HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



CAO MINH QUYỀN

NGHIÊN CỨU HỆ THỐNG THEO DÕI MẮT ĐỂ PHÁT HIỆN TÀI XẾ KHÔNG TỈNH TÁO BẰNG CÁCH SỬ DỤNG DEEP LEARNING

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CAO MINH QUYỀN

NGHIÊN CỨU HỆ THỐNG THEO DÕI MẮT ĐỂ PHÁT HIỆN TÀI XẾ KHÔNG TỈNH TÁO BẰNG CÁCH SỬ DỤNG DEEP LEARNING

Chuyên ngành: Khoa học máy tính

Mã số: 8.48.01.01

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. ĐỖ TIẾN DŨNG

HÀ NỘI - 2025

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

TÁC GIẢ



Cao Minh Quyền

LỜI CẢM ƠN

Tôi xin trân trọng bày tỏ lòng biết ơn sâu sắc đến tập thể các thầy, cô giáo Học viện Công nghệ Bưu chính Viễn thông đã tận tình giảng dạy, truyền đạt kiến thức và tạo điều kiện thuận lợi cho tôi trong suốt quá trình học tập và nghiên cứu tại Học viện. Đặc biệt, tôi xin gửi lời cảm ơn chân thành đến thầy giáo Tiến sĩ Đỗ Tiến Dũng, người đã trực tiếp hướng dẫn tôi hoàn thành đề án tốt nghiệp này. Xin được ghi nhận sự động viên và hỗ trợ từ gia đình, bạn bè và đồng nghiệp trong suốt thời gian thực hiện.

Mặc dù đã cố gắng nhiều song đề án tốt nghiệp khó tránh khỏi những hạn chế nhất định, tôi rất mong nhận được sự góp ý quý báu của quý thầy cô và các nhà nghiên cứu để công trình được hoàn thiện hơn.

Xin trân trọng cảm ơn!

HỌC VIÊN

Cao Minh Quyền

MỤC LỤC

LỜI CAM ĐOAN i
LỜI CẢM ƠN ii
MỤC LỤC iii
DANH MỤC CÁC THUẬT NGỮ, TỪ VIẾT TẮT v
DANH SÁCH BẢNG vi
DANH SÁCH HÌNH VẼ vii
MỞ ĐẦU 1
CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 2
1.1. Giới thiệu đề tài 2
1.2. Lý do chọn đề tài 3
1.3. Mục đích nghiên cứu 4
1.4. Đối tượng nghiên cứu 5
1.5. Phương pháp nghiên cứu 5
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ NGHIÊN CỨU LIÊN QUAN 7
2.1. Khái niệm buồn ngủ và các biểu hiện sinh lý 7
2.2. Các phương pháp phát hiện buồn ngủ hiện có: 8
2.3. Thuật toán PERCLOS 9
2.4. Mạng nơ-ron 10
2.4.1. Các thành phần trong mạng nơ-ron 11
2.5. Mạng nơ-ron tích chập (CNN) 15
2.5.1. Các thành phần của mạng nơ-ron tích chập 17
2.5.2. Kiến trúc mạng nơ-ron tích chập 20
2.5.3. Huấn luyện mạng nơ-ron tích chập 21
2.6. Mạng Nơ-ron Tích chập xếp tầng đa nhiệm (Multi-task Cascaded Convolutional Networks) 22
2.6.1. Mạng Đề xuất (Proposal Network - P-NET) 22
2.6.2. Mạng Tinh chỉnh (Refine Network - R-NET) 23
2.6.3. Mạng Đầu ra (Output Network - O-NET) 24

2.6.4. Kiến trúc của MTCNN 24
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG 26
3.1. Mô hình tổng thể của hệ thống 26
3.2. Mô tả chi tiết từng thành phần 30
3.2.1. Camera – Thiết bị thu nhận hình ảnh thời gian thực 31
3.2.2. MTCNN – Phát hiện khuôn mặt và xác định landmark mắt 31
3.2.3. CNN – Phân loại trạng thái mắt (mở hoặc nhắm) 32
3.2.4. PERCLOS – Tính toán mức độ buồn ngủ 33
CHƯƠNG 4: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG 35
4.1. Tập dữ liệu và Tiền xử lý (Datasets and Preprocessing) 37
4.1.1. Tập dữ liệu (Datasets) 37
4.1.2. Tiền xử lý (Preprocessing) 39
4.2. Mạng Nơ-ron Tích chập để Phát hiện Buồn ngủ 40
4.3. Hệ thống Theo dõi Mắt Phát hiện Buồn ngủ (Drowsiness Detection Tracking Eye System) 43
4.3.1 Phát hiện Khuôn mặt bằng Mạng Nơ-ron Tích chập Xếp tầng Đa nhiệm vụ (MTCNN) 44
4.3.2. Mạng Nơ-ron Tích chập Phát hiện Trạng thái Mắt (Convolutional Neural Networks Detect Eyes Status) 46
4.3.3. PERCLOS và Cảnh báo (PERCLOS and Alert) 46
4.4. Kết quả thí nghiệm 47
KẾT LUẬN 54
TÀI LIỆU THAM KHẢO 56

DANH SÁCH BẢNG

Bảng 4.1: FPS khi chạy thử nghiệm hệ thống. 44
Bảng 4.2: Kiểm tra trường hợp phát hiện buồn ngủ. 45

DANH SÁCH HÌNH VẼ

Hình 2.1: Một đơn vị trong mạng Nơ-ron 12
Hình 2.2: Hàm ReLU 13
Hình 2.3: Hàm Sigmoid 14
Hình 2.4: Các lớp của một mạng nơ-ron 15
Hình 2.5: Bước tích chập: Ảnh với kernel 3x3 và các bản đồ đặc trưng đầu ra 18
Hình 2.6: Max pooling 19
Hình 2.7: Kiến trúc cơ bản của Mạng Nơ-ron Tích chập. 21
Hình 2.8: MTCNN 23
Hình 2.9: Kiến trúc của MTCNN 25
Hình 3.1: Sơ đồ khối hệ thống 30
Hình 4.1: Các bước tổng thể của Bộ dữ liệu và Tiền xử lý 35
Hình 4.2: Các bước tổng thể của mô hình phát hiện buồn ngủ 36
Hình 4.3: Một số ví dụ về tập dữ liệu NTHU-DDD .**Error! Bookmark not defined.**
Hình 4.4: MRL Eye Dataset 38
Hình 4.5: (a) Camera thời gian thực; (b) Video có kích thước 2560x1820 pixel; (c) Tách Khung hình trong Video có kích thước 2560x1820 pixel; (d) Hộp bao quanh khuôn mặt và mắt; (e) Cắt khung bao quanh mắt có kích thước 56x64 pixel. 40
Hình 4.6: Mô hình CNN đào tạo hệ thống theo dõi mắt để phát hiện tình trạng buồn ngủ khi lái xe 41
Hình 4.7: Bộ dữ liệu đầu vào: Mắt mở và mắt đóng cho mô hình 42
Hình 4.8: Ảnh dữ liệu RGB sang Xám 42
Hình 4.9: Lớp đầu tiên: Conv2D 43
Hình 4.10: Tính năng sau khi maxpooling 43
Hình 4.11: Luồng hệ thống 44
Hình 4.12: MTCNN để trích xuất hộp giới hạn của khuôn mặt 45
Hình 4.13: Các điểm mốc khuôn mặt 46
Hình 4.14: Mắt khi MTCNN và Điểm mốc khuôn mặt được trích xuất để dự đoán trạng thái 46
Hình 4.15: Mô hình huấn luyện CNN với GPU trong Google Colab 49
Hình 4.16: Hệ thống dõi mắt để phát hiện buồn ngủ và cảnh báo 50

MỞ ĐẦU

Giấc ngủ đóng vai trò thiết yếu đối với sức khỏe thể chất và tinh thần của con người. Khi chất lượng hoặc thời lượng giấc ngủ không được đảm bảo đầy đủ, một trong những hậu quả dễ nhận thấy nhất là tình trạng mệt mỏi (fatigue) và buồn ngủ (drowsiness) trong các hoạt động đòi hỏi sự tập trung cao như lái xe. Nhiều nghiên cứu đã chỉ ra rằng áp lực công việc, sử dụng thiết bị điện tử (máy tính, điện thoại thông minh,...) kéo dài và thói quen sinh hoạt thiếu khoa học là những nguyên nhân trực tiếp dẫn đến thiếu ngủ (sleep deprivation), từ đó làm tăng đáng kể nguy cơ buồn ngủ khi điều khiển phương tiện giao thông. Trong bối cảnh giao thông hiện đại, buồn ngủ và mất tỉnh táo khi lái xe được xem là một trong những nguyên nhân quan trọng gây ra tai nạn giao thông. Tài xế có thể có các biểu hiện như: mắt nhắm lâu, tần suất chớp mắt tăng, gật đầu, phản xạ chậm với tình huống bất ngờ… nhưng bản thân họ không ý thức được mức độ nguy hiểm của tình trạng này. Những khoảng thời gian mất tập trung chỉ kéo dài vài giây cũng đủ gây ra va chạm với thiệt hại lớn về người và tài sản. Đề án này tập trung nghiên cứu và xây dựng một hệ thống theo dõi mắt nhằm phát hiện tình trạng buồn ngủ của tài xế dựa trên hình ảnh thời gian thực. Hệ thống tận dụng ưu thế của thị giác máy tính và các mô hình học sâu để nhận diện khuôn mặt, xác định vị trí mắt, phân loại trạng thái mắt (mở/nhắm) và cuối cùng tính toán chỉ số PERCLOS để đánh giá mức độ buồn ngủ. Mục tiêu cuối cùng là đưa ra cảnh báo kịp thời, hỗ trợ người lái phòng tránh tai nạn do mất tập trung. Về mặt kỹ thuật, đề tài sử dụng kết hợp mạng nơ-ron tích chập xếp tầng đa nhiệm (Multi-task Cascaded Convolutional Networks – MTCNN) để phát hiện khuôn mặt và vị trí mắt, cùng mạng nơ-ron tích chập (Convolutional Neural Network – CNN) để phân loại trạng thái mắt. Thông tin trạng thái mắt theo chuỗi thời gian được dùng làm đầu vào cho thuật toán PERCLOS, từ đó ước lượng mức độ buồn ngủ và phát cảnh báo phù hợp. Việc khai thác khả năng xử lý của AI trong một bài toán có ý nghĩa xã hội như an toàn giao thông là động lực chính dẫn đến việc lựa chọn và thực hiện đề tài này.

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu đề tài

Trong những năm trở lại đây, cùng với sự phát triển mạnh mẽ của kinh tế và đô thị hóa, mật độ phương tiện giao thông ngày càng tăng cao. Điều này kéo theo áp lực lớn lên người lái xe, đặc biệt là các tài xế đường dài, tài xế xe tải, xe khách liên tỉnh và cả người sử dụng phương tiện giao thông cá nhân trong thời gian dài. Ngoài các yếu tố khách quan như điều kiện đường sá, thời tiết hay chất lượng phương tiện, yếu tố chủ quan từ người lái – đặc biệt là tình trạng buồn ngủ – giữ vai trò then chốt trong nhiều vụ tai nạn nghiêm trọng.

Theo NHTSA (Cục Quản lý An toàn Giao thông Đường cao tốc Quốc gia Mỹ), năm 2017 có hơn 91.000 vụ tai nạn giao thông được ghi nhận có liên quan đến tình trạng buồn ngủ của tài xế, trong đó có nhiều trường hợp gây thương vong. Thực tế có thể cao hơn do khó xác định nguyên nhân trong một số vụ việc. Điều này cho thấy buồn ngủ khi lái xe là một vấn đề an toàn giao thông cần được giám sát và cảnh báo kịp thời.

Cùng lúc đó, trí tuệ nhân tạo (AI) và các nhánh ứng dụng như học sâu (Deep Learning) đang tạo ra bước thay đổi căn bản trong nhiều lĩnh vực: tài chính, sản xuất, y tế, an ninh – giám sát, và cả giao thông thông minh. Các mô hình học sâu, cụ thể là mạng nơ-ron tích chập (CNN), đã chứng minh khả năng vượt trội trong việc xử lý ảnh và video, bao gồm nhận diện khuôn mặt, theo dõi chuyển động mắt và phân loại trạng thái. Việc tận dụng các kỹ thuật này để theo dõi biểu hiện buồn ngủ của tài xế thông qua đôi mắt, tần suất chớp mắt hoặc thời gian nhắm mắt kéo dài đang trở thành một hướng tiếp cận nhiều tiềm năng.

Trong bối cảnh đó, đề tài "Hệ thống theo dõi mắt phát hiện buồn ngủ của người lái xe sử dụng MTCNN, CNN và PERCLOS" được thực hiện với mục tiêu xây dựng một giải pháp có khả năng hoạt động theo thời gian thực, không xâm lấn, chi phí triển khai thấp, nhưng vẫn đảm bảo độ chính xác đủ cao để ứng dụng trong thực tế. Hệ thống này có thể tích hợp vào các nền tảng hiện có trên xe hoặc thiết bị di động, góp phần giảm thiểu tại nạn do buồn ngủ và nâng cao an toàn giao thông.

1.2. Lý do chọn đề tài

Lý do lựa chọn đề tài có thể được xem xét dưới ba khía cạnh chính: khoa học, công nghệ và thực tiễn. Về mặt khoa học, phát hiện buồn ngủ là một bài toán mang tính liên ngành, kết hợp giữa sinh lý học, tâm lý học, xử lý tín hiệu và trí tuệ nhân tạo. Nhiều hướng tiếp cận đã được sử dụng để đánh giá mức độ tỉnh táo của người lái. Nhóm phương pháp dựa trên tín hiệu sinh lý sử dụng các tín hiệu như EEG (điện não đồ), ECG (điện tâm đồ), EOG (điện nhãn đồ) để suy luận trạng thái buồn ngủ [12]. Đặc biệt, EEG được đánh giá là có độ chính xác cao, với độ nhạy có thể vượt 90% trong nhiều nghiên cứu. Tuy nhiên, các phương pháp này cần gắn cảm biến trực tiếp lên cơ thể người lái, gây bất tiện, dễ làm họ khó chịu và khó triển khai rộng rãi trong bối cảnh giao thông thực tế. Nhóm phương pháp dựa trên hành vi điều khiển xe lại theo dõi dao động vô-lăng, độ lệch làn, sự thay đổi tốc độ để suy đoán mức độ mệt mỏi [9]. Mặc dù tận dụng được các cảm biến đã có sẵn trên xe và không xâm lấn, các phương pháp này lại rất nhạy cảm với thời tiết, điều kiện đường sá, và phong cách lái, khiến độ ổn định không cao. Bên cạnh đó, nhóm phương pháp dựa trên thị giác máy tính dùng camera và các thuật toán xử lý hình ảnh để phân tích khuôn mặt, đôi mắt, tư thế đầu và các biểu hiện đặc trưng của buồn ngủ [4]. Đây là hướng tiếp cận không xâm lấn, dễ lắp đặt và phù hợp với xu thế phát triển hệ thống hỗ trợ lái xe tiên tiến.

Về mặt công nghệ, sự tiến bộ nhanh chóng của các mô hình học sâu như CNN, cùng với các kiến trúc chuyên biệt như mạng nơ-ron tích chập xếp tầng đa nhiệm (MTCNN), và sự hỗ trợ mạnh mẽ của các thư viện như TensorFlow, OpenCV, PyTorch, Keras … đã khiến việc xây dựng và triển khai những hệ thống phát hiện buồn ngủ dựa vào thị giác máy tính trở nên dễ dàng hơn [1], [2], [7], [20], [25]. Chỉ số PERCLOS (tỷ lệ phần trăm thời gian mí mắt che phủ đồng tử trên một ngưỡng nhất định trong một khoảng thời gian) đã được công nhận là một chỉ số hữu ích trong việc định lượng mức độ buồn ngủ và được dùng trong nhiều hệ thống thực nghiệm [13], [16]. Tuy nhiên, các hệ thống truyền thống dựa trên PERCLOS thuần túy vẫn còn một

số hạn chế, ví dụ như khó hoạt động trong điều kiện ánh sáng yếu, nhạy với sự khác biệt về hình dạng mắt, hay suy giảm độ chính xác khi tài xế đeo kính [6]. Vì vậy, việc phối hợp PERCLOS với mô hình học sâu CNN và thuật toán phát hiện khuôn mặt hiện đại như MTCNN sẽ nâng cao tính chính xác và độ tin cậy của hệ thống.

Về mặt thực tiễn, tại Việt Nam và các quốc gia đang phát triển khác, tai nạn giao thông đang là vấn đề đáng lo ngại khi số vụ tai nạn do buồn ngủ và mất tập trung của tài xế vẫn chiếm tỉ lệ đáng kể. Một hệ thống phát hiện buồn ngủ với chi phí phù hợp, tận dụng thiết bị thông dụng như camera và máy tính nhúng, có thể tích hợp vào xe khách đường dài, xe tải, taxi công nghệ hay thậm chí xe cá nhân, sẽ có ý nghĩa rất lớn. Các yếu tố trên trở thành cơ sở để lựa chọn và phát triển đề tài "Hệ thống theo dõi mắt phát hiện buồn ngủ của người lái xe sử dụng MTCNN, CNN và PERCLOS".

1.3. Mục đích nghiên cứu

Đề tài được triển khai nhằm xây dựng một hệ thống hoàn chỉnh, có khả năng giám sát trạng thái mắt người lái xe và phát hiện dấu hiệu buồn ngủ dựa trên hình ảnh thu từ camera, qua đó phát cảnh báo kịp thời để góp phần nâng cao an toàn giao thông. Từ mục tiêu trên, đề tài đặt ra một số mục tiêu cụ thể hơn. Thứ nhất, hệ thống phải theo dõi được trạng thái mắt bằng thị giác máy tính, tức là xây dựng được một pipeline thu nhận hình ảnh thời gian thực từ camera gắn trong xe, sử dụng MTCNN để phát hiện khuôn mặt và tìm chính xác các điểm mốc trên khuôn mặt, nhất là vị trí mắt, tiếp theo trích xuất vùng ảnh mắt làm đầu vào cho các bước xử lý sau [25]. Thứ hai, hệ thống cần phân loại được trạng thái mắt bằng mô hình học sâu, cụ thể là thiết kế và huấn luyện một mô hình CNN có thể nhận dạng mắt mở hoặc mắt nhắm trên bộ dữ liệu đã gán nhãn, đồng thời đảm bảo mô hình đạt hiệu năng tốt và ổn định trong nhiều điều kiện khác nhau.

Bên cạnh đó, đề tài hướng đến việc phát triển một module tính toán chỉ số PERCLOS dựa trên chuỗi kết quả phân loại mắt theo từng khung hình. Thông qua việc đặt ra các ngưỡng PERCLOS, hệ thống sẽ có thể phân loại các mức độ buồn ngủ khác nhau và phát ra các mức cảnh báo phù hợp. Một mục tiêu quan trọng khác là tối ưu hệ thống để có thể triển khai trên các nền tảng phần cứng phổ thông, chẳng hạn

như CPU thông thường hoặc các thiết bị máy tính nhúng, nhằm đảm bảo hệ thống có thể hoạt động gần thời gian thực với độ trễ thấp. Cuối cùng, đề tài cũng đặt mục tiêu cải thiện độ chính xác và khả năng thích nghi của hệ thống bằng cách khảo sát thêm các đặc trưng như hình dáng mắt, tư thế đầu, khoảng cách tương đối giữa mắt và các bộ phận khác trên khuôn mặt, cũng như huấn luyện và kiểm thử trên các bộ dữ liệu đa dạng để nâng cao tính linh hoạt trong môi trường thực tế.

1.4. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính của đề tài là hành vi thị giác của tài xế, được thể hiện chủ yếu thông qua trạng thái mắt (mở hoặc nhắm) và sự thay đổi của trạng thái mắt theo thời gian trong quá trình lái xe. Hình ảnh khuôn mặt tài xế được thu thập từ camera gắn trong xe, sau đó qua bước phát hiện khuôn mặt và đánh dấu các mốc bằng MTCNN để tìm ra chính xác vị trí mắt trái và mắt phải. Dựa trên các mốc vị trí này, vùng mắt sẽ được trích xuất và chuẩn hóa, đóng vai trò là đầu vào cho mô hình CNN thực hiện nhiệm vụ phân loại trạng thái mắt. Kết quả phân loại tại từng khung hình tạo thành một chuỗi trạng thái mắt theo thời gian, là cơ sở để tính toán chỉ số PERCLOS – tỷ lệ phần trăm thời gian mắt nhắm trong một khoảng thời gian nhất định [16].

Ngoài khía cạnh nhận dạng trạng thái mắt, đề tài cũng xem xét một số tác nhân gây ảnh hưởng đến hiệu năng của hệ thống như điều kiện ánh sáng bên trong xe, sự khác nhau về hình dáng mắt giữa các cá nhân, hoặc trường hợp tài xế đeo kính thường hay kính râm. Đối tượng kỹ thuật của đề tài bao gồm kiến trúc MTCNN dùng để phát hiện khuôn mặt và đánh dấu các mốc trên khuôn mặt, mô hình CNN dùng để phân loại trạng thái mắt, thuật toán PERCLOS dùng để đánh giá mức độ buồn ngủ, tập dữ liệu mắt thu thập từ sinh viên và từ MRL Eye Dataset [5], cùng các công cụ hỗ trợ như OpenCV và các framework học sâu như TensorFlow hay PyTorch.

1.5. Phương pháp nghiên cứu

Đề tài sử dụng phương pháp tiếp cận tổng hợp, kết hợp cơ sở lý thuyết vớ thực nghiệm trên dữ liệu và triển khai hệ thống để đánh giá. Trước hết, tiến hành khảo sát và nghiên cứu các tài liệu liên quan đến sự mất tỉnh táo khi lái xe, từ các biểu hiện

sinh lý như EEG, ECG, EOG cho đến các hành vi điều khiển xe như dao động vô-lăng, lệch làn đường, tốc độ và chế độ tăng/giảm tốc [9], [12], [16]. Song song đó là việc tìm hiểu lý thuyết về mạng nơ-ron, mạng nơ-ron tích chập, kiến trúc MTCNN, cùng những khái niệm, thuật toán tối ưu và hàm mất mát liên quan [1], [2], [7], [18], [20], [25]. Từ cơ sở trên, đề tài chọn các công cụ và thư viện phù hợp như OpenCV, TensorFlow, Keras, PyTorch để hỗ trợ xây dựng mô hình và hệ thống.

Tiếp theo, đề tài thu thập và xử lý dữ liệu bằng cách ghi hình các đối tượng trong các trạng thái mắt mở và nhắm, kết hợp với việc sử dụng MRL Eye Dataset [5] và một phần dữ liệu thu thập từ Internet. Dữ liệu được xử lý qua các bước phát hiện khuôn mặt, cắt vùng mắt, chuẩn hóa kích thước, chuyển sang ảnh xám và chuẩn hóa giá trị điểm ảnh. Trên cơ sở dữ liệu đã chuẩn hóa và gán nhãn, mô hình CNN được thiết kế, huấn luyện và đánh giá bằng các thông số như độ chính xác, hàm mất mát, trong đó việc tận dụng môi trường GPU (Google Colab) giúp giảm thời gian huấn luyện. Sau khi mô hình đạt độ tin cậy cần thiết, toàn bộ pipeline hệ thống gồm camera, MTCNN, CNN và PERCLOS được tích hợp và triển khai thử nghiệm trên phần cứng thực tế, đo đạc các chỉ số như FPS, độ trễ xử lý và độ chính xác phát hiện buồn ngủ trong những kịch bản khác nhau. Cuối cùng, tác giả phân tích kết quả, chỉ ra các hạn chế, từ đó đưa ra các hướng phát triển kế tiếp cho hệ thống.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ NGHIÊN CỨU LIÊN QUAN

2.1. Khái niệm buồn ngủ và các biểu hiện sinh lý

Buồn ngủ (drowsiness) là một trạng thái sinh lý khi đó mức độ tỉnh táo và tập trung của con người bị suy giảm rõ rệt. Trạng thái này thường xuất hiện khi cơ thể thiếu ngủ, làm việc quá sức, chịu áp lực tâm lý kéo dài hoặc tham gia các hoạt động đơn điệu trong thời gian dài, chẳng hạn như lái xe trên quãng đường thẳng, ít giao cắt và ít kích thích thị giác. Về mặt chủ quan, người lái có thể cảm nhận cảm giác nặng mí mắt, khó tập trung, phản ứng chậm với những tác động từ môi trường. Mặc dù vậy, họ không phải lúc nào cũng ý thức được mức độ nguy hiểm của tình trạng này, đặc biệt khi buồn ngủ đến từ từ và kéo dài.

Về mặt biểu hiện sinh lý và hành vi, buồn ngủ thể hiện khá rõ qua đôi mắt và khuôn mặt. Tần suất chớp mắt có xu hướng tăng lên, trong khi thời gian nhắm mắt mỗi lần cũng kéo dài hơn bình thường; những nhịp "nhắm mắt" bất thường này thường được gọi là long blink. Độ mở mí mắt giảm dần, mắt trở nên lờ đờ, khó giữ được trạng thái mở hoàn toàn trong thời gian dài. Thêm vào đó, người buồn ngủ thường xuất hiện các khoảng "ngủ gật chớp nhoáng" (micro-sleep) kéo dài từ vài phần của giây đến vài giây, trong thời gian đó họ gần như không còn khả năng nhận biết môi trường xung quanh. Về mặt vận động, đầu có xu hướng gật hoặc gục xuống, tư thế thay đổi chậm chạp, nét mặt trở nên ít biểu cảm hơn.

Nếu quan sát dưới góc độ tín hiệu sinh lý, các nghiên cứu cho thấy buồn ngủ gắn liền với sự thay đổi của nhiều tín hiệu như EEG, ECG, EOG. Sóng não EEG trong trạng thái buồn ngủ có xu hướng giảm hoạt động ở dải tần số beta (liên quan đến tỉnh táo) và tăng hoạt động ở dải theta, delta (liên quan đến buồn ngủ và ngủ sâu) [12]. Nhịp tim và HRV (biến thiên nhịp tim) cũng thay đổi theo hướng giảm hoạt hóa giao cảm, tăng hoạt hóa phó giao cảm. Tín hiệu EOG phản ánh rõ các chuyển động mắt chậm, thất thường. Về mặt hành vi lái xe, người buồn ngủ điều khiển vô-lăng kém ổn định, phương tiện dễ lệch khỏi làn, tốc độ thay đổi bất thường, thời gian phản ứng với chướng ngại vật hoặc biển báo bị tăng lên. Tất cả những biểu hiện này cho

thấy buồn ngủ không chỉ là tình trạng mệt mỏi thông thường mà là một nguy cơ nghiêm trọng đối với an toàn giao thông.

2.2. Các phương pháp phát hiện buồn ngủ hiện có:

Các phương thức nhận biết buồn ngủ được biết đến có thể chia thành ba nhóm chính: dựa trên tín hiệu sinh lý, dựa trên hành vi điều khiển xe và dựa trên thị giác máy tính. Mỗi nhóm có những ưu và nhược điểm riêng, và việc lựa chọn hướng tiếp cận nào cần dựa trên yêu cầu ứng dụng, mức độ chấp nhận thiết bị đo đạc cũng như chi phí triển khai.

Nhóm phương pháp dựa trên tín hiệu sinh lý thường sử dụng các cảm biến để đo EEG, ECG, EOG nhằm đánh giá trực tiếp hoạt động của não bộ và hệ tim mạch [12]. EEG được xem là chuẩn vàng trong nghiên cứu về các giai đoạn ngủ do phản ánh trực tiếp các dao động điện trong não. Nhiều nghiên cứu cho thấy khả năng nhận biết buồn ngủ với độ chính xác cao khi phân tích đặc trưng phổ và động học của tín hiệu EEG. Tuy vậy, nhược điểm của nhóm này là tính xâm lấn: việc gắn điện cực lên đầu, ngực hoặc quanh mắt gây khó chịu, làm giảm sự thoải mái của người lái, khó áp dụng trong bối cảnh giao thông thực tế. Ngoài ra, hệ thống đo tín hiệu sinh lý thường đòi hỏi thiết bị chuyên biệt, chi phí lớn và quy trình lắp đặt, hiệu chuẩn nhiều bước.

Nhóm phương pháp thứ hai dựa trên hành vi điều khiển xe. Trong các hệ thống này, những thông số như góc quay vô-lăng, biên độ dao động tay lái, độ lệch làn đường, tốc độ và gia tốc được theo dõi liên tục để suy luận trạng thái mệt mỏi của người lái [9]. Khi tài xế buồn ngủ, khả năng duy trì ổn định tay lái suy giảm, xe dễ lệch khỏi vị trí trung tâm làn, phản ứng chậm với các tình huống bất ngờ. Ưu điểm của hướng tiếp cận này là tận dụng được các cảm biến sẵn có trên phương tiện, không cần gắn thêm thiết bị lên người lái. Tuy nhiên, kết quả bị ảnh hưởng mạnh bởi điều kiện thời tiết, đường xá, loại xe và phong cách lái của từng cá nhân, do đó khả năng tổng quát hóa và độ ổn định còn hạn chế.

Nhóm phương pháp thứ ba dựa trên thị giác máy tính sử dụng camera gắn trong cabin để ghi lại khuôn mặt và vùng mắt của người lái, sau đó áp dụng các mô

hình học sâu và thuật toán xủa lý ảnh để phân tích [4]. Các đặc trưng thường được khai thác gồm khoảng cách giữa hai mí mắt, tần suất chớp mắt, thời gian mắt nhắm liên tục, tư thế đầu và hướng nhìn. Ưu điểm cảu phương pháp này là không xâm lấn, lắp đặt đơn giản, có thể dùng các thiết bị giá thành vừa phải. Nhược điểm là bị tác động bởi điều kiện ánh sáng, góc đặt camera, sự che khuất (kính, mũ…) và sự khác biệt hình thái khuôn mặt giữa các đối tượng khác nhau. Tuy vậy, với sự tiến bộ của các mô hình học sâu chẳng hạn là CNN và những kiến trúc phát hiện khuôn mặt hiện đại, nhiều hạn chế đã dần được khắc phục, khiến hướng tiếp cận thị giác máy tính trở thành lựa chọn hấp dẫn cho bài toán phát hiện buồn ngủ trong thực tế.

2.3. Thuật toán PERCLOS

Wierwille et al. [13] đã phát triển PERCLOS như một chỉ số đo lường tình trạng buồn ngủ. PERCLOSURE được định nghĩa là tỷ lệ phần trăm thời gian mà mí mắt khép lại qua đồng tử ít nhất 80%. Các giá trị PERCLOS có thể được tính bằng cách sử dụng chuỗi video liên tục của hình ảnh mắt [6]. Để tính toán được chỉ số PERCLOS, nhiều phép đo phải được hoàn thành. Đầu tiên phải nhận dạng được khuôn mặt trong khung hình, sau đó là xác định và trích xuất hình ảnh mắt. Tiếp theo sẽ xác định mắt đang ở trạng thái mở hay nhắm. Từ chuỗi kết quả phân loại, trong một khoảng thời gian xác định gồm E_t lần đếm mắt, hệ thống đếm được E_c số lần nhắm mắt. Từ đó giá trị PERCLOS được tính như sau:

$$\text{Công thức: PERCLOS} = \frac{E_c}{E_t} \tag{2.3}$$

Ví dụ: giả sử 100 khung hình ảnh khuôn mặt được ghi lại và 20 trường hợp trong đó mắt được phát hiện nhắm (ít nhất 80% về mặt không gian). Trong trường hợp đó, giá trị PERCLOS là 20%. Mắt được phân loại là mở hoặc nhắm trong quá trình huấn luyện dựa trên tiêu chí khép 80% về mặt không gian.

Nếu giá trị PERCLOS càng cao thì thời gian mắt ở trạng thái nhắm càng nhiều, từ đó mức độ buồn ngủ càng nghiêm trọng. Nhiều thử nghiệm thực tế cho thấy khi PERCLOS vượt qua một ngưỡng nhất định, nguy cơ xảy ra sai sót, phản ứng chậm, hoặc ngủ gật trong khi lái xe tăng mạnh [6], [16].

Mặc dù là một chỉ số hiệu quả, PERCLOS vẫn đối diện với một số thử thách trong ứng dụng thực tế. Khi tài xế đeo kính, đặc biệt là kính râm đậm màu, việc xác định mức độ che phủ đồng tử gặp nhiều trở ngại, thậm chí bất khả thi nếu camera không có bước sóng phù hợp. Ngoài ra, khi người lái quay đầu, nhìn gương chiếu hậu hoặc nhìn sang hai bên, vùng mắt có thể bị che khuất hoặc nằm ngoài tầm hoạt động của camera, khiến dữ liệu đầu vào bị thiếu. Các chuyển động tự nhiên của đầu và mắt, cùng với sự thay đổi của điều kiện ánh sáng, dễ làm thuật toán nhận diện sai trạng thái mắt. Do đó, trong nhiều hệ thống, PERCLOS thường được phối hợp với các đặc điểm khác (chẳng hạn như tần suất chớp mắt, tư thế đầu, hướng nhìn) và tận dụng các mô hình học sâu nhằm nâng cao độ bền vững và độ chính xác.

2.4. Mạng nơ-ron

Trước khi đi vào kiến trúc đặc thù như mạng nơ-ron tích chập (CNN), cần nhìn nhận tổng quan về mạng nơ-ron nhân tạo như một mô hình xấp xỉ chức năng cho hoạt động của bộ não. Lấy ý tưởng từ hệ thần kinh sinh học, mạng nơ-ron nhân tạo dựa theo cách các nơ-ron trong não bộ nhận tín hiệu, tổng hợp kích thích và đưa ra quyết định có "bắn" xung hay không. Trong não người, mỗi nơ-ron sinh học nhận tín hiệu từ hàng nghìn nơ-ron khác qua synapse, tích lũy ảnh hưởng của các kích thích này và chỉ phát xung khi mức kích thích vượt qua một ngưỡng nhất định. Cách thức hoạt động này được trừu tượng hóa thành các nơ-ron nhân tạo với trọng số, hàm kích hoạt và độ lệch, cho phép mạng nắm bắt được những mỗi liên hệ phức tạp giữa đầu vào và đầu ra.

Lịch sử mạng nơ-ron bắt đầu với mô hình nơ-ron logic của McCulloch–Pitts, tiếp đến là perceptron đơn lớp, rồi mạng truyền thẳng nhiều lớp. Qua thời gian, mạng nơ-ron dần được mở rộng và tinh chỉnh, trở thành nền tảng của nhiều hệ thống thông minh hiện đại, từ nhận dạng ảnh, nhận dạng tiếng nói cho tới dịch máy. Các ứng dụng thực tế như hệ thống dịch đa ngôn ngữ, trợ lý ảo hay hệ thống gợi ý đều khai thác các kiến trúc mạng nơ-ron sâu. Mặc dù không mô phỏng chính xác mọi chi tiết sinh học của não bộ, mạng nơ-ron nhân tạo vẫn thể hiện khả năng xấp xỉ tốt nhiều hàm phi

tuyến phức tạp, đủ để giải quyết hiệu quả các bài toán phân loại và nhận dạng trong thực tế.

2.4.1. Các thành phần trong mạng nơ-ron

Một mạng nơ-ron nhân tạo được hình thành từ các đơn vị tính toán (units), lấy ý tưởng từ nơ-ron sinh học. Mỗi đơn vị nhận các tín hiệu đầu vào có thể đến từ dữ liệu bên ngoài hoặc từ đầu ra của các đơn vị ở lớp trước. Các đơn vị này được kết nối với nhau thông qua các liên kết (connections), mỗi liên kết mang một trọng số (weight) thể hiện mức ảnh hưởng của đầu vào tương ứng lên đầu ra. Trọng số càng lớn thì đóng góp của đầu vào đó vào kết quả càng mạnh; trọng số gần 0 thể hiện ảnh hưởng không đáng kể. Bên cạnh trọng số, mỗi đơn vị còn có một độ lệch (bias). Bias là một hằng số được cộng thêm vào tổng trọng số trước khi đưa qua hàm kích hoạt. Về mặt hình học, bias giúp dịch chuyển mặt phẳng quyết định, cho phép nơ-ron kích hoạt ngay cả khi tổng đầu vào bằng 0. Nói cách khác, bias đóng vai trò như một "ngưỡng" linh hoạt, giúp mạng có thêm tự do trong việc điều chỉnh biên phân tách giữa các lớp dữ liệu.

Dưới góc độ toán học, quá trình xử lý tại một nơ-ron có thể được diễn tả như sau. Giả sử x là vector đầu vào, W là vector trọng số tương ứng và b là độ lệch, giá trị hoạt hóa tuyến tính a được tính theo công thức:

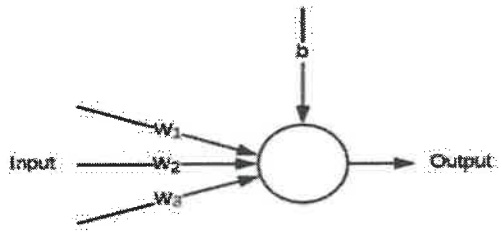
$$a = W_x^T + b \qquad (2.1)$$

W_x^T là ma trận chuyển vị của trọng số. Giá trị a này chưa phải là đầu ra cuối cùng; ở bước tiếp theo, a được truyền qua một hàm kích hoạt phi tuyến tạo ra đầu ra y. Việc chèn hàm kích hoạt vào sau bước cộng tuyến tính là chìa khóa giúp mạng nơ-ron có khả năng biểu diễn các quan hệ phi tuyến giữa đầu vào và đầu ra.

Các nơ-ron được tổ chức thành từng lớp: lớp đầu vào (input layer) nhận dữ liệu thô, một hoặc nhiều lớp ẩn (hidden layers) đảm nhiệm trích xuất và biến đổi đặc trưng, và lớp đầu ra (output layer) sinh ra kết quả cuối cùng. Khi số lượng lớp ẩn đủ lớn, mạng được gọi là mạng sâu. Độ sâu của mạng giúp mô hình học các đặc trưng từ mức đơn giản đến phức tạp, là nền tảng cho các hệ thống học sâu trong thực tế.

Giá trị chúng ta nhận được sau quá trình này chính là đầu ra (output) của đơn vị. (Hình 2.1)

Hàm kích hoạt là một phần không thể thiếu trong từng rơ-ron, quyết định cách giá trị hoạt hóa tuyến tính a được biến đổi thành đầu ra y. Nếu bỏ qua hàm kích hoạt hoặc chỉ sử dụng các hàm tuyến tính, toàn bộ nơ-ron dù có nhiều lớp cũng vẫn chỉ như một phép biến đổi tuyến tính của đầu vào, không đủ khả năng biểu diễn các quan hệ phi tuyến phức tạp trong dữ liệu. Do đó, các hàm kích hoạt phi tuyến được đưa vào để giúp mạng học tốt hơn các mô hình ẩn trong dữ liệu.

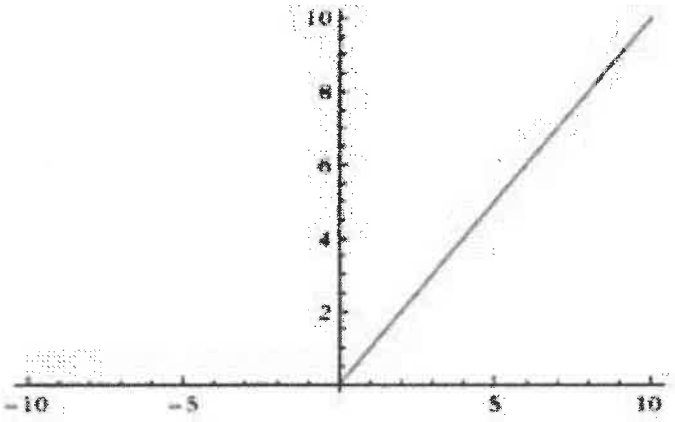


Hình 2.1: Một đơn vị trong mạng Nơ-ron

Về cơ bản, các hàm kích hoạt được chia thành hai nhóm: tuyến tính và phi tuyến. Hàm tuyến tính hiếm khi được dùng trong các lớp ẩn vì không tạo ra thêm sức biểu diễn cho mạng. Ngược lại, các hàm phi tuyến như ReLU, Tanh hay Sigmoid được sử dụng rộng rãi, đặc biệt trong các mô hình học sâu. Hàm kích hoạt cũng góp phần quyết định miền giá trị đầu ra của nơ-ron, chẳng hạn (0,1) đối với sigmoid hoặc $(-\infty, \infty)$ đối với ReLU, từ đó ảnh hưởng đến ý nghĩa diễn giải và sự ổn định của tiến trình huấn luyện. Trong các mạng nơ-ron tích chập hiện đại ReLU là một hàm rất thông dụng, đặc biệt là trong các kiến trúc CNN. Định nghĩa đơn giản của ReLU như sau:

$$A(x) = \max(0, x) \qquad (2.2)$$

Theo đó, khi đầu vào $x<0$, đầu ra bằng 0; khi $x\geq 0$, đầu ra bằng chính giá trị x. Hình dạng của hàm ReLU vì thế rất đơn giản: bằng 0 ở nửa trục âm và là một đường thẳng qua gốc ở nửa trục dương. Chính sự đơn giản này giúp ReLU ít tốn chi phí tính toán hơn so với các hàm kích hoạt khác như tanh hay sigmoid, rất phù hợp cho các mô hình sâu với hàng triệu tham số.

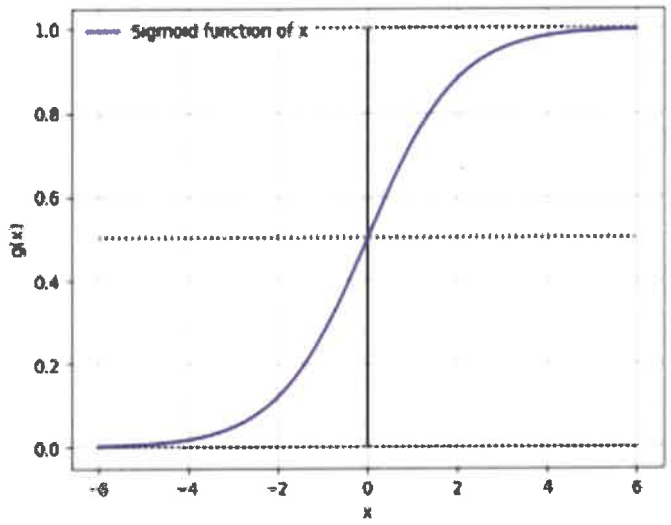


Hình 2.2: Hàm ReLU

Dưới góc độ tối ưu, ReLU khắc phục phần nào hiện tượng bão hòa gradient – khi các hàm sigmoid hoặc tanh cho gradient rất nhỏ ở những vùng giá trị lớn về độ lớn tuyệt đối, khiến quá trình lan truyền ngược trở nên chậm hoặc "đóng băng". Nhờ giữ gradient ổn định ở vùng dương, ReLU giúp các mạng sâu hội tụ nhanh hơn và đạt hiệu năng tốt hơn trong thực nghiệm. Ngoài ra, ReLU còn tạo ra tính thưa (sparsity) trong các kích hoạt vì nhiều nơ-ron cho đầu ra 0, góp phần giảm hiện tượng quá khớp và tiết kiệm tài nguyên tính toán. Vì vậy, trong đa số ứng dụng thực tế, lợi ích vượt trội của ReLU khiến nó trở thành lựa chọn ưu tiên trong nhiều kiến trúc mạng CNN.

Bên cạnh ReLU, hàm sigmoid cũng là một hàm kích hoạt kinh điển, thường xuất hiện trong lớp đầu ra của các bài toán phân loại nhị phân. Hàm Sigmoid thường dùng để chỉ hàm logistic, gọi là *logistic sigmoid function*. Đặc trưng của các hàm sigmoid là ánh xạ toàn bộ trục số thực vào một khoảng hữu hạn, điển hình như (0, 1) hoặc (−1, 1). Nhờ đặc tính này, chúng thường được dùng với mục đích chuyển đổi một giá trị thực thành một giá trị có thể diễn giải như xác suất. Trong số các hàm sigmoid, hàm logistic là dạng được ứng dụng phổ biến nhất, với tác dụng ánh xạ mọi giá trị đầu vào sang khoảng (0, 1). Hình dạng cong hình chữ S khiến nó thường được lựa chọn trong các mô hình học sâu. Các hàm sigmoid được xem là tương đồng với cơ chế kích hoạt của nơ-ron sinh học, do đó được ứng dụng nhiều trong mạng nơ-ron. Ngoài vai trò là hàm kích hoạt trong các lớp ẩn, sigmoid còn hữu ích trong nhiều thuật toán học máy khác, đặc biệt khi cần chuyển đổi một đại lượng thành giá trị xác

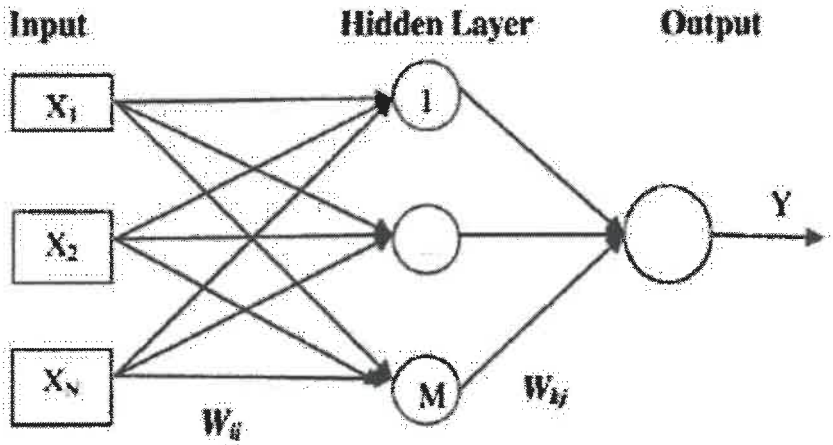
suất. Khi được sử dụng ở lớp đầu ra của mô hình, hàm sigmoid giúp chuyển đổi kết quả dự đoán thành một điểm xác suất (likelihood score) dễ diễn giải và thuận lợi cho quá trình ra quyết định.



Hình 2.3: Hàm Sigmoid

Các hàm Sigmoid là một phần quan trọng của mô hình hồi quy logistic. Hồi quy logistic là một sự điều chỉnh của hồi quy tuyến tính (linear regression) dành cho bài toán phân loại hai lớp (two-class classification). Nó chuyển đổi một hoặc nhiều đầu vào giá trị thực thành một xác suất, chẳng hạn như xác suất một người sẽ mua một vật phẩm. Giai đoạn cuối cùng của mô hình hồi quy logistic thường được thiết lập là hàm logistic (logistic function), cho phép mô hình xuất ra một giá trị xác suất.

Các đơn vị (units) trong mạng nơ-ron được sắp xếp thành các nhóm gọi là *lớp* (layers). Trong mỗi lớp, các đơn vị không kết nối trực tiếp với nhau; thay vào đó, chúng chỉ liên kết với các đơn vị thuộc lớp trước và lớp sau. Một mạng nơ-ron thông thường bao gồm ba loại lớp (Hình 2.4): lớp đầu vào (input layer), lớp ẩn (hidden layers) và lớp đầu ra (output layer). Các lớp trong mạng được sắp xếp theo thứ tự: lớp đầu vào, các lớp ẩn, và lớp đầu ra. Số lượng lớp trong một mạng được ký hiệu là L, được tính bằng cách cộng tổng số lượng lớp ẩn và số lượng lớp đầu ra. Lớp đầu tiên, hay lớp đầu vào, được ký hiệu là lớp thứ 0 (0 layer).



Hình 2.4: Các lớp của một mạng nơ-ron

Nếu trong mạng số lớp ẩn nhiều hơn một thì nó được gọi là mạng nơ-ron đa lớp. Sự kết nối đầy đủ (fully connection) giữa các lớp có nghĩa là mọi nơ-ron trong lớp này kết nối với mọi nơ-ron trong lớp kia. Trong một mạng nơ-ron truyền thẳng, ngoại trừ lớp đầu vào, các lớp nhận đầu vào từ đầu ra của lớp liền kề trước đó. Mỗi đơn vị trong một lớp gửi đầu ra của nó đến lớp tiếp theo, và đầu ra này sẽ được biến đổi thông qua các phép toán và trở thành đầu vào của lớp tiếp theo. Đầu ra của các lớp đầu ra cũng chính là đầu ra của mô hình (output of the models), tức là các giá trị dự đoán (predicted values). Đầu ra của các lớp ẩn (hidden layers) thì không được hiển thị. Do đó các lớp này được gọi là lớp "ẩn". Tốc độ học là một siêu tham số (hyperparameter) quy định mức độ thay đổi của thuật toán để phản hồi lại lỗi dự kiến mỗi khi các trọng số của mô hình được điều chỉnh. Việc chọn tốc độ học là khó khăn vì một giá trị quá nhỏ có thể dẫn đến một giai đoạn huấn luyện kéo dài và có thể bị mắc kẹt, trong khi một giá trị quá cao có thể dẫn đến việc tiếp nhận một phạm vi trọng số không hiệu quả quá nhanh hoặc gây ra một quy trình huấn luyện không thể đoán trước. Khi cấu hình mạng nơ-ron, tốc độ học (learning rate) là một trong những siêu tham số có ảnh hưởng lớn nhất đến tiền trình huấn luyện. Do đó, việc nắm bắt cách tốc độ học ảnh hưởng đến chất lượng mô hình và hiểu được động lực mà nó tạo ra đối với quá trình tối ưu hóa là yếu tố then chốt để đạt được hiệu quả huấn luyện mong muốn.

2.5. Mạng nơ-ron tích chập (CNN)

Trí tuệ nhân tạo (Artificial Intelligence – AI) và Học sâu (Deep Learning) đã phát triển mạnh mẽ trong nhiều lĩnh vực, trong đó nhận diện hình ảnh là một trong những hướng ứng dụng quan trọng. Mạng nơ-ron truyền thẳng nhiều lớp (feedforward neural network) là mô hình nền tảng, gồm lớp đầu vào, lớp đầu ra và một số lớp ẩn nằm ở giữa. Tuy nhiên, khi áp dụng trực tiếp các kiến trúc này cho dữ liệu ảnh dạng lưới hai chiều, số lượng tham số tăng lên rất lớn, khó huấn luyện và dễ dẫn đến quá khớp.

Mạng nơ-ron tích chập (Convolutional Neural Network – CNN) là một kiến trúc học sâu được đề xuất nhằm khắc phục hạn chế đó. CNN dùng cấu trúc không gian của ảnh thông qua các kết nối cục bộ và phương thức chia sẻ trọng số, cho phép trích xuất các đặc điểm quan trọng từ dữ liệu thị giác với chi phí tính toán hợp lý [7]. Một ConvNet yêu cầu bước tiền xử lý thủ công ít hơn nhiều so với các phương pháp trước đây; các bộ lọc (filters) và đặc trưng được mô hình tự động học thông qua quá trình huấn luyện, thay vì phải định nghĩa thủ công.

Kiến trúc của CNN có nhiều điểm tương đồng với cơ chế xử lý thị giác trong vỏ não người. Các lớp ở gần đầu vào học những đặc trưng đơn giản (cạnh, góc, độ tương phản), còn các lớp sâu hơn học các đặc trưng phức tạp hơn (hình dạng, cấu trúc vùng). Thay vì kết nối đầy đủ mọi nơ-ron giữa hai lớp liên tiếp, CNN chỉ sử dụng các kết nối cục bộ trên những vùng nhỏ của ảnh, giống như các "trường tiếp nhận" (receptive fields) trong hệ thị giác sinh học. Cách tổ chức này giúp số lượng tham số giảm đáng kể, đồng thời vẫn đảm bảo khả năng biểu diễn mạnh.

Thông qua việc áp dụng các bộ lọc tích chập thích hợp, một CNN có thể được huấn luyện để nhận biết nhiều loại đối tượng khác nhau trong ảnh, phân biệt chúng dựa trên các mẫu đặc trưng về kết cấu và hình dạng. Nói cách khác, CNN không chỉ đơn thuần là một mạng nơ-ron "đa lớp" mà là một kiến trúc được thiết kế chuyên biệt cho dữ liệu ảnh, tích hợp sẵn các giả định mạnh mẽ về tính cục bộ và tính bất biến dịch chuyển. Phần này trình bày các thành phần chính trong một mạng nơ-ron tích chập, kiến trúc tổng thể và quá trình huấn luyện CNN trong bối cảnh bài toán phân loại hình ảnh và phát hiện tình trạng buồn ngủ.

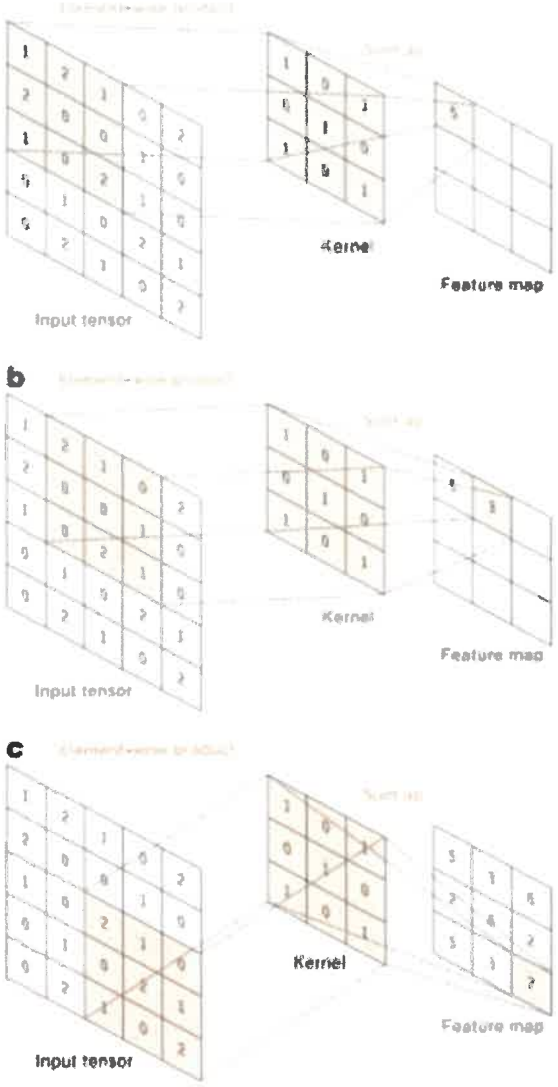
2.5.1. Các thành phần của mạng nơ-ron tích chập

Về tổng thể, một mạng nơ-ron tích chập tiêu chuẩn thường bao gồm ba nhóm lớp chính: Lớp tích chập (Convolution layer – Conv layer), Các lớp gộp (Pooling layer), Các lớp kết nối đầy đủ (Fully Connected layer).

Những lớp này được sắp xếp thành nhiều tầng nối tiếp nhau, tạo thành một kiến trúc sâu có khả năng học biểu diễn phân cấp của dữ liệu: các lớp ở gần đầu vào tập trung vào những đặc trưng cục bộ đơn giản, còn các lớp ở giai đoạn sau tổng hợp các đặc trưng này thành những biểu diễn trừu tượng hơn.

2.5.1.1. Lớp tích chập (Conv layer)

Tích chập (Convolution) là một phép toán tuyến tính chuyên biệt được sử dụng để trích xuất đặc trưng trong dữ liệu. Quá trình này được thực hiện bằng cách áp dụng một ma trận nhỏ gọi là hạt nhân (kernel) lên dữ liệu vốn là một mảng số (tensor)[24] đầu vào. Tại mỗi vị trí, kernel được "trượt" qua vùng tương ứng của tensor và thực hiện phép nhân từng phần tử (element-wise multiplication); giá trị thu được sau đó được cộng lại để tạo nên một phần tử của tensor đầu ra, hay còn gọi là bản đồ đặc trưng (feature map). Quy trình này được lặp lại với nhiều kernel khác nhau để tạo ra những bản đồ đặc trưng, mỗi bản đồ thể hiện một kiểu đặc trưng riêng của dữ liệu đầu vào. Vì vậy, mỗi kernel có thể được xem như một bộ trích xuất đặc trưng độc lập. Kích thước và số lượng kernel là hai siêu tham số quan trọng trong lớp tích chập. Kích thước kernel phổ biến nhất là 3×3, nhưng đôi khi cũng có thể là 5×5 hoặc 7×7 tùy thuộc vào yêu cầu mô hình. Số lượng kernel hoàn toàn do người thiết kế lựa chọn và quyết định độ sâu của tập bản đồ đặc trưng đầu ra. (Hình 2.5)



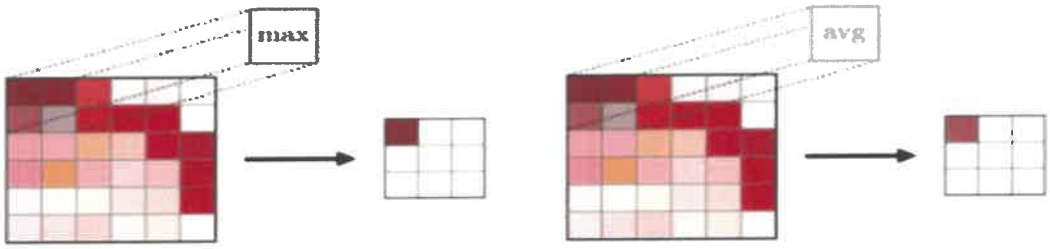
Hình 2.5: Bước tích chập: Ảnh với kernel 3x3 và các bản đồ đặc trưng đầu ra

Một khi ảnh đầu vào được đưa vào CNN, dữ liệu lần lượt đi qua các lớp để trích xuất đặc trưng và suy ra nhãn ở đầu ra. Trong CNN, mỗi nơ-ron chỉ kết nối với một vùng cục bộ của lớp trước (receptive field), giúp học đặc trưng địa phương. Các bộ lọc (filter) được chia sẻ trọng số trên toàn ảnh (weight sharing), nhờ đó mạng có thể nhận diện cùng một đặc trưng ở nhiều vị trí khác nhau.

2.5.1.2. Các lớp gộp (Pooling layer)

Bên cạnh lớp tích chập, lớp gộp (pooling) là một thành phần quan trọng trong kiến trúc CNN. Mục tiêu của pooling là giảm kích thước không gian của bản đồ đặc

trưng, giữ lại các đáp ứng quan trọng nhất trong mỗi vùng con, đồng thời tăng tính bất biến của mô hình đối với các dịch chuyển nhỏ hoặc biến thiên cục bộ trong ảnh.)



Hình 2.6: Max pooling

Phép pooling chọn giá trị lớn nhất trong vùng hiện tại (bên trái); Average pooling: phép pooling lấy trung bình các giá trị trong vùng hiện tại (bên phải).

Trong thực tế, phổ biến nhất là gộp giá trị lớn nhất (max pooling) và gộp trung bình (average pooling). Với max pooling, một cửa sổ (ví dụ 2×2 hoặc 3×3) được trượt trên bản đồ đặc trưng, và tại mỗi vị trí, giá trị lớn nhất trong cửa sổ được chọn làm đầu ra. Average pooling thì lấy giá trị trung bình của các phần tử trong vùng. Hình 2.6 minh họa cách max pooling và average pooling hoạt động: mỗi phép pooling thu gọn một vùng con thành một giá trị duy nhất, tạo ra bản đồ đặc trưng có kích thước nhỏ hơn đáng kể nhưng vẫn giữ được các đặc trưng nổi bật nhất.

Việc đưa các lớp gộp vào CNN mang lại nhiều lợi ích. Thứ nhất, pooling làm giảm số lượng tham số và phép tính, giúp mô hình nhẹ hơn và tăng tốc suy luận. Thứ hai, pooling giúp biểu diễn trở nên bất biến hơn với các dịch chuyển nhỏ của đối tượng trong ảnh; một đặc trưng quan trọng nếu đối tượng không luôn xuất hiện đúng một vị trí cố định. Trong một số kiến trúc hiện đại, convolution có bước dịch lớn (strided convolution) đôi khi được sử dụng thay thế một phần vai trò của pooling, nhưng về bản chất, mục tiêu vẫn là giảm dần kích thước không gian và tập trung vào các đặc trưng quan trọng.

2.5.1.3. Các lớp Kết nối đầy đủ

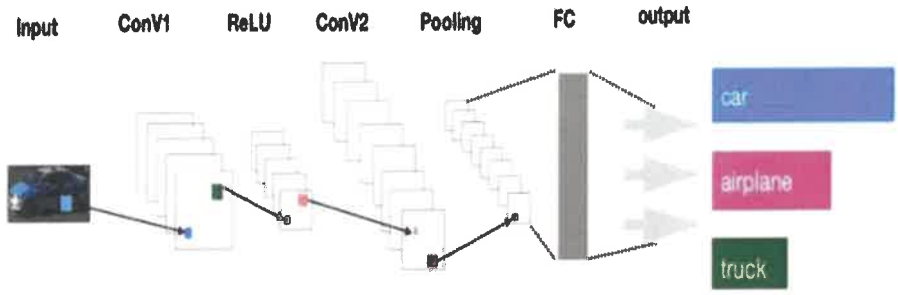
Các lớp kết nối đầy đủ thường được đặt ở phần cuối của kiến trúc CNN. Sau nhiều tầng tích chập và gộp, các bản đồ đặc trưng hai chiều được "duỗi phẳng"

(flatten) thành một vector một chiều. Vector này chứa thông tin tổng hợp về ảnh đầu vào, đã được nén qua nhiều tầng trích xuất đặc trưng.

Lớp kết nối đầy đủ nhận vector này làm đầu vào, kết nối với tất cả nơ-ron trong lớp, tương tự các mạng nơ-ron truyền thống. Một hoặc vài lớp kết nối đầy đủ được sử dụng để biến đổi biểu diễn đặc trưng thành không gian phân loại cần thiết. Lớp cuối cùng thường sử dụng hàm kích hoạt Softmax (đối với bài toán đa lớp) hoặc Sigmoid (đối với bài toán nhị phân), cho phép diễn giải đầu ra như xác suất mẫu thuộc về từng lớp. Trong bối cảnh bài toán phân loại hình ảnh, đây là nơi mô hình đưa ra dự đoán cuối cùng về nhãn của ảnh đầu vào.

2.5.2. Kiến trúc mạng nơ-ron tích chập

Trong một mạng nơ-ron truyền thẳng thông thường, mỗi nơ-ron đầu vào được kết nối với mỗi nơ-ron đầu ra trong lớp tiếp theo. Lớp này thường được gọi là lớp kết nối đầy đủ hoặc lớp Affine. Tuy nhiên, CNN không hoạt động theo cách đó. Thay vào đó, chúng ta tính toán đầu ra bằng cách sử dụng phép tích chập (convolutions) trên lớp đầu vào. Do đó, mỗi vùng của đầu vào được liên kết với một nơ-ron trong đầu ra. Mỗi lớp thêm vào các bộ lọc khác nhau, có thể là hàng trăm hoặc hàng nghìn như đã thấy ở trên, và sau đó kết hợp (blends) sản phẩm của chúng. Cũng có một thứ gọi là các lớp gộp (pooling - subsampling layers), mà tôi sẽ thảo luận sau. Một CNN học các giá trị của các bộ lọc một cách linh hoạt (dynamically) trong quá trình kiểm thử (testing process) tùy thuộc vào mục tiêu đang thực hiện. Ví dụ, trong Phân loại Hình ảnh, một CNN có thể học cách phát hiện các cạnh (edges) từ các pixel thô ở lớp đầu tiên, sau đó sử dụng những lợi thế đó để phát hiện các hình dạng cơ bản (basic shapes) ở lớp tiếp theo, và cuối cùng sử dụng những hình dạng này để tránh các đặc trưng cấp cao hơn như hình dạng khuôn mặt (facial shapes) ở các lớp tiếp theo. Lớp cuối cùng là một bộ phân loại (classifier) dùng các chức năng cấp cao này. Theo Hình 2.7, chúng ta có thể thấy kiến trúc cơ bản của CNN để phát hiện ô tô trong hình ảnh.



Hình 2.7: Kiến trúc cơ bản của Mạng Nơ-ron Tích chập.

2.5.3. Huấn luyện mạng nơ-ron tích chập

Huấn luyện mạng nơ-ron tích chập về bản chất là quá trình điều chỉnh trọng số của từng nơ-ron để thu được một mô hình có khả năng tổng quát tốt trên dữ liệu mới. Ban đầu, các trọng số của CNN được khởi tạo ngẫu nhiên. Khi một ảnh được đưa qua mạng, mô hình sinh ra dự đoán ở lớp đầu ra; dự đoán này được so sánh với nhãn đúng để tính toán hàm mất mát (loss).

Quá trình lan truyền ngược (backpropagation) được sử dụng để tính gradient của hàm mất mát theo từng trọng số trong mạng. Dựa trên gradient này, các thuật toán tối ưu như stochastic gradient descent (SGD) hoặc Adam sẽ cập nhật trọng số theo hướng làm giảm mất mát. Mỗi lần toàn bộ tập dữ liệu huấn luyện được đưa qua mô hình được gọi là một epoch. Sau một số lượng epoch đủ lớn, nếu tốc độ học được lựa chọn hợp lý, mạng dần "hội tụ", nghĩa là lỗi trên tập huấn luyện giảm xuống và hiệu suất trên tập kiểm định được nâng cao.

Sau khi quá trình huấn luyện kết thúc, mô hình CNN cần được đánh giá trên tập dữ liệu chưa từng xuất hiện trong quá trình huấn luyện để kiểm tra khả năng tổng quát hóa. Nếu mô hình hoạt động tốt trên tập huấn luyện nhưng kém trên tập kiểm thử, hiện tượng quá khớp (overfitting) có thể đã xảy ra do mạng học quá chi tiết các mẫu trong tập huấn luyện mà không tổng quát hóa được. Một số kỹ thuật như regularization, dropout hoặc tăng cường dữ liệu (data augmentation) thường được áp dụng để giảm nguy cơ quá khớp.

Hiệu suất của CNN trong thực tế thường phụ thuộc mạnh vào kích thước và chất lượng của bộ dữ liệu huấn luyện. Trong các bài toán có dữ liệu gán nhãn hạn

chế, một chiến lược phổ biến là tận dụng các mô hình đã được huấn luyện trước trên những bộ dữ liệu lớn (ví dụ ImageNet), sau đó tinh chỉnh (fine-tune) trên tập dữ liệu mục tiêu. Cách làm này cho phép kế thừa các đặc trưng chung đã học được, giảm thời gian huấn luyện và cải thiện độ chính xác trong bối cảnh dữ liệu hạn chế – đây cũng là hướng tiếp cận phù hợp khi xây dựng hệ thống phát hiện buồn ngủ dựa trên quan sát vùng mắt.

2.6. Mạng Nơ-ron Tích chập xếp tầng đa nhiệm (Multi-task Cascaded Convolutional Networks)

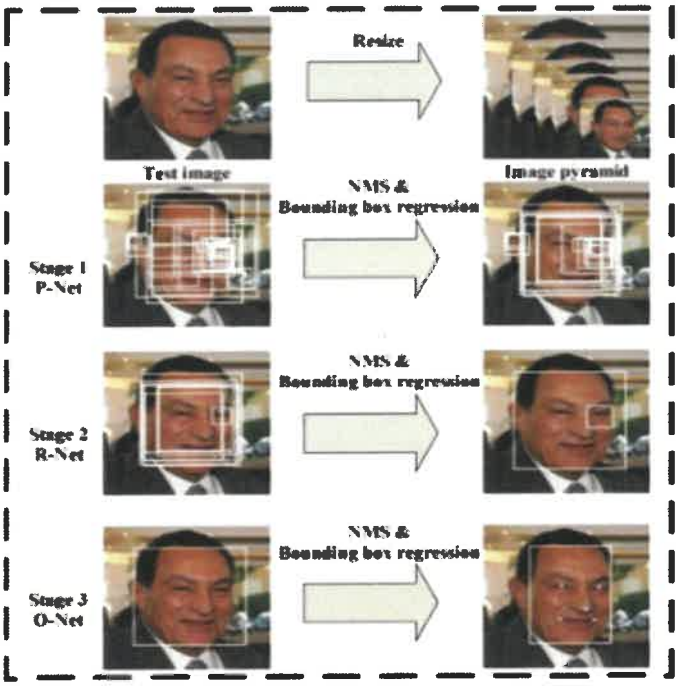
Trong khi có nhiều phương pháp phát hiện khuôn mặt [20][19][10][26][25], nhưng MTCNN (Multi-task Cascaded Convolutional Networks - Mạng Nơ-ron Tích chập Xếp tầng Đa nhiệm) được sử dụng trong nghiên cứu này vì hai lý do. Một mặt, nó đạt được độ phát hiện chính xác cao (high detection accuracy), và mặt khác, mô hình FaceNet đã cung cấp sẵn giao diện MTCNN để phát hiện khuôn mặt.

MTCNN là một thuật toán học sâu (deep learning) để nhận dạng khuôn mặt (face recognition) được xây dựng trên một Mạng Nơ-ron Tích chập xếp tầng (cascaded Convolutional Neural Network - CNN) với nhiều nhiệm vụ. Nó tăng cường hiệu quả bằng cách sử dụng mối liên kết nội tại giữa việc nhận dạng (identification) và căn chỉnh (alignment). Phương pháp trong bài báo này tận dụng một kiến trúc xếp tầng (cascaded architecture) với ba cấp độ mạng nơ-ron tích chập sâu (deep convolutional networks) được xây dựng cẩn thận để dự đoán vị trí khuôn mặt và mốc (landmark locations) theo phương pháp từ thô đến tinh (coarse-to-fine) (25). (Hình 2.8) cho thấy hình ảnh mô hình của MTCNN.

2.6.1. Mạng Đề xuất (Proposal Network - P-NET)

P-Net là một mạng nơ-ron tích chập đầy đủ (Fully Convolutional Network - FCN). Bản đồ đặc trưng (feature map) của hàm truyền thẳng (forwarding propagation function) là một vector đặc trưng 32 chiều (32-dimensional feature vector) tại mỗi điểm. Nó được dùngđể quyết định xem các ô lưới 12x12 có chứa khuôn mặt hay không. Nếu một ô lưới bao gồm một khuôn mặt, Hộp giới hạn (Bounding Box) của khuôn mặt đó sẽ được hồi quy (regressed) để thu được Hộp giới hạn tương ứng với

vùng trong bức ảnh gốc. Giai đoạn Loại bỏ Phi cực đại (Non-maximum suppression - NMS) sẽ giữ lại Hộp giới hạn có xếp hạng cao nhất (highest ranking), và tất cả các Hộp giới hạn khác có vùng chồng lấn quá lớn sẽ bị loại bỏ.



Hình 2.8: MTCNN

Các bước mà kiến trúc chuỗi tầng thực hiện để đảm bảo hiệu suất tốt nhất trong việc phát hiện khuôn mặt người và hồi quy hộp bao (bounding box regression).

2.6.2. Mạng Tinh chỉnh (Refine Network - R-NET)

R-Net là mạng con thứ hai, hoạt động như một bộ lọc tinh, nhận đầu vào là các hộp đề xuất từ P-Net đã được cắt và chuẩn hóa kích thước. Cấu trúc của R-Net phức tạp hơn P-Net, với nhiều lớp tích chập và các lớp ẩn hơn, cho phép học được các đặc trưng phân biệt tốt hơn giữa vùng có khuôn mặt và vùng nhiễu. Tương tự như giai đoạn tiếp theo (O-Net), vùng 24x24 và Hộp giới hạn (Bounding Box) tương ứng được phóng to lên 48x48. Sau đó, nó được đưa vào cấp độ R-Net, với nhiệm vụ phát hiện các Hộp giới hạn và các mốc khuôn mặt (facial landmarks) với độ tin cậy phát hiện cao nhất (highest detection confidence).

2.6.3. Mạng Đầu ra (Output Network - O-NET)

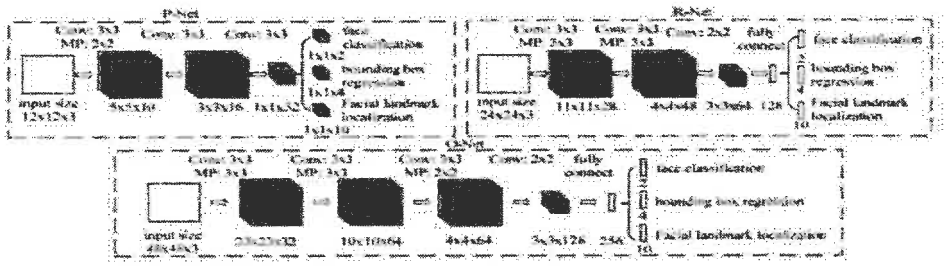
O-NET giúp cải thiện độ chính xác (improve precision). Nó là một CNN đơn giản; Hộp giới hạn được tạo ra bởi bước P-Net có thể chứa hoặc không chứa khuôn mặt người. Gói này, cùng với đầu vào 12x12, trước tiên được tăng tỷ lệ (up-scaled) lên 24x24 bằng cách sử dụng kỹ thuật nội suy song tuyến (bilinear interpolation technique), và sau đó làm đầu vào cho O-Net để đánh giá sự hiện diện của khuôn mặt người. Khi một khuôn mặt được chứa đựng (được xác nhận), hộp giới hạn được hồi quy (regressed) và trải qua cấp độ NMS (Non-maximum suppression - Loại bỏ Phi cực đại).

2.6.4. Kiến trúc của MTCNN

Trong mô hình MTCNN dạng xếp tầng, mỗi giai đoạn sử dụng một kiến trúc lớp riêng với số lượng lớp và các bộ lọc tích chập (Conv filters) khác nhau, nhưng đều tạo ra ba loại đầu ra giống nhau. Đầu tiên là đầu ra phân loại khuôn mặt, trong đó hai nơ-ron được sử dụng để tạo ra điểm phân loại: một nơ-ron xác định sự hiện diện của khuôn mặt và nơ-ron còn lại thể hiện mức độ tin cậy hay thứ hạng. Tiếp theo là đầu ra hồi quy hộp giới hạn, gồm bốn nơ-ron tương ứng với các giá trị dịch chuyển của hai góc trên trái và dưới phải của bounding box, lần lượt là là d_{x1} d_{y1}, d_{x2}, và d_{y2}. Cuối cùng, mô hình còn thực hiện định vị các mốc khuôn mặt thông qua việc hồi quy năm điểm quan trọng gồm mắt phải, mắt trái, khóe miệng phải, khóe miệng trái và mũi, giúp tăng cường độ chính xác trong việc nhận diện và theo dõi đặc trưng khuôn mặt.

Ba mạng (P-Net, R-Net, O-Net) sẽ sử dụng các vị trí mốc (landmark locations) làm tín hiệu có giám sát (supervised cues) để định hướng quá trình học của mạng trong suốt quá trình huấn luyện. Nhưng, trong quá trình dự đoán, P-Net và R-Net chỉ thực hiện phát hiện khuôn mặt và không tạo ra vị trí mốc do độ chính xác thấp ở các giai đoạn này. O-Net được sử dụng để xác định vị trí mốc. Các đầu ra của việc đồng bộ hóa hộp giới hạn và các mốc được chuẩn hóa (normalized) so với dữ liệu. Như đã đề cập ở trên, có ba nhiệm vụ mà Mạng Nơ-ron Tích chập Xếp tầng Đa nhiệm vụ (MTCNN) thực hiện. Chính xác là, nhận dạng mặt nạ (mask recognition) (ý chỉ phân

loại/phát hiện khuôn mặt), hồi quy hộp giới hạn (bounding box regression) và định vị các mốc khuôn mặt (localization of facial landmarks). Kết quả là, hàm mất mát (loss function) của thuật toán cũng có ba phần. Dưới đây là tóm tắt các điểm chính; muốn biết thêm thông tin chi tiết được hướng dẫn tham khảo [21]. (Hình 2.9) minh họa chi tiết thiết kế mô hình để phát hiện khuôn mặt và trích xuất mốc.



Hình 2.9: Kiến trúc của MTCNN

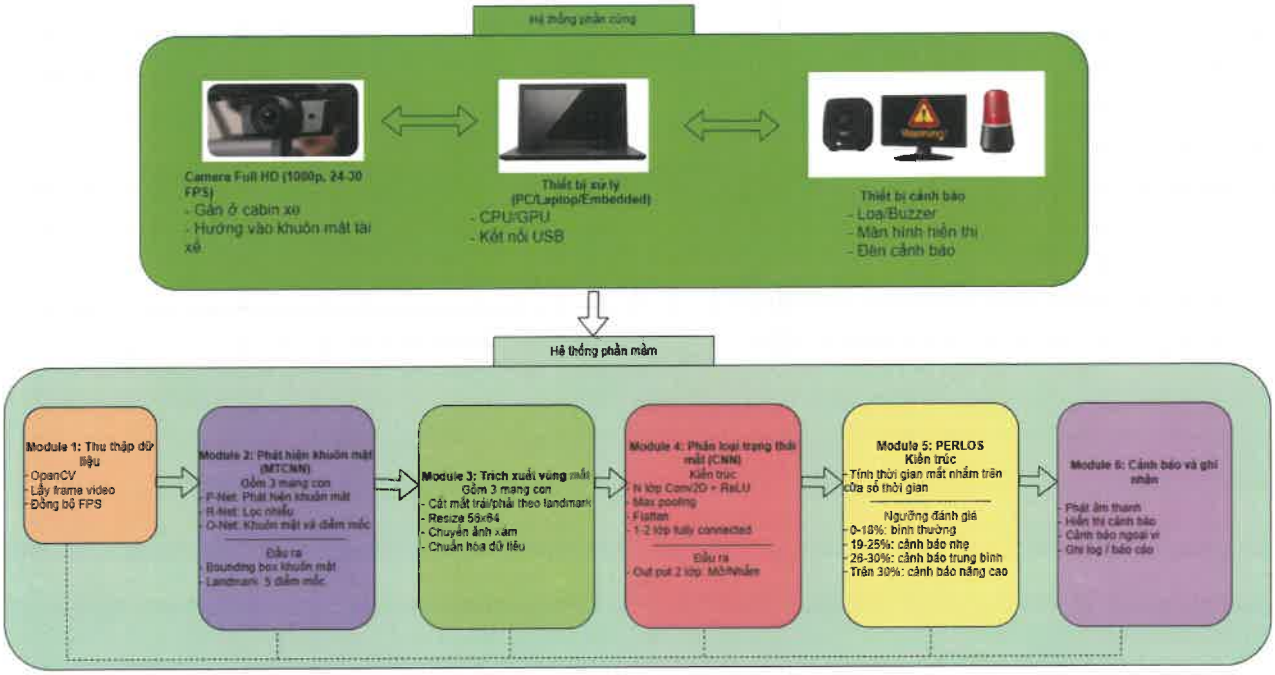
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1. Phân tích hệ thống

Trước khi tiến hành thiết kế chi tiết, cần thực hiện phân tích hệ thống nhằm làm rõ bài toán cần giải quyết, các yêu cầu chức năng và phi chức năng, cũng như xác định luồng xử lý tổng thể của hệ thống phát hiện buồn ngủ dựa trên theo dõi mắt. Việc phân tích này giúp đảm bảo hệ thống được xây dựng đúng mục tiêu, có tính khả thi khi triển khai và đáp ứng yêu cầu vận hành trong môi trường thực tế.

3.1.1. Yêu cầu của hệ thống

Xuất phát từ mục tiêu xây dựng một hệ thống theo dõi mắt nhằm phát hiện tình trạng buồn ngủ của người lái xe trong điều kiện thực tế, hệ thống cần được phân tích rõ ràng về cả yêu cầu chức năng lẫn yêu cầu phi chức năng. Việc phân tích này đóng vai trò nền tảng, định hướng cho toàn bộ quá trình thiết kế và triển khai hệ thống ở các mục tiếp theo.



Hình 3.1: Mô hình tổng thể của hệ thống

Hình 3.1 minh họa kiến trúc tổng thể của hệ thống theo dõi mắt phát hiện buồn ngủ, bao gồm hai thành phần chính là hệ thống phần cứng và hệ thống phần mềm xử

lý. Hệ thống phần cứng sử dụng camera đặt trong cabin xe để thu nhận hình ảnh khuôn mặt người lái và truyền dữ liệu đến thiết bị xử lý trung tâm. Trên cơ sở đó, hệ thống phần mềm thực hiện chuỗi xử lý gồm phát hiện khuôn mặt bằng MTCNN, trích xuất vùng mắt và phân loại trạng thái mắt bằng mạng CNN. Kết quả phân loại được tổng hợp theo thời gian thông qua chỉ số PERCLOS nhằm đánh giá mức độ buồn ngủ của người lái. Khi giá trị PERCLOS vượt qua các ngưỡng xác định, hệ thống sẽ kích hoạt cơ chế cảnh báo để hỗ trợ người lái kịp thời nhận biết và xử lý tình huống.

Về yêu cầu chức năng, hệ thống cần thực hiện được các nhiệm vụ chính sau. Trước hết, hệ thống phải thu nhận liên tục hình ảnh khuôn mặt người lái từ camera đặt trong cabin xe, đảm bảo không bị gián đoạn trong suốt quá trình lái xe. Trên cơ sở các khung hình thu được, hệ thống phải phát hiện chính xác khuôn mặt người lái và xác định được vị trí hai mắt trong từng khung hình. Tiếp theo, hệ thống cần phân loại được trạng thái mắt (mở hoặc nhắm) với độ chính xác đủ cao để phản ánh đúng hành vi thị giác của người lái. Các kết quả phân loại này phải được theo dõi theo thời gian nhằm đánh giá xu hướng buồn ngủ, thay vì chỉ dựa vào các dự đoán rời rạc tại từng thời điểm. Cuối cùng, khi phát hiện người lái có dấu hiệu buồn ngủ vượt quá ngưỡng an toàn, hệ thống phải đưa ra cảnh báo kịp thời để hỗ trợ người lái chủ động xử lý.

Bên cạnh đó, yêu cầu phi chức năng cũng đóng vai trò quan trọng trong bài toán này. Hệ thống cần hoạt động gần thời gian thực với độ trễ thấp, đảm bảo các cảnh báo được phát ra đúng thời điểm. Việc giám sát phải mang tính không xâm lấn, tức là không yêu cầu gắn thêm cảm biến sinh lý lên cơ thể người lái, nhằm tránh gây khó chịu và đảm bảo khả năng ứng dụng rộng rãi. Ngoài ra, hệ thống cần có khả năng thích nghi với sự thay đổi của điều kiện ánh sáng, tư thế khuôn mặt và sự khác biệt giữa các cá nhân. Cuối cùng, thiết kế hệ thống cần hướng tới khả năng triển khai trên các nền tảng phần cứng phổ thông với chi phí hợp lý.

3.1.2. Phân tích lựa chọn kiến trúc hệ thống

Từ các yêu cầu đã phân tích ở mục trên, có thể thấy rằng việc lựa chọn kiến trúc hệ thống cần cân bằng giữa độ chính xác, tính ổn định và khả năng triển khai

thực tế. Trong số các hướng tiếp cận phát hiện buồn ngủ đã được trình bày ở Chương 2, nhóm phương pháp dựa trên thị giác máy tính được đánh giá là phù hợp nhất với mục tiêu của đề tài do tính không xâm lấn và khả năng tận dụng các thiết bị camera thông dụng.

Tuy nhiên, các phương pháp thị giác máy tính truyền thống dựa trên đặc trưng hình học hoặc các ngưỡng cố định thường gặp hạn chế trong môi trường thực tế, đặc biệt khi điều kiện ánh sáng thay đổi hoặc khi người lái có đặc điểm khuôn mặt khác nhau. Do đó, đề tài lựa chọn kiến trúc hệ thống dựa trên sự kết hợp của các mô hình học sâu và thuật toán thống kê theo thời gian.

Cụ thể, hệ thống được thiết kế theo mô hình pipeline nhiều tầng. Ở tầng đầu tiên, MTCNN được sử dụng để phát hiện và căn chỉnh khuôn mặt. Việc lựa chọn MTCNN giúp hệ thống không chỉ xác định được vị trí khuôn mặt mà còn trích xuất chính xác các điểm mốc quan trọng, đặc biệt là vị trí hai mắt, tạo tiền đề cho các bước xử lý phía sau. Ở tầng tiếp theo, mạng nơ-ron tích chập (CNN) được sử dụng để phân loại trạng thái mắt mở hoặc nhắm. CNN cho phép mô hình tự động học các đặc trưng phù hợp từ dữ liệu, giúp cải thiện độ chính xác so với các phương pháp dựa trên đặc trưng thủ công. Cuối cùng, thuật toán PERCLOS được sử dụng để tổng hợp chuỗi kết quả phân loại theo thời gian, từ đó đánh giá mức độ buồn ngủ một cách định lượng và dễ diễn giải.

Kiến trúc này cho phép tận dụng ưu điểm của từng thành phần, đồng thời giảm thiểu các hạn chế khi sử dụng riêng lẻ từng phương pháp.

3.1.3. Phân tích luồng xử lý và tương tác giữa các module

Luồng xử lý dữ liệu trong hệ thống được thiết kế theo hướng tuyến tính và khép kín, đảm bảo mỗi khung hình đều được xử lý theo một trình tự thống nhất. Trước hết, camera thu nhận hình ảnh thời gian thực và cung cấp các khung hình đầu vào cho hệ thống. Mỗi khung hình sau đó được đưa vào mô-đun MTCNN để phát hiện khuôn mặt và xác định các điểm mốc trên khuôn mặt.

Khi khuôn mặt được phát hiện thành công, hệ thống sử dụng tọa độ landmark để trích xuất vùng mắt trái và mắt phải. Các vùng mắt này được chuẩn hóa về kích

thước và định dạng trước khi được đưa vào mô hình CNN. CNN thực hiện phân loại trạng thái mắt cho từng khung hình, tạo ra chuỗi kết quả "mở" hoặc "nhắm" theo thời gian.

Chuỗi kết quả phân loại này không được sử dụng trực tiếp để đưa ra quyết định cảnh báo, mà được chuyển đến mô-đun PERCLOS. Tại đây, hệ thống tính toán tỷ lệ thời gian mắt nhắm trong một cửa sổ thời gian trượt, từ đó đánh giá mức độ buồn ngủ của người lái. Khi giá trị PERCLOS vượt qua các ngưỡng định sẵn, mô-đun cảnh báo sẽ được kích hoạt và phát tín hiệu phù hợp.

Việc phân tách rõ ràng các mô-đun xử lý giúp hệ thống dễ bảo trì, dễ mở rộng và thuận tiện cho việc thay thế hoặc nâng cấp từng thành phần trong tương lai.

3.1.4. Đánh giá và nhận xét về thiết kế hệ thống

Từ góc độ thiết kế, hệ thống theo dõi mắt phát hiện buồn ngủ có nhiều ưu điểm nổi bật. Kiến trúc mô-đun giúp hệ thống linh hoạt và dễ mở rộng, cho phép tích hợp thêm các đặc trưng khác như tư thế đầu hoặc biểu cảm khuôn mặt trong các nghiên cứu tiếp theo. Việc kết hợp giữa học sâu và chỉ số PERCLOS giúp hệ thống giảm ảnh hưởng của nhiễu tại từng khung hình đơn lẻ, đồng thời phản ánh chính xác xu hướng buồn ngủ theo thời gian. Ngoài ra, hệ thống không yêu cầu thiết bị chuyên dụng hay cảm biến sinh lý, phù hợp với mục tiêu triển khai thực tế.

Tuy nhiên, hệ thống vẫn tồn tại một số hạn chế. Hiệu năng phụ thuộc đáng kể vào chất lượng camera và điều kiện ánh sáng trong cabin xe. Khi triển khai hoàn toàn trên CPU, tốc độ xử lý còn bị giới hạn, đặc biệt ở bước phát hiện khuôn mặt bằng MTCNN. Một số trường hợp đặc biệt như tài xế đeo kính râm đậm hoặc bị che khuất vùng mắt vẫn có thể ảnh hưởng đến độ chính xác của hệ thống.

Những hạn chế này là cơ sở để đề xuất các hướng cải tiến trong tương lai, đồng thời khẳng định rằng thiết kế hệ thống hiện tại là phù hợp với phạm vi và mục tiêu của đề tài.

3.2.Mô hình tổng thể của hệ thống

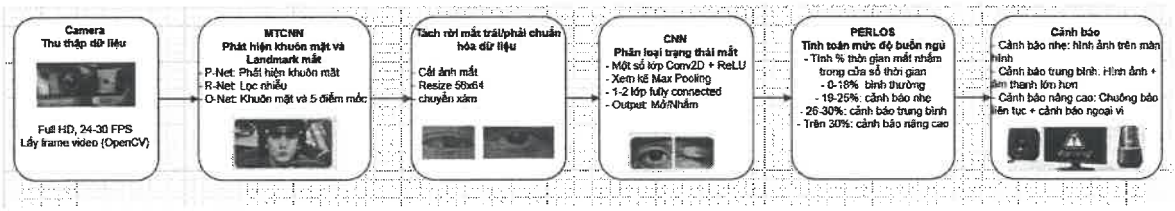
Hệ thống phát hiện buồn ngủ được xây dựng dưới dạng pipeline liên tục, xử lý từng khung hình (frame) theo trình tự từ camera đến phân tích mức độ buồn ngủ. Mô hình tổng thể gồm bốn thành phần chính:

(1) Khối thu nhận hình ảnh – Camera: thu liên tục hình ảnh khuôn mặt người lái ở tốc độ khoảng 24–30 fps và gửi trực tiếp đến hệ thống xử lý. Việc đảm bảo góc quay và chất lượng hình ảnh là yếu tố then chốt để giúp các thuật toán nhận diện hoạt động chính xác.

(2) Khối phát hiện khuôn mặt và landmark – MTCNN: dùng để phát hiện khuôn mặt, xác định chính xác năm điểm landmark (hai mắt, mũi, hai khoé miệng). Việc sử dụng MTCNN mang lại tốc độ xử lý nhanh và độ chính xác cao, phù hợp cho ứng dụng thời gian thực.

(3) Khối phân loại trạng thái mắt – CNN: Sau khi xác định được hai vùng mắt từ MTCNN, ảnh mắt được cắt và đưa vào mô hình CNN đã được huấn luyện để xác định trạng thái: mắt mở hoặc mắt nhắm. CNN được huấn luyện từ hơn 100.000 hình ảnh mắt, giúp mô hình đạt độ chính xác trên 90%.

(4) Khối đánh giá mức độ buồn ngủ – PERCLOS: được tính toán dựa trên số lượng frame mắt nhắm trong một khoảng thời gian so với tổng số frame. Từ đó hệ thống xác định mức độ buồn ngủ và đưa ra cảnh báo thích hợp.



Hình 3.2: Sơ đồ khối hệ thống

3.3. Mô tả chi tiết từng thành phần

Mục này trình bày đầy đủ và chi tiết kiến trúc cũng như chức năng của bốn thành phần cốt lõi trong hệ thống phát hiện buồn ngủ của người lái, bao gồm Camera, MTCNN, CNN và module tính toán PERCLOS. Mỗi thành phần có một vai trò độc lập nhưng có sự liên kết chặt chẽ để đảm bảo hệ thống vận hành trơn tru và chính xác trong thời gian thực.

3.2.1. Camera – Thiết bị thu nhận hình ảnh thời gian thực

Camera được xem là "con mắt" của hệ thống, là nguồn của dữ liệu đầu vào duy nhất để các thành phần phía sau hoạt động. Trong đề tài, camera được gắn cố định bên trong xe, đối diện với người lái, với góc đặt hơi thấp hơn tầm mắt một khoảng nhất định và hướng lên trên. Cách bố trí này giúp hạn chế việc vô-lăng hoặc tay người lái che khuất khuôn mặt, đồng thời đảm bảo khuôn mặt luôn nằm trong vùng nhìn chính của camera. Camera sử dụng trong thí nghiệm có độ phân giải 1080p, tốc độ khung hình khoảng 26–27 fps khi ghi hình thông thường, đủ để hệ thống quan sát được những thay đổi nhỏ trong trạng thái mắt.

Camera cần đảm bảo vùng mắt đủ rõ trong các môi trường ánh sáng khác nhau vì trong thức tế, người lái có thể lái xe ban ngày, ban đêm, trong hầm,... Do đó, ưu tiên camera có khả năng tự cân bằng sáng và dải sáng động tốt. Vị trí lắp đặt phải đảm bảo vùng mặt/mắt đủ lớn (đủ pixel) để MTCNN và CNN hoạt động ổn định.

Trong giai đoạn thử nghiệm, camera kết nối trực tiếp với máy tính xử lý qua cổng USB, các khung hình được truy xuất thông qua thư viện OpenCV. Việc xử lý video theo thời gian thực đòi hỏi không chỉ cấu hình phần cứng đủ mạnh mà còn yêu cầu tối ưu ở mức phần mềm, chẳng hạn như giảm bớt những thao tác thừa trên mỗi khung hình, sử dụng các biểu diễn dữ liệu gọn nhẹ và bố trí pipeline xử lý hợp lý để giảm độ trễ. Khi hệ thống được triển khai trong môi trường thực tế, camera có thể được tích hợp vào các thiết bị nhúng hoặc màn hình trên xe, miễn là vẫn đảm bảo đủ tốc độ và chất lượng hình ảnh.

3.2.2. MTCNN – Phát hiện khuôn mặt và xác định landmark mắt

Sau khi camera cung cấp các khung hình, bước tiếp theo là tìm ra khuôn mặt người lái trong từng ảnh. Công việc này là của MTCNN, một kiến trúc mạng nơ-ron tích chập gồm ba mạng con P O-net, R-net và P-net được sắp xếp theo dạng xếp tầng. P-Net nhận ảnh đầu vào đã được xây dựng thành tháp đa tỉ lệ và đưa ra các vùng nghi ngờ có chứa khuôn mặt. Các vùng này sau đó được tinh chỉnh và sàng lọc tiếp qua R-Net, nơi những vùng không phải khuôn mặt bị loại bỏ thêm một lần nữa và hộp giới hạn (bounding box) được điều chỉnh chính xác hơn. Cuối cùng, O-

Net nhận các vùng còn lại, vừa tiếp tục phân biệt xem đó có thực sự là khuôn mặt hay không, vừa hồi quy vị trí các điểm mốc quan trọng như hai khóe miệng, mũi và hai mắt.

Ưu điểm của MTCNN là nó đồng thời giải quyết bài toán phát hiện và căn chỉnh khuôn mặt. Ngoài việc khoanh vùng khuôn mặt, MTCNN còn cung cấp tọa độ chính xác của các điểm mốc, trong đó vị trí hai mắt là thông tin cốt lõi cho hệ thống theo dõi buồn ngủ. Từ các điểm mốc này, hệ thống xác định được vùng chứa mắt trái và mắt phải, cắt ra các ảnh mắt với kích thước phù hợp để đưa vào CNN. Nhờ khả năng xử lý khuôn mặt ở nhiều kích thước và tư thế khác nhau, MTCNN vẫn có thể phát hiện khuôn mặt ngay cả khi người lái hơi nghiêng đầu, ngước lên hoặc cúi xuống, việc thường xuyên diễn ra trong quá trình lái xe.

Trong quá trình triển khai, MTCNN hoạt động như một bộ lọc đầu tiên trên các khung hình video. Nếu ở một khung hình nào đó không tìm thấy khuôn mặt, hệ thống sẽ bỏ qua khung hình này và không thực hiện phân loại mắt, nhằm tránh các tính toán không cần thiết. Ngược lại, nếu phát hiện được khuôn mặt và các điểm mốc mắt, hệ thống sẽ sử dụng thông tin này để trích xuất vùng mắt và chuyển tiếp đến mô hình CNN. Việc áp dụng MTCNN trên nền PyTorch trong các thí nghiệm cho thấy tốc độ xử lý đạt khoảng 9–10 fps trên CPU i7, cao hơn đáng kể so với phiên bản TensorFlow với cùng điều kiện, từ đó đảm bảo hệ thống có thể hoạt động gần thời gian thực.

3.2.3. CNN – Phân loại trạng thái mắt (mở hoặc nhắm)

Sau khi vùng mắt đã được trích xuất và chuẩn hóa, nhiệm vụ tiếp theo là xác định xem mắt đang ở trạng thái mở hay nhắm. Công việc này được đảm nhiệm bởi mạng nơ-ron tích chập (CNN) được huấn luyện chuyên biệt cho bài toán phân loại trạng thái mắt. Đầu vào của CNN là ảnh mắt đã được resize về kích thước chuẩn, ví dụ 56×64 pixel, và chuyển sang mức xám để giảm số kênh màu, giúp mô hình tập trung vào các đặc trưng hình dạng thay vì thông tin màu sắc không cần thiết. Dữ liệu đầu vào cũng được chuẩn hóa giá trị điểm ảnh về một khoảng nhất định, chẳng hạn [0,1], nhằm giúp quá trình học ổn định hơn.

Kiến trúc của CNN trong đề tài được xây dựng theo kiểu điển hình, gồm một số lớp tích chập liên tiếp kết hợp với hàm kích hoạt ReLU để trích xuất các đặc trưng cục bộ như cạnh, góc, biên độ thay đổi độ sáng quanh vùng mí mắt. Sau mỗi một hoặc vài lớp tích chập là một lớp gộp (max pooling) nhằm giảm kích thước không gian của đặc trưng, tiết kiệm chi phí tính toán và giúp mô hình bền vững hơn trước những dịch chuyển nhỏ trong ảnh. Ở phần cuối mạng, các bản đồ đặc trưng được "duỗi phẳng" thành một vector, đi qua một hoặc hai lớp kết nối đầy đủ để tổng hợp thông tin, rồi đưa đến lớp đầu ra với hai nút tương ứng với hai lớp "mắt mở" và "mắt nhắm". Lớp đầu ra sử dụng hàm Softmax hoặc Sigmoid để biến đổi thành xác suất cho từng lớp, từ đó quyết định nhãn cuối cùng cho ảnh mắt.

Trong giai đoạn huấn luyện, CNN sử dụng tập dữ liệu gồm hàng chục nghìn ảnh mắt được gán nhãn, kết hợp từ dữ liệu thu thập của sinh viên và từ MRL Eye Dataset [5]. Mô hình được huấn luyện trên nền tảng Google Colab với GPU để rút ngắn thời gian. Các tham số như tốc độ học, số epoch, kích thước batch, số lượng kernel ở mỗi lớp và số lượng lớp tích chập được điều chỉnh thử nghiệm để tìm ra cấu hình cho độ chính xác và tốc độ hội tụ tốt nhất. Kết quả cho thấy mô hình đạt độ chính xác trên 90% trên tập kiểm thử, đủ tin cậy để sử dụng trong hệ thống. Trong giai đoạn vận hành, mỗi khung hình có khuôn mặt được MTCNN phát hiện sẽ tạo ra hai ảnh mắt đưa vào CNN, và kết quả phân loại "mở" hoặc "nhắm" cho từng mắt được ghi lại phục vụ cho việc tính toán PERCLOS.

3.2.4. PERCLOS – Tính toán mức độ buồn ngủ

Thành phần cuối cùng trong chuỗi xử lý là module PERCLOS, bộ phận chịu trách nhiệm chuyển đổi chuỗi trạng thái mắt theo thời gian thành một chỉ số định lượng về mức độ buồn ngủ. Sau khi CNN phân loại trạng thái mắt cho từng khung hình, hệ thống sẽ lưu trữ kết quả này trong một cửa sổ thời gian quan sát, chẳng hạn vài giây. Trong khoảng thời gian đó, hệ thống đếm số khung hình mà mắt được phân loại là nhắm và tổng số khung hình quan sát được, từ đó tính toán giá trị PERCLOS theo công thức đã nêu ở chương 2. Khi cửa sổ trượt theo thời gian, giá trị PERCLOS được cập nhật liên tục, phản ánh tức thời mức độ buồn ngủ của người lái.

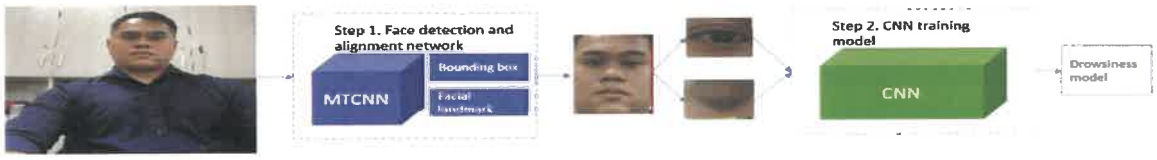
Dựa trên giá trị PERCLOS, hệ thống thiết kế các ngưỡng cảnh báo khác nhau. Khi PERCLOS nằm trong một khoảng thấp (0–18%), hệ thống có thể xem đây là trạng thái bình thường hoặc chỉ có dấu hiệu mệt nhẹ, nên không phát cảnh báo hoặc chỉ nhắc nhở nhẹ. Khi PERCLOS tăng lên một ngưỡng trung bình (19–25%), hệ thống hiểu rằng người lái đã bắt đầu buồn ngủ rõ rệt, có nguy cơ rơi vào trạng thái ngủ ngắn, nên phát cảnh báo rõ ràng hơn, chẳng hạn bằng âm thanh hoặc thông báo trên màn hình, khuyến nghị người lái nghỉ ngơi. Nếu PERCLOS vượt một ngưỡng cao (26–30% hoặc trên 30%), nguy cơ xảy ra micro-sleep hoặc ngủ gật thực sự được xem là rất lớn, hệ thống sẽ kích hoạt cảnh báo ở mức độ mạnh nhất. Các ngưỡng này được chọn dựa theo các khuyến nghị từ tài liệu nghiên cứu và điều chỉnh thông qua thực nghiệm để giảm tối đa cảnh báo giả mà vẫn không bỏ sót các trường hợp nguy hiểm.

Thông qua việc kết hợp MTCNN, CNN và PERCLOS, hệ thống không chỉ quan sát được trạng thái mắt tại từng thời điểm mà còn đánh giá được xu hướng buồn ngủ trong một khoảng thời gian, từ đó đưa ra cảnh báo hợp lý và kịp thời. Đây là bước chuyển quan trọng, biến các dự đoán rời rạc trên từng khung hình thành một quyết định mang ý nghĩa thực tiễn đối với an toàn giao thông.

CHƯƠNG 4: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG

Hệ thống phát hiện buồn ngủ bằng thị giác máy tính cần được xây dựng như một chuỗi xử lý có tổ chức, trong đó mỗi bước đều phục vụ trực tiếp cho mục tiêu cuối cùng là đánh giá đúng mức độ buồn ngủ và đưa ra cảnh báo kịp thời. Trong đề tài này, giải pháp được xây dựng bao gồm hai bước. Bước đầu tiên là xây dựng tập dữ liệu và thực hiện tiền xử lý để tạo ra các ảnh mắt có chất lượng và định dạng phù hợp cho mô hình học sâu. Bước thứ hai là phát triển mô hình buồn ngủ dựa trên sự kết hợp giữa MTCNN, CNN và thuật toán PERCLOS, từ đó tạo nên một hệ thống theo dõi mắt có thể hoạt động gần thời gian thực. Hai bước này được thể hiện trong các sơ đồ khối ở Hình 4.1 và Hình 4.2, minh họa rõ luồng xử lý từ dữ liệu thô cho đến tín hiệu cảnh báo buồn ngủ gửi tới người lái.

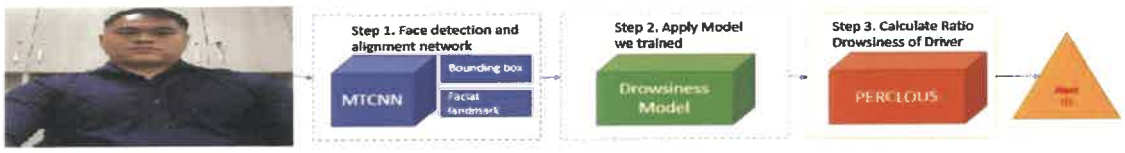
Quy trình tổng thể được mô tả ở hình 4.1. Tình trạng buồn ngủ có thể được phát hiện thông qua việc phát hiện đặc trưng trên khuôn mặt (facial feature recognition) và ánh xạ khuôn mặt (facial mapping) [23]. Tuy nhiên, để tìm được các đặc trưng khuôn mặt tiêu chuẩn nhằm liên kết các mốc khuôn mặt (joint landmarks) và kiểm tra xem chúng đã nằm đúng vị trí không, cần phải tuân theo một quy trình cụ thể và nhất quán. Trong bối cảnh này, Bộ phân loại xếp tầng đa nhiệm (Multi-Task Cascade Classifier – MTCNN) [25] thường được áp dụng đồng thời cho cả việc phát hiện và căn chỉnh khuôn mặt, nhằm tối ưu đồng thời cả thời gian xử lý và độ chính xác, nên thường được ưu tiên sử dụng cho nhiệm vụ này. Bằng cách khai thác cấu trúc mạng sâu và đa nhiệm, MTCNN đạt hiệu suất tốt trong việc nhận diện và phân đoạn khuôn mặt, cung cấp chính xác vị trí hai mắt.



Hình 4.1: Các bước tổng thể của Bộ dữ liệu và Tiền xử lý

Bước 1 bao gồm khuôn mặt và điểm mốc để tiếp cận, trạng thái mở mắt và bước 2 bao gồm mô hình buồn ngủ.

Sau khi quá trình nhận diện khuôn mặt hoàn tất, hai tọa độ mắt trái và mắt phải được xác định và sử dụng để suy ra hướng nhìn cũng như vùng mắt cần quan sát. Dựa trên hai tọa độ này, hệ thống xây dựng khung bao (bounding box) và áp dụng các phép mở rộng cần thiết để tạo nhãn cho dữ liệu, từ đó hình thành các khung giới hạn và vùng cắt (cropping landmark) thể hiện trạng thái mắt trái – mắt phải tại từng thời điểm. Các ảnh mắt đã được cắt và gán nhãn này sau đó được đưa vào mô hình CNN, sử dụng cách thức đã nhắc đến ở phần trước để huấn luyện và tạo ra mô hình tổng hợp (summary model) dùng cho việc phát hiện tình trạng buồn ngủ. Bước tiếp theo là áp dụng mô hình đã huấn luyện để phát hiện trạng thái buồn ngủ khi lái xe. Theo Hình 4.2, có thể quan sát toàn bộ hệ thống trong giai đoạn suy luận: hệ thống áp dụng các trọng số đã được học ở giai đoạn đầu, sử dụng thuật toán PERCLOS [13] để tính toán tỷ lệ khung hình (frame ratio) mà mắt được phát hiện ở trạng thái nhắm từ luồng camera, sau đó cảnh báo người lái khi chỉ số mệt mỏi vượt ngưỡng. Quy trình đánh giá này được lặp lại thêm hai lần nữa trước khi kết luận rằng tỷ lệ giữa các cảm biến đã lệch đi đáng kể so với giá trị ban đầu của chúng, nhằm giảm thiểu sai số và hiện tượng cảnh báo giả.



Hình 4.2: Các bước tổng thể của mô hình phát hiện buồn ngủ

Bước 1 nhận diện khuôn mặt bao gồm khuôn mặt và điểm mốc của mắt và bước áp dụng mô hình buồn ngủ. Bước 3: Sử dụng PERCLOS để cảnh báo buồn ngủ.

Trọng tâm của chương này là trình bày chi tiết tập dữ liệu và các bước tiền xử lý, kiến trúc mạng nơ-ron tích chập dùng để phát hiện buồn ngủ, cũng như mô tả cách triển khai hệ thống theo dõi mắt phát hiện buồn ngủ. Cuối chương, các kết quả thí nghiệm về tốc độ xử lý, độ chính xác và khả năng triển khai thực tế sẽ được thảo luận, qua đó đánh giá mức độ đáp ứng của hệ thống đối với các yêu cầu đặt ra ở những chương trước.

4.1. Tập dữ liệu và Tiền xử lý (Datasets and Preprocessing)

4.1.1. Tập dữ liệu (Datasets)

Dữ liệu ban đầu mà hệ thống thu được là dữ liệu thô dưới dạng ảnh hoặc video, và bản thân máy tính không thể hiểu trực tiếp ý nghĩa của những dữ liệu này nếu chúng không được tổ chức và gán nhãn một cách thích hợp. Vì vậy, trước khi xây dựng mô hình học sâu, cần hình thành một khái niệm rõ ràng về "tập dữ liệu" mà mô hình sẽ sử dụng. Trong một số ngữ cảnh, tập dữ liệu có thể được hiểu là một tập hợp các ví dụ hoặc các cặp đầu vào – đầu ra dùng để huấn luyện và đánh giá mô hình. Mỗi phần tử trong tập dữ liệu thường được mô tả bởi một số thuộc tính (features) cùng với nhãn tương ứng, thể hiện thông tin mà mô hình cần học.

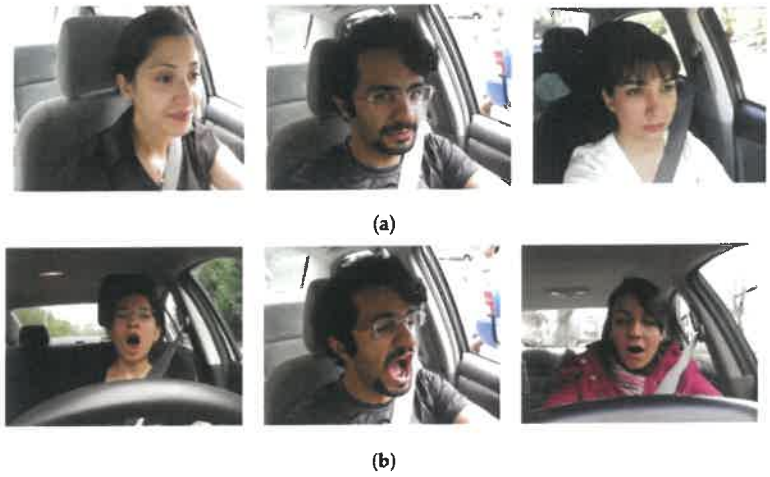
Mô hình học sâu trong đề tài này được phát triển trên cơ sở các ảnh mắt đã được cắt và tiền xử lý. Các ảnh này xuất phát từ nguồn tập dữ liệu mắt được thu thập từ Internet, gồm:

1, MRL Eye Dataset [5]. Đây là một tập dữ liệu công khai, quy mô lớn, chứa ảnh mắt thu thập trong nhiều điều kiện khác nhau: ánh sáng thường, ánh sáng hồng ngoại, các độ sáng – tối khác nhau, nhiều kiểu camera và trạng thái mắt đa dạng như mắt mở, mắt nhắm, mắt một phần, có vật cản.

2, Closed Eyes In The Wild (CEW): Đây là tập dữ liệu chứa 2423 đối tượng, trong đó 1192 đối tượng nhắm cả hai mắt và 1231 đối tượng mở mắt được chọn từ Labeled Face in the Wild. Kích thước ảnh mặt 100x100 và ảnh mắt đã cắt là 56x64.

3, UTA-RLDD (UTA Real-Life Drowsiness Dataset): Đây là bộ dữ liệu buồn ngủ của người lái xe được thu thập trong điều kiện thực tế, với các video ghi lại khuôn mặt tài xế trong quá trình lái xe. Dữ liệu có sự đa dạng về điều kiện ánh sáng, tư thế khuôn mặt và môi trường, đồng thời được gán nhãn theo trạng thái tỉnh táo hoặc buồn ngủ của người lái.

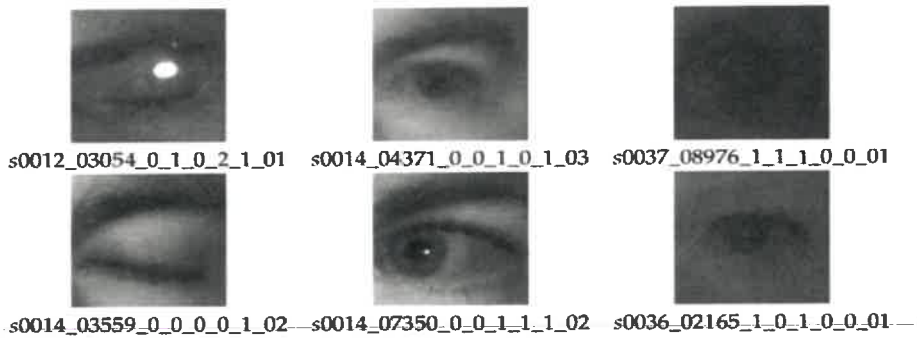
4, NTHU-DDD (Drive Drowsiness Detection Dataset): Đây là là các video được quay trong cabin xe, thường được dùng trong các nghiên cứu phát hiện buồn ngủ trong khi lái xe. Video có độ phân giải 640x480.



Hình 4.3: Một số ví dụ về tập dữ liệu NTHU-DDD

5, YawDD (Yawing Detection Dataset): Đây là các video (khoảng 350 video) tập trung vào ngáp của nhiều tài xế (cả nam và nữ) làm các hoạt động như nói, hát, ngáp,... khi đang lái xe. Video có độ phân giải 640x480.

Dữ liệu từ Internet sau khi được tải về sẽ trải qua quá trình sàng lọc, loại bỏ các mẫu không phù hợp, chuẩn hóa kíchD thước và gán nhãn "mắt mở" hoặc "mắt nhắm" theo đúng mục tiêu của đề tài. Hình 4.4 là mô tả cho tập dữ liệu này.



Hình 4.4: MRL Eye Dataset

Việc kết hợp tập dữ liệu sinh viên với MRL Eye Dataset giúp tăng đáng kể số lượng mẫu huấn luyện và kiểm thử, đồng thời làm phong phú thêm các kiểu mắt, điều kiện ánh sáng và góc chụp, qua đó nâng cao khả năng tổng quát hóa của mô hình phân loại mắt.

4.1.2. Tiền xử lý (Preprocessing)

Sau khi lựa chọn và thu thập được tập dữ liệu phù hợp, bước tiếp theo là tiền xử lý dữ liệu để chuyển những ảnh thô thành đầu vào chuẩn hóa cho mô hình CNN. Với sự ra đời và phát triển mạnh mẽ của mạng nơ-ron tích chập (CNN), việc chuẩn hóa dữ liệu đầu vào trở nên đặc biệt quan trọng, bởi hiệu suất của mô hình phụ thuộc không chỉ vào kiến trúc CNN mà còn vào cách tổ chức, kích thước và không gian màu của ảnh [15], [24]. Nếu các mẫu ảnh có kích thước quá khác nhau, độ phân giải không đồng nhất, hoặc không được chuẩn hóa về cùng một không gian màu, mô hình sẽ khó học được những đặc trưng ổn định và dễ bị ảnh hưởng bởi nhiễu.

Trong đề tài, quy trình tiền xử lý được triển khai theo các bước nhất quán. Trước hết, từ các video ban đầu chứa khuôn mặt người tham gia, hệ thống tiến hành tách khung hình và sử dụng các kỹ thuật phát hiện khuôn mặt để xác định vùng chứa khuôn mặt trên mỗi khung. Từ vùng mặt này, vùng mắt được cắt ra để tạo thành các ảnh mắt độc lập, bảo đảm mỗi ảnh đầu vào cho mô hình chỉ tập trung vào phần có ý nghĩa nhất đối với bài toán phân loại trạng thái mắt. Các video ban đầu có độ phân giải 2560×1820 pixel, nhưng trong quá trình tiền xử lý, chỉ vùng mắt được giữ lại; vùng này sau đó được thay đổi kích thước về 56×64 pixel với tốc độ trích xuất 24 khung hình mỗi giây. Các khung hình mắt được lưu dưới định dạng PNG, không kèm âm thanh, để chuẩn bị cho các bước xử lý và huấn luyện tiếp theo (hình 4.5).

Với các tập dữ liệu hình ảnh thu thập từ Internet dựa vào kỹ thuật crawl, kích thước ảnh ban đầu rất không đồng nhất, có thể là 185×111 pixel, 104×58 pixel hoặc nhiều kích thước khác. Sự không đồng nhất này gây cản trở cho việc xây dựng các tập dữ liệu để huấn luyện và kiểm thử phù hợp cho mô hình CNN, bởi kiến trúc này yêu cầu một kích thước đầu vào cố định. Vì vậy, toàn bộ các ảnh mắt từ nguồn Internet đều được reshape về cùng kích thước chuẩn 56×64 pixel nhằm tạo sự thống nhất trong đầu vào của CNN. Nhờ đó, mô hình có thể xử lý đồng nhất mọi mẫu dữ liệu, tránh những sai lệch do sự khác biệt về kích thước ảnh gây ra.

Tổng thể bộ dữ liệu sau tiền xử lý được chia làm ba phần: tập kiểm thử, tập đánh giá và tập huấn luyện. Tập huấn luyện chứa hơn 35.000 khung hình mắt mở và

hơn 35.000 khung hình mắt nhắm. Tập đánh giá gồm khoảng 5.000 khung hình mắt mở và 5.000 khung hình mắt nhắm, trong khi tập kiểm thử bao gồm 10.000 khung hình mắt mở và 10.000 khung hình mắt nhắm. Trong suốt quá trình huấn luyện và đánh giá, mỗi khung hình đều được gán một nhãn nhị phân tương ứng với hai trạng thái "mở" hoặc "nhắm", bảo đảm mô hình học được ranh giới phân biệt rõ ràng giữa hai lớp.



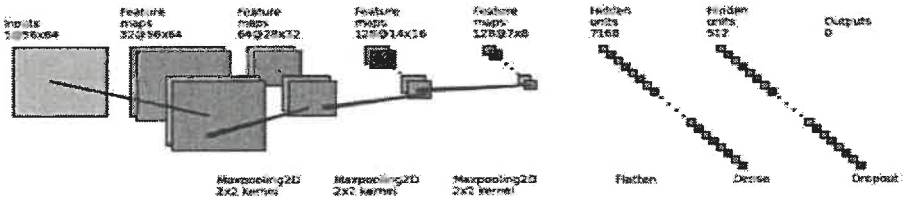
Hình 4.5: (a) Camera thời gian thực; (b) Video có kích thước 2560x1820 pixel; (c) Tách Khung hình trong Video có kích thước 2560x1820 pixel; (d) Hộp bao quanh khuôn mặt và mắt; (e) Cắt khung bao quanh mắt có kích thước 56x64 pixel.

Sau bước chuẩn hóa kích thước, các ảnh màu được chuyển sang không gian mức xám (grayscale) để giảm số kênh dữ liệu từ ba kênh RGB xuống còn một kênh. Điều này vừa giảm khối lượng tính toán, vừa giúp mô hình tập trung vào các đặc trưng hình học và sự thay đổi cường độ sáng – tối quanh vùng mí mắt thay vì thông tin màu sắc vốn ít liên quan tới trạng thái mắt mở/nhắm. Cuối cùng, giá trị pixel được chuẩn hóa về một khoảng nhất định, chẳng hạn [0,1], nhằm tránh sự chênh lệch quá lớn về biên độ dữ liệu, đồng thời giúp quá trình lan truyền ngược và tối ưu trọng số trong mô hình học sâu diễn ra ổn định hơn. Dữ liệu sau tiền xử lý được lưu trữ theo cấu trúc thư mục chuẩn, phân tách rõ ràng giữa các lớp "mắt mở" và "mắt nhắm", sẵn sàng cho giai đoạn huấn luyện CNN ở các mục tiếp theo.

4.2. Mạng Nơ-ron Tích chập để Phát hiện Buồn ngủ

Như đã trình bày ở đầu chương, mạng nơ-ron tích chập (Convolutional Neural Network – CNN) được lựa chọn làm mô hình lõi cho bài toán theo dõi mắt và phát hiện buồn ngủ. Phần cơ sở lý thuyết ở chương trước đã mô tả tổng quan kiến trúc và cơ chế hoạt động của CNN; tại đây, nội dung tập trung làm rõ cấu trúc cụ thể được

sử dụng trong hệ thống và cách chuẩn hóa dữ liệu trước khi đưa vào huấn luyện. Mô hình CNN được thiết kế gồm một chuỗi các lớp tích chập và gộp, tiếp theo là các lớp kết nối đầy đủ và lớp đầu ra nhị phân, được minh họa trong Hình 4.6. Cấu trúc này đảm bảo đủ khả năng trích xuất đặc trưng phục vụ phân loại, đồng thời vẫn giữ được tốc độ suy luận phù hợp với yêu cầu gần thời gian thực của hệ thống theo dõi mắt.



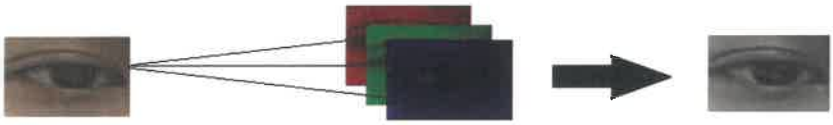
Hình 4.6: Mô hình CNN đào tạo hệ thống theo dõi mắt để phát hiện tình trạng buồn ngủ khi lái xe

Nguồn dữ liệu đầu vào cho mô hình CNN là các ảnh mắt đã được chuẩn bị ở mục 3.2.1 (Hình 4.7). Ban đầu, dữ liệu thu được dưới dạng video màu độ phân giải cao, trong đó khuôn mặt người tham gia được ghi lại ở các trạng thái mắt mở và mắt nhắm. Từ các video này, hệ thống tiến hành tách khung hình, phát hiện khuôn mặt, cắt vùng mắt và chuẩn hóa kích thước về 56×64 pixel. Tuy nhiên, ở giai đoạn này, các ảnh mắt vẫn ở không gian màu RGB nên chưa hoàn toàn phù hợp với định dạng đầu vào mong muốn của mô hình. Trong bước tiền xử lý tiếp theo, toàn bộ ảnh mắt được chuyển sang ảnh xám (gray image), tạo thành các tensor có kích thước 56×64x1 như minh họa ở Hình 4.8. Nhờ đó, dữ liệu đầu vào được rút gọn còn một kênh cường độ sáng, vừa giảm chi phí tính toán, vừa giúp mô hình tập trung vào hình dạng và biên độ sáng–tối quanh vùng mí mắt – những yếu tố then chốt khi phân biệt mắt mở và mắt nhắm. Khi ảnh đầu vào đã được chuyển đổi sang định dạng 56×64×1, dữ liệu được đưa vào lớp tích chập hai chiều đầu tiên (conv2d). Nhiệm vụ của lớp này là trích xuất các đặc trưng cục bộ cơ bản từ vùng mắt, chẳng hạn như cạnh, đường biên và các biến thiên cường độ quanh mí mắt. Để làm được điều đó, một tập các nhân tích chập (kernel) kích thước 3×3 được sử dụng, trượt trên toàn bộ ảnh với bước dịch bằng 1 pixel (stride = 1). Việc lựa chọn kernel 3×3 kết hợp với stride = 1 giúp mô hình "quét" được chi tiết tinh trên toàn bộ

vùng ảnh, tránh bỏ sót những biến thiên nhỏ nhưng có thể mang nhiều thông tin về trạng thái đóng–mở của mắt (Hình 4.7, Hình 4.8).



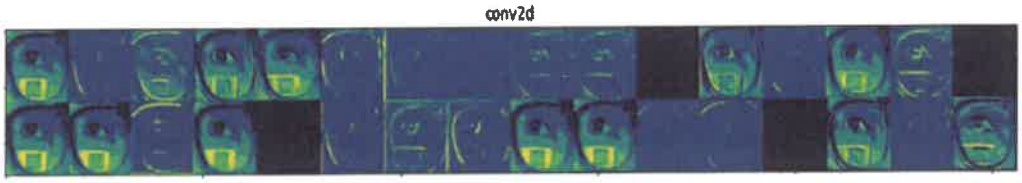
Hình 4.7: Bộ dữ liệu đầu vào: Mắt mở và mắt đóng cho mô hình



Hình 4.8: Ảnh dữ liệu RGB sang Xám

Ngay sau lớp tích chập đầu tiên, hàm kích hoạt ReLU (Rectified Linear Unit) được áp dụng nhằm đưa tính phi tuyến vào mạng. ReLU có đặc tính giữ các giá trị dương và đưa các giá trị âm về 0, đồng thời không bị bão hòa ở hai đầu như Sigmoid hay Tanh, nhờ đó giúp gradient lan truyền tốt hơn và tăng tốc độ hội tụ trong khi huấn luyện. Hình ảnh đầu vào ở Hình 4.8 và kết quả sau lớp conv2d kết hợp ReLU được thể hiện tại Hình 4.9, cho thấy các vùng đặc trưng quanh mí mắt bắt đầu được nhấn mạnh rõ rệt.

Để tối ưu hóa hơn nữa các đặc trưng đã trích xuất, sau mỗi lớp tích chập thường đi kèm một lớp gộp cực đại (max pooling). Cơ chế gộp giúp giảm kích thước của bản đồ đặc trưng, giữ lại những giá trị nổi bật, qua đó giảm số lượng tham số cần huấn luyện và rút ngắn thời gian tính toán. Hình 4.10 minh họa các đặc trưng sau khi thực hiện max pooling, cho thấy thông tin quan trọng vẫn được bảo toàn trong khi biểu diễn đã gọn hơn đáng kể. Nhờ đó, mô hình có thể học được các mẫu đặc trưng ổn định, ít bị nhiễu bởi các dịch chuyển nhỏ hoặc biến thiên không quan trọng trong ảnh mắt.



Hình 4.9: Lớp đầu tiên: Conv2D



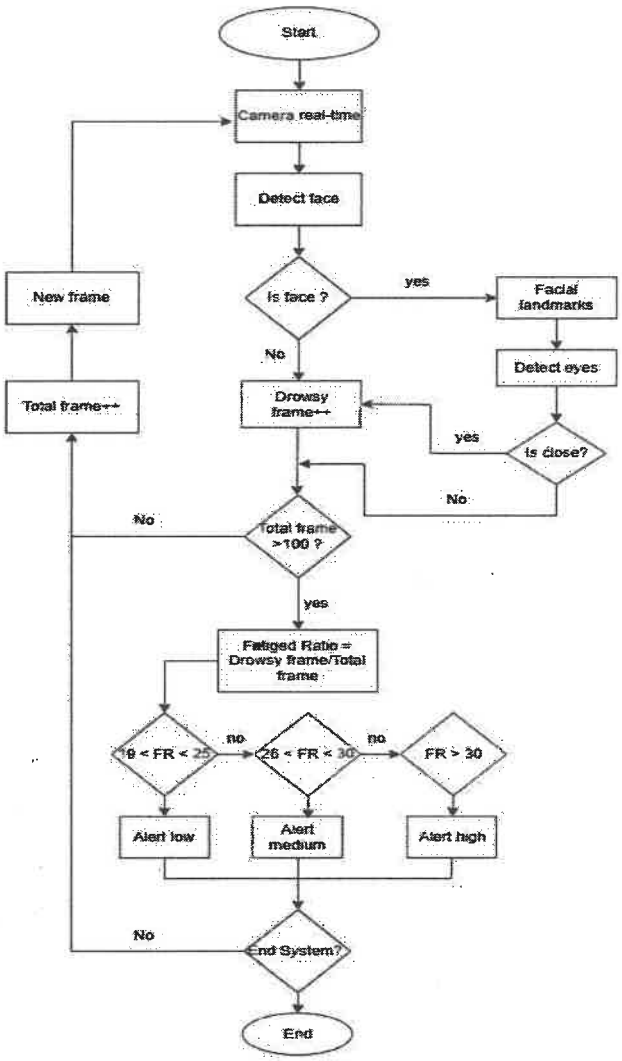
Hình 4.10: Tính năng sau khi maxpooling

Các lớp tiếp theo trong kiến trúc CNN tiếp tục lặp lại mô-típ "tích chập – kích hoạt – gộp" với số lượng bộ lọc tăng dần, cho phép mô hình học được các đặc trưng trừu tượng hơn dựa trên các đặc trưng thấp cấp đã được rút trích ở những lớp đầu. Ở phần cuối mạng, các bản đồ đặc trưng được "duỗi phẳng" thành một vector, đưa qua một hoặc nhiều lớp kết nối đầy đủ để tổng hợp thông tin, trước khi đi vào lớp đầu ra nhị phân dự đoán hai trạng thái "mắt mở" và "mắt nhắm". Nhờ kiến trúc này, mô hình CNN có khả năng tự động học các đặc trưng phù hợp nhất cho bài toán, hỗ trợ trực tiếp cho việc phát hiện buồn ngủ trong quá trình giám sát mắt người lái xe.

4.3. Hệ thống Theo dõi Mắt Phát hiện Buồn ngủ (Drowsiness Detection Tracking Eye System)

Dựa trên cơ sở các mục trước, hệ thống phát hiện buồn ngủ được xây dựng bằng cách kết hợp MTCNN, CNN và thuật toán PERCLOS trong một chuỗi xử lý thống nhất. Camera gắn trong cabin xe liên tục thu nhận khung hình chứa khuôn mặt người lái; MTCNN được dùng để phát hiện và căn chỉnh khuôn mặt, đồng thời xác định vị trí hai mắt; vùng mắt sau đó được cắt ra, chuẩn hóa và đưa vào CNN để phân loại trạng thái mở hoặc nhắm; chuỗi kết quả phân loại theo thời gian được chuyển đến mô-đun PERCLOS nhằm tính toán tỷ lệ thời gian mắt nhắm, từ đó suy luận mức độ buồn ngủ và phát cảnh báo nếu cần thiết. Luồng xử lý tổng thể của hệ thống được

minh họa trong Hình 4.11, thể hiện rõ lộ trình từ dữ liệu thô cho đến tín hiệu cảnh báo cuối cùng.



Hình 4.11: Luồng hệ thống

4.3.1 Phát hiện Khuôn mặt bằng Mạng Nơ-ron Tích chập Xếp tầng Đa nhiệm vụ (MTCNN)

Bước đầu tiên sau khi thu nhận dữ liệu từ camera là phát hiện khuôn mặt người lái trên từng khung hình. Nhiệm vụ này được đảm nhiệm bởi MTCNN – một kiến trúc CNN xếp tầng đa nhiệm được thiết kế chuyên biệt cho bài toán phát hiện và căn chỉnh khuôn mặt [25]. Kiến trúc MTCNN gồm ba mạng con: P-Net, R-Net và O-Net. P-Net xử lý ảnh đầu vào ở nhiều tỉ lệ khác nhau, đưa ra các vùng nghi ngờ là khuôn mặt; R-Net tinh chỉnh và lọc bớt các vùng sai, điều chỉnh chính xác hơn hộp giới hạn

(bounding box); cuối cùng, O-Net xác nhận lại các vùng khuôn mặt còn lại và đồng thời dự đoán vị trí các điểm mốc đặc trưng trên khuôn mặt như mắt trái, mắt phải, mũi và miệng. Nhờ cấu trúc xếp tầng và cơ chế đa nhiệm, MTCNN không chỉ phát hiện khuôn mặt với độ chính xác cao mà còn thực hiện luôn nhiệm vụ căn chỉnh khuôn mặt thông qua hệ thống landmark chuẩn. Trong hệ thống đề xuất, MTCNN được sử dụng để trích xuất hộp giới hạn của khuôn mặt (Hình 4.12) và xác định các điểm mốc khuôn mặt (Hình 4.13). Dựa trên các điểm mốc này, đặc biệt là vị trí hai mắt, hệ thống xác định chính xác vùng mắt cần theo dõi và cắt ra các ảnh mắt đầu vào cho mô hình CNN (Hình 4.14). Việc sử dụng MTCNN giúp vùng mắt được xác định ổn định hơn dưới nhiều tư thế, góc nhìn và điều kiện ánh sáng khác nhau, tạo nền tảng quan trọng cho các bước phân loại trạng thái mắt ở phía sau.



Hình 4.12: MTCNN để trích xuất hộp giới hạn của khuôn mặt



Hình 4.13: Các điểm mốc khuôn mặt



Hình 4.14: Mắt khi MTCNN và Điểm mốc khuôn mặt được trích xuất để dự đoán trạng thái

4.3.2. Mạng Nơ-ron Tích chập Phát hiện Trạng thái Mắt (Convolutional Neural Networks Detect Eyes Status)

Sau khi MTCNN và các điểm mốc khuôn mặt đã cung cấp chính xác vị trí hai mắt, vùng mắt tương ứng được trích xuất và chuẩn hóa về dạng ảnh xám kích thước 56×64 pixel. Các ảnh mắt này là đầu vào trực tiếp cho mô hình CNN. Dựa trên kiến trúc đã trình bày ở mục 4.2, CNN tiến hành trích xuất đặc trưng và phân loại mỗi ảnh mắt vào một trong hai trạng thái "mắt mở" hoặc "mắt nhắm". Quá trình này được lặp lại cho từng khung hình mà MTCNN phát hiện được khuôn mặt, tạo thành một chuỗi dự đoán liên tục theo thời gian. Chuỗi nhãn "mở/nhắm" thu được phản ánh trực tiếp hành vi chớp mắt và mức độ duy trì trạng thái mở mắt của người lái trong suốt quá trình điều khiển xe. Khi người lái bắt đầu mệt mỏi, tần suất xuất hiện các khoảng mắt nhắm kéo dài hoặc các giai đoạn mắt nhắm liên tục sẽ tăng lên; những thay đổi này đều được mô hình CNN ghi nhận thông qua kết quả phân loại trên từng khung hình. Có thể xem CNN như một "cảm biến phần mềm" giám sát trạng thái mắt, cung cấp dữ liệu đầu vào chính xác và kịp thời cho mô-đun PERCLOS nhằm đánh giá mức độ buồn ngủ và kích hoạt cơ chế cảnh báo phù hợp cho người lái.

4.3.3. PERCLOS và Cảnh báo (PERCLOS and Alert)

Khi hệ thống đã có khả năng phát hiện trạng thái mắt là mở hoặc nhắm trên từng khung hình, thuật toán PERCLOS [13], [16] được sử dụng để chuyển đổi chuỗi nhãn này thành một chỉ số định lượng về mức độ buồn ngủ. Mô-đun PERCLOS tính toán tỷ lệ phần trăm số khung hình mà mắt ở trạng thái nhắm trong một cửa sổ thời gian quan sát, qua đó phản ánh mức độ giảm tỉnh táo của người lái. Khi chỉ số

PERCLOS vượt qua các ngưỡng đã định, hệ thống sẽ kích hoạt một trong ba mức cảnh báo tương ứng với các cấp độ buồn ngủ khác nhau. Đầu tiên là cảnh báo Mức thấp (Low Warning), cảnh báo này xuất hiện là khi người lái xe đang ở trạng thái buồn ngủ ở mức nhẹ. Với thời gian nhắm mắt ngắn, nhưng tần suất liên tục, cho thấy người lái đang bị thiếu sự tỉnh táo. Mục 2.3 đã đề cập đến PERCLOS, trong trường hợp này tỷ lệ P của người lái xe nằm trong khoảng 19% - 25%, hệ thống sẽ đưa ra cảnh báo ở mức thấp để người lái biết rằng họ đang có dấu hiệu buồn ngủ. Tiếp theo là cảnh báo mức trung bình (Mid-level Warning), khi người lái xe nhắm mắt trong một khoảng thời gian dài hơn và khi hệ thống đã cảnh báo mức thấp nhưng không thấy người lái xe có dấu hiệu mở mắt, thì người lái đang ở trạng thái ngủ gật ngắn và cơ thể không thể cưỡng lại được cơn buồn ngủ. Trong trường hợp này, tỷ lệ P sẽ nằm trong khoảng 26% - 30%. Hệ thống sẽ phát cảnh báo ở mức trung bình với âm lượng lớn hơn và tần suất dày hơn giúp người lái xe tập trung trở lại. Cuối cùng là cảnh báo Nâng cao (Advanced Warning), cảnh báo này xuất hiện khi tín hiệu cảnh báo trung bình không thể giúp người lái tỉnh táo. Khi đó, cảnh báo nâng cao sẽ có thể kết hợp với các biện pháp giảm tốc độ xe hay đồng thời cảnh báo người xung quanh bằng đèn hoặc âm thanh. Trong trường hợp này, tỷ lệ P của người lái trên 30%, đây là một trường hợp nguy hiểm vì người lái đã quá mệt mỏi và cảnh báo không thể khiến nhọ tỉnh táo. Vì thế, hệ thống sẽ cảnh báo cho người xung quanh bằng đèn cảnh báo hay tín hiệu tương tự để chủ động xử lý.

4.4. Kết quả thí nghiệm

Để đánh giá hiệu quả của hệ thống theo dõi mắt phát hiện buồn ngủ, một loạt thí nghiệm đã được thực hiện trên cả phương diện tốc độ xử lý và độ chính xác nhận diện. Về môi trường triển khai, hệ thống được cài đặt trên một máy tính sử dụng bộ xử lý đa nhân hiệu năng cao kết hợp với webcam độ phân giải Full HD. Cấu hình này cho phép thu nhận và xử lý hình ảnh khuôn mặt người dùng trong thời gian gần với thời gian thực. Đồng thời, kiến trúc phần mềm được thiết kế theo hướng linh hoạt, có thể chuyển đổi sang các nền tảng khác như thiết bị nhúng, điện thoại di động hoặc

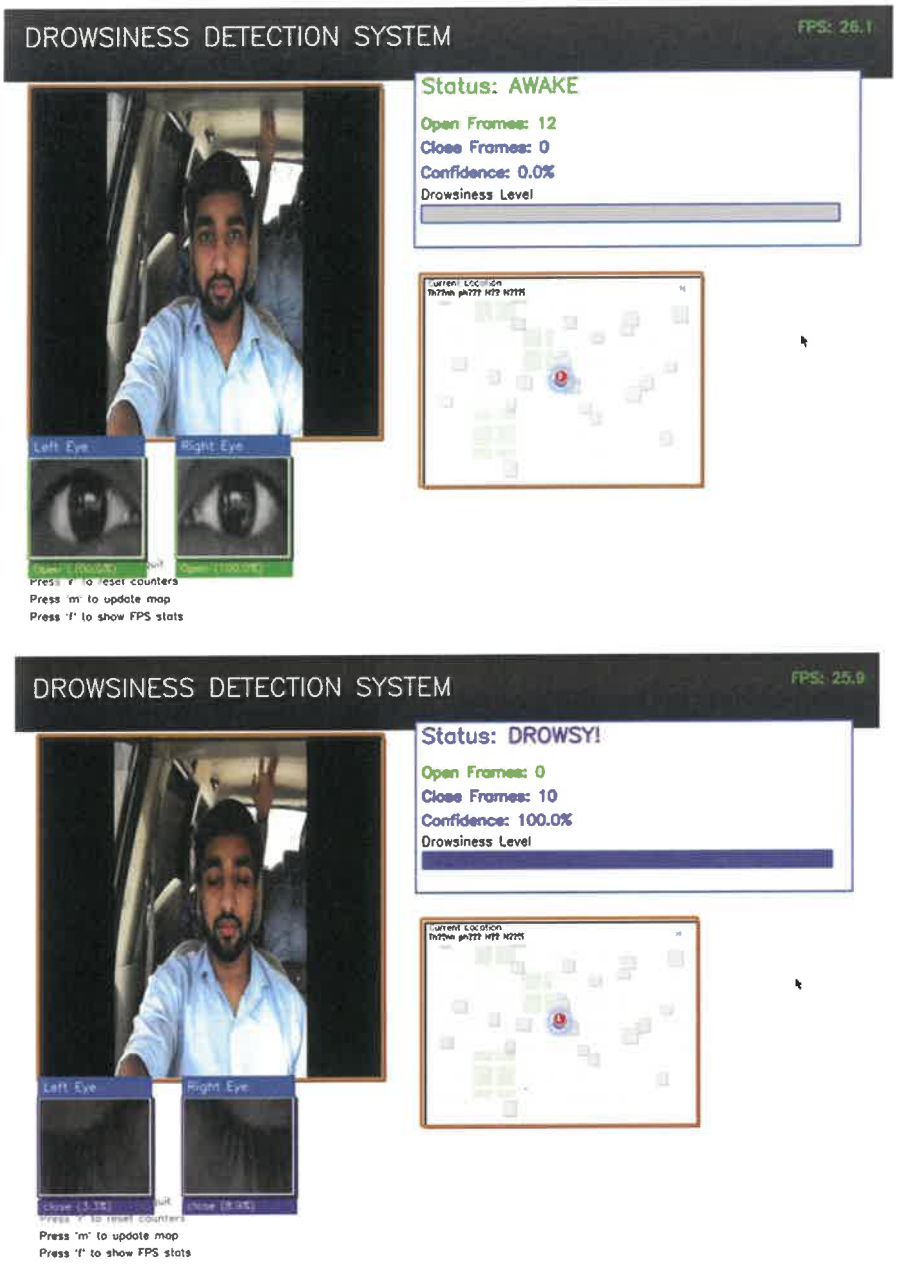
tích hợp trực tiếp vào ô tô với chi phí tối ưu mà không đòi hỏi thay đổi quá lớn về mặt thuật toán.

Hệ thống được đo tốc độ xử lý dưới các cấu hình khác nhau (bảng 4.1) để đánh giá tốc độ xử lý khung hình. Kết quả cho thấy khi chỉ sử dụng camera mà chưa tích hợp bước phát hiện khuôn mặt bằng MTCNN, tốc độ xử lý đạt mức đủ cao để hiển thị video mượt mà. Tuy nhiên, khi đưa MTCNN trên nền TensorFlow vào pipeline, tốc độ suy giảm đáng kể, chỉ còn vài khung hình mỗi giây, khiến việc theo dõi trạng thái mắt khó bắt kịp các biến thiên nhanh. Sau khi chuyển sang triển khai MTCNN trên PyTorch với cách tối ưu phù hợp hơn, tốc độ xử lý tăng lên khoảng 9–10 khung hình mỗi giây. So sánh giữa TensorFlow và PyTorch trong cùng điều kiện cho thấy PyTorch mang lại hiệu năng tốt hơn rõ rệt đối với bài toán cụ thể này.

Bảng 4.1: FPS khi chạy thử nghiệm hệ thống.

Kiểm tra FPS			
Tường hợp	Khung hình	Thời gian	FPS
Camera bình thường	100	3.8333(s)	26-27
Camera sử dụng MTCNN với TensorFlow	100	28.333(s)	3-4
Camera sử dụng MTCNN với PyTorch	100	10.667(s)	9-10

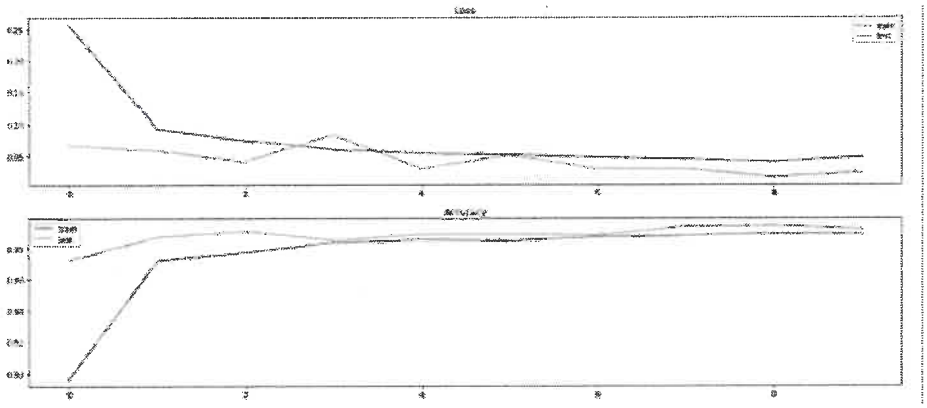
Mặc dù CPU sử dụng có tám lõi vật lý và thuộc nhóm vi xử lý cao cấp, thời gian xử lý vẫn là một thách thức khi phải đồng thời thực hiện các bước thu nhận hình ảnh, phát hiện khuôn mặt, trích xuất mắt và phân loại trạng thái. Để khắc phục hạn chế này trong giai đoạn huấn luyện, hạ tầng tính toán đám mây đã được tận dụng bằng cách sử dụng Google Colab với GPU K80. Mô hình CNN được huấn luyện trên tập dữ liệu gồm hơn một trăm nghìn ảnh mắt, thời gian huấn luyện khoảng hai giờ. Kết quả quá trình huấn luyện được thể hiện trong Hình 4.15, cho thấy hàm mất mát giảm dần và hội tụ trên cả tập huấn luyện và tập đánh giá, trong khi độ chính xác tăng lên vượt ngưỡng 90%. Điều này chứng tỏ mô hình CNN hoạt động



Hình 4.16: Hệ thống theo dõi mắt để phát hiện buồn ngủ và cảnh báo

Kết quả trong Bảng 4.2 cho thấy hệ thống phát hiện trạng thái mắt và mức độ buồn ngủ đạt hiệu năng cao ở cả hai điều kiện. Khi không đeo kính, mô hình đạt Accuracy 92.852%, đồng thời Precision 92.722%, Recall 93.237% và F1-score 92.961%; các chỉ số tương đối cân bằng, cho thấy mô hình vừa hạn chế báo nhầm vừa giảm bỏ sót và hoạt động ổn định. Khi có đeo kính, hiệu năng giảm nhẹ xuống Accuracy 90.469%, với Precision/Recall 90.446% và F1-score 90.473% (giảm

ổn định, tránh được hiện tượng quá khớp nghiêm trọng và đủ tin cậy để đưa vào hệ thống phát hiện buồn ngủ.



Hình 4.15: Mô hình huấn luyện CNN với GPU trong Google Colab

Từ kết quả thể hiện ở Hình 4.15 có thể kết luận rằng mô hình CNN đã đáp ứng được các yêu cầu về độ chính xác trong việc phân loại trạng thái mắt mở/nhắm. Sau khi mô hình được huấn luyện, toàn bộ pipeline MTCNN – CNN – PERCLOS được tích hợp và triển khai thử nghiệm trong môi trường gần với thực tế. Hệ thống được kiểm tra bằng việc quay video của các sinh viên tại các trường đại học khác nhau trong các tình huống mắt mở, mắt nhắm, chớp mắt bình thường và giả lập trạng thái buồn ngủ; đồng thời, một số video trên YouTube có nội dung tài xế ngủ gật cũng được sử dụng để kiểm tra khả năng tổng quát hóa.

Bảng 4.2: Kiểm tra trường hợp phát hiện buồn ngủ.

Trường hợp thử nghiệm				
Trường hợp	Accuracy	Precision	Recall	F1-score
Không đeo kính	92.852%	92.722%	93.237%	92.961%
Đeo Kính	90.469%	90.446%	90.446%	90.473%

khoảng 2.383% so với không kính). Sự suy giảm này nhiều khả năng đến từ phản xạ ánh sáng và hiện tượng che khuất vùng mắt do kính, khiến bước trích xuất đặc trưng và phân loại khó khăn hơn. Dù vậy, hệ thống vẫn theo dõi được các trường hợp nhắm mắt kéo dài, làm PERCLOS tăng và kích hoạt cảnh báo đúng thời điểm; Hình 4.16 minh họa trực quan quá trình theo dõi và cảnh báo, qua đó khẳng định tính khả thi của mô hình và gợi ý tiềm năng tối ưu thêm cho triển khai thực tế.

Để đánh giá tương quan, đề tài so sánh mô hình đề xuất với ba hướng tiếp cận đại diện. Thứ nhất là EAR + Dlib + ngưỡng là nhóm dựa trên đặc trưng hình học: dùng Dlib xác định landmark quanh mắt, tính EAR và so sánh với ngưỡng để suy ra trạng thái mở/nhắm, đồng thời có thể suy luận buồn ngủ qua tần suất/thời lượng nhắm mắt. Thứ hai, Haar Cascade + heuristics là nhóm thị giác máy tính truyền thống: phát hiện vùng mặt/mắt bằng Haar Cascade và ra quyết định bằng các tiêu chí kinh nghiệm (đếm khung hình mắt nhắm liên tiếp, ngưỡng hóa…). Ưu điểm là đơn giản, chạy nhanh; nhược điểm là nhạy với ánh sáng, góc mặt và che khuất. Cuối cùng, MobileNetV2 transfer learning là nhóm học sâu gọn nhẹ: dùng MobileNetV2 làm bộ trích xuất đặc trưng và fine-tune/ghép lớp phân loại để nhận diện mắt mở/nhắm; hiệu quả phụ thuộc đáng kể vào độ đa dạng dữ liệu, đặc biệt với các tình huống khó như ánh sáng thay đổi hoặc đeo kính.

Trong điều kiện không kính (Bảng 4.3), mô hình MTCNN–CNN–PERCLOS đạt kết quả tốt nhất với Accuracy 92.852% và F1-score 92.961%, đồng thời Precision và Recall gần tương đương (92.722% và 93.237%), cho thấy tính cân bằng giữa giảm báo nhầm và hạn chế bỏ sót. Các phương pháp truyền thống (EAR + Dlib, Haar Cascade) thấp hơn rõ rệt, phù hợp với nhận định rằng chúng phụ thuộc mạnh vào chất lượng landmark/đặc trưng thủ công nên dễ suy giảm khi góc mặt lệch, ánh sáng thay đổi hoặc nhiễu ảnh. Với MobileNetV2 TL, dù Precision cao (93.333%), Recall giảm (85.667%) khiến F1-score không vượt mô hình đề xuất; điều này phản ánh xu hướng dự đoán "chặt" (ít báo nhầm) nhưng vẫn bỏ sót một số trường hợp mắt nhắm cần phát hiện.

Bảng 4.3: So sánh các mô hình khi đối tượng không đeo kính

Mô hình	Accuracy	Precision	Recall	F1-score
MTCNN–CNN–PERCLOS	92.852%	92.722%	93.237%	92.961%
EAR + Dlib + ngưỡng	88.235%	87.308%	89.216%	88.253%
Haar Cascade + heuristics	86.436%	85.333%	87.463%	86.233%
MobileNetV2 TL cho mắt	90.223%	93.333%	85.667%	89.335%

Trong điều kiện có kính (Bảng 4.4), hiệu năng của các mô hình đều giảm do kính gây phản xạ và che khuất làm suy giảm chất lượng vùng mắt đầu vào. Tuy nhiên, MTCNN–CNN–PERCLOS vẫn dẫn đầu với Accuracy 90.469% và F1-score 90.473%, đồng thời Precision–Recall cân bằng (90.446%–90.446%), cho thấy độ ổn định tốt hơn khi điều kiện quan sát khó. Ngược lại, các mô hình dựa trên landmark/heuristic giảm mạnh hơn (EAR + Dlib và Haar Cascade), hợp lý vì kính có thể làm sai lệch landmark hoặc khiến vùng mắt khó được phát hiện chính xác. Tương tự trường hợp không kính, MobileNetV2 TL vẫn giữ Precision cao (93.127%) nhưng Recall thấp (84.416%), do đó F1-score vẫn kém hơn mô hình đề xuất.

Bảng 4.4: So sánh các mô hình khi đối tượng có đeo kính

Mô hình	Accuracy	Precision	Recall	F1-score
MTCNN–CNN–PERCLOS	90.469%	90.446%	90.446%	90.473%
EAR + Dlib + ngưỡng	81.683%	81.212%	82.677%	82.013%
Haar Cascade + heuristics	80.454%	79.587%	80.777%	80.152%
MobileNetV2 TL cho mắt	88.215%	93.127%	84.416%	88.515%

Từ hai bảng so sánh, có thể kết luận MTCNN–CNN–PERCLOS cho hiệu năng tổng thể tốt và ổn định hơn trong cả hai điều kiện. Việc kết hợp phát hiện/căn chỉnh vùng mắt bằng MTCNN, phân loại bằng CNN và tổng hợp theo thời gian bằng PERCLOS giúp giảm ảnh hưởng của nhiễu theo từng khung hình, nhờ đó phù hợp hơn với bài toán phát hiện buồn ngủ trong bối cảnh thực tế, đặc biệt khi có các yếu tố gây nhiễu như đeo kính và biến thiên ánh sáng.

KẾT LUẬN

Tóm lại, hệ thống theo dõi mắt phát hiện tình trạng buồn ngủ của người lái xe trong đề tài được xây dựng trên cơ sở kết hợp nhiều mô hình mạng nơ-ron, trong đó trọng tâm là Mạng nơ-ron (NN), Mạng nơ-ron tích chập (CNN) và Mạng nơ-ron tích chập xếp tầng đa nhiệm (MTCNN). Luận văn đã trình bày rõ kiến trúc, cơ chế học và cách áp dụng các mô hình này cho một bài toán học có giám sát, từ khâu phát hiện và căn chỉnh khuôn mặt, trích xuất vùng mắt cho đến phân loại trạng thái mắt mở/nhắm và tính toán chỉ số PERCLOS để đánh giá mức độ buồn ngủ. Trên cơ sở đó, một luồng xử lý hoàn chỉnh cho hệ thống đã được đề xuất, trong đó mỗi mô hình đảm nhận một nhiệm vụ cụ thể và phát huy được thế mạnh riêng, góp phần nâng cao hiệu quả tổng thể của hệ thống. Các kết quả thực nghiệm cho thấy mô hình kết hợp dựa trên MTCNN và CNN đạt hiệu năng tốt hơn so với việc sử dụng riêng lẻ các mô hình nơ-ron cơ bản, cả về độ chính xác phân loại trạng thái mắt và khả năng theo dõi buồn ngủ trong chuỗi thời gian. Điều này khẳng định tính phù hợp của hướng tiếp cận sử dụng kết hợp nhiều mạng nơ-ron chuyên biệt trong cùng một hệ thống. Tuy nhiên, bên cạnh những kết quả đạt được, hệ thống vẫn còn tồn tại những hạn chế nhất định cần tiếp tục được nghiên cứu và hoàn thiện trong tương lai.

Về hướng phát triển, trước hết cần chú trọng đến khả năng mở rộng và triển khai thực tế. Hệ thống hiện mới dừng lại ở mức cài đặt dưới dạng mã nguồn chạy trên máy tính cá nhân; trong các nghiên cứu tiếp theo, việc nhúng (embed) mô hình vào các nền tảng phần cứng như thiết bị di động, hệ thống giải trí trên ô tô hoặc các bộ điều khiển nhúng chuyên dụng sẽ được xem xét. Khi đó, hệ thống có thể hoạt động trực tiếp trên phương tiện, giám sát liên tục trạng thái mệt mỏi của người lái và đưa ra cảnh báo kịp thời, đồng thời có thể tích hợp thêm các chức năng hỗ trợ như gợi ý điểm dừng nghỉ, nhà nghỉ phù hợp giúp tài xế tạm thời nghỉ ngơi trước khi tiếp tục hành trình, góp phần giảm thiểu nguy cơ tai nạn giao thông. Bên cạnh khả năng triển khai, độ chính xác của hệ thống cũng là yếu tố cần được cải thiện. Trong tương lai, có thể khảo sát và thử nghiệm thêm các kiến trúc học sâu tiên tiến khác hoặc các mô hình lai ghép nhằm nâng cao hơn nữa tỷ lệ nhận diện đúng trạng thái mắt và

mức độ buồn ngủ. Ngoài ra, việc kết hợp thêm các đặc trưng khuôn mặt khác, chẳng hạn như tương quan giữa vị trí lông mày và mắt, giữa miệng và mắt hay các chỉ báo vi biểu cảm (micro-expression), cũng là một hướng tiềm năng để tăng độ tin cậy của mô hình. Sự kết hợp đa dạng các nguồn thông tin hình ảnh sẽ giúp hệ thống đánh giá chính xác hơn trạng thái tỉnh táo của người lái, qua đó nâng cao mức độ an toàn trong quá trình điều khiển phương tiện.

TÀI LIỆU THAM KHẢO

[1] AGGARWAL, C. *Neural Networks and Deep Learning: A Textbook*, 1st ed. 2018 ed. Springer, 2018.

[2] ALBAWI, S., MOHAMMED, T. A., AND AL-ZAWI, S. Tìm hiểu về một mạng nơ-ron tích chập. Trong *Hội nghị Quốc tế về Kỹ thuật và Công nghệ (ICET) 2017* (2017), tr. 1–6.

[3] ARBIB, M. Warren mcculloch's search for the logic of the nervous system. *Perspectives in biology and medicine 43* (02 2000), tr. 193–216.

[4] DANISMAN, T., BILASCO, I. M., DJERABA, C., AND IHADDADENE, N. Hệ thống phát hiện tài xế buồn ngủ dựa trên mẫu nháy mắt. Trong *Hội nghị Quốc tế về Trí tuệ Máy và Web 2010* (2010), tr. 230–233.

[5] FUSEK, R. Pupil localization using geodesic distance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11241 LNCS* (2018), 433–444.

[6] JUNAEDI, S., AND AKBAR, H. Driver Drowsiness Detection Based on Face Feature and PERCLOS. *Journal of Physics: Conference Series 1090* (2018), 012037.

[7] KETKAR, N., AND MOOLAYIL, J. Mạng nơ-ron tích chập. 04 2021, tr. 197–242.

[8] KONSTANTINIDIS, D., DIMITROPOULOS, K., AND DARAS, P. Cách tiếp cận học sâu để phân tích video và đặc trưng khung xương trong nhận dạng ngôn ngữ ký hiệu. Trong *Hội nghị IEEE Quốc tế về Hệ thống và Kỹ thuật Ảnh (IST) 2018* (2018), tr. 1–6.

[9] KRAJEWSKI, J., SOMMER, D., TRUTSCHEL, U., EDWARDS, D., AND GOLZ, M. Steering wheel behavior based estimation of fatigue.

[10] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems 25* (01 2012).

[11] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM 60*, 6 (2017), 84–90.

[12] MARDI, Z., ASHTIANI, S., AND MIKAEILI, M. Eeg-based drowsiness detection for safe driving using chaotic features and statistical tests. *Journal of medical signals and sensors 1* (05 2011), 130–7.

[13] MARQUART, G., CABRALL, C., AND DE WINTER, J. Tổng quan các thước đo liên quan đến mắt về tải công việc tinh thần của người lái. *Procedia Manufacturing* 3 (2015), 2854–2861. Hội nghị Quốc tế lần thứ 6 về Các Yếu tố Con người và Công thái học Ứng dụng (AHFE 2015) và các hội nghị liên kết, AHFE 2015.

[14] CƠ QUAN QUẢN LÝ AN TOÀN GIAO THÔNG ĐƯỜNG BỘ QUỐC GIA (CHÍNH PHỦ). Lái xe khi buồn ngủ. *https://www.nhtsa.gov/risky-driving/drowsy-driving.*

[15] OLISAH, C., AND SMITH, L. Understanding unconventional preprocessors in deep convolutional neural networks for face identification. *SN Applied Sciences 1*, 11 (2019).

[16] QING, W., BINGXI, S., BIN, X., AND JUNJIE, Z. Ứng dụng nhận dạng mệt mỏi của tài xế dựa trên PERCLOS cho không gian xe thông minh. *Hội thảo Quốc tế lần thứ 3 về Xử lý Thông tin 2010* (2010).

[17] SANIL, N., VENKAT, P. A. N., RAKESH, V., MALLAPUR, R., AND AHMED, M. R. ác kỹ thuật học sâu cho phát hiện và tránh chướng ngại vật trong xe tự lái. Trong *Hội nghị Quốc tế về Trí tuệ Nhân tạo và Xử lý Tín hiệu (AISP) 2020* (2020), tr. 1–4.

[18] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks 61* (04 2014).

[19] SUN, Y., LIANG, D., WANG, X., AND TANG, X. Deepid3: Face recognition with very deep neural networks.

[20] SUN, Y., WANG, X., AND TANG, X. Hybrid deep learning for face verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 38* (10 2016), 1997– 2009.

[21] TAIGMAN, Y., YANG, M., RANZATO, M., AND WOLF, L. Deepface: Thu hẹp khoảng cách tới mức hiệu năng xác thực khuôn mặt tương đương con người. Trong *Hội nghị IEEE về Thị giác Máy tính và Nhận dạng Mẫu (CVPR) 2014* (2014), tr. 1701–1708.

[22] WADHAWAN, A., AND KUMAR, P. Deep learning-based sign language recognition system for static signs. *Neural Computing and Applications 32*, 12 (2020), 7957–7968.

[23] WU, Y., AND JI, Q. Khung hồi quy tầng (cascade regression) liên kết có ràng buộc cho nhận dạng đồng thời các đơn vị hành động khuôn mặt (AU) và phát hiện mốc đặc trưng khuôn mặt. Trong *Kỷ yếu Hội nghị IEEE về Thị giác Máy tính và Nhận dạng Mẫu (CVPR)* (06 2016).

[24] YAMASHITA, R., NISHIO, M., DO, R., AND TOGASHI, K. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging 9*, 4 (2018), 611–629.

[25] ZHANG, K., ZHANG, Z., LI, Z., AND QIAO, Y. Joint face detection and alignment using multitask cascaded convolutional networks. vol. 23, pp. 1499–1503.

[26] ZHANG, S., CHI, C., LEI, Z., AND LI, S. Z. Refineface: Refinement neural network for high performance face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1.

15% Tính tương đồng nói chung

Tổng cộng của tất cả các kết quả trùng khớp, bao gồm cả các nguồn trùng lặp, cho mỗi c...

Đã lọc khỏi Báo cáo

- Mục lục tham khảo
- Văn bản được trích dẫn
- Kết quả trùng khớp nhỏ (ít hơn 8 từ)

Nguồn hàng đầu

9% Nguồn Internet
6% Ấn bản
8% Bài tập được nộp (bài của học sinh)

Đỗ Tiến Dũng

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Họ và tên học viên: Cao Minh Quyền

Chuyên ngành: Khoa học máy tính

Khóa: 2024

Tên đề tài: Nghiên cứu hệ thống theo dõi mắt để phát hiện tài xế không tỉnh táo bằng cách sử dụng Deep Learning

Người hướng dẫn khoa học: TS. Đỗ Tiến Dũng

Ngày bảo vệ: 24/0162026

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Bổ sung phân tích thiết kế hệ thống	Đã bổ sung vào chương 3, mục 3.1
2	Chỉnh sửa bộ dữ liệu sử dụng cho đề án	Lược bỏ bộ dữ liệu quay các sinh viên và thêm bộ dữ liệu UTA Real-Life Drowsiness Dataset

Hà Nội, ngày 06 tháng 02 năm 2026

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
PGS.TS. Lê Hữu Lập	TS. Nguyễn Xuân Đức	TS. Đỗ Tiến Dũng	Cao Minh Quyền

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 3299/QĐ-HV ngày 26 tháng 12 năm 2025 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi ..8.. giờ ..15.. phút, ngày 24 tháng 01 năm 2025 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Cao Minh Quyền**

Tên đề án tốt nghiệp: **Nghiên cứu hệ thống theo dõi mắt để phát hiện tài xế không tỉnh táo bằng cách sử dụng Deep Learning**

Ngành: **Khoa học máy tính** Mã số: **8.48.01.01**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ..05../ 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**PGS.TS. Lê Hữu Lập**	Chủ tịch	
2	**TS. Nguyễn Xuân Đức**	Thư ký	
3	**PGS.TS. Đỗ Năng Toàn**	Phản biện 1	
4	**TS. Lê Văn Vịnh**	Phản biện 2	
5	**PGS.TS. Nguyễn Quang Uy**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên (Có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

Rằng buộc của hình ảnh thu nhận là gì?
Nêu lý do lựa chọn phân tích thiết kế hệ thống như trong đề án
Giải thích bảng 4.1: tại sao tốc độ của Tensor Flow suy giảm đáng kể?

7. Trả lời của học viên:

Sử dụng hình ảnh từ dữ liệu thực tế của Mỹ

Hệ thống chưa chạy mô phỏng, chưa đưa vào ứng dụng thực tế

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: TS. Nguyễn Xuân Đức
  2. Uỷ viên Ban kiểm phiếu: PGS.TS. Nguyễn Quang Vy
  3. Uỷ viên Ban kiểm phiếu: TS. Lê Văn Vinh
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 7,0

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

Bổ sung phân tích thiết kế hệ thống

Chỉnh sửa bộ dữ liệu tự xây dựng cho phù hợp đề án

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên: Đề nghị công nhận và cấp bằng

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.... Không

Buổi làm việc kết thúc vào... 9h00..... cùng ngày.

Chủ tịch	**Thư ký**
PGS.TS. Lê Hữu Lập	**TS. Nguyễn Xuân Đức**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

(Dùng cho người phản biện)

Tên đề tài đề án tốt nghiệp: NGHIÊN CỨU HỆ THỐNG THEO DÕI MẮT ĐỂ PHÁT HIỆN TÀI XẾ KHÔNG TỈNH TÁO BẰNG CÁCH SỬ DỤNG DEEP LEARNING

Chuyên ngành: Hệ thống thông tin

Mã chuyên ngành: 8.48.01.04

Họ và tên học viên: CAO MINH QUYỀN

Họ và tên người nhận xét: Lê Văn Vịnh

Học hàm, học vị: Tiến sĩ

Chuyên ngành: Mạng máy tính

Cơ quan công tác: Học viện Công nghệ Bưu chính Viễn thông

Số điện thoại: 0988928728 E-mail: vinhlv@ptit.edu.vn

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

Khi chất lượng hoặc thời lượng giấc ngủ không được đảm bảo đầy đủ, một trong những hậu quả dễ gây ra tình trạng mệt mỏi và buồn ngủ, trong các hoạt động đòi hỏi sự tập trung cao như lái xe, buồn ngủ và mất tỉnh táo khi lái xe được xem là một trong những nguyên nhân quan trọng gây ra tai nạn giao thông. Tài xế có thể có các biểu hiện như: mắt nhắm lâu, tần suất chớp mắt tăng, gật đầu, phản xạ chậm với tình huống bất ngờ.

Những khoảng thời gian mất tập trung chỉ kéo dài vài giây cũng đủ gây ra va chạm với thiệt hại lớn về người và tài sản. Đề án này tập trung nghiên cứu và xây dựng một hệ thống theo dõi mắt nhằm phát hiện tình trạng buồn ngủ của tài xế dựa trên hình ảnh thời gian thực.

II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được:

Về mặt lý thuyết, đề án đã trình bày các khái niệm buồn ngủ và các phương pháp phát hiện buồn ngủ, các phương pháp trong mạng No-ron, thiết kế được mô hình hệ thống và triển khai thử nghiệm.

Đề án đã xây dựng được hệ thống thiết kế theo pipeline xử lý khung hình (frame) từ camera đến quyết định cảnh báo gồm 4 khối: (1) thu nhận hình ảnh; (2) phát hiện khuôn mặt và landmark bằng MTCNN; (3) trích xuất vùng mắt và phân loại mở/nhắm bằng CNN; (4) tính PERCLOS theo cửa sổ thời gian và kích hoạt cảnh báo.

Kế quả thực nghiệm cho thấy mô hình kết hợp dựa trên MTCNN và CNN của đề án đạt hiệu năng tốt hơn so với việc sử dụng riêng lẻ các mô hình nơ-ron cơ bản, cả về độ chính xác phân loại trạng thái mắt và khả năng theo dõi buồn ngủ trong chuỗi thời gian.

III/ Những vấn đề cần giải thích thêm:

Trong bảng 4.1, từ viết tắt FPS không có trong danh mục từ viết tắt. Tác giả hãy giải thích từ viết tắt đó có ý nghĩa là gì.

Tác giả hãy giải thích trong trang 44, kết quả đo tốc độ xử lý của hệ thống khi: chỉ sử dụng camera, khi kết hợp với MTCNN và TensoFlow và khi kết hợp MTCNN với Pytorch. "Tốc độ suy giảm đáng kể, chỉ còn vài khung hình mỗi giây".

IV/ Kết luận:

Đề án đã xây dựng được hệ thống theo dõi mắt phát hiện tình trạng buồn ngủ của người lái xe được xây dựng trên cơ sở kết hợp nhiều mô hình mạng nơ-ron như: Mạng nơ-ron tích chập (CNN) và Mạng nơ-ron tích chập xếp tầng đa nhiệm (MTCNN).

Đề án đã chứng minh được mô hình đề xuất của đề án đạt hiệu năng tốt hơn về độ chính xác phân loại trạng thái mắt và khả năng theo dõi buồn ngủ so với việc sử dụng riêng lẻ các mô hình nơ-ron cơ bản.

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp.

Ngày 19 tháng 01 năm 2026

NGƯỜI NHẬN XÉT



Lê Văn Vịnh

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT LUẬN VĂN TỐT NGHIỆP THẠC SĨ

(Dũng cho người phản biện)

Tên đề tài luận văn: Nghiên cứu hệ thống theo dõi mắt để phát hiện tài xế không tỉnh táo bằng cách sử dụng deep learning

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.01

Tên học viên: Cao Minh Quyền

Họ và tên người nhận xét: Đỗ Năng Toàn

Học hàm, học vị: PGS. TS

Chuyên ngành: Khoa học máy tính

Cơ quan công tác: Viện Công nghệ thông tin–Viện Hàn lâm KH&CN Việt Nam

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài

Phát hiện buồn ngủ và các dấu hiệu mất tập trung trong khi lái xe là vấn đề có tính thực tiễn cao.

Luận văn nhằm hệ thống hóa một số vấn đề trong xây dựng hệ thống theo dõi mắt để phát hiện tài xế không tỉnh táo là vấn đề có ý nghĩa.

II/ Về nội dung, chất lượng của luận văn, các kết quả đạt được (so với đề cương đã được duyệt)

Nội dung chính của luận văn là trình bày khái quát về mạng tích chập CNN và bài toán phát hiện buồn ngủ, thiết kế mô hình hệ thống đánh giá mức độ buồn ngủ dựa vào camera, cài đặt chương trình thử nghiệm theo mô hình đã hệ thống hóa được.

Bố cục luận văn tương đối phù hợp, nêu được nội dung chính của luận văn. Tài liệu tham khảo đầy đủ cần hiệu chỉnh và trích dẫn đầy đủ trong luận văn.

Nội dung luận văn phù hợp với đề cương đã được duyệt.

Không nên đánh số với nội dung chỉ có 01 mục.

III/ Những vấn đề cần giải thích thêm

Thiết kế hệ thống cần mô tả những gì?

Tính toán trạng thái buồn ngủ dựa trên trạng thái mắt sử dụng bao nhiêu thuộc tính, làm thế nào bảo đảm yêu cầu thời gian thực?

IV/ Kết luận

Những kết quả nêu ra trong bản luận văn của học viên Cao Minh Quyền đáp ứng được những yêu cầu cơ bản của một bản luận văn Thạc sĩ kỹ thuật định hướng ứng dụng, chuyên ngành: Khoa học máy tính, Mã số: 8.48.01.01.

Tôi đồng ý để học viên Cao Minh Quyền được bảo vệ luận văn trước Hội đồng chấm luận văn thạc sĩ.

Hà Nội, ngày 19 tháng 01 năm 2026

CÁN BỘ PHẢN BIỆN

(ký và ghi rõ họ tên)



PGS. TS. Đỗ Năng Toàn

Câu hỏi:

1) Thiết kế hệ thống ở chương 2 có cần phân tích thiết kế các ca sử dụng của hệ thống phần mềm như thông thường không? Tại sao?

2) Ràng buộc hình ảnh thu nhận nhận phục vụ cho hệ thống là thế nào?