VI. Thành tựu số liệu nghiên cứu hệ thống ADAS	23
VII. Kết luận	24
VIII. Nguồn tham khảo	25

TÓM TẮT DỰ ÁN

“HỆ THỐNG HỖ TRỢ LÁI XE AN TOÀN-ADAS”

I. Câu hỏi hay vấn đề đặt ra

1. Vấn đề tình trạng giao thông.

Trong năm 2022, tai nạn giao thông đường bộ xảy ra 11.450 vụ tai nạn giao thông (TNGT), làm chết 6.384 người, bị thương 7.804 người. So với cùng kỳ năm 2021, giảm 44 vụ (0,38%), tăng 585 người chết (9,16%), giảm 214 người bị thương (2,74%) (Theo nguồn Báo điện tử Đảng cộng sản Việt Nam)

Diễn hình như toàn cảnh một số vụ TNGT, trong đó có vụ TNGT làm 04 thành viên CLB Hoàng Anh Gia Lai thương vong. (Theo nguồn Báo điện tử Đảng cộng sản Việt Nam)



Một số vấn đề tình trạng giao thông của nước ta hiện nay:

✦ **Ùn tắc giao thông:** Các thành phố lớn và khu vực đô thị của Việt Nam thường đối mặt với tình trạng ùn tắc giao thông nghiêm trọng. Số lượng xe cộ gia tăng nhanh chóng, trong khi hệ thống đường không phát triển tương xứng. Điều này dễ dẫn đến tắc nghẽn, sự chen chúc và tốn thời gian di chuyển.

✦ **An toàn giao thông:** Việt Nam ghi nhận một số lượng lớn tai nạn giao thông hàng năm, với số lượng người thương vong cao. Nguyên nhân từ việc không tuân thủ quy tắc giao thông: Chạy quá tốc độ; Đi sai phần đường, làn đường; Trách vượt sai quy định; Chuyển hướng không nhường đường; Không chú ý quan sát; Sử dụng rượu, bia và các chất kích thích thần kinh khác khi lái xe; Không đội mũ bảo hiểm và không thắt dây an toàn; Sử dụng xe cũ kỹ hết hạn đăng kiểm,

✦ **Quản lý giao thông:** Công tác quản lý giao thông ở Việt Nam đang gặp nhiều khó khăn. Sự phát triển nhanh chóng của xe cộ và cơ sở hạ tầng giao thông tạo

ra áp lực lớn cho cơ quan chức năng. Đồng thời, việc thực thi quy định giao thông và xử lý vi phạm chưa đạt hiệu quả cao, dẫn đến sự thiếu trật tự và an toàn trên đường.

✦ **Ý thức tham gia giao thông:** Ý thức và nhận thức về quy tắc giao thông của một số người tham gia giao thông vẫn còn hạn chế. Việc vi phạm quy tắc giao thông, đi lạng lách, lấn làn, không tôn trọng người đi bộ và người đi xe đạp, là những vấn đề phổ biến.

✦ **Phát triển giao thông công cộng:** Việc phát triển hệ thống giao thông công cộng như xe buýt, đường sắt đô thị và hệ thống giao thông thông minh vẫn còn hạn chế. Sự lựa chọn và ưu tiên di chuyển cá nhân, đặc biệt là ô tô cá nhân, vẫn là xu hướng chính.

2. Vấn đề về công nghệ

Công nghệ ADAS được coi là bước đột phá của ngành công nghiệp ô tô. ADAS sử dụng mạng máy tính và các dữ liệu thông minh để giúp người lái có hành trình di chuyển thuận tiện và an toàn.

Vậy hệ thống ADAS là gì?

Công nghệ ADAS là viết tắt của **Advanced Driver Assistance Systems**, có nghĩa là hệ thống hỗ trợ lái xe nâng cao. Là các hệ thống an toàn chủ động và thụ động được thiết kế để loại bỏ thành phần lỗi của con người khi vận hành nhiều loại phương tiện.

Các hệ thống công nghệ ADAS sử dụng các công nghệ tiên tiến để hỗ trợ người lái trong quá trình lái xe và từ đó cải thiện hiệu suất của người lái. Công nghệ ADAS sử dụng kết hợp các công nghệ cảm biến để nhận biết thế giới xung quanh xe, sau đó cung cấp thông tin cho người lái hoặc thực hiện hành động khi cần thiết. Đây là một hệ thống điện tử hỗ trợ người điều khiển phương tiện lái xe an toàn và thuận tiện hơn.

Hệ thống hỗ trợ lái nâng cao công nghệ ADAS đã chứng minh giúp giảm tỷ lệ tai nạn trong khi tham gia giao thông bằng cách hạn chế khả năng mắc lỗi của người điều khiển phương tiện thông qua các chức năng cảnh báo thông minh, như: cảnh

báo va chạm phía trước và sau, cảnh báo lệch làn đường, phát hiện điểm mù, phát hiện tài xế buồn ngủ, kiểm soát đồ đèo, hệ thống nhìn ban đêm, nhận diện biển báo tốc độ,...

Một số vấn đề công nghệ liên quan đến hệ thống Advanced Driver Assistance System (ADAS) mà cần được xem xét. Dưới đây là một số vấn đề chính:

✦ **Độ tin cậy và độ chính xác:** ADAS dựa vào các cảm biến, camera và hệ thống xử lý để thu thập và phân tích dữ liệu. Để hệ thống hoạt động hiệu quả, các cảm biến và hệ thống xử lý phải đảm bảo độ tin cậy và độ chính xác cao. Sự chính xác của ADAS có thể bị ảnh hưởng bởi môi trường lái xe, điều kiện thời tiết, nhiễu và sự cố kỹ thuật.

✦ **Tương tác và tích hợp:** ADAS thường bao gồm nhiều tính năng và mô-đun khác nhau, chẳng hạn như hỗ trợ giữ làn đường, hệ thống cảnh báo điểm mù và phanh khẩn cấp. Việc tích hợp các tính năng này và tạo ra một tương tác nhất quán giữa chúng có thể là một thách thức. Cần có sự tương thích và tích hợp tốt giữa các thành phần phần cứng và phần mềm của ADAS để đảm bảo hoạt động ổn định và hiệu quả.

✦ **Học máy và trí tuệ nhân tạo:** ADAS ngày càng sử dụng các công nghệ học máy và trí tuệ nhân tạo để phân tích dữ liệu và dự đoán hành vi của người lái và môi trường. Tuy nhiên, việc huấn luyện và triển khai mô hình học máy phức tạp và đòi hỏi dữ liệu lớn và đa dạng. Ngoài ra, cần đảm bảo tính công bằng và an toàn trong việc sử dụng dữ liệu và thuật toán học máy để tránh các vấn đề liên quan đến đạo đức và quyền riêng tư.

✦ **Phát triển và cập nhật liên tục:** Công nghệ liên quan đến ADAS đang phát triển nhanh chóng. Cần có sự cập nhật và nâng cấp liên tục của phần mềm và phần cứng ADAS để cải thiện hiệu suất, tính năng và sự an toàn. Đồng thời, cần đảm bảo khả năng tương thích với các hệ thống mới và tiêu chuẩn công nghệ.

3. Nghiên cứu bắt đầu dự án “ADVANCED-DRIVER-ASSISTANCE-SYSTEM”

Từ vấn đề thực tiễn “**tình trạng giao thông**” và “**công nghệ**” nêu trên, tôi bắt đầu nghiên cứu thực hiện dự án “**ADVANCED-DRIVER-ASSISTANCE-**

SYSTEM” với ý nghĩa **“Là một hệ thống AI giúp hỗ trợ người lái xe-Giải pháp khoa học công nghệ”** đi vào đời sống giúp cho cuộc cách mạng ô tô ngày càng phát triển vượt bậc hỗ trợ cho người tham gia giao thông an toàn và thuận tiện.

Để thực hiện hai vấn đề trên và đảm bảo tính hiệu quả và phát triển của **ADVANCED-DRIVER-ASSISTANCE-SYSTEM**. Nên tôi nghiên cứu và là người lập dự án. Chia làm hai hình thức sau:

- Hình thức lập trình mô phỏng thực tế ảo trên máy tính
- Triển khai nhân rộng vào thực tiễn đời sống với sự trợ giúp của các thiết bị.

II. Mục tiêu kết quả mong đợi 1. Hệ thống ADVANCED-DRIVER-ASSISTANCE-SYSTEM mô phỏng trên

máy tính

- **Mục tiêu thứ nhất:** xây dựng cơ sở dữ liệu và nền tảng hệ thống Sử dụng các mã nguồn mở trên nền tảng công nghệ của Open AI.

Lập trình phần mềm bằng ngôn ngữ python liên kết với các thư viện hỗ trợ: OpenCV, Mediapipe, Numpy,...

Tận dụng được khả năng nhanh, nhạy xử lý ảnh (gọi là thị giác máy tính)

- **Mục tiêu thứ hai:** đưa lên nền tảng facebook các hội nhóm công nghệ nhằm xin ý kiến mọi người về dự án và tiến hành tối ưu dự án.

- **Mục tiêu thứ ba:** tiến hành thực nghiệm mô phỏng trên máy tính và thống kê các thông số đạt được.

- **Mục tiêu thứ tư:** có thể liên kết với cơ sở dữ liệu quốc gia nhằm quản lí theo định danh điện tử.

- **Mục tiêu thứ năm:** đưa ra công nghệ tiên tiến với chi phí thấp cho những người có nhu cầu có thể tiếp cận sản phẩm.

- **Mục tiêu tiến hành chung:** phát triển liên tục và hoàn thiện tối ưu dự án.

2. Hệ thống ADVANCED-DRIVER-ASSISTANCE-SYSTEM triển khai vào thực tiễn

- **Mục tiêu thứ nhất:** phát triển trí tuệ nhân tạo và hệ thống cơ sở dữ liệu quản lý định danh điện tử.

- **Mục tiêu thứ hai:** phát triển ứng dụng dựa trên lập trình (với ngôn ngữ python và cơ sở dữ liệu đã được huấn luyện từ nguồn github (với cơ sở dữ liệu có sẵn: lane.pt, obstacle.pt, traffic_sign.pt). Phát triển phần mềm qua đóng gói dự án với (ADAS.*) để có thể đồng bộ hóa với hệ điều hành (OS) của từng loại xe.

- **Mục tiêu thứ ba:** một trong những hệ thống giúp giảm TNGT ở nước ta và phần mềm đứng số 1 trong hỗ trợ an toàn giao thông thời kỳ công nghiệp 4.0.

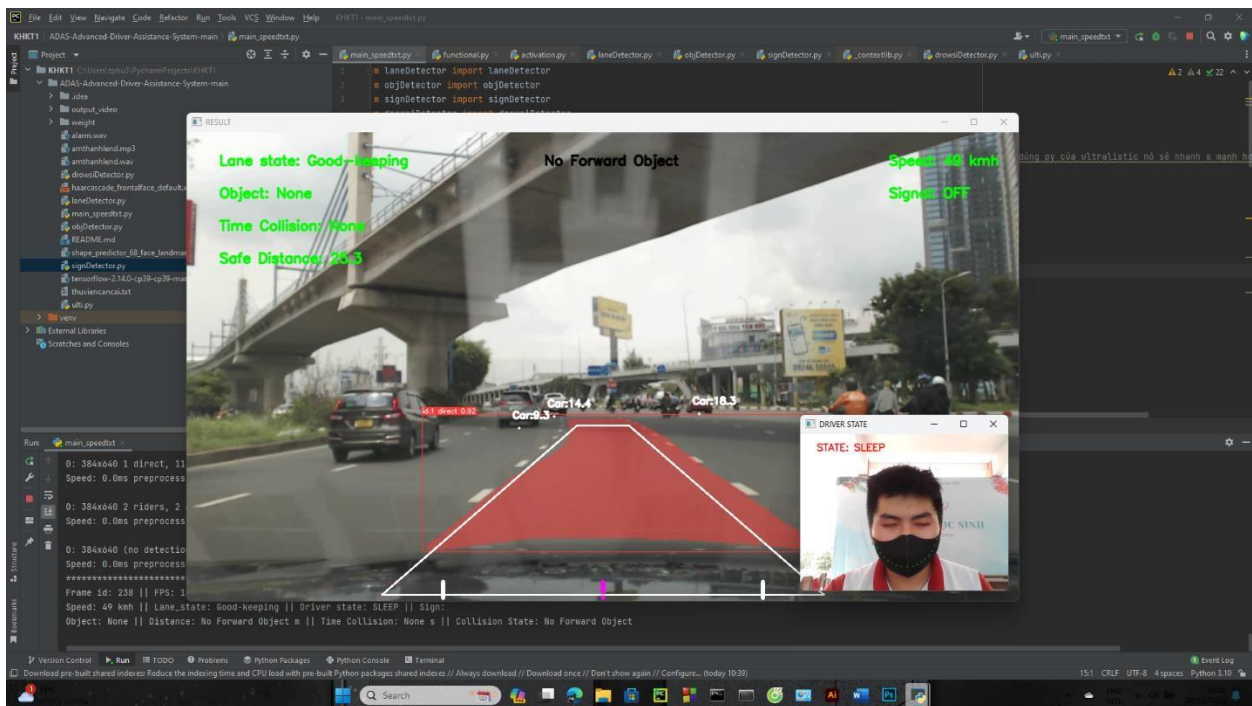
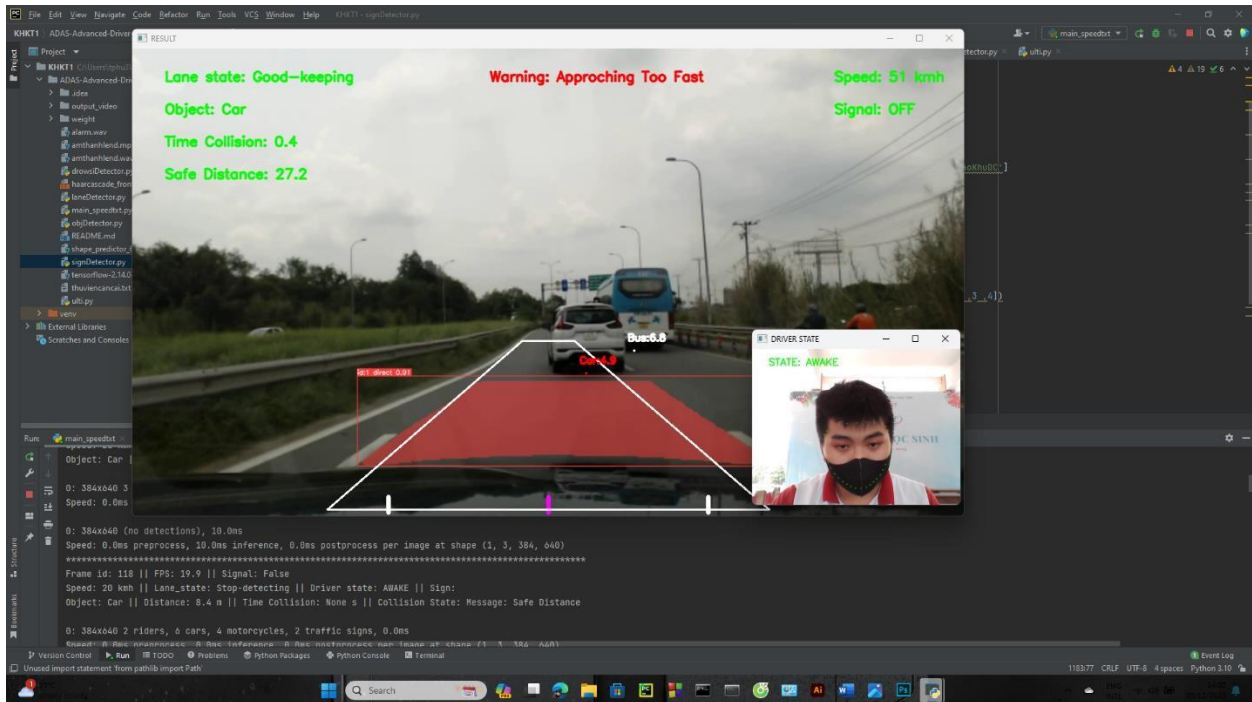
- **Mục tiêu thứ tư:** phát triển mã nguồn mở hỗ trợ cá nhân hóa cho từng loại phương tiện.

- **Mục tiêu thứ năm:** nhằm cải tiến hệ thống giao thông đường bộ Việt Nam trở nên văn minh tiến đến xã hội tri thức và công nghệ số.

- **Mục tiêu tiến hành chung:** chuyên đổi số đóng một vai trò quan trọng trong cuộc Cách mạng công nghiệp 4.0, phát triển tiến đến xã hội tri thức, hỗ trợ người lái xe thuận tiện hơn, giúp giảm tình trạng TNGT do các yếu tố chủ quan, quản lý thông qua cơ sở dữ liệu quốc gia đồng bộ định danh biển số điện tử.

3. Tổng thể về hệ thống ADVANCED-DRIVER-ASSISTANCE-SYSTEM-ADAS

Đưa ra ADVANCED-DRIVER-ASSISTANCE-SYSTEM có tham vọng và mục tiêu lớn là trở thành một hệ thống thiết bị hữu ích cho các tài xế với công nghệ trí tuệ nhân tạo đem lại sự yên tâm thoải mái cho người sử dụng. Chính vì điều đó ADVANCED-DRIVER-ASSISTANCE-SYSTEM gọi tắt là ADAS đã được em chọn làm dự án nghiên cứu và phát triển chặt chẽ theo quy trình logic. Với khả năng tận dụng được tốc độ tính toán của máy tính hiện nay qua thị giác máy tính để có thể dễ dàng phát hiện được những yếu tố tác nhân bên ngoài như: vật cản, khoảng cách an toàn, hệ thống nhận diện làn đường, ...



III. Giả thuyết khoa học và những điều ADVANCED-DRIVERASSISTANCE-SYSTEM-ADAS đã làm được

1. Giả thuyết khoa học

- Từ vấn đề “tình trạng giao thông” đã đặt ra các câu hỏi:
Về tính an toàn giao thông

Về xây dựng hệ thống thiết bị như thế nào - Từ vấn đề “công nghệ” đã đặt ra các câu hỏi:

Có cần thiết hay không

Nền tảng lập trình trên máy tính

Hệ thống hoạt động như thế nào

Chi phí phần mềm như thế nào

- Từ các câu hỏi vấn đề trên giả thuyết được đặt ra: tính an toàn giao thông có cần thiết? xây dựng như thế nào? hệ thống hoạt động ra sao? Chi phí phần mềm như thế nào? Sẽ có thể giải quyết được một phần hoặc tất cả các câu hỏi vấn đề trên.

2. Những điều ADVANCED-DRIVER-ASSISTANCE-SYSTEM đã làm được

a) Tổng thể ADVANCED-DRIVER-ASSISTANCE-SYSTEM

- Xây dựng thành công và đi hoạt động mô hình thực hiện trên mô phỏng trên máy tính :

- Hệ thống thiết bị mô phỏng
- Hệ thống thiết bị áp dụng thực tiễn có các thiết bị hỗ trợ
- Hệ thống camera nhận diện buồn ngủ tích hợp

- Tổng quan về công nghệ

- Hệ thống thiết bị mô phỏng trên máy tính: hỗ trợ hầu hết các OS hiện nay trên thị trường tạo sự thuận tiện cho người sử dụng, nhận diện chính xác khoảng 95% các đối tượng và trong tương lai đồng bộ được với cơ sở dữ liệu quốc gia về định danh biển số điện tử
- Hệ thống thiết bị áp dụng thực tiễn sẽ được tích hợp với OS của xe: thông qua các thiết bị hỗ trợ và nền tảng mô phỏng trên máy tính sẽ đem lại sự thuận tiện cho người sử dụng, nhận diện các đối tượng chính xác khoảng 99% nhờ có sự hỗ trợ của radar, cảm biến, ... và tương lai đồng bộ được với cơ sở dữ liệu quốc gia về định danh biển số điện tử

- Tổng quan về các tính năng và dữ liệu hiện hành gồm:

- Nhận diện làn đường – thông báo – cảnh báo

```

37     self.laneBoxes = self.laneTrack(frame, tracker='bytetrack.yaml')
38     if len(self.laneBoxes) != 0:
39         # Lấy vị trí của hai đường viền bên trái và bên phải
40         x_tl, y_tl, x_br, y_br = self.laneBoxes[0]
41         # self.center = (int((x_tl + x_br)/2), int((y_br + y_tl)/2))
42
43         if (x_br - x_tl)//self.dict_setup['shape'][1] > 0.9 \
44             and (y_br - y_tl)//self.dict_setup['shape'][0] > 0.9:
45             self.state = "Stop-Detecting"
46             return self.state
47
48         # Lệch lane + signal_bool
49         if x_tl < self.dict_setup['laneThresh'][0] and x_br > self.dict_setup['laneThresh'][1]:
50             # Nếu xe đang đi giữa hai làn đường đúng
51             self.state = "Good-keeping"
52             self.count = 0
53         else:
54             if not signal_bool:
55                 if self.count < 10:
56                     self.count +=1
57                     self.state = "Good-keeping"
58
59                 else:
60                     # Nếu xe đang lệch khỏi làn đường
61                     if self.dict_setup['laneThresh'][0] < x_tl and x_br < self.dict_setup['laneThresh'][1]:
62                         self.state = "Wrong-both"
63                     else:
64                         if x_tl > self.dict_setup['laneThresh'][0]:
65
66                             self.state = "Wrong-left"
67                         elif x_br < self.dict_setup['laneThresh'][1]:
68
69                             self.state = "Wrong-right"
70                         else: pass
71                     else:
72                         self.state = "Changing-lane"
73             else:
74                 self.state = "Lane-Missing"
75     return self.state

```

- Nhận diện tài xế buồn ngủ và cảnh báo âm thanh

```

class drowsiDetector():
    def __init__(self) -> None:
        self.face_mesh = mp.solutions.face_mesh
        self.draw_utils = mp.solutions.drawing_utils
        self.landmark_style = self.draw_utils.DrawingSpec((0, 255, 0), thickness=2, circle_radius=2)
        self.connection_style = self.draw_utils.DrawingSpec((0, 0, 255), thickness=2, circle_radius=2)

        self.STATIC_IMAGE = False
        self.MAX_NO_FACES = 2
        self.DETECTION_CONFIDENCE = 0.6
        self.TRACKING_CONFIDENCE = 0.5

        self.COLOR_RED = (0, 0, 255)
        self.COLOR_BLUE = (255, 0, 0)
        self.COLOR_GREEN = (0, 255, 0)

        self.LIPS = [61, 146, 91, 181, 84, 17, 314, 405, 321, 375, 291, 308, 324, 318, 402, 317, 14, 87, 178, 88, 95,
                    185, 40, 39, 37, 0, 267, 269, 270, 409, 415, 310, 311, 312, 13, 82, 81, 42, 183, 78]

        self.RIGHT_EYE = [33, 7, 163, 144, 145, 153, 154, 155, 133, 173, 157, 158, 159, 160, 161, 246]
        self.LEFT_EYE = [362, 382, 381, 380, 374, 373, 390, 249, 263, 466, 388, 387, 386, 385, 384, 398]

        self.LEFT_EYE_TOP_BOTTOM = [386, 374]
        self.LEFT_EYE_LEFT_RIGHT = [263, 362]

        self.RIGHT_EYE_TOP_BOTTOM = [159, 145]
        self.RIGHT_EYE_LEFT_RIGHT = [133, 33]

        self.UPPER_LOWER_LIPS = [13, 14]
        self.LEFT_RIGHT_LIPS = [78, 308]

        self.FACE = [10, 338, 297, 332, 284, 251, 389, 356, 454, 323, 361, 288, 397, 365, 379, 378, 400,
                    377, 152, 148, 176, 149, 150, 136, 172, 58, 132, 93, 234, 127, 162, 21, 54, 103, 67, 109]

        self.face_model = self.face_mesh.FaceMesh(static_image_mode=self.STATIC_IMAGE,
                                                  max_num_faces=self.MAX_NO_FACES,
                                                  min_detection_confidence=self.DETECTION_CONFIDENCE,
                                                  min_tracking_confidence=self.TRACKING_CONFIDENCE)

```

- Nhận diện đối tượng xung quanh.
- Nhận diện khoảng cách an toàn và tiền tố va chạm
- Nhận diện signal
- Kiểm soát tốc độ

```

class objDetector(YOLO):
    def __init__(self, model: str | Path = 'weight/obstacle.pt', task='detect', dict_setup = {},
                 focal_length:float=3.67, sensor_height:float = 8.45, #8.45
                 react_time= 0.5, deceleration= 5):

        super().__init__(model, task)

        self.dict_setup = dict_setup

        self.Height = { # mm
            0: 1700, # Person
            1: 1800, # Rider
            2: 1700, # Car
            3: 3000, # Truck
            4: 2800, # Bus
            # 5: 2500, # Train
            # 6: 1300, # Moto
            # 7: 1300, # Bicycle
        }

```

```

def collisionLogic(self, time_collision):
    if time_collision is not None:
        if 0.8 < time_collision <= 2.5:
            color = (255, 255, 0)
            collision_state = 'Attention: Forward Obstruction'
        elif 0.5 < time_collision <= 0.8:
            color = (0, 255, 255)
            collision_state = 'Attention: Imminent Collision'
        elif time_collision <= 0.5:
            color = (0, 0, 255)
            collision_state = "Warning: Approaching Too Fast"
        else:
            color = (0, 255, 0)
            collision_state = "Message: Safe Distance"
    else:
        color = (0, 255, 0)
        collision_state = "Message: Safe Distance"

    return collision_state, color

```

Hình ảnh: Tổng quan hiện tại ADAS: đã có cơ sở dữ liệu, hệ thống nhận diện tốt

b) Hệ thống mô phỏng trên máy tính

- Về công nghệ: xây dựng thành công và đi vào hoạt động một cách chính xác ổn định sau thời gian thực hiện nghiên cứu: ○ Lập trình trên nền tảng ngôn ngữ lập trình Python, hỗ trợ nhiều thư viện dễ dàng bảo trì chỉnh sửa tối ưu và nâng cấp.

- Giao diện rành mạch rõ ràng.

- Cơ sở dữ liệu hoạt động truy vấn nhanh chóng.
- Thuật toán:

Sử dụng các thư viện: OpenCV, Mediapipe, Mumpy, ...

Thuật toán trong class objDetector

Khởi tạo các tham số cần thiết như chiều cao của đối tượng, từ đó tính toán khoảng cách gần đúng, thời gian phản ứng, tốc độ giảm tốc.

Hàm objTrack sử dụng object detector để phát hiện và theo dõi các đối tượng trong frame. Trả về bounding boxes của các đối tượng.

Hàm calDistance tính toán khoảng cách gần đúng dựa trên chiều cao của đối tượng trong hình.

Hàm calSafeDistance tính toán khoảng cách an toàn dựa trên tốc độ, thời gian phản ứng và tốc độ giảm tốc.

Hàm processResult lọc ra thông tin các đối tượng và biên báo trong frame.

Hàm getClosestObject tìm ra đối tượng gần nhất và tính toán thời gian va chạm.

Hàm getResult trả về kết quả gồm thông tin đối tượng gần nhất, khoảng cách an toàn.

Các hàm còn lại dùng để hiển thị kết quả, theo dõi vận tốc, lưu trữ thông tin qua các frame.

Nói chung, thuật toán sử dụng phát hiện và theo dõi đối tượng để cung cấp cảnh báo va chạm, khoảng cách an toàn dựa trên các tham số đầu vào. Thuật toán phát hiện lái xe buồn ngủ gồm các bước chính:

Khởi tạo các tham số cần thiết như cấu hình object detection, kiểu vẽ, màu sắc, ...

Xác định các điểm trên khuôn mặt: mắt, môi cần theo dõi.

Sử dụng MediaPipe để phát hiện và lấy ra landmark của khuôn mặt trong khung hình camera.

Tính toán khoảng cách giữa các điểm landmark để xác định tỷ lệ chiều rộng/chiều dài của mắt.

So sánh tỷ lệ mắt với ngưỡng định sẵn để xác định trạng thái mắt mở/ít mở.

Hiển thị thông báo trạng thái và phát âm thanh báo động khi phát hiện lái xe buồn ngủ (khoảng sau 5 giây).

Lặp lại quá trình trên cho từng khung hình.

Thuật toán nhận diện biển báo tốc độ

Khởi tạo camera và lấy kích thước của khung hình. Định nghĩa giới hạn làn đường và hình thoi kiểm soát tốc độ.

Tạo lớp Yolo để phát hiện biển báo giao thông trong khung hình.

Hàm getFPS tính tốc độ khung hình để điều chỉnh độ chính xác.

Tạo videoWriter để ghi dữ liệu nhận diện ra khung hình.

Lớp SQLConnector để lưu trữ dữ liệu nhận diện vào CSDL.

Vòng lặp xử lý từng khung hình: Phát hiện biển báo giao thông và tìm kiếm biển cần theo dõi; Cập nhật tốc độ phương tiện, kiểm tra vi phạm; Ghi kết quả nhận diện lên khung hình; Cập nhật CSDL.

Sau mỗi vòng lặp cập nhật fps và kiểm tra điều kiện dừng.

Thuật toán chính trong class laneDetector:

Khởi tạo đối tượng Yolo để nhận diện và theo dõi đường viền.

Hàm laneTrack sử dụng Yolo để phát hiện và trả về bounding boxes của đường viền trong khung hình.

Hàm laneLogic xử lý logic điều khiển: Kiểm tra tốc độ và tín hiệu xe; Lấy vị trí đường viền; Phân tích vị trí xe so với làn đường; Trả về trạng thái xe theo quy tắc;

Hàm laneDraw vẽ thông tin lên khung hình: Vẽ kết quả xử lý từ Yolo ; Vẽ đường giới hạn làn, tốc độ, tín hiệu; Hiển thị thông báo trạng thái; Sử dụng concurrent thread để phát tiếng cảnh báo khi vi phạm.

c) Hệ thống thiết bị áp dụng thực tiễn

- Về công nghệ: tích hợp vào ô tô

- Kết hợp với cái modum cảm biến và radar cho kết quả nhận dạng chính xác tuyệt đối
- Về thuật toán
 - o Sử dụng các thư viện OpenCV, Ultralytics, Mumpy, Mediapipe, ...
 - o Thuật toán trong class objDetector:

Khởi tạo các tham số cần thiết như chiều cao của đối tượng, từ đó tính toán khoảng cách gần đúng, thời gian phản ứng, tốc độ giảm tốc.

Hàm objTrack sử dụng object detector để phát hiện và theo dõi các đối tượng trong frame. Trả về bounding boxes của các đối tượng.

Hàm calDistance tính toán khoảng cách gần đúng dựa trên chiều cao của đối tượng trong hình.

Hàm calSafeDistance tính toán khoảng cách an toàn dựa trên tốc độ, thời gian phản ứng và tốc độ giảm tốc.

Hàm processResult lọc ra thông tin các đối tượng và biển báo trong frame.

Hàm getClosestObject tìm ra đối tượng gần nhất và tính toán thời gian va chạm. Hàm getResult trả về kết quả gồm thông tin đối tượng gần nhất, khoảng cách an toàn.

Các hàm còn lại dùng để hiển thị kết quả, theo dõi vận tốc, lưu trữ thông tin qua các frame.

Nói chung, thuật toán sử dụng phát hiện và theo dõi đối tượng để cung cấp cảnh báo va chạm, khoảng cách an toàn dựa trên các tham số đầu vào

- o Thuật toán phát hiện lái xe buồn ngủ gồm các bước chính:

Khởi tạo các tham số cần thiết như cấu hình object detection, kiểu vẽ, màu sắc, ...

Xác định các điểm mặt, mắt, môi cần theo dõi.

Sử dụng MediaPipe để phát hiện và lấy ra landmark của khuôn mặt trong khung hình camera.

Tính toán khoảng cách giữa các điểm landmark để xác định tỷ lệ chiều rộng/chiều dài của mắt.

So sánh tỷ lệ mắt với ngưỡng định sẵn để xác định trạng thái mắt mở/ít mở.

Hiện thị thông báo trạng thái và phát âm thanh báo động khi phát hiện buồn ngủ (khoảng 5 giây)

Lặp lại quá trình trên cho từng khung hình.

o Thuật toán nhận diện biển báo tốc độ

Khởi tạo camera và lấy kích thước của khung hình. Định nghĩa giới hạn làn đường và hình thoi kiểm soát tốc độ.

Tạo lớp Yolo để phát hiện biển báo giao thông trong khung hình.

Hàm getFPS tính tốc độ khung hình để điều chỉnh độ chính xác.

Tạo videoWriter để ghi dữ liệu nhận diện ra khung hình.

Lớp SQLConnector để lưu trữ dữ liệu nhận diện vào CSDL.

Vòng lặp xử lý từng khung hình: Phát hiện biển báo giao thông và tìm kiếm biển cần theo dõi; Cập nhật tốc độ phương tiện, kiểm tra vi phạm; Ghi kết quả nhận diện lên khung hình; Cập nhật CSDL

Sau mỗi vòng lặp cập nhật fps và kiểm tra điều kiện dừng.

o Thuật toán chính trong class laneDetector:

Khởi tạo đối tượng Yolo để nhận diện và theo dõi đường viền.

Hàm laneTrack sử dụng Yolo để phát hiện và trả về bounding boxes của đường viền trong khung hình.

Hàm laneLogic xử lý logic điều khiển: Kiểm tra tốc độ và tín hiệu xe; Lấy vị trí đường viền; Phân tích vị trí xe so với làn đường; Trả về trạng thái xe theo quy tắc

Hàm laneDraw vẽ thông tin lên khung hình: Vẽ kết quả xử lý từ Yolo ; Vẽ đường giới hạn làn, tốc độ, tín hiệu; Hiện thị thông báo trạng thái

Sử dụng concurrent thread để phát tiếng cảnh báo khi vi phạm.

- Về thiết bị ○ Cảm biến radar ô tô được sử dụng để xác định khoảng cách và tốc độ của các đối tượng đứng im hoặc di chuyển xung quanh ô tô. Thiết bị radar phát ra sóng vô tuyến, chạy với tốc độ cực nhanh và phản xạ trở lại radar khi có vật thể trên đường đi của nó.

Các radar ô tô cho tầm xa điển hình có thể đo các vật thể cách xa

○ Camera trên ô tô

- Lập trình trên các thiết bị cảm biến: Lập trình arduino để nhúng dữ liệu vào cảm biến sử dụng ngôn ngữ C ++.

IV. Phạm vi nghiên cứu

1. Tổng thể hệ thống mô phỏng trên máy tính

Được nghiên cứu và thực hiện trên phạm vi mô phỏng trên máy tính, chia sẻ trên nhóm cộng đồng facebook **J2TEAM Community** và **hội những anh em thích ăn Mì AI** cung cấp một cách miễn phí. Tổng thể dự án luôn đề cao khả năng hoàn thiện và chính xác của các modul hỗ trợ và tính năng tiện ích của hệ thống mang lại cho người sử dụng.

Được thiết kế trên ngôn ngữ lập trình Python và các CSDL (nguồn github) được đã được huấn luyện

2. Tổng thể hệ thống thiết bị áp dụng vào thực tiễn

- Nghiên cứu điều kiện làn đường mờ
- Thực nghiệm điều kiện có thời tiết xấu, mưa, ...
- Tích hợp công nghệ cao với cảm biến
- Được thiết kế với ngôn ngữ lập trình python và C++, lập trình nhúng
- Hoạt động với cơ sở dữ liệu, với nền tảng quản lí, truy vấn, truy xuất dữ liệu đưa ra tốc độ nhanh chóng

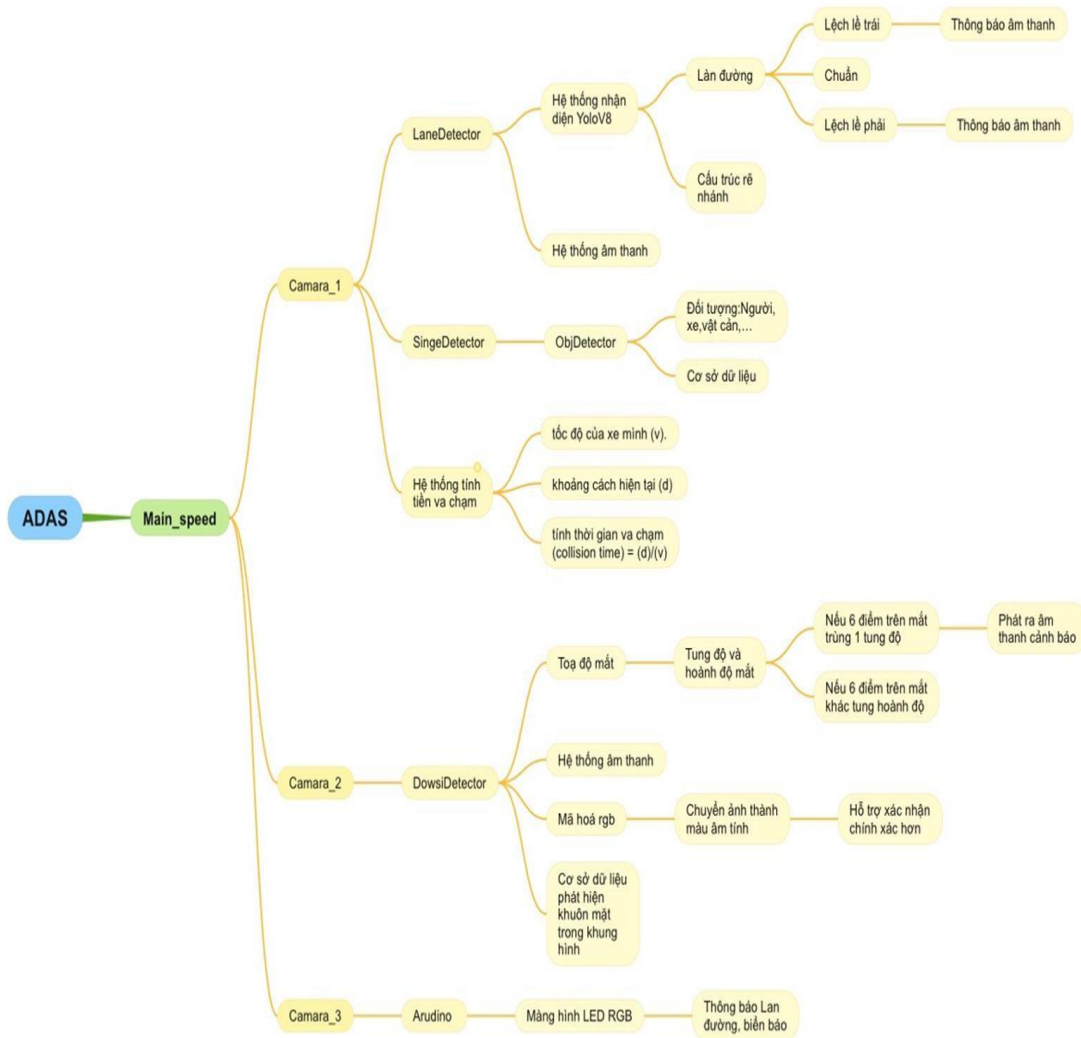
V. Kế hoạch nghiên cứu và chuẩn bị thực hiện

1. Kế hoạch:

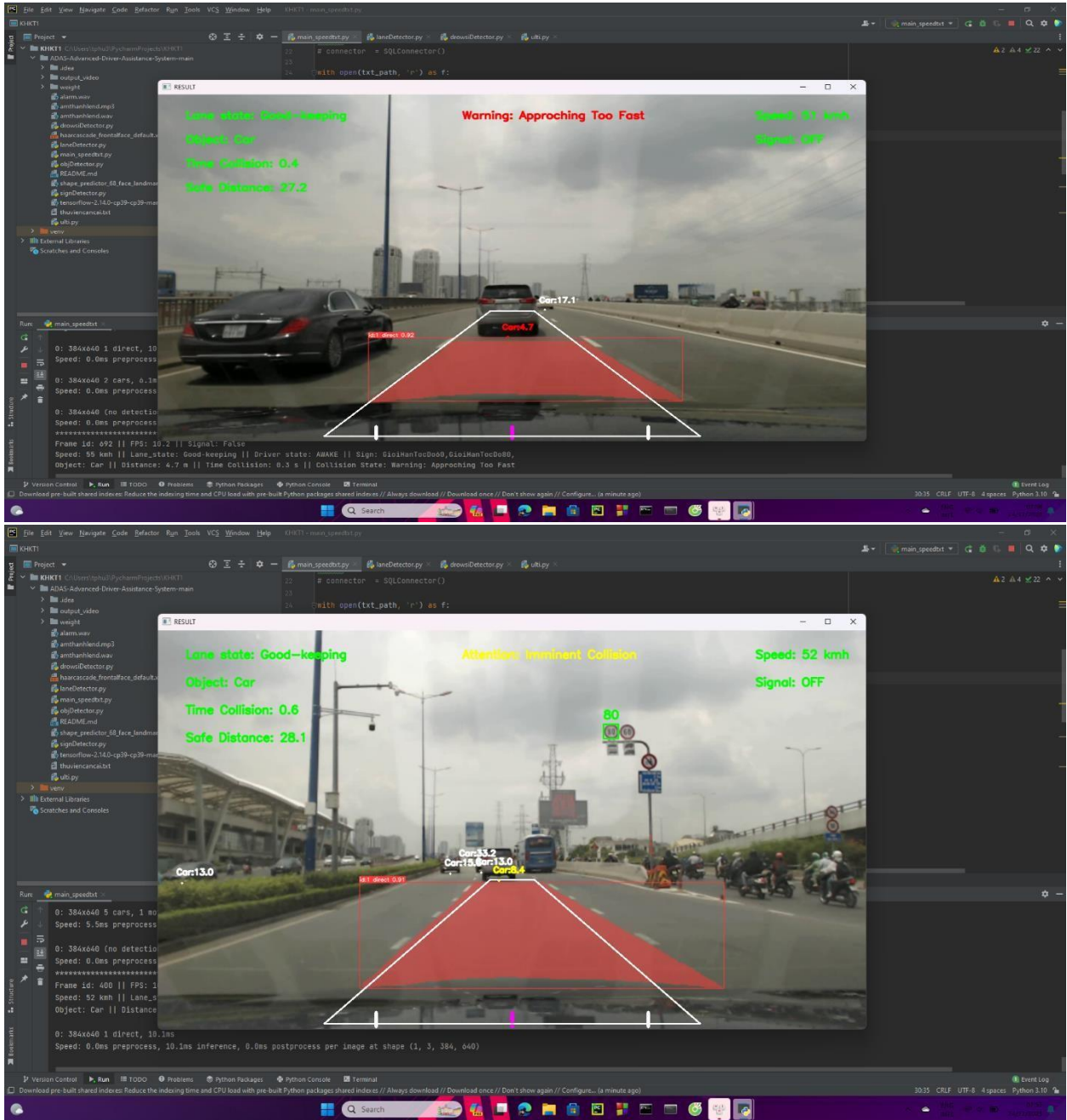
a) Nền tảng mô phỏng trên máy tính

- Nghiên cứu vấn đề tình trạng giao thông ở Việt Nam
- Nghiên cứu dự án

- Tạo sơ đồ nguyên lí cho dự án



- Xây dựng giao diện



- Thu thập ý kiến

< Hội những anh em thích ăn Mì AI

Xin chào mọi người.

Hiện tại em đang làm project môn học về sản phẩm hỗ trợ tài xế (ADAS based on camera).

Hiện tại em đang gặp vấn đề về cảnh báo tiê... Xem thêm



👍 21

19 bình luận 4 lượt chia sẻ



Luu Trung Hieu

Đâu có tính chính xác tốc độ của xe phía trước và khoảng cách giữa xe của mình với xe phía trước qua hình ảnh. Có tính được thì cũng tương đối thôi, phải dùng cảm biến radar mới có vận tốc và khoảng cách có giá trị về thực địa.

29 tuần Thích Phản hồi



Nguyễn Văn Trường

Nếu có vận tốc, gia tốc thì phải có imu, và khoảng cách phải có lidar chứ

29 tuần Thích Phản hồi



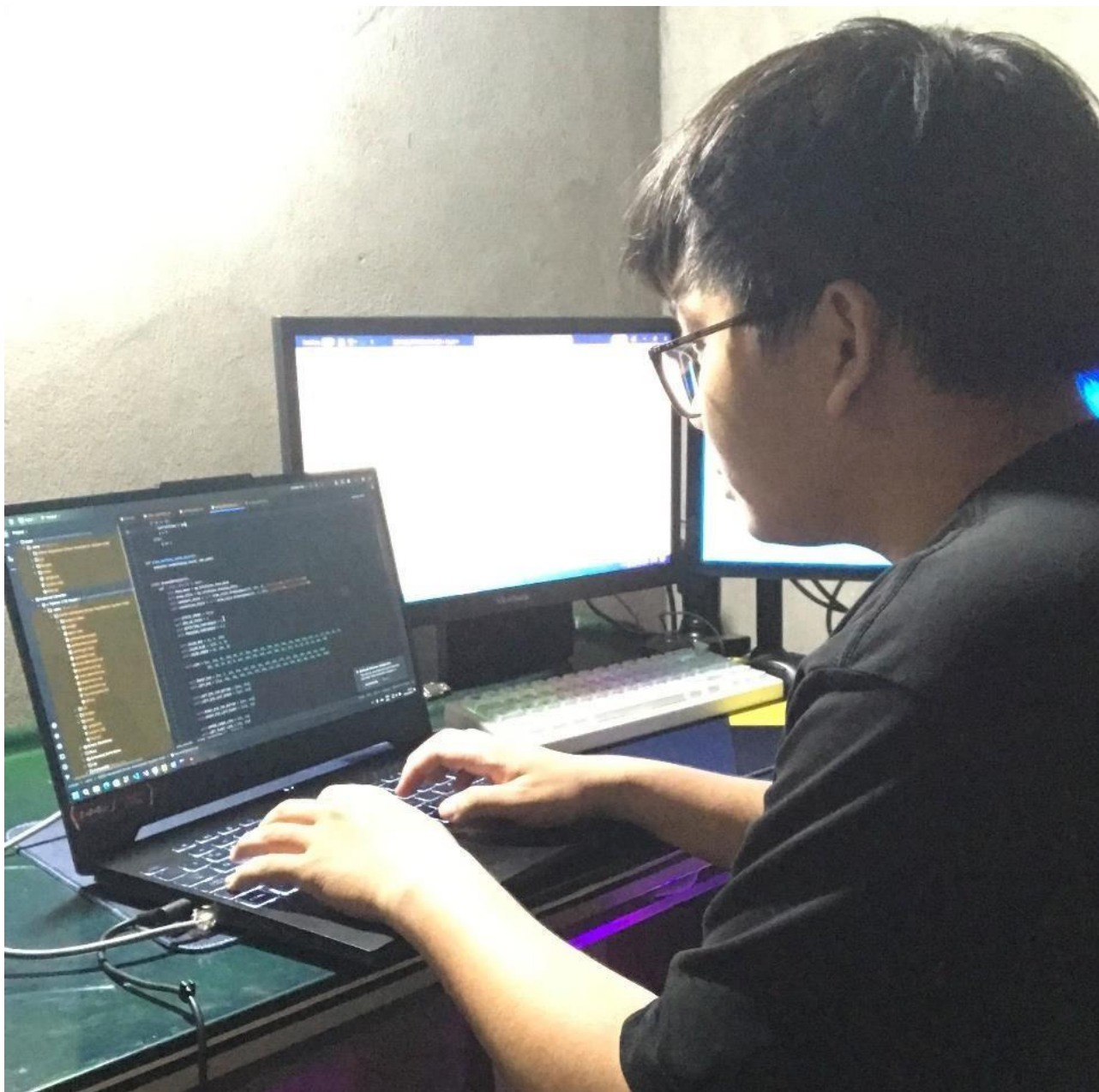
Nguyễn Quốc Khánh

Mình mới làm xong FCWS bằng distance b cần gửi cho tham khảo

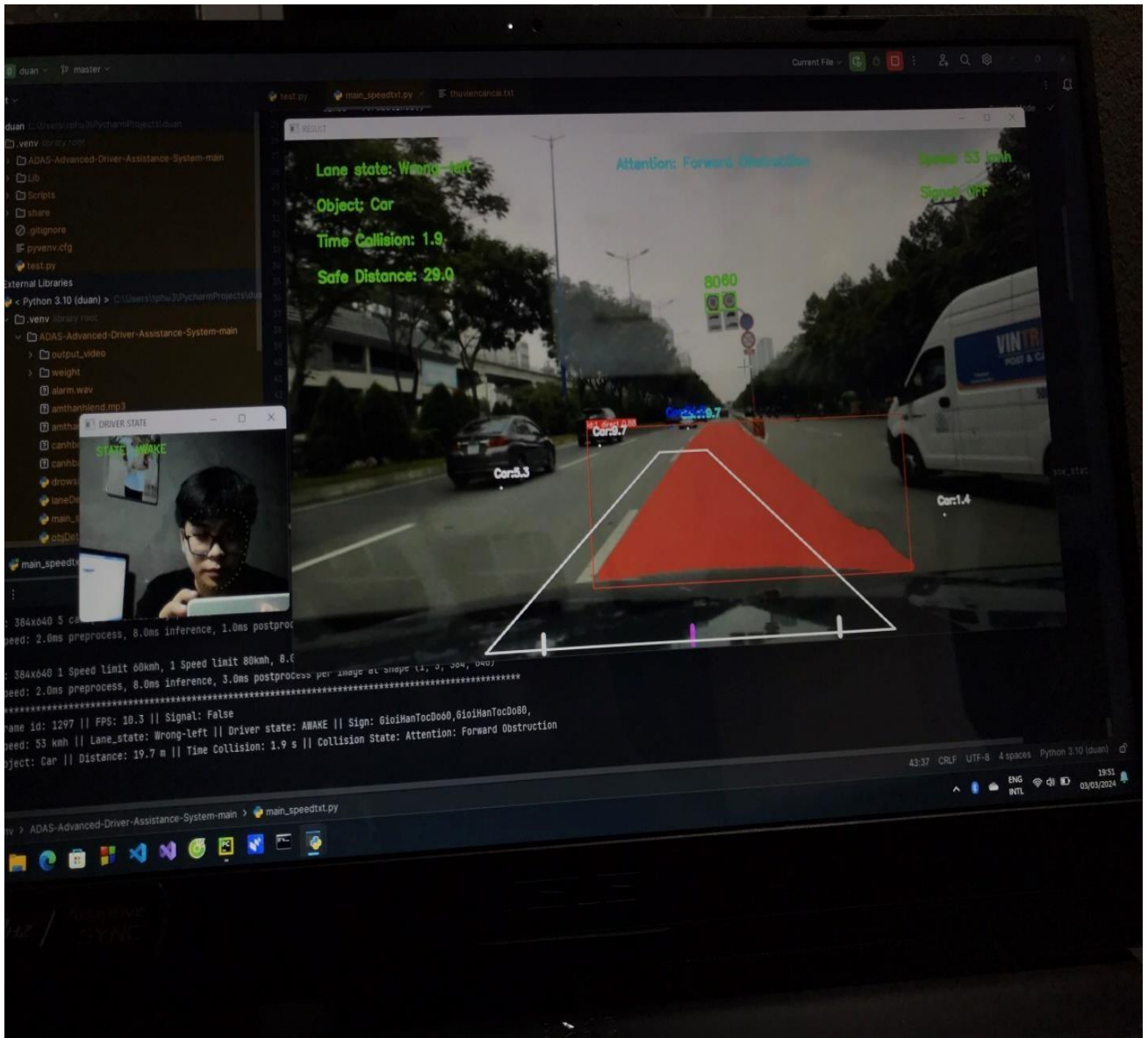
28 tuần Thích Phản hồi



- Thu thập dữ liệu cho dự án
- Nghiên cứu xây dựng hệ thống, chuẩn bị các nguồn tài nguyên (thư viện hỗ trợ, các thuật toán cơ sở ban đầu, thu thập các nguồn tài nguyên trên internet hoặc tự thu thập)
- Bắt đầu lập trình dự án



- Tối ưu và sửa chữa dự án (tối ưu các modul hoạt động và tối ưu hóa code để dễ dàng sửa chữa và bảo trì nâng cao trong tương lai)
- Bắt đầu thực hiện test hệ thống một cách hoàn chỉnh



b) Nền tảng hệ thống thiết bị thực tiễn đã tiến hành trên mô phỏng trên máy tính

- Nghiên cứu phân tích từ kết quả đạt được từ mô phỏng trên máy tính
- Phân tích xây dựng dữ liệu tối ưu từ mô phỏng
- Xây dựng phát triển công nghệ
- Kết hợp với các modul cảm biến radar tiến hành lập trình arduino - Xây dựng giao diện đơn giản thuận tiện cho người sử dụng.
- Tiến hành thực nghiệm phát triển tối ưu lắp đặt các thiết bị modul cảm biến, radar.

2. Chuẩn bị:

- Hệ thống máy tính có hiệu năng cao.
- Sử dụng các ngôn ngữ lập trình (Python, C++, ...) để phát triển dự án.
- Các kho dữ liệu và thư viện lập trình python cần thiết.
- Các hình ảnh để huấn luyện AI nhận diện biển báo.
- Các nguồn tài liệu dữ liệu phân tích nghiên cứu.
- Các nguồn tài liệu khám khảo và phát triển.
- Nguồn kinh phí và tài trợ.
- Một webcam có độ sắc nét cao.

3. Thực hiện:

- Hệ thống mô phỏng
 - Giai đoạn 1: Thực hiện nghiên cứu dự án.
 - Giai đoạn 2: Tiến hành thu thập CSDL cho dự án .
 - Giai đoạn 3: Tiến hành xây dựng cấu trúc dự án ban đầu .
 - Giai đoạn 4: Thử nghiệm dự án.
 - Giai đoạn 5: Tối ưu, chỉnh sửa các lỗi mà AI nhận diện chưa chính xác và sử dụng nhiều thư viện.
 - Giai đoạn 6: Hoàn chỉnh dự án
- Hệ thống thiết bị áp dụng thực tiễn.
 - Giai đoạn 1: Phân tích những kết quả thử nghiệm trong mô phỏng.
 - Giai đoạn 2: Lựa chọn vị trí thích hợp để lắp đặt cảm biến, radar, camera nhận diện một cách hợp lí.
 - Giai đoạn 3: Tiến hành lập trình nhúng vào các thiết ngoại vi.
 - Giai đoạn 4: Thử nghiệm dự án.
 - Giai đoạn 5: Từ các kết quả đã thử nghiệm tiến hành tối ưu dự án nâng cấp điều chỉnh những sai sót.

- Giai đoạn 6: Hoàn chỉnh dự án.

VI. Thành tựu số liệu nghiên cứu hệ thống ADAS

- Hệ thống nhận diện biển báo và tiền va chạm tương đối chính xác
- Tuy nhiên hệ thống nhận diện được các đối tượng, phương tiện và làn đường khi xe lệch làn phía trước một cách chính xác tuyệt đối
- Nhận diện tốt gần như là hoàn hảo trong điều kiện môi trường bình thường
- Hệ thống nhận diện được biển báo, tài xế buồn ngủ, làn đường, xe phía trước và thời gian va chạm.
 - Khi vận tốc xe dưới 30km/h sẽ tắt chức năng nhận diện làn đường để tránh nhiễu và cảnh báo sai gây mất tập chung cho tài xế.
 - Thực nghiệm trên quốc lộ 51 chức năng nhận diện làn đường bị nhiễu do làn đường bị mờ hoặc bị mất một phần hay toàn bộ (đoạn Video kèm theo)
 - Hệ thống phát hiện chính xác tài xế buồn ngủ liền đưa ra thông tin và cảnh báo bằng âm thanh để tài xế kịp thời nghỉ ngơi nhằm đảm bảo an toàn giao thông
 - Hệ thống có thể phát hiện và nhận diện biển báo khá chính xác nhằm cung cấp thêm thông tin về vận tốc và cung đường cho tài xế tuân thủ đúng tốc độ nhằm tránh xảy ra tai nạn.
 - Với tình huống xảy ra khá phổ biến trên các cung đường Việt Nam đó là xe máy vượt ẩu trước đầu ô tô hệ thống gần như phát hiện ngay lập tức và thông báo đến tài xế để tránh gây tai nạn. (có đoạn Video kèm theo)
 - Tình huống thực nghiệm môi trường âm u, thiếu sáng, trời nhiều mây đen hệ thống vẫn nhận diện khá tốt tất cả các chức năng vẫn nhận diện khá tốt làn đường, tài xế buồn ngủ, tiền va chạm và biển báo tốc độ
 - Trong môi trường thiếu sáng và có mưa to hệ thống bị nhiễu nhẹ do mưa và lượng nước mưa kết hợp với gạt nước của xe làm gián đoạn hệ thống, tuy nhiên hệ thống vẫn hoạt động tương đối chính xác tất cả các chức năng do cần gạt nước thi thoảng lướt qua khung hình không ảnh hưởng nhiều đến hệ thống

Tóm lại: Hệ thống ADAS vẫn hoạt động tốt trong mọi điều kiện thời tiết và đem lại sự thuận tiện cho tài xế và an toàn cho tình trạng giao thông nước ta.

VII. Kết luận

- Công nghệ ADAS được coi là bước đột phá của ngành công nghiệp ô tô. ADAS sử dụng mạng máy tính và các dữ liệu thông minh để giúp người lái có hành trình di chuyển thuận tiện và an toàn.

- Công nghệ ADAS mang đến sự hiện đại, khẩn cấp, chính xác và tiện lợi cho người dùng. Đây được coi là “đòn bẩy” để ngành công nghiệp ô tô tiến vào giai đoạn cách mạng công nghiệp 4.0. Trong tương lai gần, xe ô tô có khả năng hoàn toàn tự lái, giúp người dùng thư giãn trên mọi hành trình di chuyển.

- Khi sử dụng tính tiền va chạm hệ thống sau khi nghiên cứu thì hoạt động một cách chính xác hơn nhờ sử dụng kiến thức vật lý cơ bản. Tính khoảng cách an toàn (safe distance) dựa vào tốc độ của xe mình (v : vận tốc m/s). Sau đó tính khoảng cách hiện tại (d) với (safe distance) tính thời gian va chạm (collision time) = $(d)/(v)$. Công thức ở chỗ này sử dụng công thức mối liên hệ tốc độ giữa 2 xe (relatite v). Sử dụng chuyển động thẳng biến đổi đều.

Do thời gian nghiên cứu ngắn, lượng kiến thức có giới hạn nên em rất mong ban giám khảo và mọi người tham gia cuộc thi giúp đỡ, hướng dẫn để ứng dụng ngày càng hoàn thiện hơn và có thể áp dụng rộng rãi.

VIII. Nguồn tham khảo

1. Wikipedia, youtube, ...
2. Kho dữ liệu hình ảnh video, Adobe Photoshop, Adobe Illustrator, ...
3. Cử nhân Đại học Sư Phạm Kỹ Thuật- Hồ Chí Minh: Nguyễn Ngọc Hiếu
4. Tiến sĩ: Dũng, lập trình viên.
5. Nhóm cộng đồng facebook **J2team Community**, nhóm **Hội những anh em thích ăn Mì AI**.