

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP
NGÀNH KHOA HỌC MÁY TÍNH

Đề tài

DATM: ĐIỀU KHIỂN THIẾT BỊ IOT VỚI SỰ
GIÁM SÁT CỦA TRÍ TUỆ NHÂN TẠO

Sinh viên thực hiện:
Trần Minh Đạt
Mã số: B1913221
Khóa: 45

Cần Thơ, 12/2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP
NGÀNH KHOA HỌC MÁY TÍNH

Đề tài

**DATM: ĐIỀU KHIỂN THIẾT BỊ IOT VỚI SỰ
GIÁM SÁT CỦA TRÍ TUỆ NHÂN TẠO**

Giảng viên hướng dẫn:
TS. Mã Trường Thành

Sinh viên thực hiện:
Trần Minh Đạt
Mã số: B1913221
Khóa: 45

Cần Thơ, 12/2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

XÁC NHẬN CHỈNH SỬA LUẬN VĂN
THEO YÊU CẦU CỦA HỘI ĐỒNG

Tên luận văn (tiếng Việt và tiếng Anh): **DATM: Điều khiển thiết bị IoT với sự giám sát của Trí tuệ nhân tạo (Controlling IoT Devices with Artificial Intelligence Monitoring)**

Họ tên sinh viên: **Trần Minh Đạt**

MASV: **B1913221**

Mã lớp: **DI19Z6A1**

Đã báo cáo tại hội đồng ngành: **Khoa học máy tính**

Ngày báo cáo: **14/12/2023**

Hội đồng báo cáo gồm:

1. **TS. Trần Việt Châu**

Chủ tịch hội đồng

2. **ThS. Huỳnh Ngọc Thái Anh**

Thành viên

3. **TS. Mã Trường Thành**

Thư ký

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

Cần Thơ, ngày tháng 12 năm 2023

Giáo viên hướng dẫn

(Ký và ghi họ tên)

NHẬN XÉT CỦA GIẢNG VIÊN

LỜI CẢM ƠN

Để có được bài luận văn này, tôi xin được bày tỏ lòng biết ơn chân thành và sâu sắc đến Thầy Mã Trường Thành, người đã trực tiếp tận tình hướng dẫn, giúp đỡ tôi. Trong suốt quá trình thực hiện luận văn, nhờ những sự chỉ bảo và hướng dẫn quý giá đó mà bài luận văn này được hoàn thành một cách tốt nhất.

Ngoài ra, tôi xin chân thành cảm ơn bạn Trần Thanh Huy là bạn đã tạo điều kiện cho tôi quay video để có được dữ liệu về ba tư thế, bao gồm: đứng, nằm, ngồi. Điều này góp phần cho bài luận văn được hoàn thiện một cách tốt nhất.

Đồng thời, tôi cũng xin gửi lời cảm ơn chân thành đến các Thầy, Cô trường Đại học Cần Thơ, đặc biệt là các Thầy, Cô ở Trường CNTT&TT, những người đã truyền đạt những kiến thức, giúp tôi tích lũy kinh nghiệm quý báu qua hơn bốn năm đại học để đủ khả năng, kinh nghiệm hoàn thành đề tài: “**DATM: điều khiển thiết bị IoT với sự giám sát của Trí tuệ nhân tạo**”.

Cuối cùng, tôi xin chân thành cảm ơn gia đình, bạn bè đã luôn động viên, khích lệ và tạo điều kiện giúp đỡ trong suốt quá trình thực hiện để tôi có thể hoàn thành bài luận văn một cách tốt nhất.

Cần Thơ, ngày tháng năm 2023

Người viết

Trần Minh Đạt

LỜI CAM ĐOAN

Tôi tên Trần Minh Đạt, sinh viên ngành Khoa học máy tính, khóa 45. Tôi xin cam đoan luận văn với đề tài “**DATM: điều khiển thiết bị IoT với sự giám sát của Trí tuệ nhân tạo**” (*Controlling IoT Devices with Artificial Intelligence Monitoring*) là công trình luận văn của bản thân tôi, được sự hướng dẫn bởi TS. Mã Trường Thành.

Các thông tin được sử dụng tham khảo trong luận văn được thu thập từ các nguồn đáng tin cậy, đã được kiểm chứng, được công bố rộng rãi và được tôi trích dẫn nguồn gốc rõ ràng ở phần tài liệu tham khảo. Các kết quả nghiên cứu được trình bày trong luận văn này là do chính tôi thực hiện một cách nghiêm túc, trung thực và không trùng lặp với các đề tài khác đã được công bố trước đây.

Cần Thơ, ngày tháng năm 2023

Người cam đoan

Trần Minh Đạt

MỤC LỤC

MỤC LỤC	1
DANH MỤC HÌNH	3
DANH MỤC BẢNG	4
DANH MỤC BIỂU ĐỒ	5
DANH MỤC CÁC TỪ VIẾT TẮT	6
TÓM TẮT	7
ABSTRACT	8
PHẦN GIỚI THIỆU.....	9
1. Đặt vấn đề	9
2. Lịch sử giải quyết vấn đề.....	10
2.1 Nhận diện hành động của người với Deep Learning	10
2.2. Xây dựng huấn luyện viên thể hình AI bằng Mediapipe để phân tích Squats (Build an AI Fitness Trainer using Mediapiipy to Analyze Squats).....	12
3. Mục tiêu đề tài	12
3.1. Mục tiêu chung	12
3.2. Mục tiêu chi tiết.....	13
4. Phạm vi nghiên cứu	13
4.1. Đối tượng nghiên cứu	13
4.2. Phạm vi nghiên cứu	13
5. Phương pháp nghiên cứu	14
5.1. Phương pháp nghiên cứu tài liệu	14
5.2. Phương pháp thực nghiệm.....	14
5.3. Phương pháp lấy ý kiến	15
6. Kết quả đạt được	16
7. Bố cục luận văn	17
PHẦN NỘI DUNG	18
CHƯƠNG 1.....	18
MÔ TẢ BÀI TOÁN	18
1.1. Mô tả chi tiết bài toán	18
1.2. Vấn đề và giải pháp liên quan đến bài toán	18
1.2.1. Các tư thế ngủ quên thông thường	18
1.2.2. Mạng nơ-ron	19
1.2.2.1. Mạng nơ-ron nhân tạo	19
1.2.2.2. Mạng nơ-ron tích chập	20
1.2.2.3. Kiến trúc LeNet-5 CNN	20
1.2.3. SVM (Support vector machines)	21
1.2.4. KNN (K-nearest neighbor)	22
1.2.5. YOLOv7	23

1.2.6. Human Pose Estimation	24
1.2.7. Tuya API	25
CHƯƠNG 2.....	27
THIẾT KẾ VÀ CÀI ĐẶT	27
2.1. Thiết kế hệ thống	27
2.1.1. Quá trình cài đặt và thực hiện đề tài	27
2.1.2. Thư mục các tư thế	27
2.1.3. Xây dựng mô hình huấn luyện phân lớp tư thế	28
2.1.4. Xây dựng mô hình huấn luyện xác định ngủ	29
2.1.5. Thiết kế giải thuật xác định ngủ hay thức dựa vào việc tính ngưỡng:	31
2.2. Mã giả xây dựng bộ đặc trưng và cài đặt hệ thống.....	31
2.2.1 Mã giả xây dựng bộ đặc trưng.....	31
2.2.2 Mã giả cài đặt hệ thống.....	33
2.3. Cài đặt giải thuật.....	34
2.3.1. Dụng cụ thu thập dữ liệu	34
2.3.2. Thu thập dữ liệu.....	34
2.3.3. Cắt ảnh và tiền xử lý dữ liệu.....	36
2.3.4. Phân bổ dữ liệu huấn luyện	37
2.2.5. Xây dựng và huấn luyện mô hình.....	37
2.2.5.1. Mô hình LeNet-5	37
2.2.6. Cài đặt Tuya API	38
CHƯƠNG 3.....	45
KẾT QUẢ, ĐÁNH GIÁ VÀ GIAO DIỆN	45
3.1. Môi trường thực nghiệm.....	45
3.2. Kết quả kiểm tra, đánh giá	63
3.2.1. Kết quả phân lớp ba tư thế.....	63
3.2.2. Kết quả mô hình xác định trạng thái ngủ.....	67
3.2.3. Kết quả xác định trạng thái ngủ theo phương pháp chọn ngưỡng.....	73
3.3. Giao diện	66
PHẦN KẾT LUẬN	69
1. Kết quả đạt được	69
2. Hướng phát triển	69
TÀI LIỆU THAM KHẢO.....	70

DANH MỤC HÌNH

Hình 1 - Kiến trúc tổng quát cho bài toán nhận diện Human Pose.	11
Hình 2 - Minh họa 6 hành động.....	11
Hình 3 - Giao diện hệ thống phân tích Squats	12
Hình 4 - Bảng thống kê tham khảo ý kiến.....	16
Hình 5 - Mô tả hoạt động hệ thống.....	18
Hình 6 - Các tư thế ngủ quên.....	19
Hình 7 - Minh họa cấu trúc của một mạng nơ ron nhân tạo	20
Hình 8 - Sơ đồ trên mô tả về kiến trúc LeNet-5.....	20
Hình 9 - Thực hiện phân loại nhị phân và đa lớp trên tập dữ liệu	22
Hình 10 - KNN cho bài toán regression	23
Hình 11 - Minh họa so sánh các phiên bản của YOLO	24
Hình 12 - Minh họa độ tin cậy và so sánh giữa các giải thuật.....	25
Hình 13 - Cấu trúc liên kết pose landmarks	25
Hình 14 - Nhận yêu cầu và phản hồi về các thiết bị điện	26
Hình 15 - Sơ đồ quá trình cài đặt và thực hiện đề tài	27
Hình 16 - Mô tả cây thư mục các tư thế để huấn luyện	27
Hình 17 - Quá Trình huấn luyện mô hình	28
Hình 18 - Quá trình tạo dữ liệu mới cho mô hình phân lớp ngủ	29
Hình 19 - Tập dữ liệu sau khi xây dựng	30
Hình 20 - Tập dữ liệu đã được gom 5 dòng và được gán nhãn	30
Hình 21 - Xây dựng mô hình huấn luyện xác định ngủ hay thức	31
Hình 22 - Minh họa quá trình thu dữ liệu.....	35
Hình 23 - Minh họa 3 tư thế đã được crop ra từ video thu được	35
Hình 24 - Ảnh minh họa cắt và tiền xử lý	36
Hình 25 - Tổng số lượng hình ảnh thu thập.....	37
Hình 26 - Mô hình huấn luyện LeNet-5	37
Hình 27 - Compile và fit() mô hình LeNet-5.....	38
Hình 28 - Trang chủ Tuya IoT	39
Hình 29 - Giao diện trang dự án mới tạo Tuya.....	39
Hình 30 - Giao diện chọn devices kết nối	40
Hình 31 - Giao diện thêm device.....	41
Hình 32 - Giao diện tạo account cho app mobile	41
Hình 33 - Giao diện thêm account mobile thành công	41
Hình 34 - Giao diện đăng nhập trên app mobile.....	42
Hình 35 - Giao diện add device kết nối wifi app mobile.....	43
Hình 36 - Giao diện web khi kết nối thiết bị thành công.....	43
Hình 37 - Giao diện app mobile khi kết nối thành công.....	44
Hình 38 - Giao diện hệ thống	66
Hình 39 - Giao diện khi nhấn nút mở file.....	67
Hình 40 - Giao diện khi mở file thành công.....	68
Hình 41 - Giao diện khi hệ thống đã xác định được trạng thái của đối tượng sau một khoảng thời gian	68

DANH MỤC BẢNG

Bảng 1 - Thư viện sử dụng	45
Bảng 2 - Thể hiện so sánh đánh giá các mô hình phân lớp ba tư thế	63
Bảng 3 - Thể hiện so sánh thời gian huấn luyện và dự đoán phân lớp ba tư thế	65
Bảng 4 - Mô hình kết quả SVM	67
Bảng 5 - Mô hình kết quả KNN	68
Bảng 6 - Mô hình kết quả RandomForest.....	68
Bảng 7 - Mô hình kết quả 1D-ANN	69
Bảng 8 - Bảng thống kê kết quả với các ngưỡng khác nhau đối với file ngũ.....	73
Bảng 9 - Bảng thống kê kết quả với các ngưỡng khác nhau đối với file thức	73

DANH MỤC BIỂU ĐỒ

Biểu đồ 1 - So sánh kết quả các đánh giá giữa các mô hình phân lớp ba tư thế.....	64
Biểu đồ 2 - Biểu đồ so sánh thời gian huấn luyện và kiểm tra mô hình phân lớp ba tư thế.....	66
Biểu đồ 3 - Biểu đồ kết quả đánh giá các mô hình xác định ngủ với 12 đặc trưng.....	70
Biểu đồ 4 - Biểu đồ kết quả đánh giá các mô hình xác định ngủ với 14 đặc trưng.....	71
Biểu đồ 5 - Biểu đồ kết quả đánh giá các mô hình xác định ngủ với 20 đặc trưng.....	72

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Từ viết tắt	Ngôn ngữ	Diễn giải
1	AI	Tiếng Anh	Artificial Intelligence
2	IOT	Tiếng Anh	Internet of Things
3	DATM	Tiếng Anh	Devices – AI – IoT - Monitoring
4	LSTM	Tiếng Anh	Long Short-Term Memory
5	NLP	Tiếng Anh	Natural Language Processing
6	CNN	Tiếng Anh	Convolutional Neural Networks
7	ML	Tiếng Anh	Machine Learning
8	DL	Tiếng Anh	Deep Learning
9	GPU	Tiếng Anh	Graphics Processing Unit
10	SVM	Tiếng Anh	Support Vector Machines
11	KNN	Tiếng Anh	K-Nearest Neighbors
12	MNIST	Tiếng Anh	Modified National Institute Of Standards And Technology
13	GUI	Tiếng Anh	Graphical User Interface
14	HPE	Tiếng Anh	Human Pose Estimation
15	SSD	Tiếng Anh	Solid State Drive
16	AMD	Tiếng Anh	Advanced Micro Devices
17	ACC	Tiếng Anh	Accuracy
18	REC	Tiếng Anh	Recall
19	PREC	Tiếng Anh	Precision
20	DGMH	Tiếng Anh	Đánh giá mô hình
21	ROI	Tiếng Anh	Region Of Interest
22	HAR	Tiếng Anh	Human Action Recognition
23	YOLO	Tiếng Anh	You Only Look One
24	RGB	Tiếng Anh	Red, Green, Blue
25	BB	Tiếng Anh	Bounding Box
26	CNTT	Tiếng Việt	Công nghệ thông tin
27	CNTT & TT	Tiếng Việt	Công nghệ thông tin và truyền thông
28	XDUD	Tiếng Việt	Xây dựng ứng dụng

TÓM TẮT

Việc tiêu thụ điện năng ở Trái Đất nói chung và Việt Nam chúng ta nói riêng đang ngày càng được báo động bởi việc tiêu thụ điện một cách lãng phí và gây ra hậu quả ngày càng ảnh hưởng đến môi trường cũng như thời tiết biến đổi thất thường. Đôi khi chính chúng ta đã quên mất việc tắt thiết bị điện khi không sử dụng lại là một yếu tố góp phần hạn chế tiêu thụ điện năng cho gia đình. Vì vậy đề tài này tập trung vào việc sẽ áp dụng Trí tuệ nhân tạo (AI) để giải quyết vấn đề giúp gia đình hạn chế việc tiêu thụ điện năng lãng phí cụ thể là thông qua bài toán để dự đoán trạng thái ngủ hay thức của con người và tự động tắt các thiết bị điện khi không còn nhu cầu sử dụng chẳng hạn như tivi, đèn, ... Đề tài tuy có vẻ đơn giản khi chỉ dự đoán trạng thái ngủ hay thức của con người nhưng lại là một thách thức cho những người nghiên cứu như chúng tôi vì để xác định một người có ngủ hay không qua góc nhìn của camera là rất khó. Vì vậy chúng tôi đề xuất một hệ thống mang tên “DATM”, có thể dự đoán con người có ngủ hay không qua một khoảng thời gian được theo dõi từ camera sau đó sẽ ra lệnh tắt thiết bị điện khi con người đã ở trạng thái ngủ. Ý tưởng chính của DATM là tận dụng Human Pose Estimation (HPE) để xác định các điểm trên cơ thể có khả năng vận động, di chuyển nhiều khi còn thức hay thói quen sinh hoạt hằng ngày. Sau khi đã xác định các điểm hoàn thành DATM sẽ tiến hành theo dõi các điểm này trong một khoảng thời gian xem các điểm có di dời rời khỏi phạm vi DATM quy định hay không từ đó sẽ xác định người đó có ngủ hay không. Trước khi sử dụng HPE, DATM sẽ qua một mô hình phân lớp 3 tư thế cơ bản của một người đó là đứng, nằm, ngồi. Khi tư thế là nằm thì sẽ được thực thi tiếp với HPE, ngược lại sẽ luôn xác định người đó đang thức. Chúng tôi đã xây dựng mô hình phân lớp tư thế với tổng dữ liệu là 10100 ảnh. Sau khi thực nghiệm với các kiến trúc khác nhau của mô hình CNN thì tôi nhận thấy độ chính xác của kiến trúc Lenet-5 đạt kết quả có độ chính xác tổng thể với tập kiểm tra từ 98% - 100% thì phù hợp để kết hợp với mô hình xác định trạng thái ngủ của SVM. Hơn nữa, chúng tôi cũng đã đóng góp một Framework chung với việc giám sát camera, cụ thể là giám sát các tư thế của con người và xác định họ có đang trong trạng thái ngủ hay không cũng như giám sát luôn cả thiết bị điện đang tắt hay mở. Để sử dụng hệ thống DATM một cách đơn giản chúng tôi đã xây dựng một giao diện windows thân thiện dễ sử dụng và theo dõi cho việc tiếp cận ứng dụng lần đầu tiên.

ABSTRACT

The escalating energy consumption globally, and particularly in Vietnam, has sparked alarm due to the increasingly wasteful use of electricity and its consequential environmental impacts, including unpredictable weather patterns. At times, we may overlook the fact that turning off electrical devices when not in use is a contributing factor to reducing electricity consumption in households. Hence, this research project is dedicated to the application of Artificial Intelligence (AI) to address this issue, with the goal of assisting families in curbing unnecessary electricity usage. Specifically, the project addresses the problem by predicting the sleep or wake state of person and autonomously powering down electronic devices no longer in demand, such as televisions and lights... The topic introduces a system named 'DATM,' engineered to forecast whether the person is asleep based on an observation period from a camera, followed by issuing a directive to power off electronic devices once that person is identified to be in a sleep state. The core concept of DATM revolves around harnessing Human Pose Estimation (HPE) to recognize points on the body indicative of movement, signifying routine activities during wakefulness. Post identifying these points, DATM tracks their movement over a designated period to discern whether the individual is asleep or not. Preceding the use of HPE, DATM undergoes a posture classification model for fundamental human postures: standing, lying, and sitting. When the posture is lying down, it earns with HPE; otherwise, it continuously concludes that the person is awake. We constructed a posture classification model with a comprehensive dataset including **10,100 images**. After experimenting with various architectures of CNN models, I observed that the accuracy of the Lenet-5 architecture yielded overall results with an accuracy range of **98% - 100% on the test set**. This makes it suitable for integration with the SVM-based sleep state detection model. Furthermore, we have also contributed to a common framework by monitoring cameras, specifically observing human postures and determining whether they are in a sleep state, as well as monitoring the status of electrical devices (on or off). To facilitate the straightforward use of the DATM system, we have constructed a user-friendly Windows interface for easy operation and monitoring for users during their first encounter with the application.

PHẦN GIỚI THIỆU

1. Đặt vấn đề

Trong bối cảnh nguồn năng lượng có hạn và là mối lo ngại về biến đổi khí hậu ngày càng nghiêm trọng. Việc tiết kiệm điện năng là không thể “coi nhẹ”, nó ảnh hưởng trực tiếp đến khía cạnh về cuộc sống và xã hội ngày nay. Để có thể tránh sử dụng lãng phí điện năng trong gia đình là một thách thức ngày càng lớn và quan trọng. Nhìn chung, gia đình truyền thống ở Việt Nam thường tiêu thụ điện năng một cách lãng phí. Cụ thể, các thiết bị điện phổ biến như: đèn, quạt, máy lạnh, và ấm đun nước thường hoạt động liên tục với nhu cầu của gia đình hoặc được sử dụng ở chế độ tiêu tốn năng lượng cao. Vậy chuyện gì sẽ xảy ra khi các thiết bị điện sẽ được hoạt động với công suất cao và tần suất liên tục như vậy mà lại không có người sử dụng? Trong trường hợp này việc đầu tiên là sẽ dẫn đến tiêu tốn rất nhiều điện năng không cần thiết, ngoài ra chi phí điện tiêu dùng hàng tháng cũng sẽ tăng theo. Từ đó, nó ảnh hưởng trực tiếp đến tài chính của gia đình. Trường hợp thứ hai khi sử dụng với công suất cao và tần suất liên tục như vậy rất dễ xảy ra việc quá tải nhiệt dẫn đến chuyện không mong muốn đến gia đình và các hàng xóm xung quanh.

Nhìn chung, một trong những việc tiết kiệm điện năng khi không sử dụng trong gia đình là một trách nhiệm quan trọng trong mỗi gia đình và cá nhân. Nó không chỉ giúp tiết kiệm tài chính cho gia đình mà còn tác động tích cực lên môi trường và xã hội. Ngoài ra việc tiết kiệm điện lại là một trong những khía cạnh quan trọng nhất vì tiết kiệm điện năng cũng sẽ tiết kiệm được tài chính. Tiết kiệm chi tiêu không cần thiết có thể dẫn đến áp lực tài chính đáng kể, đặc biệt đối với các hộ gia đình có nguồn thu nhập hạn chế. Việc quản lý năng lượng một cách thông minh và tiết kiệm có thể giúp giảm thiểu chi phí này. Không chỉ thế tiết kiệm điện năng cũng đóng góp rất tích cực vào việc bảo vệ môi trường. Việc tiêu tốn nhiều năng lượng không cần thiết gây ra lượng khí nhà kính lớn và góp phần vào biến đổi khí hậu toàn cầu. Bằng cách giảm lãng phí điện năng, chúng ta giúp giảm tác động tiêu cực đối với môi trường tự nhiên, đảm bảo rằng tài nguyên năng lượng tái tạo được bảo tồn và sử dụng hiệu quả. Tóm lại, việc tiết kiệm điện năng không chỉ là một phần quan trọng của cuộc sống hàng ngày mà còn là một trách nhiệm đối với môi trường và xã hội.

Bài toán đặt ra trong nghiên cứu này sẽ liên quan đến “ngủ quên”. Sau khi một ngày làm việc và học tập chăm chỉ và mệt nhọc thì việc “ngủ quên” sẽ không thể tránh khỏi. Một vài tình huống cụ thể như sau: gia đình sẽ xem phim vào buổi tối, khi đó các thiết bị điện như máy điều hoà, tivi, quạt, đèn,... sẽ được bật để phục vụ gia đình. Thông thường, họ sẽ điều chỉnh nhiệt độ và tắt các thiết bị điện cần thiết trước khi ngủ, nhưng một vài trường hợp, họ đã chìm sâu vào giấc ngủ mà các thiết bị vẫn hoạt động. Trường hợp khác, khi đi ra khỏi phòng nhưng lại quên tắt các thiết bị điện hoặc khi đang học bài thì lại ngủ quên. Tuy rằng những tình huống này không thường xảy ra, tuy nhiên, nó

cũng tiêu phí điện năng khá lớn nếu suy xét trong một phạm vi lớn. Câu hỏi đặt ra là “Làm sao ngủ quên mà vẫn có thể hiệu chỉnh các thông số và tắt các thiết bị điện?” “Làm sao để biết chủ nhà có ngủ quên hay không để điều khiển các thiết bị điện?”. Để trả lời cho câu hỏi này, chúng tôi đã lấy cảm hứng từ việc điều khiển các thiết bị IoT.

Hơn nữa, trong thực tế, chính mỗi chúng ta cũng không dễ dàng biết được một người có ngủ quên hay không khi nhìn họ đang trong tư thế nằm. Một cách tự nhiên và cảm tính, chúng ta sẽ dựa vào kinh nghiệm về thời gian, cử động để dự đoán họ có ngủ hay không? Ở đây, kết quả của dự đoán dựa trên “suy nghĩ” của con người vẫn chưa hoàn toàn tin cậy. Vì vậy, làm sao máy tính có thể phát hiện được “chủ nhân” đang ngủ quên?. Thật vậy, hầu hết các gia đình hiện đại cũng đã tin tưởng và cài đặt các thiết bị IoT có tích hợp AI nhằm có một cuộc sống tốt hơn. Nhưng với việc để xác định khi nào nên tắt thiết bị điện và khi nào nên mở thiết bị điện lại là một trong những quyết định cần cân nhắc để tránh lãng phí điện khi không dùng. Để làm được điều này, chúng tôi đã đưa dòng suy nghĩ đến “trí tuệ nhân tạo” để giải quyết bài toán “ngủ quên” này. Cụ thể, tận dụng những camera an ninh quan sát, theo dõi và đưa ra quyết định về ngủ quên dựa trên trí tuệ nhân tạo. Chính xác hơn là thay thế con mắt người bằng camera và thay thế não người bằng các giải thuật trí tuệ nhân tạo.

Vì các lý do đã nêu trên nên tôi chọn thực hiện đề tài “DATM: điều khiển thiết bị IoT với sự giám sát của Trí tuệ nhân tạo”. Ý tưởng chính của đề tài là xây dựng một framework / hệ thống có thể điều khiển được thiết bị điện gia dụng trong gia đình. Đề tài sẽ tập trung chủ yếu vào việc phân tích tư thế nằm của người trong nhà và đưa ra thuật toán để xác định người đang nằm ngủ hay thức dựa trên hình ảnh được thu trực tiếp từ Camera, từ đó sẽ quyết định là tắt hay mở trên thiết bị điện một cách cần thiết. Từ kiến thức tốt nhất của chúng tôi, hiện tại có một số đề tài làm về khía cạnh này, tuy nhiên, chưa có đề tài nào đề cập đến “điều khiển thiết bị IoT với sự giám sát trí tuệ nhân tạo” cho vấn đề về “ngủ quên”. Trong phần kế tiếp chúng tôi sẽ đề cập những nghiên cứu liên quan đến việc nhận diện các cử động của con người.

*Lưu ý: Đề tài này được thực hiện và cài đặt bởi chính tôi, sinh viên Trần Minh Đạt. Tuy nhiên, vì có nhiều người đóng góp vào nghiên cứu về ý tưởng, giải pháp, và kiểm tra tính đúng đắn. Do vậy, những nội dung trong phần kế tiếp của luận văn này (*một số phần*) tôi sẽ sử dụng cụm từ “**chúng tôi**” để thể hiện những ý tưởng, giải pháp do nhiều người đóng góp và sử dụng “**tôi**” để diễn đạt phần thực hiện cá nhân của chính mình.

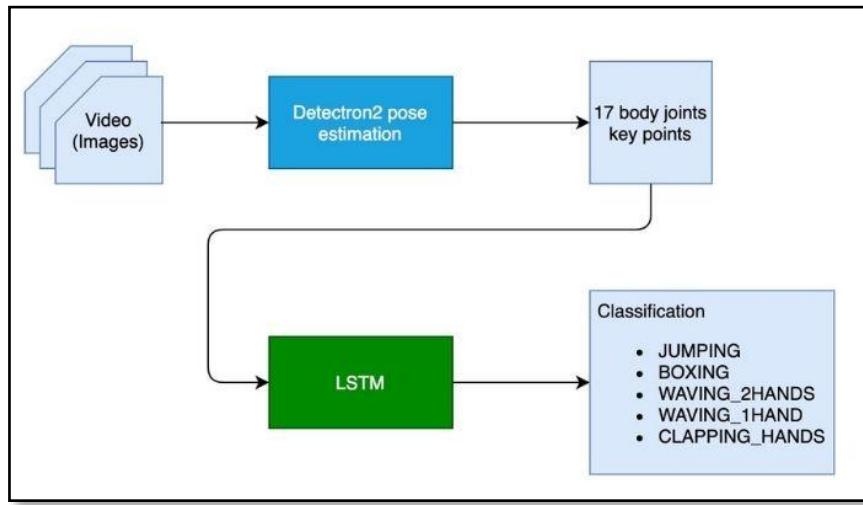
2. Lịch sử giải quyết vấn đề

Sau đây là một số hệ thống, ứng dụng liên quan đến các cử chỉ của con người được vận dụng CNTT trong việc phân tích tư thế:

2.1 Nhận diện hành động của người với Deep Learning

Human Action Recognition (*HAR*) [5] là quá trình sử dụng những cảnh quay trong video để nhận diện, phân loại các hành động khác nhau được thực hiện bởi người trong video. Nó được ứng dụng rất rộng rãi trong các lĩnh vực như giám sát, thể dục, thể thao.

Tổng quan sơ đồ kiến trúc



Hình 1 - Kiến trúc tổng quát cho bài toán nhận diện Human Pose.

Để phân loại một hành động, trước tiên chúng ta cần xác định vị trí các bộ phận cơ thể khác nhau trong mọi khung hình, sau đó phân tích chuyển động của các bộ phận đó theo thời gian.

Bước đầu tiên đạt được bằng cách sử dụng Detectron2, nó xuất ra tư thế của cơ thể (17 Keypoints) sau khi quan sát một khung hình trong video.

Bước thứ hai là phân tích chuyển động của cơ thể theo thời gian và đưa ra dự đoán được thực hiện bằng mạng LSTM [10]. Đầu vào là các Keypoints từ một chuỗi khung được, đầu ra là loại hành động được dự đoán.



Hình 2 - Minh họa 6 hành động

Hình 2 mô tả bao gồm 6 hành động: Nhảy, nhảy dội hai tay, đánh bốc, hai tay vuông góc, một tay vuông góc, vỗ tay.

2.2. Xây dựng huấn luyện viên thể hình AI bằng Mediapipe để phân tích Squats (Build an AI Fitness Trainer using Mediapiipy to Analyze Squats).



Hình 3 - Giao diện hệ thống phân tích Squats

Giới thiệu cách xây dựng một ứng dụng đơn giản để phân tích squats bằng giải pháp Pose của MediaPipe. Các thành phần quan trọng của ứng dụng bao gồm:

- Tính các góc như đường vai-hông, hông-đầu gối và đầu gối-mắt cá chân với các phương thẳng đứng tương ứng của chúng;
- Duy trì các trạng thái khác nhau để hiển thị phản hồi phù hợp và phân biệt giữa squats đúng và sai;
- Tính toán không hoạt động trong thời gian đó các bộ đếm tương ứng được đặt lại.

Ứng dụng giả định rằng người đó nên duy trì góc nhìn tốt của máy ảnh. Nếu một người hoàn toàn quay mặt về phía trước máy ảnh, chúng tôi sẽ hiển thị một thông báo cảnh báo thích hợp.

3. Mục tiêu đề tài

3.1. Mục tiêu chung

Xây dựng hệ thống “Điều khiển thiết bị IoT với sự giám sát của Trí tuệ nhân tạo”, cụ thể là máy học, thị giác máy tính và ước lượng dáng của con người (Human Pose Estimation). Thông qua hình ảnh tư thế, trạng thái của người trong ảnh, hệ thống sẽ xác định đâu là tư thế đứng, tư thế ngồi và cuối cùng là tư thế nằm một cách tự động dựa trên máy học (cụ thể, thị giác máy tính). Tiếp theo đối với riêng tư thế nằm DATM tiếp tục phân tích và dự đoán xem người đó đang ngủ hay còn thức. Hơn nữa, hệ thống tận dụng

thiết bị IoT của hệ sinh thái “Tuya” điều khiển thiết bị điện khi mà người đó đang ngủ. Cuối cùng, một giao diện người dùng sẽ được cài đặt.

Đề tài được chia ra làm 3 giai đoạn:

1. Xây dựng framework nền tảng sử dụng mô hình AI;
2. Tích hợp và phát triển bộ dữ liệu nhiều góc độ. Phát triển thêm với việc xác định các tư thế khác;
3. Triển khai mô hình ứng dụng với mô hình thực tế ở các ngôi nhà.

Lưu ý, trong đề tài này chúng tôi sẽ tập trung vào giai đoạn một để xây dựng khung nền tảng (fundamental framework). Đây sẽ là khởi đầu cho các nghiên cứu tiếp theo để phát triển và nâng cao độ chính xác của các giải thuật học máy.

3.2. Mục tiêu chi tiết

- Thu thập và xử lý hình ảnh các tư thế, cụ thể là đứng, nằm, ngồi;
- Huấn luyện mô hình CNN với 3 tư thế trên;
- Uớc tính, dò tìm các điểm dựa theo tư thế người bằng việc sử dụng MediaPipe;
- Tính toán, đưa ra thuật toán, đánh dấu tư thế đó có đang ngủ không;
- Phân tích các tính chất đặc trưng của từng tư thế;
- Tìm hiểu về trí tuệ nhân tạo, các kỹ thuật và công cụ học sâu phục vụ cho việc xây dựng mô hình thị giác máy tính phân loại ảnh và chấm điểm;
- Tìm hiểu các phương pháp huấn luyện, các kỹ thuật tối ưu trong quá trình huấn luyện nhằm tăng tính chính xác cho mô hình;
- Xây dựng mô hình thị giác máy tính xác định trạng thái con người có đang ngủ không;
- Đề xuất Framework chung.

4. Phạm vi nghiên cứu

4.1. Đối tượng nghiên cứu

- Sinh viên đại học, người trưởng thành từ 18-60 tuổi;
- Tập trung xác định trạng thái ngủ hay chưa ngủ chủ yếu vào tư thế nằm;
- Phân lớp theo các tư thế đứng, nằm, ngồi.

4.2. Phạm vi nghiên cứu

- Chúng tôi tập trung vào phân lớp 3 tư thế đứng, nằm, ngồi.
- Thực hiện cài đặt trên Camera 2D.

- Tính toán điểm số với toạ độ cố định của camera để đảm bảo quan sát các cử chỉ của đối tượng (sinh viên). Cụ thể, cung cấp một vị trí cố định cho camera với góc nghiêng 30 - 60° từ trên xuống, ở khoảng cách từ camera đến đối tượng là 3 mét, chiều cao từ mặt đất đến camera khoảng 2 – 2,1 mét, để lấy toàn bộ cơ thể người một cách bao quát nhất.

- Cho phép phân loại tư thế, khi tư thế là nằm (cụ thể là nằm ngửa) xác định trạng thái ngủ hay thức của 01 người trên khung camera tại cùng thời điểm.

- Chỉ thực hiện việc tắt đèn trên thiết bị Tuya khi người trong khung camera được xác định là tư thế nằm ngửa và đang trong trạng thái ngủ.

- Cho phép thực hiện offline (sử dụng video) và online (sử dụng webcam/camera thu hình trực tiếp).

5. Phương pháp nghiên cứu

Chúng tôi tập trung chủ yếu vào 02 phương pháp chính: nghiên cứu tài liệu và thực nghiệm. Tuy nhiên, để đảm bảo kiến thức không sai sót trong quá trình nghiên cứu, chúng tôi đã thực hiện lấy ý kiến của nhiều người khi ở đang trong trạng thái ngủ.

5.1. Phương pháp nghiên cứu tài liệu

Khảo sát các phương pháp giải quyết bài toán có liên quan đến đề tài nghiên cứu về Hỗ trợ giảng dạy môn Giáo dục thể chất với máy học và Human pose Estimation. Cụ thể, chúng tôi tập trung chủ yếu với bao gồm 02 phần chính: kiến thức được tổng hợp lại khi tham khảo ý kiến của nhiều người và kiến thức trí tuệ nhân tạo để xây dựng mô hình.

- **Kiến thức tổng hợp được từ nhiều người:** tham khảo các ý kiến chung được cho rằng là phổ biến khi 1 người đang xem tivi mà ngủ quên sẽ có những đặc điểm như thế nào.

- **Kiến thức Trí tuệ nhân tạo:** các kiến trúc mạng nơron CNN để ứng dụng trong việc phát hiện động tác bằng các mô hình: LeNet-5, MobileNets, Densenet121, InceptionV3. Từ đó, nhận xét, đánh giá, so sánh các phương pháp nhằm lựa chọn cách tiếp cận thích hợp, tiếp đến sẽ tạo dữ liệu để huấn luyện trạng thái ngủ hay thức khi mô hình xác định là tư thế nằm qua các kiến thức tìm hiểu liên quan đến thư viện OpenCV, thư viện Tensorflow, Mediapipe, Numpy, Keras, Scikit-learn lập trình Python, Tkinter và những thư viện khác liên quan. Tìm hiểu phương pháp huấn luyện mô hình, tối ưu trong quá trình huấn luyện và các phương pháp đánh giá mô hình.

5.2. Phương pháp thực nghiệm

Xây dựng bộ dữ liệu dùng để huấn luyện mô hình xác định 3 tư thế, bằng cách thu thập dữ liệu hình ảnh bằng cách quay video (recording video) 3 tư thế (đứng, nằm, ngồi) với điện thoại cá nhân (Iphone 11, hệ điều hành IOS 16.7.2).

Xây dựng mô hình xác định trạng thái ngủ hay thức của tư thế nằm bằng cách: tạo dữ liệu huấn luyện sử dụng dữ liệu quay được ở trên kết hợp thư viện MediaPipe để detection 4 điểm (cổ tay trái, cổ tay phải, đầu gối trái, đầu gối phải) của con người rồi tính khoảng cách 4 điểm này cứ sau 5 frame ảnh, nếu khoảng cách của 1 trong 4 điểm vượt ra ngưỡng được đặt ban đầu thì sẽ gán nhãn 0 (thức), ngược lại gán nhãn 1 (ngủ). Sau khi gán nhãn hoàn thành, sẽ sử dụng bộ dữ liệu vừa gán nhãn đi huấn luyện mô hình xác định trạng thái. Cài đặt môi trường huấn luyện, tinh chỉnh các tham số, kỹ thuật để nâng cao kết quả của mô hình đề xuất. Nhận xét đánh giá mô hình dựa vào các tiêu chí cụ thể như độ chính xác, hiệu suất huấn luyện, tốc độ thực thi, thời gian xử lý việc huấn luyện. Chạy thực nghiệm trên giao diện Tkinter để phát hiện tư thế gì? Trên những hình ảnh mới từ video quay được qua điện thoại cá nhân và từ camera ngoài thực tế và ghi nhận kết quả lên giao diện. Ngoài ra sau khi mô hình đã xác định trạng thái ngủ của đối tượng thì thiết bị Tuya sẽ nhận tín hiệu và sẽ tắt thiết bị điện.

5.3. Phương pháp lấy ý kiến

Để đảm bảo cho tính đúng đắn của kết quả nghiên cứu về mặt kiến thức chuyên môn ít sai sót nhất có thể. Chúng tôi đã sử dụng phương pháp lấy ý kiến của nhiều người. Cụ thể, chúng tôi đã liên hệ với 03 sinh viên và 01 giảng viên hướng dẫn luận văn (Tiến sĩ Mã Trường Thành). Chúng tôi đã thực hiện một khảo sát nhỏ là “khi ngủ các bộ phận nào trên cơ thể sẽ có khả năng di chuyển sang nhiều chỗ mà dễ dàng phát hiện”. Bài khảo sát này sẽ được thực hiện trên 33 điểm tương ứng với thư viện MediaPipe sẽ được trình bày ở phần bên dưới và các điểm này được đánh giá theo thang điểm 0 – 5 trên 33 điểm đó theo tần suất mà bộ phận đó có khả năng di chuyển nhiều. Chúng tôi sẽ thực hiện tính tổng cho từng điểm mà các bạn đánh giá và có cài đặt trọng số cho giảng viên hướng dẫn để đánh giá theo thang điểm, cụ thể riêng ý kiến của giảng viên sẽ giữ hệ số 2. Từ đó chúng tôi thống kê lại như bảng sau:

điểm tàn suất từ 0 - 5						
name point	point	Huy	Ngoan	Duy	thầy Thành	total
nose	0	0	0	0	0	0
left_eye_inner	1	0	0	0	0	0
left_eye	2	0	0	0	0	0
left_eye_outer	3	0	0	0	0	0
right_eye_inner	4	0	0	0	0	0
right_eye	5	0	0	0	0	0
right_eye_outer	6	0	0	0	0	0
left_ear	7	0	0	0	0	0
right_ear	8	0	0	0	0	0
mouth_left	9	0	0	1	1	3
mouth_right	10	0	0	1	1	3
left_shoulder	11	1	0	1	1	4
right_shoulder	12	1	0	1	1	4
left_elbow	13	1	2	1	1	6
right_elbow	14	1	2	1	1	6
left_wrist	15	3	3	2	3	14
right_wrist	16	3	3	2	3	14
left_pinky	17	0	0	0	0	0
right_pinky	18	0	0	0	0	0
left_index	19	0	0	0	0	0
right_index	20	0	0	0	0	0
left_thumb	21	1	1	1	0	3
right_thumb	22	1	1	1	0	3
left_hip	23	1	0	0	1	3
right_hip	24	1	0	0	1	3
left_knee	25	2	2	3	4	15
right_knee	26	2	2	3	4	15
left_ankle	27	0	0	1	0	1
right_ankle	28	0	0	1	0	1
left_heel	29	0	0	0	0	0
right_heel	30	0	0	0	0	0
left_foot_index	31	0	0	0	0	0
right_foot_index	32	0	0	0	0	0

Hình 4 - Bảng thống kê tham khảo ý kiến

Sau khi thống kê lại chúng tôi nhận thấy được có 4 điểm trên 10 và cũng là điểm lớn nhất được các bạn và thầy đánh giá. Chúng tôi quyết định sẽ lấy 4 điểm này áp dụng cho bài toán để có thể phát hiện được một người có đang trong trạng thái ngủ hay không.

6. Kết quả đạt được

Đề tài đã vận dụng các thuật toán trí tuệ nhân tạo vào thực tế. Cụ thể, xây dựng một hệ thống “Điều khiển thiết bị IoT với sự giám sát của Trí tuệ nhân tạo”. Những đóng góp chính như sau:

- Tập dữ liệu gồm 3 tư thế đứng nằm ngồi;
- Tập dữ liệu gồm 2 trạng thái của tư thế nằm là ngủ và thức;
- Xây dựng mô hình CNN với những kiến trúc hiện nay, LeNet-5 CNN with Keras phân loại các tư thế độ chính xác cao;
- Xây dựng được giải thuật xác định trạng thái ngủ hay thức khi kết hợp với MediaPipe;

- Xây dựng mô hình SVM phân lớp trạng thái ngủ hay thức để so sánh với giải thuật xác định thủ công thông qua ngữ cảnh;
- Kết nối với thiết bị IoT (Tuya) để điều khiển thiết bị khi nhận thấy đối tượng đang ở trạng thái ngủ;
- Xây dựng giao diện người dùng.

7. Bố cục luận văn

Nội dung của quyển luận văn gồm các phần sau đây:

Phần giới thiệu

Phần này trình bày các vấn đề phát sinh và lịch sử giải quyết vấn đề của đề tài, mục tiêu đề tài, những nghiên cứu được thực hiện trong lúc thực hiện đề tài.

Phần nội dung

Phần này trình bày chi tiết bài toán, thiết kế và cài đặt hệ thống, đồng thời nêu lên quy trình kiểm thử, đánh giá phần mềm. Bao gồm các phần:

Chương 1: Mô tả bài toán.

Chương 2: Thiết kế, cài đặt giải thuật, trình bày các bước xây dựng hệ thống và kết nối với IoT

Chương 3: Kiểm thử hệ thống và đánh giá độ chính xác, tốc độ của hệ thống.

Phần kết luận

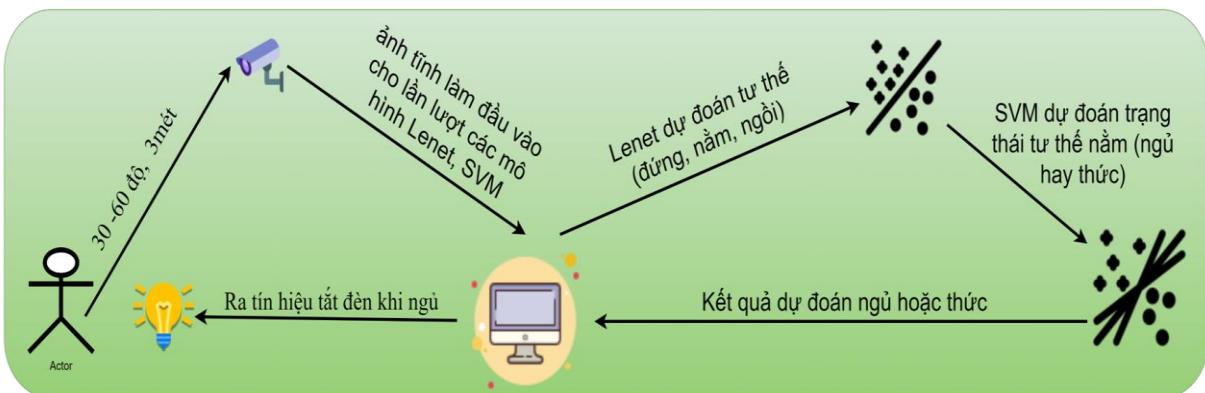
Phần này trình bày kết quả đạt được của đề tài cũng như những hạn chế mà đề tài chưa thực hiện được, ngoài ra cũng đưa ra hướng phát triển sau này.

PHẦN NỘI DUNG

CHƯƠNG 1 MÔ TẢ BÀI TOÁN

1.1. Mô tả chi tiết bài toán

Đề tài: “**DATM: điều khiển thiết bị IoT với sự giám sát của Trí tuệ nhân tạo**” được nghiên cứu nhằm đáp ứng lăng phí điện khi không sử dụng, hệ thống DATM được xử lý khi hình ảnh đọc từ camera / video làm đầu vào. Đầu tiên sẽ được xử lí qua mô hình xác nhận là “người”. Tiếp theo sẽ được thông qua mô hình phân lớp 3 tư thế đứng, nằm, ngồi. Khi tư thế được xác định là nằm sẽ tiếp tục sử dụng hình ảnh này làm đầu vào cho mô hình tiếp theo để xác định chủ thể người trong ảnh có đang trong trạng thái ngủ không. Nếu chủ thể người đang ngủ kết quả sẽ được phản hồi về và máy tính ra tín hiệu tắt thiết bị điện khi chủ thể đã ngủ không còn sử dụng điện, ngược lại các thiết bị điện sẽ được giữ nguyên.



Hình 5 - Mô tả hoạt động hệ thống

Hình trên cho ta thấy được máy quay sẽ cách chủ thể khoảng 3 mét. Máy quay được đặt phía bên phải lệch góc $30 - 60^{\circ}$, khoảng cách từ mặt đất đến máy quay khoảng 2 – 2,1 mét để giúp hệ thống có thể quan sát, và lấy được chủ thể người một cách bao quát và toàn diện nhất. Lưu ý rằng, chúng tôi sẽ giới hạn hệ thống như hình trên cho giai đoạn nghiên cứu này. Mục tiêu chính của nghiên cứu là trả lời câu hỏi ngủ quên hay không với một framework được đề xuất và với những giải thuật sau.

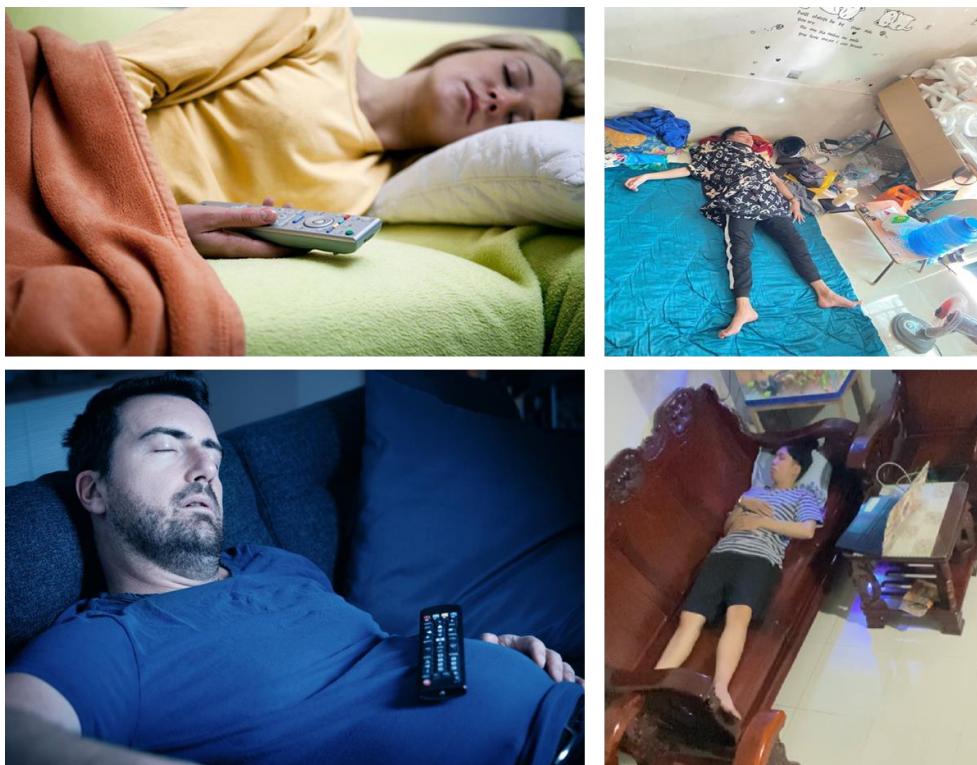
1.2. Vấn đề và giải pháp liên quan đến bài toán

Để thực hiện bài toán trên tôi đã sử dụng một số mô hình máy học, thư viện hỗ trợ.

1.2.1. Các tư thế ngủ quên thông thường

Ngủ quên hay được diễn đạt một cách thông thường khi con người đã rơi vào trạng thái ngủ khá lâu nhưng họ lại không biết mình đã ngủ từ khi nào. Con người sẽ hay ngủ quên khi đang làm việc nhưng quá mệt mỏi hoặc chẳng hạn khi đang xem điện thoại, ngoài ra việc ngủ quên khi xem tivi cũng là một trường hợp phổ biến có thể nói đến họ

roi vào giấc ngủ khi nào mà bản thân họ cũng không hề biết khi đang thoái mái với tư thế nằm để thư giãn. Sau đây là các tư thế nằm ngủ quên phổ biến:



Hình 6 - Các tư thế ngủ quên

Nhìn chung Hình 6 cho ta thấy được các tư thế nằm khi mà con người ta rơi vào trạng thái ngủ quên thì các cử chỉ như tay chân các bộ phận khác sẽ bất động trong thời gian dài hoặc chỉ cử động nhưng rất ít và rồi sẽ bất động tại vị trí mới đó. Từ nhận xét trên chúng tôi nhận thấy được rằng khi thức con người sẽ hay cử động các bộ phận như là cổ tay và đầu gối chân để có thể vận động, nên chúng tôi quyết định sẽ tập trung vào 4 bộ phận đó để xác định một người có đang trong trạng thái ngủ hay không. Để vận dụng Trí tuệ nhân tạo cho bài toán này, chúng tôi đã đề xuất vận dụng mô hình máy học để thực hiện phân lớp cho 2 trạng thái ngủ và thức. Phần kế tiếp chúng tôi sẽ trình bày ngắn gọn về máy học.

1.2.2. Mạng nơ-ron

1.2.2.1. Mạng nơ-ron nhân tạo

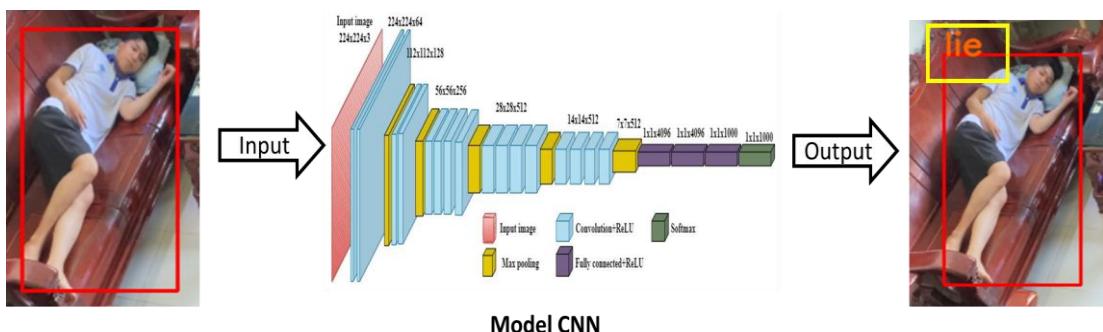
Mạng nơ-ron nhân tạo [2] (*Artificial Neural Network - ANN*) là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thần kinh sinh vật. Một mạng nơ-ron bao gồm nhiều nơ-ron được liên kết với nhau theo kiến trúc dạng đồ thị. Thông tin di chuyển vào một nơ-ron gọi là đầu vào của nơ-ron, sau khi được tính toán xử lý, thông tin đầu ra của một nơ-ron sẽ tiếp tục làm đầu vào của một nơ-ron khác.

Cấu trúc chung của một mạng nơ-ron nhân tạo gồm 3 thành phần tầng đầu vào (Input Layer), tầng ẩn (Hidden Layer) và tầng đầu ra (Output Layer), có thể có 1 hoặc

nhiều tầng ẩn. Tầng ẩn gồm các nơ-ron trung gian nhận dữ liệu đầu vào từ các nơ-ron ở tầng trước đó và chuyển đổi các đầu vào này thành đầu ra cho các tầng xử lý tiếp theo.

1.2.2.2. Mạng nơ-ron tích chập

Mạng nơ-ron tích chập được phát triển từ ý tưởng ban đầu lấy cảm hứng từ cách hoạt động của mắt người, chia nhỏ hình ảnh thành các chập chia cho các nơ-ron xử lý. Năm 1998, Mạng Nơ-ron tích chập được giới thiệu bởi Bengio, Le Cun, Bottou và Haffner. Mô hình đầu tiên của họ được gọi tên là LeNet-5 [8].

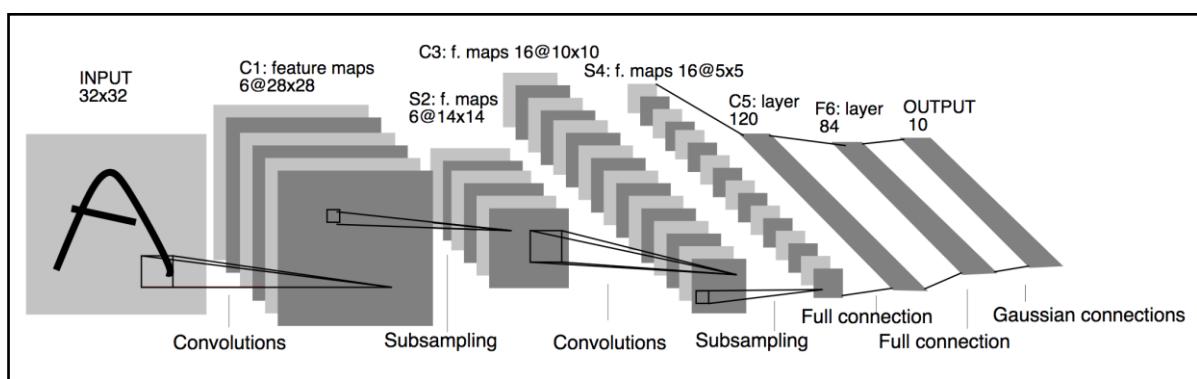


Hình 7 - Minh họa cấu trúc của một mạng nơ-ron nhân tạo

Cấu trúc chung của một mạng nơ-ron nhân tạo được minh họa trong Hình 1.4. Mạng nơ-ron tích chập cũng là một mạng nơ-ron, cấu trúc của một mạng nơ-ron tích chập bao gồm tầng đầu vào, các tầng ẩn và các tầng phân lớp để cho kết quả đầu ra. Các tầng ẩn dùng để rút trích đặc trưng ảnh, các tầng ẩn là một loạt các phép tích chập (convolution) và hợp nhất (pooling) để phát hiện đặc trưng trong ảnh đầu vào.

1.2.2.3. Kiến trúc LeNet-5 CNN

Kiến trúc CNN của LeNet-5 [4][9] được tạo thành từ 7 lớp. Thành phần lớp bao gồm 3 lớp tích chập, 2 lớp lấy mẫu con và 2 lớp được kết nối đầy đủ.



Hình 8 - Sơ đồ trên mô tả về kiến trúc LeNet-5

Lớp đầu tiên là lớp đầu vào - đây thường không được coi là một lớp của mạng vì không có gì được học trong lớp này. Lớp đầu vào được xây dựng để có kích thước 32x32 và đây là kích thước của hình ảnh được chuyển vào lớp tiếp theo. Những người đã thuộc về tập dữ liệu MNIST sẽ biết rằng hình ảnh tập tin MNIST có kích

thước 28x28. Để kích thước hình ảnh MNIST đáp ứng các yêu cầu của lớp đầu vào, hình ảnh 28x28 được đệm.

Các hình ảnh thang độ xám được sử dụng trong bài báo nghiên cứu có các giá trị pixel được chuẩn hóa từ 0 đến 255, thành các giá trị trong khoảng 0,1 đến 1,175. Lý do chuẩn hóa là để chắc chắn rằng lô ảnh có giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1, lợi ích của công việc này là giảm thời gian đào tạo.

1.2.3. SVM (Support vector machines)

Support vector machines (SVM) [6][1] là một tập hợp các phương pháp học có giám sát được sử dụng để phân loại, hồi quy và phát hiện giá trị ngoại lệ.

Ưu điểm của SVM

- Hiệu quả trong không gian chiều cao;
- Vẫn hiệu quả trong trường hợp số chiều lớn hơn số mẫu;
- Sử dụng một tập hợp con các điểm đào tạo trong chức năng quyết định do đó nó cũng hiệu quả về bộ nhớ;
- Linh hoạt: các chức năng Kernel khác nhau có thể được chỉ định cho chức năng quyết định. Các hạt nhân thông thường được cung cấp, nhưng cũng có thể chỉ định các hạt nhân tùy chỉnh.

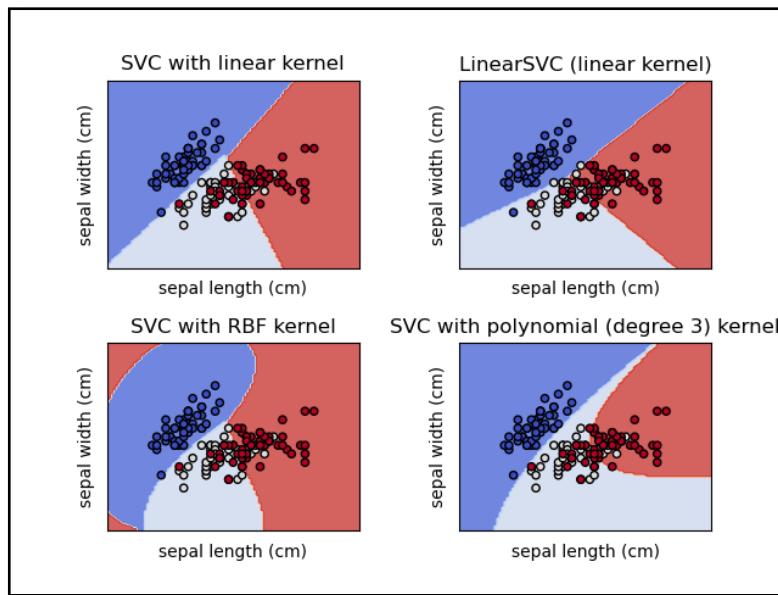
Nhược điểm của SVM

- Nếu số lượng tính năng lớn hơn nhiều so với số lượng mẫu, tránh quá khóp trong việc chọn các hàm Kernel và thuật ngữ chính quy hóa là rất quan trọng;
- Các SVM không trực tiếp cung cấp các ước tính xác suất, chúng được tính toán bằng cách sử dụng xác thực chéo năm lần đắt tiền.

Các SVM trong scikit-learning hỗ trợ cả vectơ mẫu dày đặc và thưa thớt làm đầu vào. Tuy nhiên, để sử dụng SVM để đưa ra dự đoán cho dữ liệu thưa thớt, nó phải phù hợp với dữ liệu đó.

Phân loại SVM

SVC và NuSVC là LinearSVC các lớp có khả năng thực hiện phân loại nhị phân và đa lớp trên tập dữ liệu.



Hình 9 - Thực hiện phân loại nhị phân và đa lớp trên tập dữ liệu

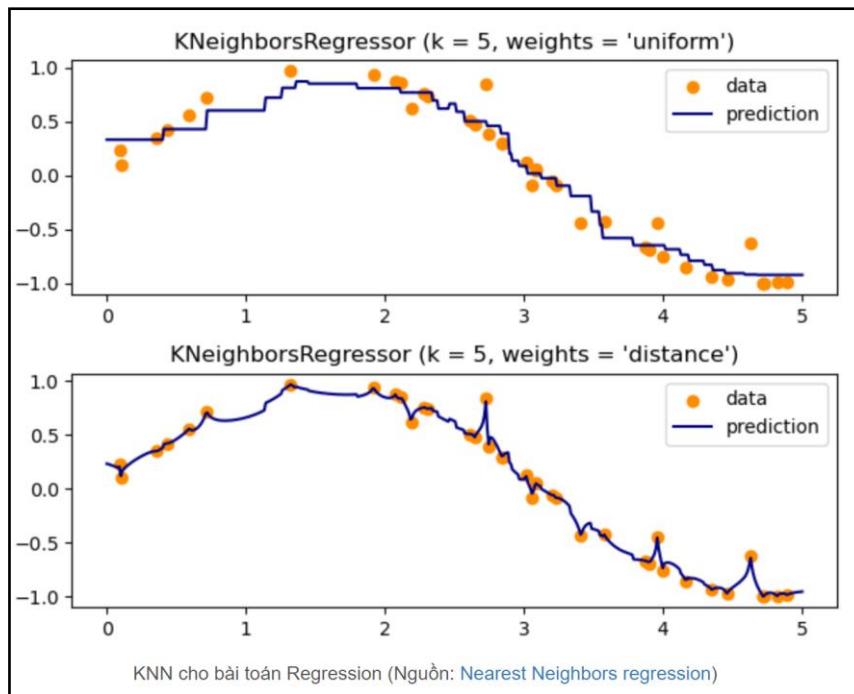
SVC và NuSVC là các phương pháp tương tự, nhưng chấp nhận các bộ tham số hơi khác nhau và có các công thức toán học khác nhau. Mặt khác, LinearSVC là một triển khai khác của Phân loại vectơ hỗ trợ cho trường hợp hạt nhân tuyến tính.

1.2.4. KNN (K-nearest neighbor)

K-nearest neighbor [3] là một trong những thuật toán supervised-learning đơn giản nhất trong machine learning. Khi training, thuật toán này không học một điều gì từ dữ liệu training, mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là classification và regression.

Trong bài toán classification, label của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong training set. label của một test data có thể được quyết định giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra label.

Trong bài toán regression, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó.



Hình 10 - KNN cho bài toán regression

Ưu điểm của KNN

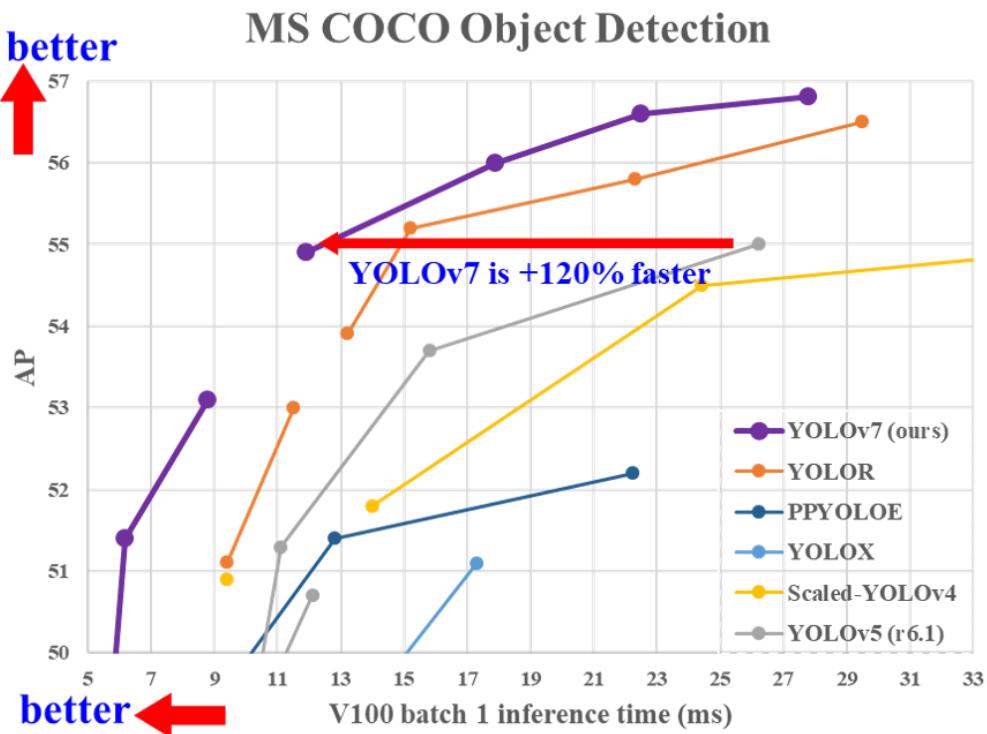
- Độ phức tạp tính toán của quá trình training là bằng 0;
- Việc dự đoán kết quả của dữ liệu mới rất đơn giản;
- Không cần giả sử gì về phân phối của các class;

Nhược điểm của KNN

- KNN rất nhạy cảm với nhiễu khi K nhỏ;
- Như đã nói, KNN là một thuật toán mà mọi tính toán đều nằm ở khâu test. Trong đó việc tính khoảng cách tới từng điểm dữ liệu trong training set sẽ tốn rất nhiều thời gian, đặc biệt là với các cơ sở dữ liệu có số chiều lớn và có nhiều điểm dữ liệu. Với K càng lớn thì độ phức tạp cũng sẽ tăng lên. Ngoài ra, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN.

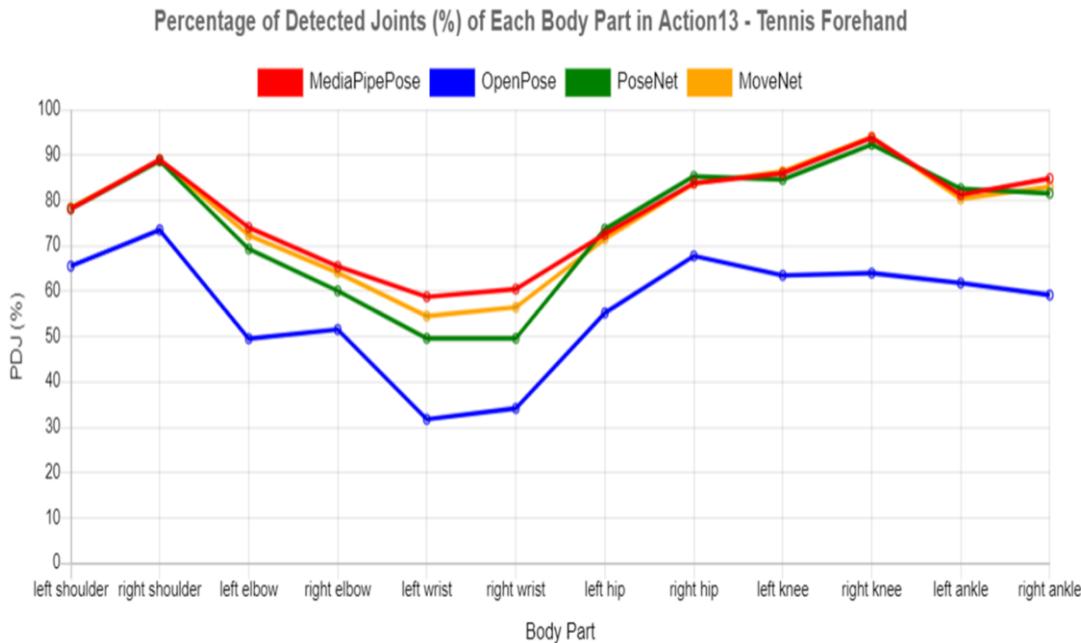
1.2.5. YOLOv7

YOLO [13] là viết tắt của "You Only Look Once," và đây là một mô hình object detection (nhận diện đối tượng) trong lĩnh vực thị giác máy tính. Mô hình này được thiết kế để nhận diện và xác định vị trí của đối tượng trong một hình ảnh hoặc video. Nổi bật hơn YOLO có khả năng nhận diện đối tượng nhanh chóng và hiệu quả còn giữ được tốc độ xử lý cao phù hợp cho ứng dụng thời gian thực. Ngoài ra YOLO còn dự đoán được các tọa độ x, y, w, h nhằm tạo ra được một bb bao quanh đối tượng. Mô hình YOLO chỉ thực hiện xem qua một lần các bức ảnh và thực hiện dự đoán. Trong các phiên bản YOLO thì nhận thấy được YOLOv7 có nhiều điểm được cải tiến về độ chính xác, tốc độ cũng như khả năng nhận diện trong thời gian thực

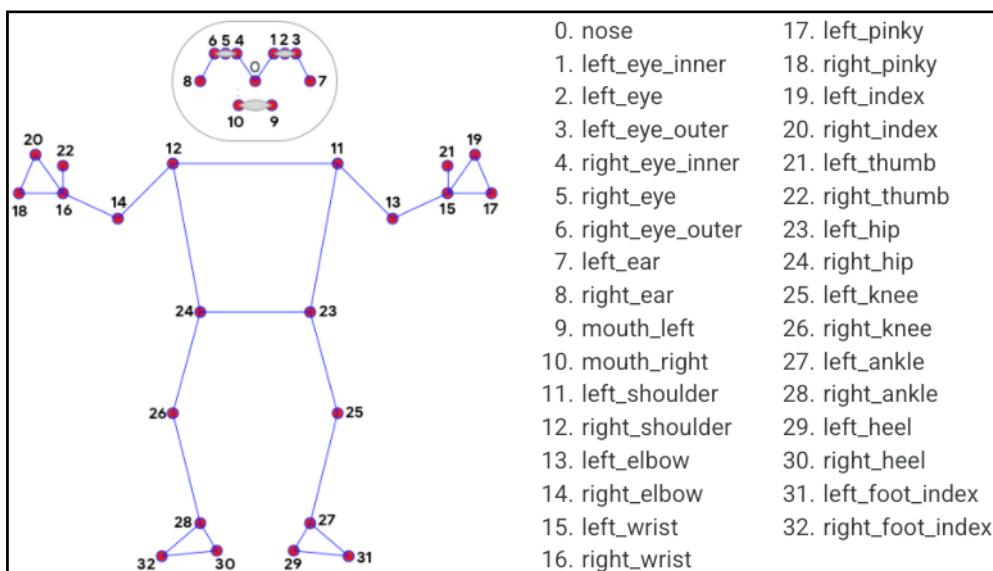
*Hình 11 - Minh họa so sánh các phiên bản của YOLO*

1.2.6. Human Pose Estimation

Human pose estimation (HPE) [11] là tác vụ thị giác máy tính nhằm biểu diễn tư thế của một người ở định dạng đồ họa. Một mô hình pose estimation có nhiệm vụ xác định và phân biệt các khớp của cơ thể người, liên kết giữa các keypoint được gọi là pairs. Khi kết nối keypoints và pairs ta thu được một sơ đồ giống khung xương cơ thể người và bài toán xác định nhiều skeleton trong 1 khung ảnh còn được gọi là Multi - person pose estimation. Dựa trên độ tin cậy và so sánh giữa các giải thuật, chúng tôi quyết định lựa chọn MediaPipe.



Hình 12 - Minh họa độ tin cậy và so sánh giữa các giải thuật



Hình 13 - Cấu trúc liên kết pose landmarks

MediaPipe Pose sẽ thực hiện dò tìm tư thế của người có liên quan mật thiết với hình ảnh và vẽ tất cả các điểm mốc trong tư thế sẽ được dò tìm trong hình ảnh. Nó sử dụng cấu trúc liên kết pose landmarks là siêu tập hợp các điểm chính COCO, pose landmarks hoạt động trong hai giai đoạn – phát hiện và theo dõi.

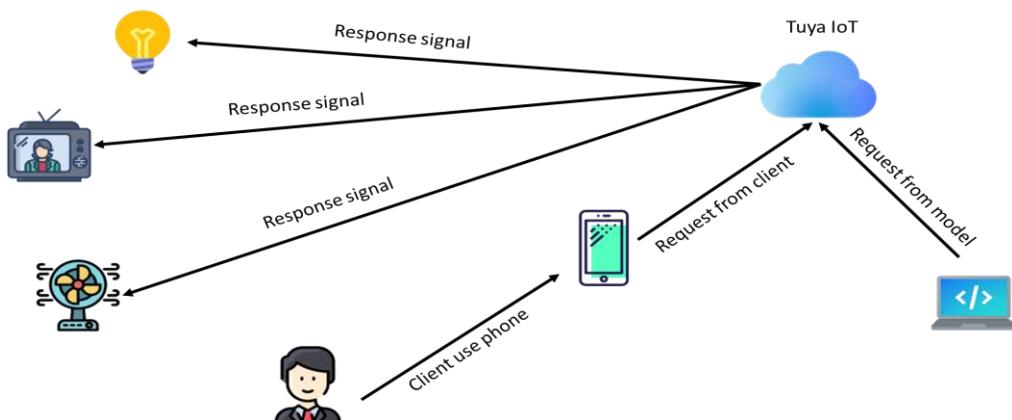
1.2.7. Tuya API

Tuya là nền tảng IoT (Internet of Things) toàn cầu và là nền tảng tương tác AI lớn nhất trên thế giới. Dựa trên chuẩn công nghệ Tuya, các nhà sản xuất có thể tạo nên các sản phẩm công nghệ IoT và phần mềm điều khiển của mình. Nó cung cấp cho người dùng giải pháp giúp các sản phẩm thông minh có thể làm việc với nhau trên cùng 1 APP

chung tạo thành một hệ sinh thái. Trong hệ sinh thái Tuya có rất nhiều thiết bị: điều khiển hồng ngoại từ xa, máy lạnh, công tắc thông minh, bóng đèn, ô cấm thông minh, các cảm biến nhiệt độ, cảm biến chuyển động, cảm biến khói...

Tuya Smart hỗ trợ tất cả các tiêu chuẩn kỹ thuật chủ đạo, như Wi-Fi, Bluetooth, GPRS, Zigbee, vv... Hợp tác với các đối tác phần cứng hàng đầu, Tuya Smart cung cấp cho bạn các sản phẩm đáng tin cậy và được ủy quyền.

Những tiện ích mà Tuya API mang lại thì chúng tôi có thể tích hợp vào đề tài này để thể hiện rõ sự ứng dụng của trí tuệ nhân tạo vào đời sống như sau:



Hình 14 - Nhận yêu cầu và phản hồi về các thiết bị điện

- Người dùng có thể sử dụng ứng dụng trên thiết bị di động để điều khiển các thiết bị điện từ xa

- Ngoài ra chúng tôi sẽ sử dụng API này để gửi request khi mô hình xác định người trong video / camera là trong tình trạng ngủ để ra tín hiệu tắt thiết bị điện.

CHƯƠNG 2

THIẾT KẾ VÀ CÀI ĐẶT

2.1. Thiết kế hệ thống

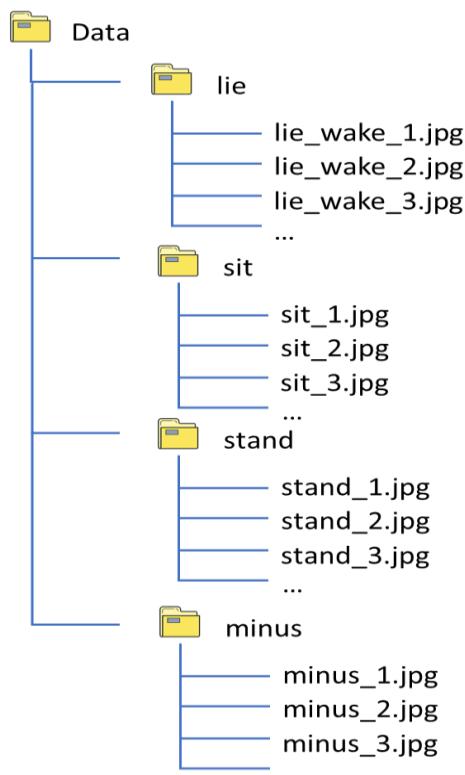
Thiết kế hệ thống là thiết kế ra các bước thực hiện theo trình tự trước sau để cấu thành nên một hệ thống.

2.1.1. Quá trình cài đặt và thực hiện đề tài



Hình 15 - Sơ đồ quá trình cài đặt và thực hiện đề tài

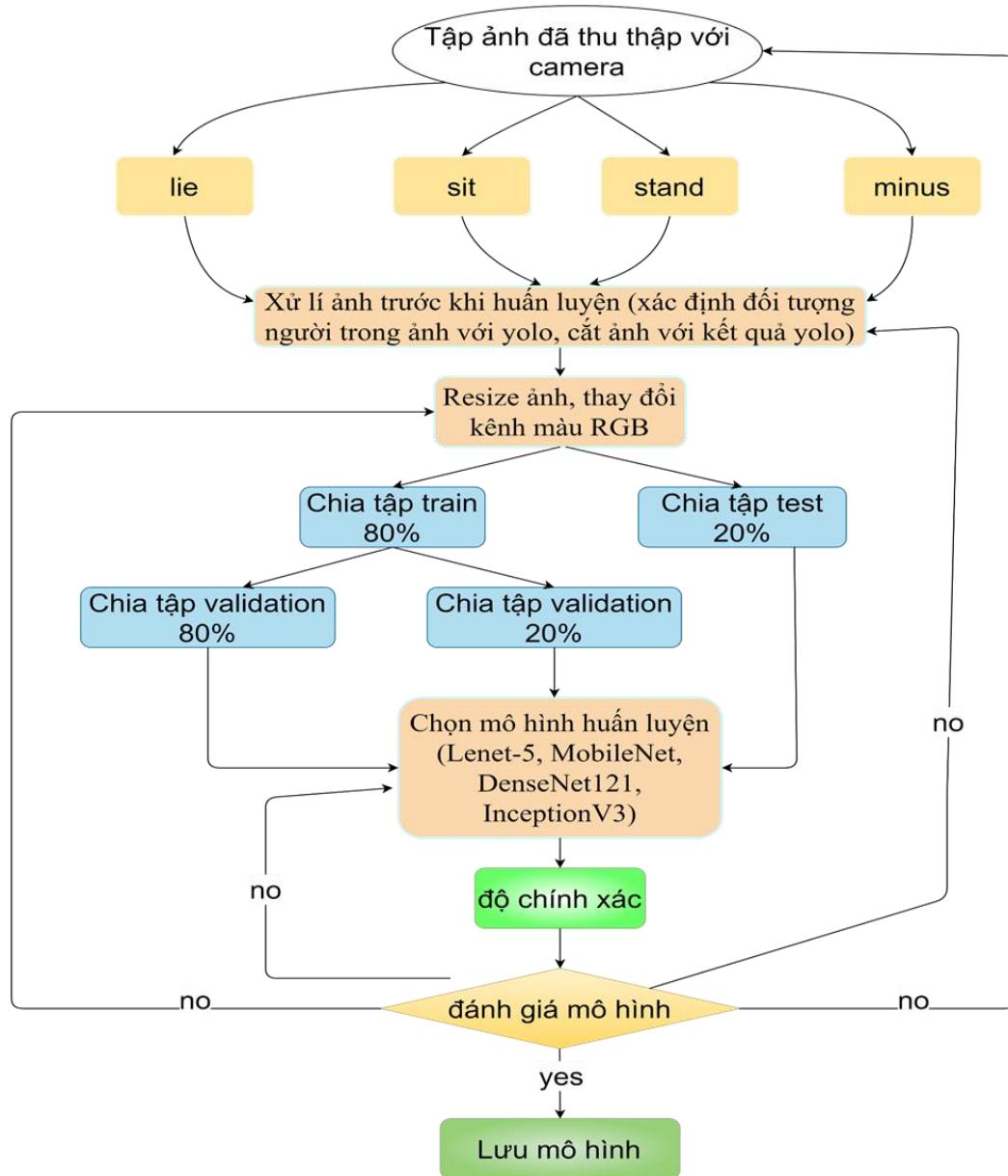
2.1.2. Thư mục các tư thế



Hình 16 - Mô tả cây thư mục các tư thế để huấn luyện

Hình 15 bao gồm 3 tư thế và có cả phần bù dùng để huấn luyện mô hình phân lớp tư thế đứng, nằm, ngồi.

2.1.3. Xây dựng mô hình huấn luyện phân lớp tư thế



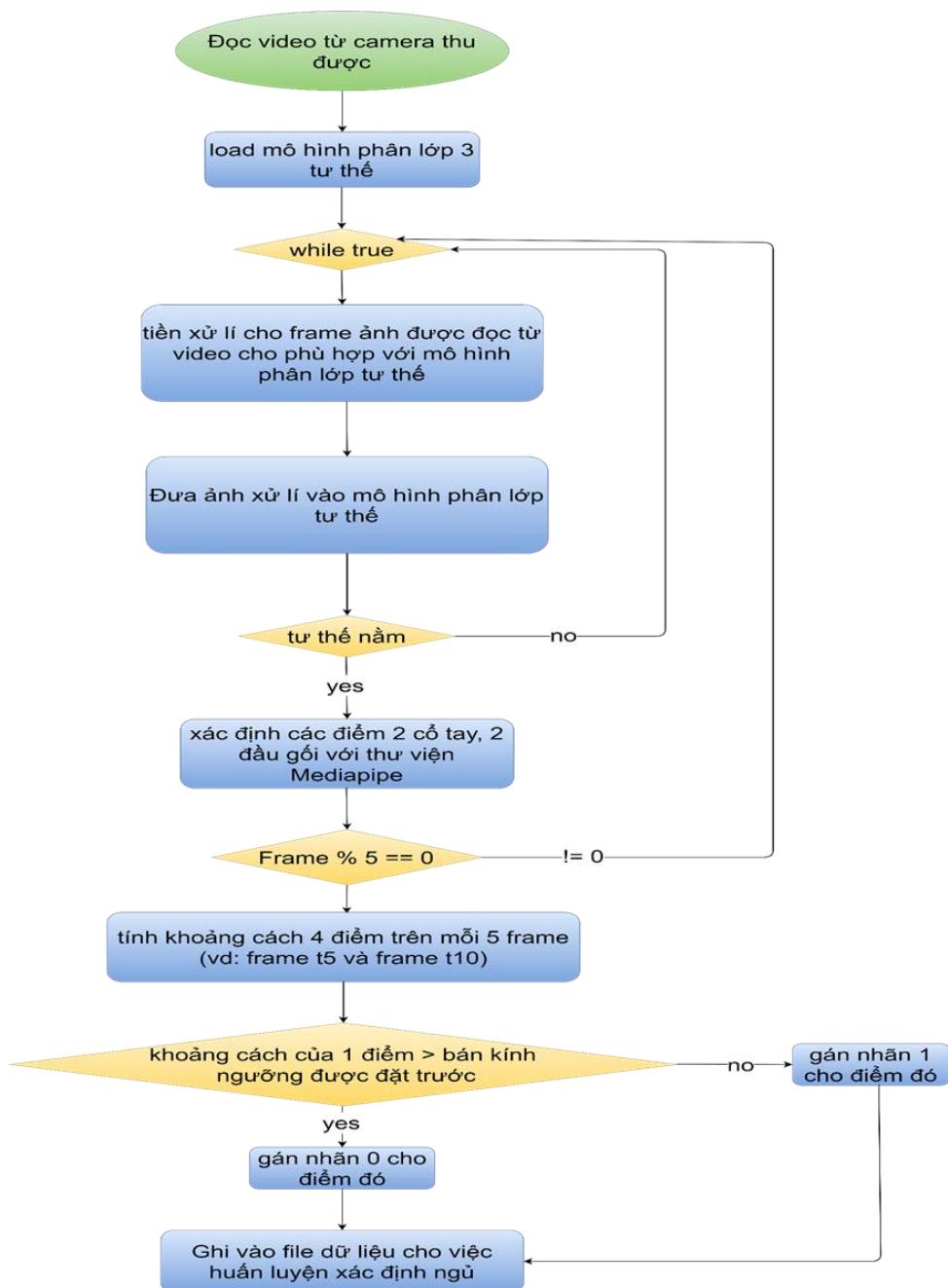
Hình 17 - Quá Trình huấn luyện mô hình

Khi nhìn hình 16, chúng ta biết được ảnh đầu vào được thu bằng camera, tạo ra 4 thư mục hình ảnh, trải qua quá trình tiền xử lý dữ liệu (cắt ảnh đúng cơ thể người), sau quá trình tiền xử lý xong resize ảnh và kênh RGB, sử dụng 80% để train, 20% để test, tiếp tục lấy 80% tập train này chia ra 80% tập validation train, 20% validation test chọn mô hình (CNN với 4 kiến trúc: Lenet-5, MobileNet, DenseNet121, InceptionV3), hiển thị độ chính xác mô hình và đánh giá mô hình đúng thì lưu lại mô hình nếu sai chúng ta kiểm tra lại ảnh đầu vào hoặc quá trình tiền xử lý hoặc kích thước ảnh hoặc mô hình cho phù hợp với đề bài.

2.1.4. Xây dựng mô hình huấn luyện xác định ngủ

Trước khi huấn luyện mô hình cần xây dựng dữ liệu để huấn luyện mô hình xác định ngủ hay thức của đối tượng người, quá trình xây dựng dữ liệu này sẽ chia thành 2 giai đoạn:

- Giai đoạn 1: sử dụng mô hình phân lớp tư thế vào tập dữ liệu thư mục nằm bao gồm nằm ngủ và nằm thức, Sử dụng thư viện MediaPipe để bắt được tọa độ của 4 điểm trên cơ thể
- Giai đoạn 2: xây dựng giải thuật xác định đối tượng người có đang cử động không, nếu có gán nhãn 0 (không ngủ) ngược lại gán nhãn 1 (ngủ)



Hình 18 - Quá trình tạo dữ liệu mới cho mô hình phân lớp ngủ

Sau khi đã xây dựng được tập dữ liệu với 2 giá trị 0, 1 ở 4 điểm trên cơ thể như sau:

Dữ liệu xây dựng được với video ngủ	Dữ liệu xây dựng được với video thức

Hình 19 - Tập dữ liệu sau khi xây dựng

Từ tập dữ liệu trên nhận thấy được với video ngủ số lượng nhãn 1 sẽ được gán nhiều hơn 0 và ngược lại.

Do 1 dòng dữ liệu trên chỉ được tính khoảng cách sau mỗi 5 frame nên thời gian ngắn này để xác định 1 người có đang trong trạng thái ngủ hay chưa là chưa đủ độ tin cậy. Chúng tôi đã quyết định sẽ thực hiện việc gom các dòng lại theo lần lượt 3 dòng, 4 dòng, 5 dòng cho phù hợp với mong muốn dự đoán với thời gian dài hay ngắn, như vậy nghĩa là cứ 3 dòng sẽ phải qua 15 frame tính toán tương tự với 4 dòng và 5 dòng. Ngoài ra lần lượt 3 dòng, 4 dòng, 5 dòng cũng sẽ tương ứng với 12 đặc trưng, 16 đặc trưng và 20 đặc trưng để dự đoán dựa trên các đặc trưng này. Sau đây là tập dữ liệu đã được gom nhóm lại

Dữ liệu với video ngủ	Dữ liệu với video thức

Dữ liệu với video ngủ

Dữ liệu với video thức

Hình 20 - Tập dữ liệu đã được gom 5 dòng và được gán nhãn

Khi đã có đầy đủ tập dữ liệu, việc huấn luyện mô hình dự đoán đối tượng người trong trạng thái ngủ hay không ngủ được thực hiện 1 cách tuần tự như sau:



Hình 21 - Xây dựng mô hình huấn luyện xác định ngủ hay thức

2.1.5. Thiết kế giải thuật xác định ngủ hay thức dựa vào việc tính ngưỡng:

Ngoài phương pháp xác định đối tượng người trong trạng thái ngủ hay không ngủ bằng việc huấn luyện mô hình nói trên chúng tôi còn đưa ra 1 đề xuất đến việc thống kê số lượng đặc trưng của giá trị 0 và 1 trong từng file đã được xây dựng dữ liệu ở trên. Từ đó chúng tôi có thể đưa vào 1 ngưỡng dựa theo thống kê đó và sẽ xác định luôn đối tượng người đó đang trong trạng thái ngủ hay thức mà không cần thông qua việc huấn luyện mô hình.

2.2. Mã giả xây dựng bộ đặc trưng và cài đặt hệ thống

2.2.1 Mã giả xây dựng bộ đặc trưng

a. Giải thuật tính khoảng cách cho 2 điểm

Function CalcDistance2PointOnFrame (t, coordi_f1, coordi_f2):

$$d = \sqrt{(coordi_{f1x1} - coordi_{f2x2})^2 + (coordi_{f1y1} - coordi_{f2y2})^2}$$

If d > t: return 0

return 1

Trong đó:

t là giá trị bán kính ngưỡng, để xác định độ vị trí của điểm đó có ra khỏi bán kính ngưỡng được đưa ra không;

coordi_f1 và coordi_f2 là tọa độ 4 điểm trên cơ thể của 2 frame trước và sau sẽ tính khoảng cách.

b. Giải thuật trích đặc trưng trên 2 frame ảnh

Function *ExtractFeatureFrame* (*points* [], *frame1*, *frame2*):

```
 $\beta_1 = \text{get\_coordinates\_mediapipe}(\text{points}, \text{frame1})$ 
 $\beta_2 = \text{get\_coordinates\_mediapipe}(\text{points}, \text{frame2})$ 
 $l = \text{CalcDistance2PointOnFrame}(t, \beta_1, \beta_2)$ 
return  $l$ 
```

Trong đó:

points là mảng các điểm [15, 16, 25, 26] sẽ cần lấy tọa độ để tính khoảng cách trên 2 frame;

frame1 và *frame2* là 2 frame ảnh sau 1 khoảng thời gian được trích từ video.

c. Giải thuật trích đặc trưng trên hai frame trên video

Function *ExtractFeatureVideo* (*video*, *points* []):

```
while video_is_not_empty:
    frame1 = get_frame_video(video)
    frame2 = get_frame_video(video)
     $\beta_1 = \text{get\_coordinates\_mediapipe}(\text{points}, \text{frame1})$ 
     $\beta_2 = \text{get\_coordinates\_mediapipe}(\text{points}, \text{frame2})$ 
     $l = \text{CalcDistance2PointOnFrame}(t, \beta_1, \beta_2)$ 
     $D = D \cup l$ 
return  $D$ 
```

Tại đây, chúng tôi sẽ định nghĩa cách thu thập tất cả các đặc trưng để biểu diễn ngữ quên hay không như sau:

Định nghĩa 1 (Toàn bộ đặc trưng): Gọi D là tập của những đặc trưng được trích từ Video, q là số đặc trưng thu thập trong một ảnh. Cho r là độ lớn của đặc trưng khi thu thập. Toàn bộ các đặc trưng trích xuất với $q \times r$ thuộc tính thì được định nghĩa như sau:

$$\Sigma_r = \{\boldsymbol{\theta} \mid \boldsymbol{\theta} \subset D, |\boldsymbol{\theta}| = r\}$$

Trong đó:

video: là đường dẫn đến video cần trích đặc trưng;

points: là mảng của 4 điểm trên cơ thể [15, 16, 25, 26];

Σ_r : là toàn bộ các đặc trưng được trích mỗi 5 frame trong video;

Lưu ý: i là giá trị sẽ được thay đổi tùy theo ý muốn, có ý nghĩa gộp 3, 4 hoặc 5 dòng đặc trưng để tăng thời gian cho việc xác định được chính xác hơn.

Ý tưởng: Thực hiện việc tính khoảng cách lần lượt 4 điểm [15, 16, 25, 26] của thư viện MediaPipe trên 2 frame, nếu điểm nào vượt ra ngoài bán kính ngưỡng sẽ thực hiện gán 0 và ngược lại gán 1. Việc này sẽ được lặp qua từng frame trên video và khi hết video sẽ thu lại được toàn bộ đặc trưng cho 1 video ngủ hoặc thức đó là Σ_r . Chúng tôi đã thực hiện trích đặc trưng trên cả 2 video là ngủ và thức. Sau khi có đủ 2 đặc trưng tiếp tục gộp cả 2 đặc trưng này và sẽ sử dụng cho việc huấn luyện mô hình máy học xác định đối tượng người đang ngủ hay thức.

Ngoài ra chúng tôi còn thực hiện đánh giá xác định ngủ hay thức của đối tượng người thông qua đặt ngưỡng.

Function AssessmentByThreshold(Σ_r , ratio):

$$Y_{pred} = 0$$

$$T = ratio * \Sigma_r[0].length$$

for $\varepsilon \in \Sigma_r$:

if (sum(ε) \geq | ε | * T:

$$Y_{pred} += 1$$

$$acc = \frac{Y_{pred}}{\Sigma_r - 1}$$

return acc, T

Trong đó:

Σ_r : là toàn bộ các đặc trưng được trích mỗi 5 frame trong video và đã được gộp nhóm theo 3, 4 hoặc 5 dòng;

ratio: là tỉ lệ phần trăm được truyền vào;

T: là ngưỡng sẽ được tính theo phần trăm ratio mong muốn.

2.2.2 Mã giả cài đặt hệ thống

Hệ thống sẽ bao gồm 2 phần chính được thực hiện cụ thể như sau:

Ý tưởng: Hệ thống sẽ bắt đầu với 1 frame ảnh được trích từ camera hoặc video, và sẽ được tiền xử lí từng frame mới đọc vào qua Yolov7 để xác định đối tượng người. Đầu

tiên frame ảnh sau khi tiền xử lí sẽ được đi qua mô hình máy học nhận dạng tư thế đứng hay nằm hay ngồi. Nếu là tư thế ngồi hoặc đứng sẽ được tiếp tục với frame tiếp theo, ngược lại nếu là tư thế nằm, frame ảnh này sẽ đi qua thư viện MediaPipe để trích các tọa độ 4 điểm chúng tôi quy định trên cơ thể ở frame này. Tiếp theo nếu frame này ở thứ tự mà có thể chia hết cho 5, chúng tôi thực hiện tính khoảng cách giữa 2 frame: hiện tại và trước đó nếu có đủ tọa độ ở 2 frame đây là bước trích đặc trưng. Sau khi trích đặc trưng cho 2 frame chúng tôi cho các đặc trưng này vào mảng và khi mảng này có số lượng = số điểm * số dòng gộp, thì sẽ đưa vào mô hình máy học xác định trạng thái ngủ hay thức. Khi ở trạng thái ngủ chúng tôi sẽ thực hiện gọi TuyaAPI cho việc tắt thiết bị điện, ngược lại sẽ không làm gì. Sau khi xác định được ngủ hay thức chúng tôi làm trống mảng đặc trưng và bắt đầu với các frame tiếp theo.

```
Function ProcessDetectSleep( $\beta$ , quantity_features):
    #  $\beta$  : Chế độ thu hình, bao gồm: 0: camera, "path": link video
     $\varphi \leftarrow \text{list}([ ])$  # danh sách đặc trưng
     $\theta \leftarrow \text{int}$  # thứ tự frame (bắt đầu là 0)
     $\rho \leftarrow \text{list}(15, 16, 25, 26)$  # danh sách 4 điểm trên cơ thể
     $\alpha \leftarrow \text{quantity\_features}$  #  $4 * \text{số dòng gộp}$ 
    while true:
         $\omega \leftarrow \text{read\_frame}(\beta)$ 
         $\omega \leftarrow \text{preprocessing}(\omega)$ 
         $\mu \leftarrow \text{predict\_pose}(\omega)$ 
        if  $\mu$  is 'Lie':
             $\vartheta \leftarrow \text{ExtractFeatureFrame}(\rho, \text{frame1}, \text{frame2})$ 
             $\varphi \leftarrow \text{update\_list}(\vartheta)$ 
            if  $\text{len}(\varphi) == \alpha$ :
                 $\tau \leftarrow \text{predict\_sleep\_model}(\varphi)$ 
                if  $\tau == \text{sleep}$ :
                    call tuyaAPI # tắt thiết bị điện
                 $\varphi \leftarrow \text{update\_list}([ ])$  # cập nhật danh sách đặc trưng thành rỗng
```

2.3. Cài đặt giải thuật

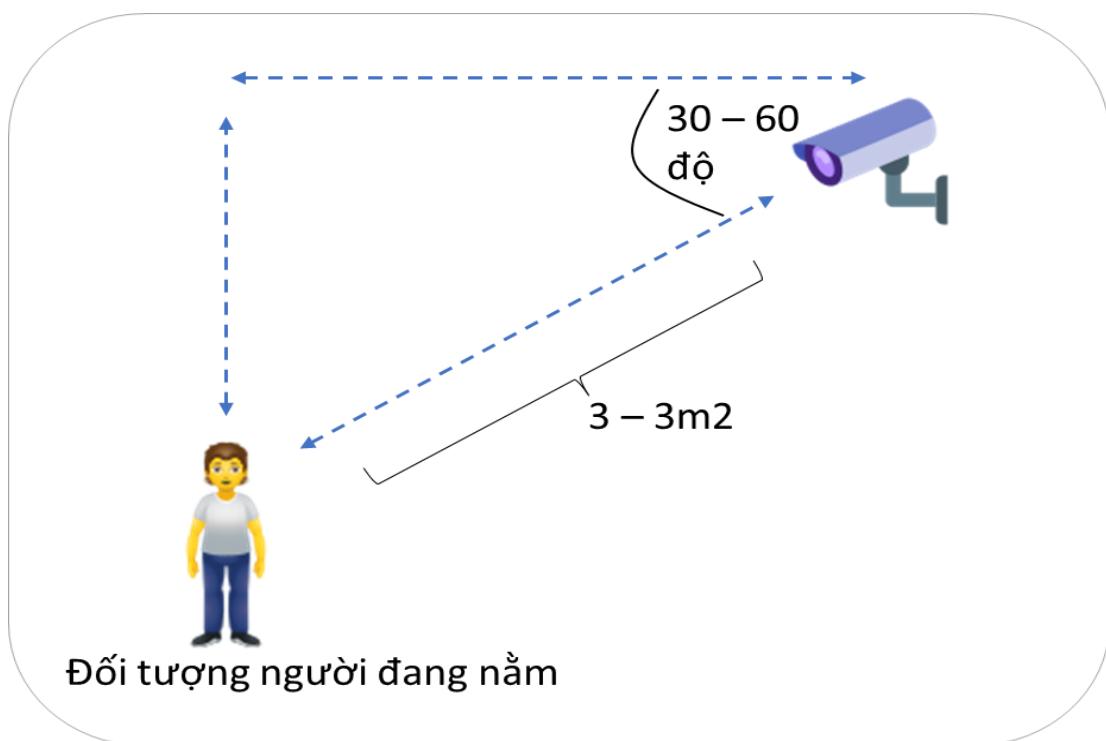
Sau khi đã có bản thiết kế xây dựng hệ thống chúng ta tiến hành cài đặt hệ thống.

2.3.1. Dụng cụ thu thập dữ liệu

- Chuẩn bị cây quay có chiều cao khoảng 50cm hỗ trợ cố định điện thoại khi ghi hình ảnh;
- Điện thoại để ghi lại hình ảnh động làm dữ liệu đầu vào cho phân lớp tư thế;
- Cần có một người thực hiện các tư thế hằng ngày như đứng, nằm , ngồi một cách bình thường;

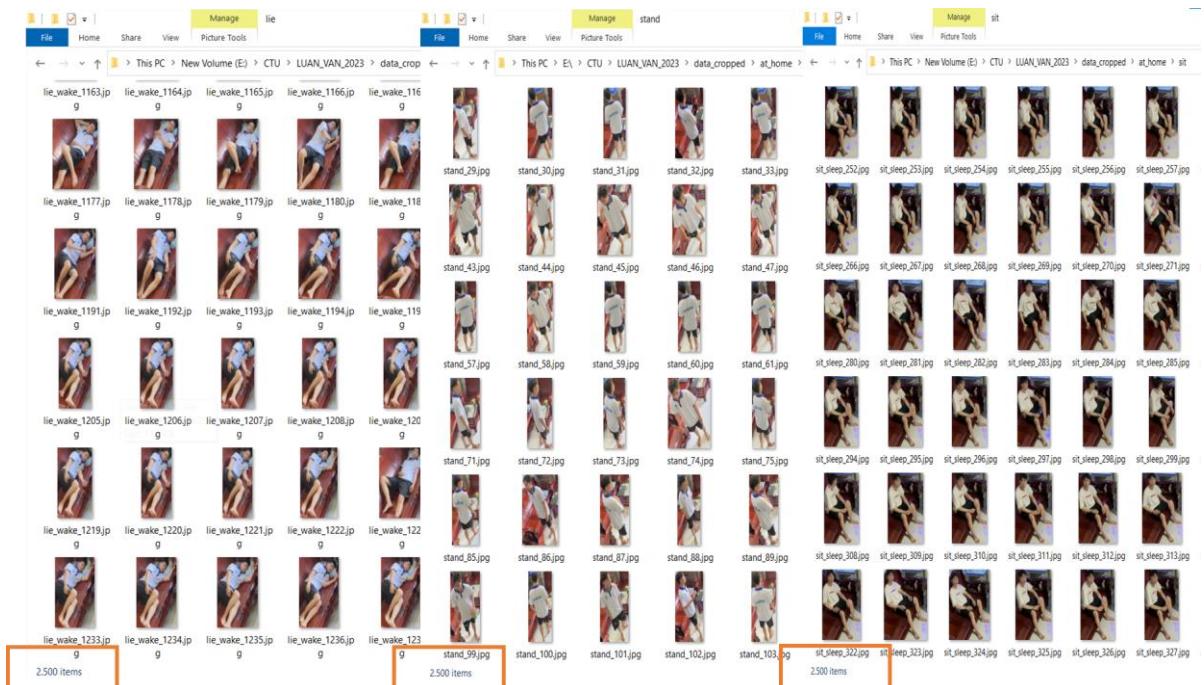
2.3.2. Thu thập dữ liệu

- Các góc độ đặt camera: góc 30 - 60⁰, người tập đứng cách máy khoảng 3m – 3m2, máy quay được đặt bên tay phải người thực hiện và điều kiện ánh sáng tốt không bị trói ánh nắng trực tiếp (ban ngày).



Hình 22 - Minh họa quá trình thu dữ liệu

- Quá trình thu thập dữ liệu ghi hình lại bằng những đoạn video từ những bạn hỗ trợ dữ liệu, được quay từ điện thoại mỗi đoạn video có độ dài khoảng 30 – 32 phút đối với 3 tư thế đứng, ngồi, riêng tư thế nằm sẽ chia ra 2 lớp là nằm ngủ và nằm thức với thời lượng 1 video khoảng 30 – 240 phút độ phân giải HD. Vị trí thu hình là nhà tôi, đây là vị trí mà ánh sáng, cũng như không gian thoáng thích hợp cho việc thu thập dữ liệu;



Hình 23 - Minh họa 3 tư thế đã được crop ra từ video thu được

Qua hình ảnh 23 cho chúng ta biết được 1 tư thế sẽ có 2500 tấm ảnh được cắt ra với Yolov7.

2.3.3. Cắt ảnh và tiền xử lý dữ liệu

a. Cắt ảnh

- Tôi sử dụng thư viện YoloV7 để detection đối tượng (con người);
- Sử dụng thư viện OpenCV2 để đọc video;
- Sau khi detection được con người tôi sử dụng Roi để cắt, 1 frame thì lấy một ảnh với tọa độ khung chứa đối tượng theo thứ tự sau:

$$Roi = [Top_Y: Bottom_Y, Top_X:Bottom_X]$$

và ghi xuống đúng thư mục động tác.

b. Tiền xử lý dữ liệu

Trong bài toán này tôi thực hiện tiền xử lí từng bước như sau

- Do ảnh được thu thập từ điện thoại full HD nên khi đưa vào máy tính các frame ảnh sẽ to, nên tôi thực hiện resize ảnh cho phù hợp
- Ngoài ra ảnh đầu vào sẽ có nhiều vật dụng trong nhà sẽ làm ảnh hướng đến kết quả mô hình, tôi thực hiện detect người sau đó crop đúng vị trí người.
- Riêng đối với mô hình xác định đối tượng người có đang trong trạng thái ngủ không, thì tôi cần sử dụng thư viện MediaPipe đã được trình bày ở phần xây dựng mô hình huấn luyện xác định ngủ ở trên có nội dung xây dựng dữ liệu huấn luyện



Hình 24 - Ảnh minh họa cắt và tiền xử lý

Hình 24 minh họa cho quá trình cắt ảnh và tiền xử lý dữ liệu.

2.3.4. Phân bổ dữ liệu huấn luyện

Tên tư thế	Số lượng ảnh	Số lượng ảnh train	Số lượng ảnh test	Số lượng ảnh validation
đứng	2500	6464	2020	1616
nằm	2500			
ngồi	2500			
chưa xác định	2600			
Tổng số lượng ảnh	10100			

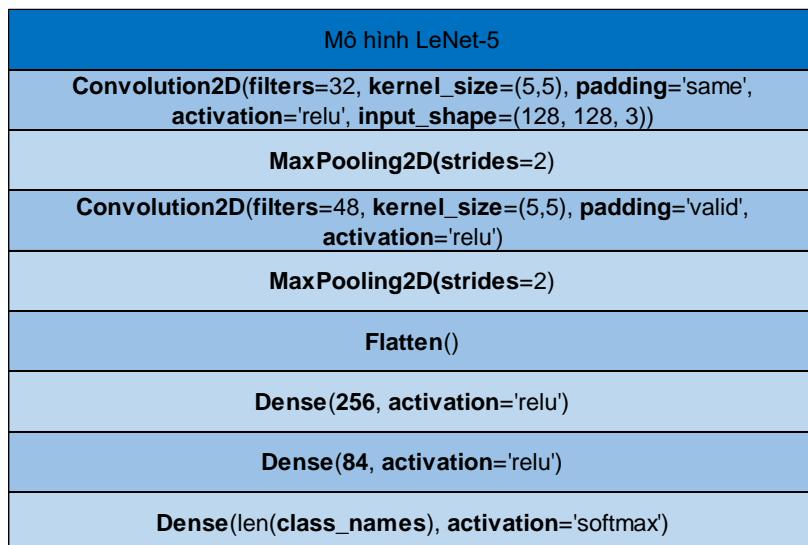
Hình 25 - Tổng số lượng hình ảnh thu thập

Kết quả cuối cùng chúng tôi thu thập được mỗi một lớp ảnh cho một tư thế là 2500 ảnh riêng lớp bù là 2600 ảnh. Lưu ý, để đánh giá mô hình, chúng tôi không đưa lớp bù vào thực hiện huấn luyện, chỉ thực hiện huấn luyện trên các nhịp, động tác bắt đầu và kết thúc. Chúng tôi đã lấy lớp “chưa phát hiện động tác” để xây dựng ứng dụng. Hơn nữa, để mô hình tốt và có tính tổng quát, tức là mô tả được dữ liệu cả trong lần ngoài tập huấn luyện, chúng tôi phân bổ tập dữ liệu thành 2 tập huấn luyện và tập đánh giá: **80%** cho tập huấn luyện. **20%** cho tập kiểm thử đánh giá. Chúng tôi tiếp tục sử dụng **80%** tập huấn luyện chia thành **80%** cho tập huấn luyện validation và **20%** cho tập kiểm thử validation.

2.2.5. Xây dựng và huấn luyện mô hình

2.2.5.1. Mô hình LeNet-5

Trong đề tài này, tôi chọn mô hình LeNet-5 làm mô hình nhận dạng chính, được xây dựng đơn giản, không phức tạp. Bởi mô hình này có độ chính xác cao và thời gian kiểm thử nhanh để biết thêm thông tin chi tiết của mô hình xem tiếp chương kế tiếp. Mô hình thiết kế như sau:



Hình 26 - Mô hình huấn luyện LeNet-5

Conv2D: Biến đổi hình ảnh bằng cách sử dụng 32 bộ lọc mỗi bộ lọc có kích thước 5×5 nhân tích chập với dạng đầu vào dạng $128 \times 128 \times 3$, padding = ‘same’, sử dụng chức năng kích hoạt ReLU;

MaxPooling2D: Lớp rút gọn, giúp làm giảm các bản đồ đặt trưng, giảm số lượng tham số / tính toán cho các lớp sau, tăng tính ổn định với các biến đổi nhỏ trong đầu vào với kích thước 2;

Conv2D_1: Biến đổi hình ảnh từ lớp Conv2D, ở lớp này sử dụng 48 bộ lọc có kích thước 5×5 và vẫn sử dụng , sử dụng padding = ‘valid’, chức năng kích hoạt ReLU;

MaxPooling2D_1: Cũng tương tự như MaxPooling2D;

Flatten: Làm phẳng dữ liệu đầu vào mà không ảnh hưởng đến kích thước, đầu ra làm phẳng này được chuyển đến lớp kế tiếp;

Dense: Là lớp ẩn trong mạng Neuron, bao gồm 256 đơn vị và sử dụng chức năng kích hoạt ReLU;

Dense_1: Là lớp ẩn trong mạng Neuron, bao gồm 84 đơn vị và sử dụng chức năng kích hoạt ReLU;

Cuối cùng là lớp đầu ra (output) sử dụng hàm kích hoạt **softmax** để mô hình dự đoán cho các lớp đầu ra.

Như vậy chúng ta đã xây dựng xong mô hình LeNet-5. Tiếp theo cần tổng hợp lại mô hình với các tham số tự thiết kế như sau: learning rate=0,001 (tốc độ học), epochs = 15, batch_size = 24, sử dụng thuật toán tối ưu Adam() có sẵn của keras, theo Kingma và cộng sự, 2014 [7] phương pháp này hiệu quả về mặt tính toán yêu cầu ít bộ nhớ, bất biến với thay đổi tỷ lệ theo đường chéo của gradient và rất phù hợp cho các vấn đề lớn về dữ liệu / tham số. Hoàn tất định cấu hình cho mô hình bằng compile(), trong đó hàm tổn thất loss được sử dụng, optimizer = adam là tên của thuật toán tối ưu và metrics=[‘accuracy’] là chỉ số đánh giá mô hình trong quá trình huấn luyện.

```
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])

model.fit(trainX, trainY, batch_size=BS, epochs=EPOCHS, verbose=1)
```

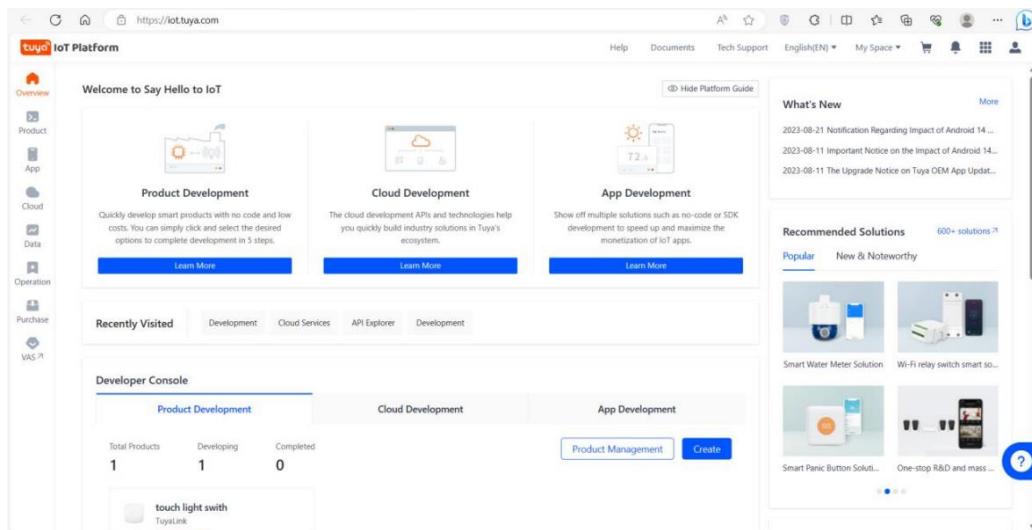
Hình 27 - Compile và fit() mô hình LeNet-5

Cuối cùng, hàm fit() được gọi để bắt đầu huấn luyện mô hình.

2.2.6. Cài đặt Tuya API

Bước 1: Import thư viện hỗ trợ kết nối với Tuya API

Bước 2: Truy cập vào [Tuya IoT Platform](https://iot.tuya.com), đăng nhập tài khoản

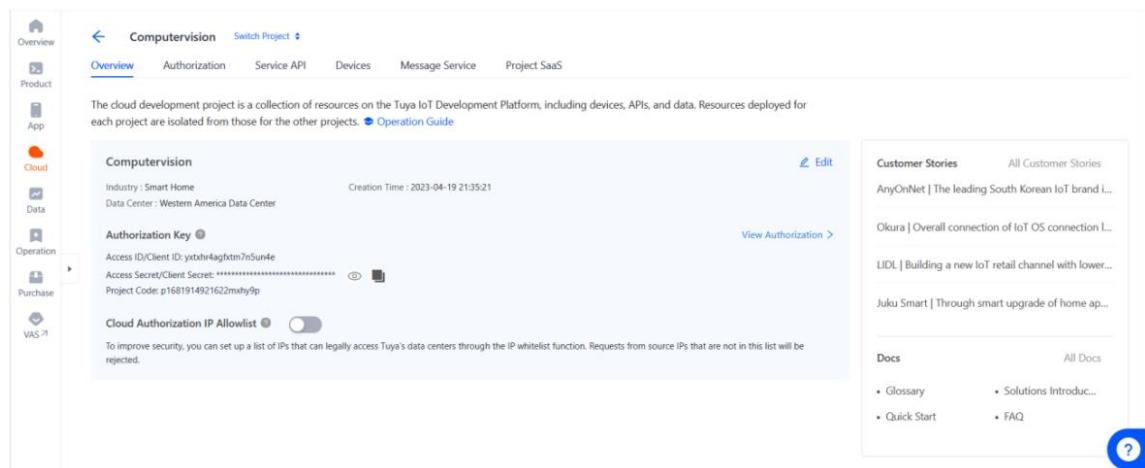


Hình 28 - Trang chủ Tuya IoT

Bước 3: Tiếp theo nhấp vào Cloud chọn Development và chọn create cloud project

Chọn vào dự án mới hoặc dự án cũ trước đó

Bước 4: Truy cập vào dự án vừa tạo

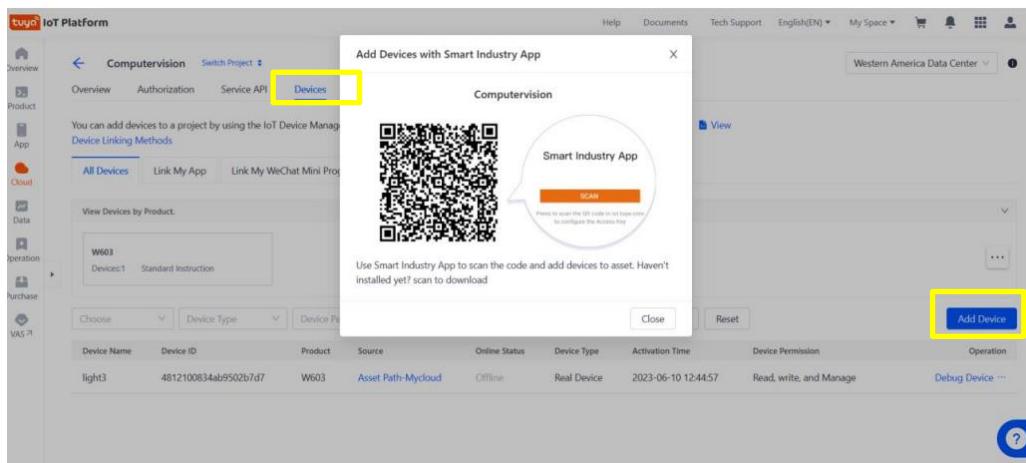


Hình 29 - Giao diện trang dự án mới tạo Tuya

Ở đây cần chú ý đến Access ID và Access Secret

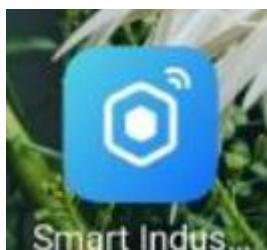
Bước 5: Chọn Device -> Add Devices-> scan mã QR để tải App điều khiển về

Điều khiển thiết bị IoT với sự giám sát của Trí tuệ nhân tạo



Hình 30 - Giao diện chọn devices kết nối

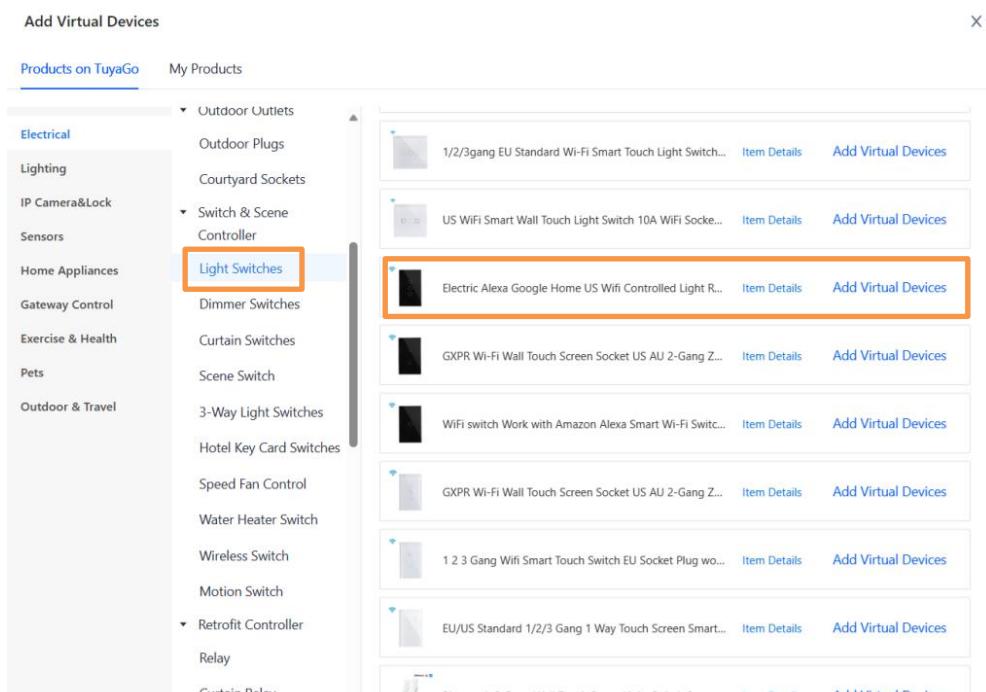
Giao diện app mobile như sau



Chúng ta cần sử dụng app mobile để kết nối wifi cho thiết bị tuya

Bước 6: đăng ký tài khoản đăng nhập app mobile lưu ý khi đăng nhập sử dụng chung 1 email cho việc đăng nhập tránh dùng 2 email

Bước 7: ở web tiếp tục thực hiện chọn vào Add Devices chọn Add Virtual Devices tiếp theo chọn thiết bị thích hợp với thiết bị IOT.

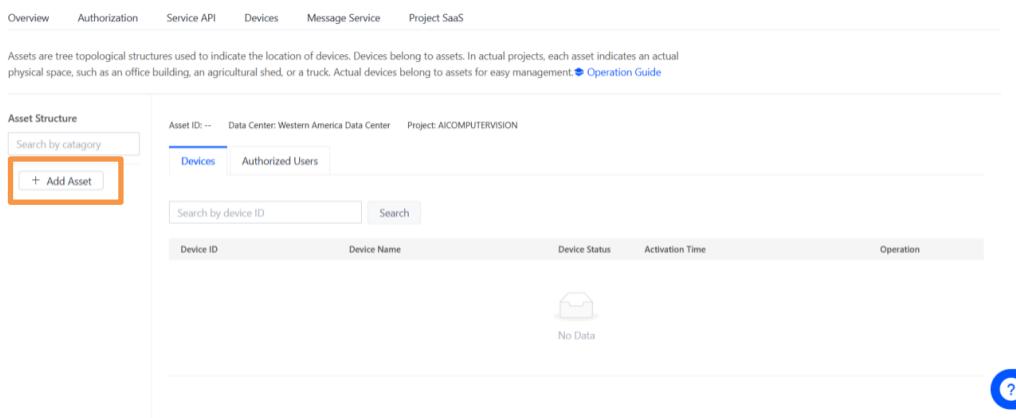


Hình 31 - Giao diện thêm device

Chọn vào Add Virtual Devices với thiết bị tương ứng.

Bước 8: chọn vào create an asset

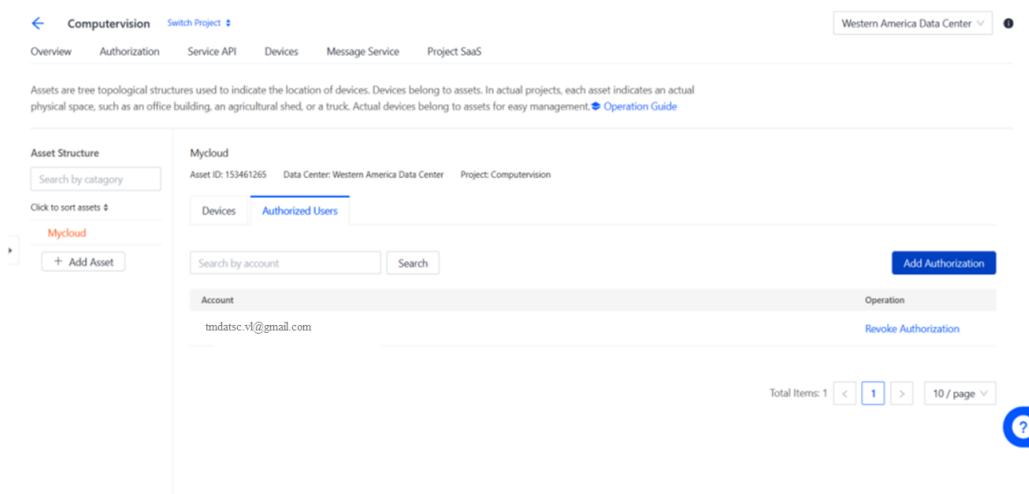
Bước 9: chọn vào add asset



Hình 32 - Giao diện tạo account cho app mobile

Tạo tài khoản cho app mobile theo hướng dẫn

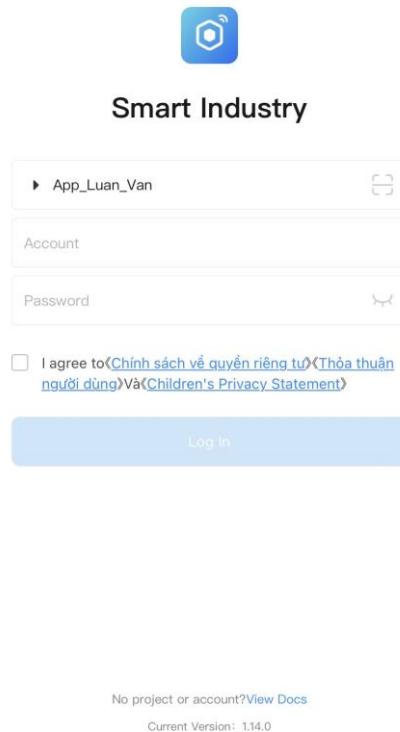
Sau khi tạo tài khoản theo yêu cầu thành công ta vào Authorization chọn add Authorization thêm email đăng ký trên app vào



Hình 33 - Giao diện thêm account mobile thành công

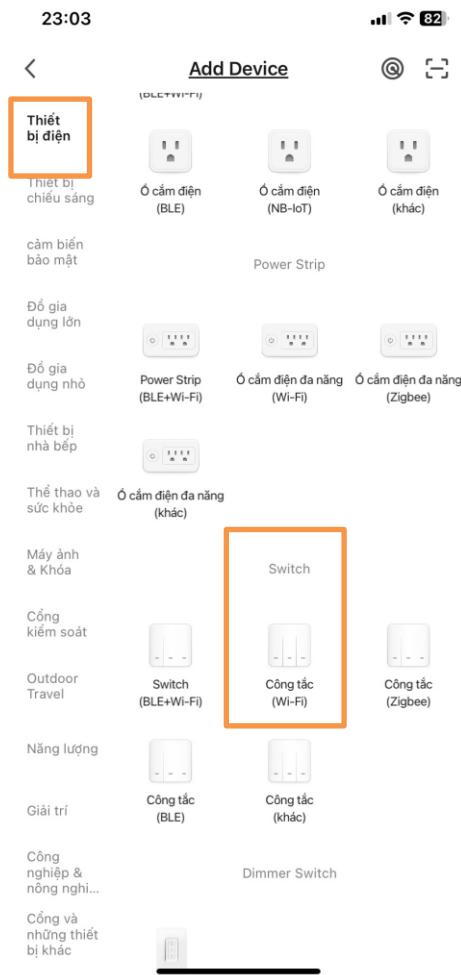
Bước 10: Sau khi cài đặt app mobile thành công, mở app vào thực hiện scan mã ở bước 4

23:33 78



Hình 34 - Giao diện đăng nhập trên app mobile

Bước 11: Sau khi đăng nhập, chọn icon dấu cộng góc trên bên phải để add device và kết nối wifi cho device đó.



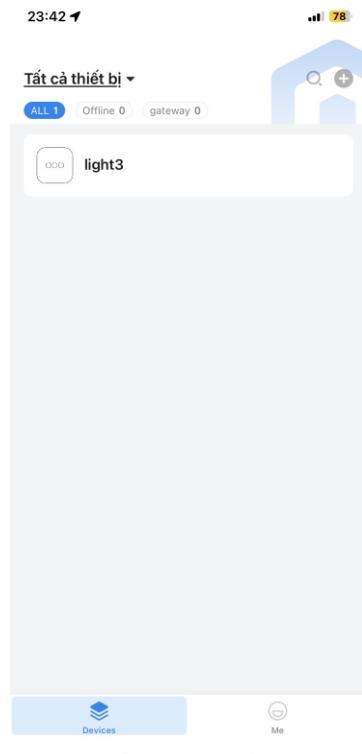
Hình 35 - Giao diện add device kết nối wifi app mobile

Lưu ý khi kết nối wifi cần bật thiết bị tuya ở chế độ dò khi đèn chớp liên tục cho đến khi đèn tắt thì sẽ kết nối thành công

Bước 12: Khi kết nối thành công web và app mobile sẽ như sau:

Device Name	Device ID	Product	Source	Online Status	Device Type	Activation Time	Device Permission	Operation
light3	4812100834ab9502b7d7	W603	Asset Path-my_luan_van	Online	Real Device	2023-11-29 23:04:14	Controllable	Debug Device

Hình 36 - Giao diện web khi kết nối thiết bị thành công



Hình 37 - Giao diện app mobile khi kết nối thành công

CHƯƠNG 3

KẾT QUẢ, ĐÁNH GIÁ VÀ GIAO DIỆN

3.1. Môi trường thực nghiệm

Để đáp xây dựng và kiểm tra hệ thống và mô hình huấn luyện, tôi đã thực hiện cài đặt các thư viện và chạy thực nghiệm trên máy tính có cấu hình như sau:

3.1.1. Cấu hình máy:

- AMD Quad Core R5-2500U, 3.6 GHZ, 16 GB RAM;
- Máy sử dụng hệ điều hành Window, 64 bit;

3.1.2. Các thư viện được sử dụng:

Bảng 1 - Thư viện sử dụng

Tên thư viện	Phiên bản
Opencv-python	4.7.0.72
Mediapipe	0.9.0.1
Keras	2.11.0
Tensorflow	2.11.0
Numpy	1.21.6
Matplotlib	3.5.3
Pandas	1.3.5
Metrics	0.3.3
Tuya-connector-python	0.1.2
Math	31.0
Tkinter	0.2

Lưu ý: Với mong muốn phát triển hệ thống cho cộng đồng, do vậy, mã nguồn cài đặt và dữ liệu (dataset) của tất cả các động tác (ảnh đã thu thập) sẽ không mang tính cá nhân và bản quyền, chúng tôi đã cung cấp DATM trên github:

https://github.com/tmdat17/DATM_Predict_Person_Sleep_App.git

3.2. Kết quả kiểm tra, đánh giá

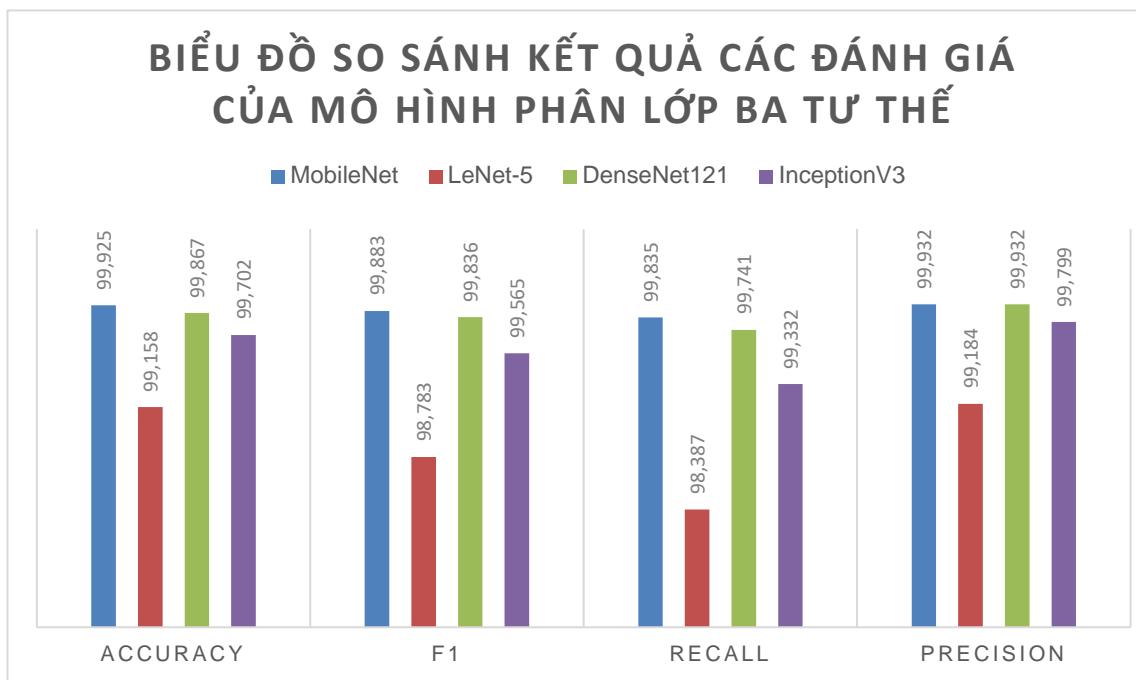
3.2.1. Kết quả phân lớp ba tư thế

Để lựa chọn mô hình phân lớp tư thế, chúng tôi đã thực hiện chạy mô hình với nhiều tham số khác nhau. Ở đây, tôi đã kiểm tra với 04 mô hình phân lớp, cụ thể bao gồm: CNN (MobileNet, DenseNet121, LeNet-5, InceptionV3). Các mô hình được chạy trung bình 6 lần và có áp dụng kỹ thuật early stopping với tham số patience = 4

Bảng 2 - Thể hiện so sánh đánh giá các mô hình phân lớp ba tư thế

	AVG = 6, PATIENCE = 4, EPOCH = 54											
	BS 24				BS 32				BS 64			
	Acc	F1	Rec	Pre	Acc	F1	Rec	Pre	Acc	F1	Rec	Pre
MobileNet	99,917	99,9	99,834	99,695	99,892	99,836	99,737	99,935	99,925	99,883	99,835	99,932
Lenet-5	99,158	98,783	98,387	99,184	98,589	98,747	98,022	99,492	98,952	98,798	98,29	99,334
DenseNet121	99,785	99,661	99,392	99,392	99,867	99,836	99,741	99,932	99,768	99,627	99,324	99,933
InceptionV3	99,43	99,226	98,936	99,522	99,702	99,565	99,332	99,799	99,612	99,533	99,202	99,867

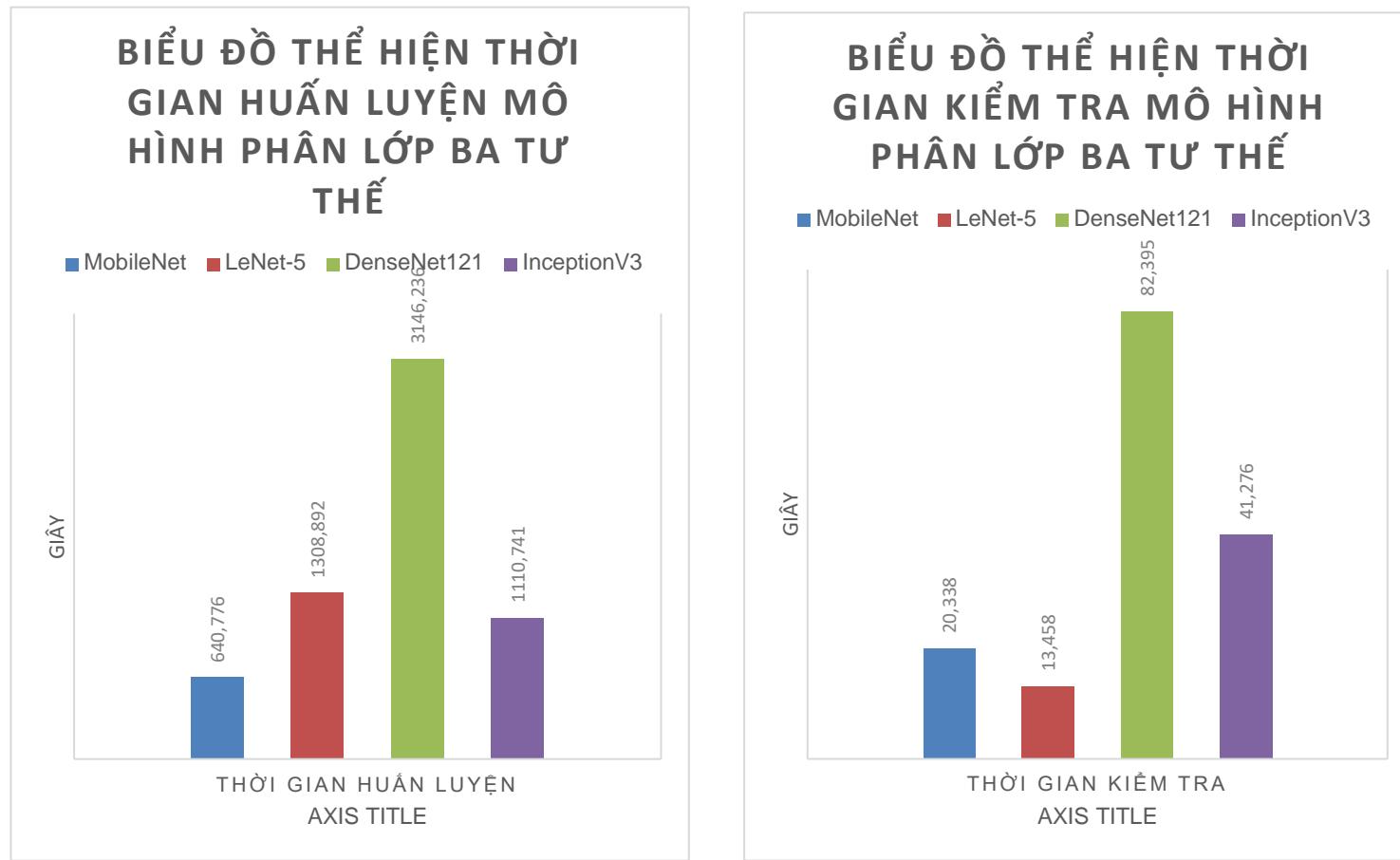
- Để so sánh kết quả độ chính xác mô hình với các tư thế, biểu đồ thể hiện độ chính xác của mô hình máy học với tư thế được thể hiện như sau:



Biểu đồ 1 - So sánh kết quả các đánh giá giữa các mô hình phân lớp ba tư thế

Bảng 3 - Thể hiện so sánh thời gian huấn luyện và dự đoán phân lớp ba tư thế

	AVG = 6, PATIENCE = 4, EPOCH = 54								
	BS 24			BS 32			BS 64		
	Time_train (sec)	Time_test (sec)	Early_Stopping	Time_train (sec)	Time_test (sec)	Early_Stopping	Time_train (sec)	Time_test (sec)	Early_Stopping
MobileNet	640,776	20,44	Epo 6 - 11	771,675	23,803	Epo 5 - 14	665,674	20,338	Epo 6 – 12
Lenet-5	1550,208	15,898	Epo 5 - 9	5442,719	14,485	Epo 5 – 9	1308,892	13,458	Epo 5 – 12
DenseNet121	3294,14	85,772	Epo 7- 12	3417,948	82,395	Epo 6 – 14	3146,236	90,924	Epo 7 – 13
InceptionV3	1355,239	41,276	Epo 5 - 12	1110,741	49,734	Epo 5 - 8	2050,819	43,328	Epo 9 - 17



Biểu đồ 2 - Biểu đồ so sánh thời gian huấn luyện và kiểm tra mô hình phân lớp ba tư thế

Kết Luận: Qua quá trình huấn luyện mô hình phân lớp ba tư thế thì độ chính xác mô hình CNN (MobileNet) có độ chính cao nhất và xét về thời gian thì mô hình MobileNet có thời gian huấn luyện tốt nhất. Tuy nhiên độ chính xác của từng mô hình có khoảng cách không đáng kể, nhưng về thời gian kiểm tra mô hình thì LeNet-5 là hơn tất cả. Vì vậy với hệ thống này chúng tôi chọn mô hình LeNet-5 (có BS = 24 và EPOCH = 54) làm mô hình chính phân lớp các tư thế bởi mô hình có thời gian phù hợp để tiếp tục cho mô hình tiếp theo hoạt động

3.2.2. Kết quả mô hình xác định trạng thái ngủ

Để lựa chọn mô hình xác định trạng thái ngủ, chúng tôi đã thực hiện chạy mô hình với nhiều tham số khác nhau. Ở đây, tôi đã kiểm tra với 04 mô hình cụ thể bao gồm: CNN (SVM, KNN, RandomForest, 1D-ANN), các mô hình được chia theo Kfold với K = 50 và chạy trung bình 50 lần

Bảng 4 - Mô hình kết quả SVM

Kfold = 50, Avg = 50									
SVM	3 lines (12 features)			4 lines (16 features)			5 lines (20 features)		
Parameters	C = 100000	C = 10000	C = 1000	C = 100000	C = 10000	C = 1000	C = 100000	C = 10000	C = 1000
	Gamma = 0.001	Gamma = 0.0001	Gamma = 0.00001	Gamma = 0.001	Gamma = 0.0001	Gamma = 0.00001	Gamma = 0.001	Gamma = 0.0001	Gamma = 0.00001
Accuracy	97.393	97.515	96.969	97.5	98.916	97.500	98.2	97.4	98.1
F1	97.296	97.432	96.879	97.413	98.859	97.357	98.134	97.183	98.034
Recall	97.313	97.547	97.012	97.425	98.876	97.625	98.256	97.179	98.050
Precision	97.433	97.441	96.932	97.605	99.001	97.306	98.221	97.418	98.217
TIME	10.624	1.784	1.210	4.798	0.767	0.680	3.004	0.572	0.511

Bảng 5 - Mô hình kết quả KNN

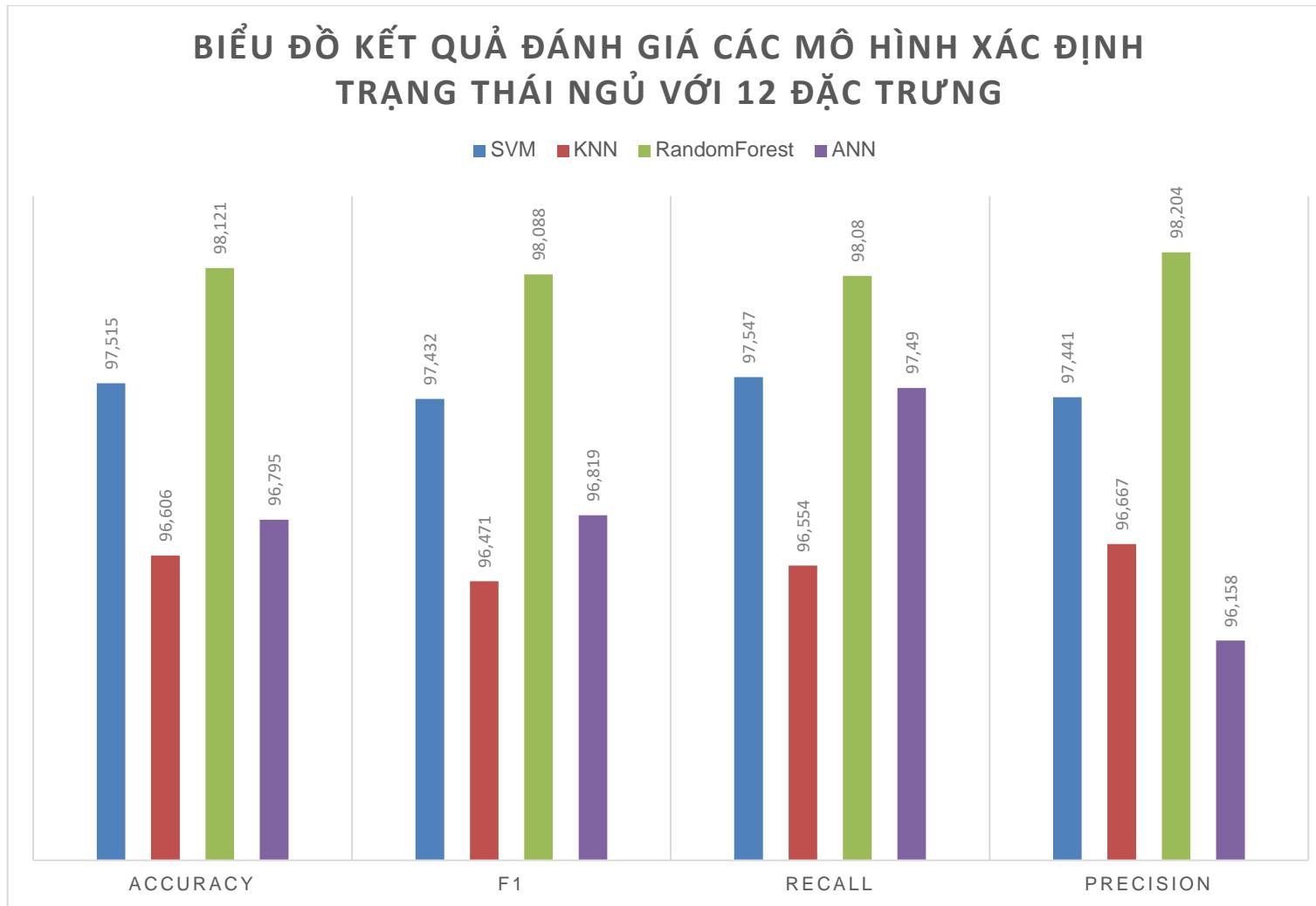
Kfold = 50, Avg = 50									
KNN	3 lines (12 features)			4 lines (16 features)			5 lines (20 features)		
Parameters	K-Neighbors = 20	K-Neighbors = 30	K-Neighbors = 40	K-Neighbors = 20	K-Neighbors = 30	K-Neighbors = 40	K-Neighbors = 20	K-Neighbors = 30	K-Neighbors = 40
Accuracy	96.060	96.606	95.636	96.5	96.166	96.083	96.1	96.2	96.1
F1	95.958	96.471	95.464	96.310	95.995	95.918	95.871	96.067	95.818
Recall	96.064	96.554	95.740	96.371	96.053	95.960	96.066	96.346	95.957
Precision	96.218	96.667	95.545	96.773	96.450	96.437	96.184	96.273	96.141
TIME	1.337	0.360	0.353	0.123	0.125	0.119	0.108	0.109	0.110

Bảng 6 - Mô hình kết quả RandomForest

Kfold = 50, Avg = 50									
RandomForest	3 lines (12 features)			4 lines (16 features)			5 lines (20 features)		
Parameters	N_ESTIMATORS = 100	N_ESTIMATORS = 250	N_ESTIMATORS = 450	N_ESTIMATORS = 100	N_ESTIMATORS = 250	N_ESTIMATORS = 450	N_ESTIMATORS = 100	N_ESTIMATORS = 250	N_ESTIMATORS = 450
	Max_Depth = 15	Max_Depth = 55	Max_Depth = 75	Max_Depth = 15	Max_Depth = 55	Max_Depth = 75	Max_Depth = 15	Max_Depth = 55	Max_Depth = 75
Accuracy	98.121	96.606	96.666	97.583	97.416	98.333	98.0	98.5	97.6
F1	98.088	96.507	96.584	97.489	97.330	98.287	97.846	98.341	97.496
Recall	98.080	96.660	96.724	97.666	97.509	98.396	97.843	98.488	97.471
Precision	98.204	96.668	96.632	97.587	97.431	98.347	98.172	98.390	97.876
TIME	8.98	19.683	34.112	7.494	17.760	32.569	7.027	16.923	30.551

Bảng 7 - Mô hình kết quả 1D-ANN

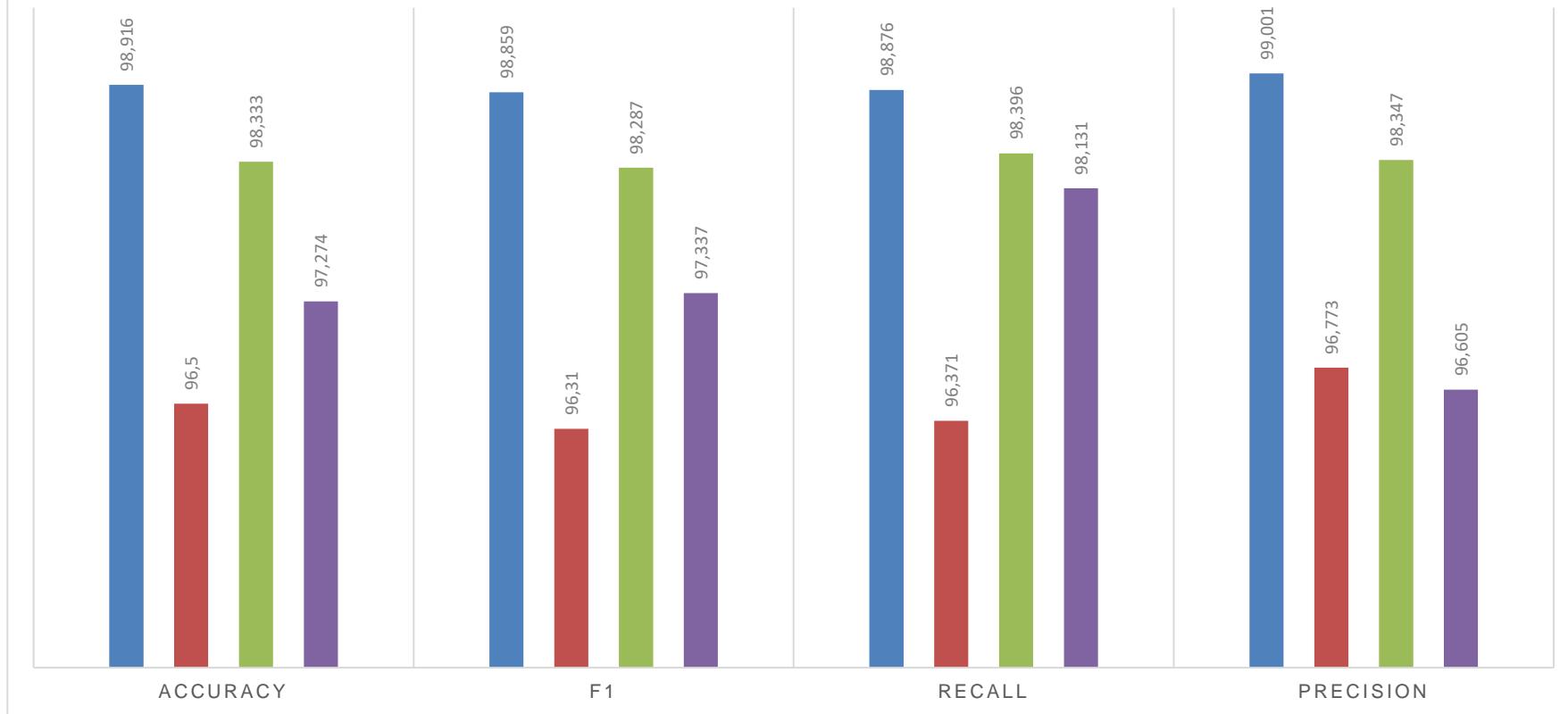
Kfold = 50, Avg = 50									
ANN	3 lines (12 features)			4 lines (16 features)			5 lines (20 features)		
Parameters	Epoch = 20	Epoch = 50	Epoch = 70	Epoch = 20	Epoch = 50	Epoch = 70	Epoch = 20	Epoch = 50	Epoch = 70
	BS = 24	BS = 64	BS = 128	BS = 24	BS = 64	BS = 128	BS = 24	BS = 64	BS = 128
Accuracy	96.602	96.759	96.795	97.129	97.193	97.274	97.191	97.717	97.434
F1	96.568	96.847	96.819	97.153	97.274	97.337	97.264	97.722	97.494
Recall	97.393	97.524	97.490	97.824	97.954	98.131	98.596	98.625	98.335
Precision	95.812	96.227	96.188	96.557	96.647	96.605	96.008	96.889	96.724
TIME (sec)	183.929	254.667	276.158	165.595	233.255	246.635	140.410	215.629	268.728



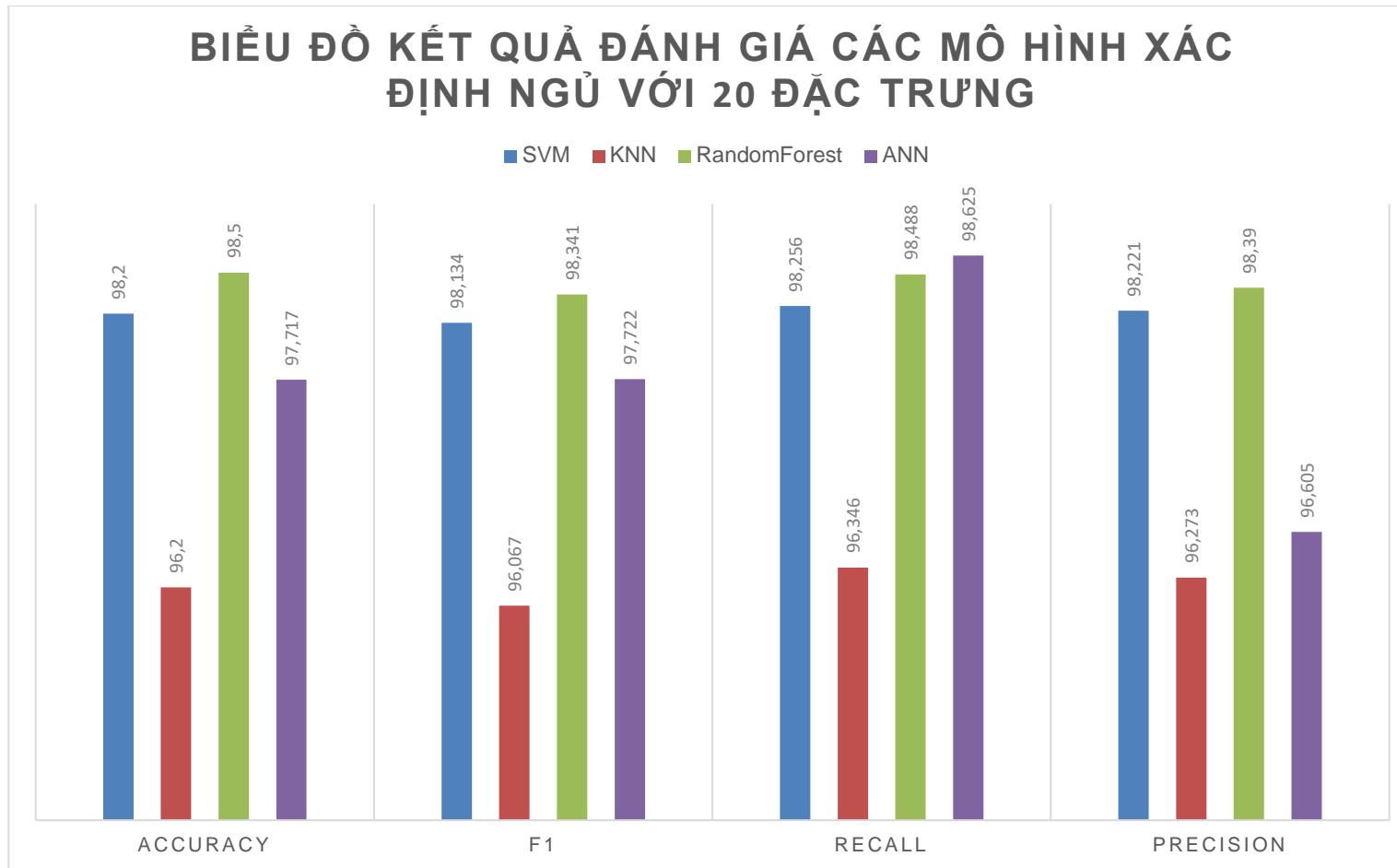
Biểu đồ 3 - Biểu đồ kết quả đánh giá các mô hình xác định ngủ với 12 đặc trưng

BIỂU ĐỒ KẾT QUẢ ĐÁNH GIÁ CÁC MÔ HÌNH XÁC ĐỊNH NGỦ VỚI 16 ĐẶC TRƯNG

SVM ■ KNN ■ RandomForest ■ ANN



Biểu đồ 4 - Biểu đồ kết quả đánh giá các mô hình xác định ngủ với 14 đặc trưng



Biểu đồ 5 - Biểu đồ kết quả đánh giá các mô hình xác định ngủ với 20 đặc trưng

Kết Luận: Qua quá trình huấn luyện mô hình xác định trạng thái ngủ thì độ chính xác của mô hình RandomForest có độ chính cao nhất và nhưng xét về thời gian thì mô hình SVM có thời gian huấn luyện tốt nhất. Đối với hệ thống này chúng tôi chọn mô hình LeNet-5 (có BS = 24 và EPOCH = 54) có thời gian thấp do đó mô hình xác định trạng thái ngủ tôi chọn mô hình SVM làm mô hình chính để xác định trạng thái ngủ cho đối tượng người.

3.2.3. Kết quả xác định trạng thái ngủ theo phương pháp chọn ngưỡng

Bảng 8 - Bảng thống kê kết quả với các ngưỡng khác nhau đối với file ngủ

A=2500lines		total 0: 567	total 1: 9433	Sleep file									
Feature		r=3 lines (12 features)				4 lines (16 features)				5 lines (20 features)			
Threshold Beta		70%	80%	90%	100%	70%	80%	90%	100%	70%	80%	90%	100%
quantity label 1 in 3 4 5 lines T		8.399	9.600	10.8	12	11.2	12.8	14.4	16	14	16	18	20
accuracy		98,4375	97,2355	92,6682	64,9038	98,0769	96,9551	87,3397	61,6987	97,9959	95,1903	82,9659	57,5150
mean		11,3269	11,3269	11,3269	11,3269	15,1057	15,1057	15,1057	15,1057	18,8857	18,8857	18,8857	18,8857
median		12,0	12,0	12,0	12,0	16,0	16,0	16,0	16,0	20,0	20,0	20,0	20,0
count pass quantity		819	809	771	540	612	605	545	385	489	475	414	287
total count line pass		int(A/r)=832	832	832	832	624	624	624	624	499	499	499	499

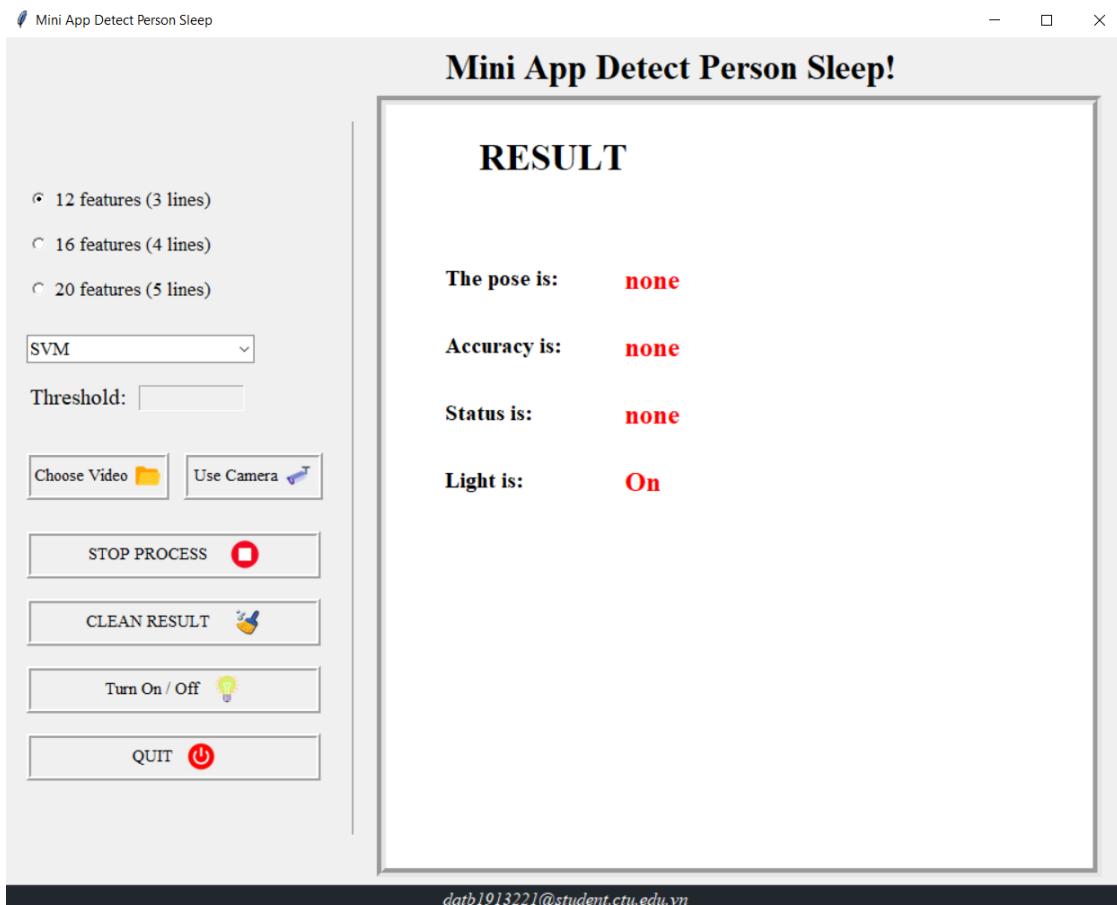
Bảng 9 - Bảng thống kê kết quả với các ngưỡng khác nhau đối với file thức

A=2500lines		total 0: 6603	total 1: 3397	Wake file									
Feature		r=3 lines (12 features)				4 lines (16 features)				5 lines (20 features)			
Threshold Beta		70%	80%	90%	100%	70%	80%	90%	100%	70%	80%	90%	100%
quantity label 0 in 3 4 5 lines T		8.399	9.600	10.8	12	11.2	12.8	14.4	16	14	16	18	20
accuracy		59,4950	44,5910	28,8460	5,1680	54,8070	40,3840	16,5060	2,7240	49,8990	28,0560	9,6190	1,6030
mean		7,9200	7,9200	7,9200	7,9200	10,5570	10,5570	10,5570	10,5570	13,1940	13,1940	13,1940	13,1940
median		8,0	8,0	8,0	8,0	11,0	11,0	11,0	11,0	13,0	13,0	13,0	13,0
count pass quantity		495	371	240	43	342	252	103	17	249	140	48	8
total count line pass		832	832	832	832	624	624	624	624	499	499	499	499

Do hệ thống chỉ mong muốn xác định được trạng thái ngủ của con người nên chúng tôi sẽ tập trung chủ yếu vào file dữ liệu huấn luyện của trạng thái ngủ để thống kê và đặt ngưỡng cho ứng dụng chính của hệ thống DATM

3.3. Giao diện

- Giao diện là một phần không thể thiếu đối với hệ thống, dù hệ thống có ít tính năng thì vẫn có giao diện. Giao diện nhằm giúp cho người dùng tương tác với hệ thống thuận tiện và dễ dàng hơn.



Hình 38 - Giao diện hệ thống

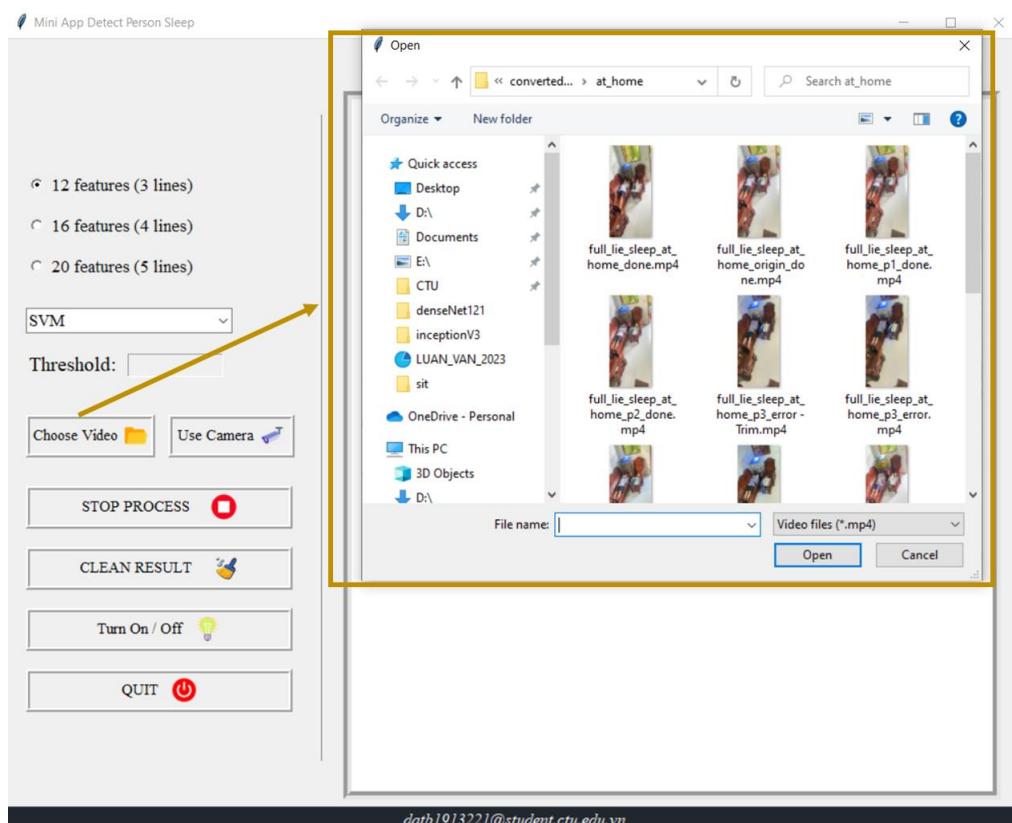
Qua hình 45 chúng ta thấy được giao diện hệ thống có các tính năng:

- Nút chọn số lượng đặc trưng cho đầu vào mô hình dự đoán, đặc trưng càng nhiều khoảng thời gian càng lâu và ngược lại;
- Nút selection cho việc lựa chọn 1 trong 2 mô hình là SVM và theo ngưỡng thủ công cho việc dự đoán;
- Ô input nhập ngưỡng khi người dùng chọn lựa chọn 2 là theo ngưỡng;
- Nút mở file: Dùng để mở file từ thư mục có video;
- Nút mở camera: Dùng để mở camera từ hệ thống;
- Nút tắt: dùng để tắt camera hoặc tắt video;
- Nút làm sạch kết quả: sẽ làm mới kết quả về lúc bắt đầu;

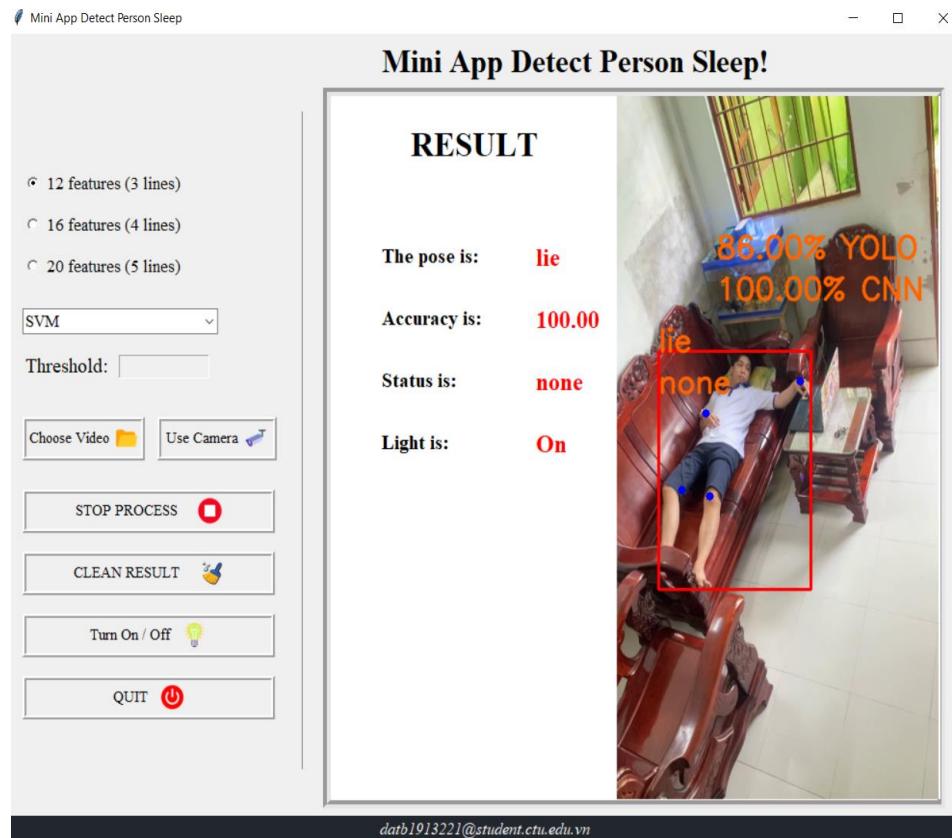
- Nút bật/tắt thiết bị điện: điều khiển thiết bị điện Tuya qua nút này;
- Nút đóng: Dùng để thoát toàn bộ hệ thống.

Khung bên phải là nơi sẽ hiển thị kết quả gồm các kết quả sau:

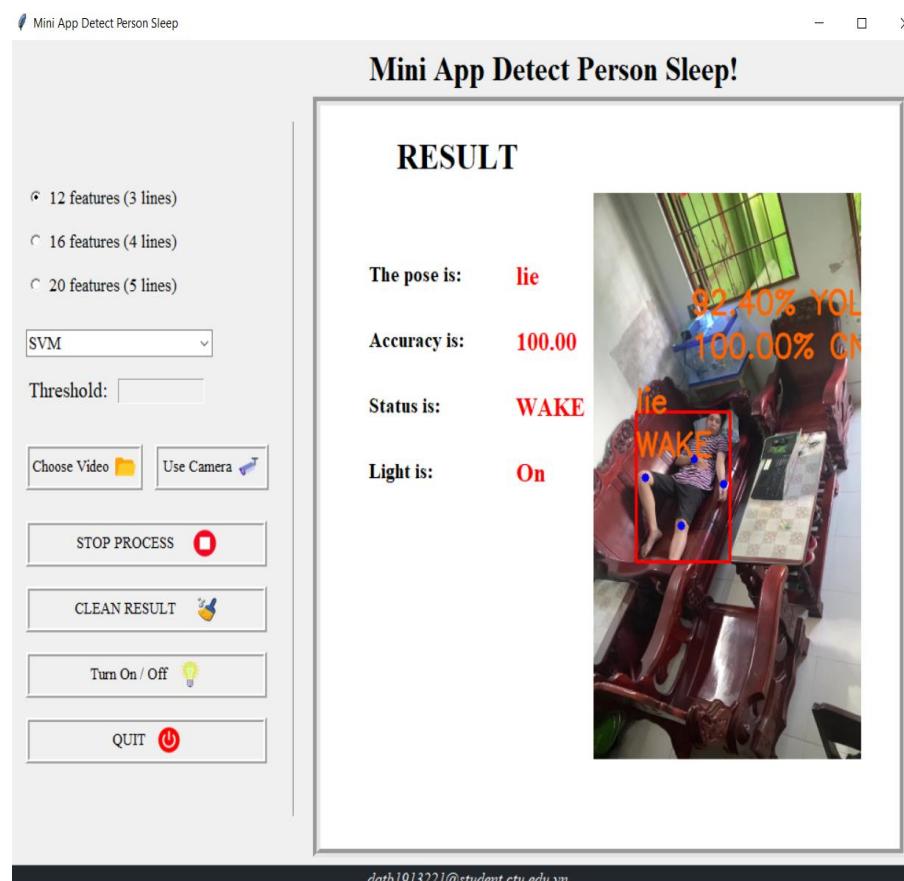
- Tư thế của người trong video hay camera là gì;
- Độ chính xác của mô hình phân loại tư thế;
- Trạng thái ngủ hay thức của người trong video hoặc camera được xác định khi người đó đang ở tư thế nằm;
- Trạng thái của thiết bị điện đang mở hay đang tắt;
- Và cuối cùng là khu vực trống là nơi hiển thị video hoặc camera.



Hình 39 - Giao diện khi nhấn nút mở file



Hình 40 - Giao diện khi mở file thành công



Hình 41 - Giao diện khi hệ thống đã xác định được trạng thái của đối tượng sau một khoảng thời gian

PHẦN KẾT LUẬN

1. Kết quả đạt được

Để giải quyết bài toán xác định đối tượng người đang ngủ và điều khiển thiết bị điện khi đối tượng ngủ, tôi đã thực hiện quá trình cài đặt và lựa chọn mô hình máy học. Ở đây, bài toán đã được xử lý với thị giác máy tính. Cụ thể là vận dụng Human Pose Estimation (HPE) để lấy các điểm trên cơ thể và xây dựng đặc trưng mới cho từng điểm qua từng khoảng thời gian khác nhau. Những đóng góp chính của luận văn này là

- Xây dựng được ứng dụng giám sát tư thế cũng như xác định trạng thái ngủ của con người và sau đó sẽ thực hiện ra lệnh điều khiển thiết bị điện với Tuya API;
- Thu thập bộ dữ liệu ảnh với 3 tư thế bao gồm: đứng, nằm, ngồi;
- Thu thập bộ dữ liệu ảnh với tư thế nằm gồm 2 trạng thái: ngủ và thức;
- Xây dựng mô hình phân lớp cho từng tư thế;
- Tận dụng được mô hình Human Pose Estimation để xác định các điểm trên cơ thể;
- Đề xuất một giải thuật để xây dựng bộ dữ liệu có đặc trưng mới hoàn toàn;
- Xây dựng mô hình xác định trạng thái ngủ hay thức của con người từ bộ dữ liệu đặc trưng trên;
- Kết nối được thiết bị Tuya với Cloud Tuya và triển khai chúng sau khi xác định được trạng thái của con người;
- Xây dựng được giao diện người dùng trên nền tảng ứng dụng.

2. Hướng phát triển

- Hiện tại, đề xuất của chúng tôi chỉ thực hiện trên ảnh 2D được thu nhận từ Camera thông thường nên một số khuyết điểm có thể được cải tiến: (1) sử dụng camera ở nhiều góc khác nhau trong nhà để có góc nhìn rộng, (2) sử dụng camera có chất lượng tốt cũng như chuyên dụng để có thể quan sát, theo dõi người trong gia đình thay vì chỉ dùng điện thoại.
- Phát triển trên nhiều nền tảng web và mobile;
- Xây dựng thêm cơ sở dữ liệu;
- Xây dựng mô hình xác định trạng thái ngủ với tư thế ngồi;
- Xây dựng mô hình xác định khu vực trong nhà không có người để điều khiển các thiết bị điện.

TÀI LIỆU THAM KHẢO

- [1]. DEWANGAN, Rajeshwar Kumar; CHOUHAN, Yamini. A Review on Object Detection using Open CV Method. *International Research Journal of Engineering and Technology (IRJET)*, 2020, 7.09: 3859-3861.
- [2]. Ian Goodfellow, Yoshua Bengio, Aaron Courville, in Deep Learning, The MIT Press, 2016, p. 9.
- [3]. KRAMER, Oliver; KRAMER, Oliver. K-nearest neighbors. *Dimensionality reduction with unsupervised nearest neighbors*, 2013, 13-23.
- [4]. LECUN, Yann, et al. LeNet-5, convolutional neural networks. URL: <http://yann.lecun.com/exdb/lenet>, 2015, 20.5: 14.
- [5]. LÊ, Việt Tuân. Nhận diện hành động của người trong video. 2012.
- [6]. PAL, Mahesh; MATHER, Paul M. Support vector machines for classification in remote sensing. *International journal of remote sensing*, 2005, 26.5: 1007-1011.
- [7]. VERGROESEN, Pieter-Paul A., et al. Intradiscal pressure depends on recent loading and correlates with disc height and compressive stiffness. *European Spine Journal*, 2014, 23: 2359-2368.
- [8]. Yann LeCun Leon Bottou Joshua Bengio and Patrick Haffner (1998), Gradient Based Learning Applied to Document Recognition.
- [9]. ZHANG, Chuan-Wei, et al. Pedestrian detection based on improved LeNet-5 convolutional neural network. *Journal of Algorithms & Computational Technology*, 2019, 13: 1748302619873601.
- [10]. ZHOU, Chunting, et al. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*, 2015.
- [11]. ZHANG, Dong; SHAH, Mubarak. Human pose estimation in videos. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015. p. 2012-2020.
- [12]. Li, Wen, Fei Jiang, and Ruimin Shen. "Sleep gesture detection in classroom monitor system." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.
- [13]. Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.