NGUYỄN QUANG HUY

KHOA HỌC MÁY TÍNH

2024 – 2025

HÀ NỘI – 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN QUANG HUY

ỨNG DỤNG MÔ HÌNH TRÍCH XUẤT KHUNG XƯƠNG VÀ HỌC MÁY TRONG NHẬN DIỆN TƯ THẾ YOGA THỜI GIAN THỰC

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(***Theo định hướng ứng dụng***)

HÀ NỘI - NĂM 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN QUANG HUY

ỨNG DỤNG MÔ HÌNH TRÍCH XUẤT KHUNG XƯƠNG VÀ HỌC MÁY TRONG NHẬN DIỆN TƯ THẾ YOGA THỜI GIAN THỰC

Chuyên ngành: Khoa học máy tính
Mã số: 8.48.01.01

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT
(KHOA HỌC MÁY TÍNH)
(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN DUY PHƯƠNG

HÀ NỘI - NĂM 2025

LỜI CAM ĐOAN

Tôi cam kết rằng công trình nghiên cứu này là sản phẩm trí tuệ của tôi. Các số liệu và các kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng xuất hiện trong bất kỳ tài liệu khoa học nào của tác giả khác.

Hà Nội, tháng 12 năm 2025

Tác giả đề án

Nguyễn Quang Huy

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn tới các thầy cô giáo thuộc Khoa Công nghệ Thông tin 1 và và Khoa Đào tạo Sau đại học – PTIT. Suốt chặng đường học tập, rèn luyện và làm việc tại Học viện, em đã nhận được không chỉ những kiến thức chuyên ngành vững chắc mà còn là các bài học giá trị về tư duy, cách làm việc và tinh thần nghiên cứu khoa học. Đây chính là điểm tựa vững chắc giúp em thực hiện tốt đề án này.

Em xin gửi lời tri ân đặc biệt tới thầy **TS. Nguyễn Duy Phương**, người thầy đã tận tâm dẫn dắt em trong suốt thời gian thực hiện đề án. Những định hướng chuyên môn, góp ý chi tiết cùng sự hỗ trợ tận tâm của thầy đã giúp em từng bước cải thiện đề tài, cũng như rèn luyện được phương pháp nghiên cứu và viết đề án nghiêm túc và khoa học hơn.

Ngoài ra, mình xin cảm ơn các đồng nghiệp cùng tham gia tư vấn, đóng góp và các bạn sinh viên hỗ trợ thu thập, xây dựng dữ liệu cho đề tài. Đồng thời, con xin bày tỏ lòng biết ơn đến gia đình đã luôn khích lệ con vượt qua mọi thử thách và hoàn thiện công trình này. Đề án này không chỉ là minh chứng cho thành quả học tập mà còn là tình cảm, sự quan tâm và giúp sức quý báu mà em may mắn nhận được từ mọi người.

Hà Nội, tháng 12 năm 2025

Tác giả đề án

Nguyễn Quang Huy

MỤC LỤC

LỜI CAM ĐOAN ..... **i**

LỜI CẢM ƠN ..... **ii**

MỤC LỤC ..... **iii**

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT ..... **v**

DANH SÁCH BẢNG ..... **viii**

DANH SÁCH HÌNH VẼ ..... **ix**

MỞ ĐẦU **1**

1. Lý do chọn đề tài 1

2. Tổng quan về vấn đề nghiên cứu 2

3. Mục đích nghiên cứu 3

4. Đối tượng và phạm vi nghiên cứu 4

5. Phương pháp nghiên cứu 4

CHƯƠNG 1: TỔNG QUAN VỀ NHẬN DIỆN TƯ THẾ YOGA **5**

1.1. Khái quát về nhận diện tư thế người 5

1.2. Bài toán nhận diện tư thế yoga 6

1.3. Các phương pháp và mô hình nhận diện tư thế người 7

1.4. Nghiên cứu liên quan đến nhận diện tư thế yoga 8

1.5. Kết luận chương 1 11

CHƯƠNG 2: NGHIÊN CỨU ĐỀ XUẤT HỆ THỐNG NHẬN DIỆN TƯ THẾ YOGA THỜI GIAN THỰC **13**

2.1. Mô hình đề xuất cho bài toán 13

2.2. Mô đun trích xuất khung xương 14

2.2.1. Các mô hình được áp dụng 14

2.2.2. Nhận diện và xử lý keypoints.....19
2.3. Mô đun xử lý và trích xuất đặc trưng.....21
2.4. Mô đun phân loại tư thế.....22
2.5. Mô đun phản hồi và hiển thị.....24
2.6. Kết luận chương 2.....25
CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ.....27
3.1. Bộ dữ liệu thực nghiệm.....27
3.2. Cài đặt thực nghiệm.....29
3.2.1. Môi trường phần cứng và phần mềm.....29
3.2.2. Thiết lập mô hình HPE.....30
3.2.3.Thiết lập đào tạo mô hình ML.....31
3.2.4. Tiêu chí đánh giá hiệu năng.....33
3.3. Kết quả và đánh giá thực nghiệm.....34
3.3.1. Hiệu suất ước lượng tư thế người.....34
3.3.2. So sánh hiệu suất kết hợp HPE và thuật toán học máy.....36
3.3.3. Đánh giá hiệu suất thời gian thực.....38
3.4. Kết luận chương 3.....40
KẾT LUẬN.....42
CÔNG BỐ KHOA HỌC.....43
TÀI LIỆU.....44
PHỤ LỤC.....49

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tên đầy đủ (Tiếng Anh)	Tên đầy đủ (Tiếng Việt)
3D-CNN	Three-Dimensional Convolutional Neural Network	Mạng nơ-ron tích chập ba chiều
AI	Artificial Intelligence	Trí tuệ nhân tạo
AR	Augmented Reality	Thực tế tăng cường
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
CPU	Central Processing Unit	Bộ xử lý trung tâm
CV	Computer Vision	Thị giác máy tính
DL	Deep Learning	Học sâu
DT	Decision Tree	Cây quyết định
FPN	Feature Pyramid Network	Mạng kim tự tháp đặc trưng
FPS	Frames Per Second	Số khung hình trên giây
GAUSSIANNB	Gaussian Naive Bayes	Bộ phân loại Naive Bayes với phân phối Gauss
HAR	Human Action Recognition	Nhận diện hành động của con người
HCI	Human–Computer Interaction	Tương tác người–máy
HOG	Histogram of Oriented Gradients	Biểu đồ hướng của gradient
HPE	Human Pose Estimation	Ước lượng tư thế người
HPR	Human Pose Recognition	Nhận diện tư thế người

HRNet	High-Resolution Network	Mạng biểu diễn độ phân giải cao
KNN	K-Nearest Neighbors	Thuật toán K láng giềng gần nhất
LIGHTGBM	Light Gradient Boosting Machine	Máy tăng cường độ dốc nhẹ (LightGBM)
LSTM	Long Short-Term Memory Network	Mạng bộ nhớ dài–ngắn hạn
ML	Machine Learning	Học máy
MLP	Multi-Layer Perceptron Classifier	Bộ phân loại mạng perceptron nhiều lớp
MPPE	Multi-Person Pose Estimation	Ước lượng tư thế đa người
NMS	Non-Maximum Suppression	Phương pháp khử cực đại không cần thiết
ONNX	Open Neural Network Exchange	Chuẩn trao đổi mô hình mạng nơ-ron mở
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
RBF	Radial Basis Function	Hàm cơ sở xuyên tâm (dùng cho SVM)
RF	Random Forest	Rừng ngẫu nhiên
RGB	Red, Green, Blue	Hệ màu Đỏ, Lục, Lam
ROI	Region of Interest	Vùng quan tâm
RTMO	Real-Time Multi-Person One-Stage Pose Estimation	Ước lượng tư thế nhiều người trong thời gian thực, một giai đoạn
RTMPose	Real-Time Multi-Person Pose Estimation	Ước lượng tư thế nhiều người trong thời gian thực

SIFT	Scale-Invariant Feature Transform	Phép biến đổi đặc trưng bất biến theo tỉ lệ
SVM	Support Vector Machine	Máy vectơ hỗ trợ
TensorRT	Tensor Runtime	Trình tăng tốc suy luận TensorRT
ViT	Vision Transformer	Bộ biến đổi thị giác
ViTPose	Vision Transformer Pose	Mô hình tư thế dựa trên Vision Transformer
XGBOOST	Extreme Gradient Boosting	Thuật toán tăng cường độ dốc cực đại (XGBoost)

DANH SÁCH BẢNG

Bảng 3.1: Số lượng ảnh cho từng tư thế Yoga.....28
Bảng 3.2: Các gói thư viện được sử dụng trong quá trình thực nghiệm.....29
Bảng 3.3: Không gian tìm kiếm siêu tham số cho các mô hình ML.....32
Bảng 3.4: So sánh kết quả thực nghiệm của các mô hình HPE.....35
Bảng PL.0.1: Phụ lục tổng hợp kết quả Cross-Validation và Test của mô hình HPE+ML.....49

DANH SÁCH HÌNH VẼ

Hình 2.1: Pipeline của giải pháp đề xuất cho bài toán nhận diện tư thế Yoga. 13
Hình 2.2: Các mô hình HPE được sử dụng trong đề án.....15
Hình 2.3: Danh sách các điểm khớp của COCO dataset (17 điểm) và BlazePose (33 điểm) bằng tiếng Việt.....20
Hình 3.1: Minh họa 12 tư thế Yoga trong bộ dữ liệu 28
Hình 3.2: Kết quả độ chính xác (Accuracy) của các tổ hợp mô hình HPE và ML..36
Hình 3.3: Kết quả điểm F1-Score của các tổ hợp mô hình HPE và ML 37
Hình 3.4: Kết quả nhận diện và phân loại tư thế Yoga thời gian thực trên máy cây Gigabyte (32 GB RAM) sử dụng MoveNet TFLite và SVM. 39
Hình 3.5: Kết quả nhận diện và phân loại tư thế Yoga thời gian thực trên laptop Acer (16 GB RAM) sử dụng MoveNet TFLite và SVM. 40

MỞ ĐẦU

1. Lý do chọn đề tài

Yoga, với những lợi ích vượt trội về thể chất và tinh thần [1] [2], đang ghi nhận mức độ phổ biến tăng mạnh trên toàn thế giới, nhất là mô hình tập luyện tại nhà thu hút nhiều người dùng hơn trước. Sự bùng nổ của các nền tảng trực tuyến, ứng dụng di động và video hướng dẫn đã tạo điều kiện thuận lợi cho việc tiếp cận bộ môn này mà không cần đến phòng tập truyền thống. Tuy nhiên, cùng với sự tiện lợi đó, một vấn đề lớn đã nảy sinh: thiếu đi sự hướng dẫn và phản hồi trực tiếp từ người dạy. Người tập tại nhà, nhất là mới bắt đầu tập Yoga, họ thường không dễ dàng xác định liệu tư thế mình đang thực hiện đã đúng chuẩn hay chưa. Tư thế sai kéo dài vừa làm suy yếu kết quả tập luyện, vừa âm thầm mang theo rủi ro tổn thương cơ thể. Người dùng vẫn phải tự mình đối chiếu với hình ảnh mẫu, hoặc ghi lại video và xem lại sau, một quy trình tốn thời gian và không mang lại hiệu quả tức thì. Từ thực trạng trên, đề tài "Ứng dụng mô hình trích xuất khung xương và học máy trong nhận diện tư thế Yoga thời gian thực" được lựa chọn với mục tiêu nghiên cứu, tổng hợp và thực nghiệm để tìm ra giải pháp hiệu quả, phù hợp cho các ứng dụng nhận diện tư thế dáng yoga. Đây là một chủ đề giao thoa giữa sức khỏe cộng đồng và công nghệ thông tin, cho thấy tiềm năng đóng góp cả trong phương diện học thuật lẫn ứng dụng.

Về mặt khoa học, đề tài góp ý nghĩa trong thị giác máy tính và học máy. Nó không chỉ là việc ứng dụng đơn thuần các công nghệ hiện có mà còn đi sâu vào việc phân tích, đánh giá và so sánh hiệu suất của các tổ hợp phương pháp khác nhau trong một bài toán cụ thể là nhận diện tư thế người trong điều kiện thực tế. Kết quả nghiên cứu có thể làm bệ phóng cho những hướng nghiên cứu chuyên biệt về sau trong huấn luyện thể thao thông minh, chăm sóc sức khỏe từ xa. Về tính cấp thiết, trong bối cảnh lĩnh vực trong đời sống đều chịu ảnh hưởng từ công nghệ, đưa AI vào phục vụ mục tiêu nâng cấp chất lượng tập luyện thể thao tại nhà không những giúp người dân nâng cao sức khỏe mà còn tạo điều kiện hình thành một lối sống lành mạnh, chủ động. Một hệ thống nhận diện tư thế yoga hiệu quả, có khả năng hoạt động mượt mà trên các thiết bị phổ biến (webcam, camera điện thoại), sẽ là công cụ đắc lực hỗ trợ hàng triệu

người yêu thích yoga trên toàn thế giới. Cuối cùng, về tính khả thi, đề tài có tính khả thi cao nhờ vào sự phát triển mạnh mẽ của công nghệ trích xuất khung xương với độ chính xác và tốc độ xử lý cao, cùng các thư viện học máy dễ sử dụng. Bên cạnh đó, bộ dữ liệu là ảnh dễ dàng thu thập và đánh giá. Hệ thống có thể triển khai trên máy tính cá nhân dùng webcam thông thường, giúp tiết kiệm chi phí và tăng tính ứng dụng. Phạm vi nghiên cứu được giới hạn hợp lý, tập trung vào nhận diện tư thế cơ bản cho một người, đảm bảo hoàn thành trong thời gian của một luận văn thạc sĩ.

2. Tổng quan về vấn đề nghiên cứu

Trong những năm gần đây, nhiều nghiên cứu đã tiếp cận bài toán phân loại tư thế Yoga bằng mô hình học sâu (Deep Learning – DL) trực tiếp trên ảnh RGB [3], tiêu biểu như ResNet [4]. Tuy những mô hình này đạt hiệu quả tốt trên các bộ dữ liệu quy mô lớn, chúng vẫn tồn tại nhược điểm: dễ bị ảnh hưởng bởi nhiễu từ phông nền, điều kiện ánh sáng và trang phục người tập, trong khi lại chưa khai thác được các đặc trưng hình học cốt lõi của tư thế, người dùng không thể biết mình tập đúng sai như thế nào. Để khắc phục, các nghiên cứu sau đó chuyển sang hướng skeleton-based, sử dụng các mô hình HPE như YOLOPose [5], Movenet [6], BlazePose [7] để trích xuất tọa độ khớp, sau đó tính toán các đặc trưng hình học (khoảng cách, góc) để phân loại, từ đó loại bỏ phần lớn thông tin nền gây nhiễu, giảm yêu cầu xử lý hình ảnh phức tạp và giữ lại cấu trúc tư thế thiết yếu. Tuy nhiên, các tư thế Yoga có mức độ biến thiên lớn do sự khác biệt về thể trạng và độ linh hoạt giữa người tập, khiến phương pháp dựa vào tính toán đặc trưng thủ công dễ bị giảm độ chính xác khi gặp tư thế phức tạp hoặc dữ liệu không hoàn chỉnh. Gần đây, hướng kết hợp trích xuất khung xương với các thuật toán ML như KNN [8], SVM [9], Random Forest [10]... đã được đề xuất như một giải pháp hiệu quả. Các mô hình này học trực tiếp từ dữ liệu khung xương, tận dụng ưu điểm nhẹ, nhanh, và khả năng triển khai tốt trên các thiết bị hạn chế tài nguyên. Ngoài ra, do chỉ sử dụng các tọa độ khớp làm đầu vào, chúng không bị ảnh hưởng bởi yếu tố nền ảnh, từ đó tăng tính ổn định trong phân loại tư thế Yoga. Dẫu vậy, không ít vấn đề vẫn chưa được xử lý hoàn toàn. Những vấn đề còn tồn tại bao gồm: (1) thiếu bộ dữ liệu đặc thù cho người mới bắt đầu, đặc biệt tại Việt Nam; (2)

thiếu so sánh toàn diện giữa các mô hình HPE (BlazePose, YOLOPose, RTMPose,...) và thuật toán học máy để tìm ra tổ hợp tốt nhất; (3) Chưa có phân tích, đánh giá toàn diện của mô hình về tốc độ nhận diện trong môi trường thực tế.

3. Mục đích nghiên cứu

Đề án hướng đến xây dựng và đánh giá hệ thống nhận diện tư thế yoga thời gian thực, sử dụng sự kết hợp giữa các mô hình trích xuất khung xương và thuật toán học máy.

Về mặt lý luận: Đề án khảo sát nền tảng lý thuyết của phương pháp trích xuất khung xương HPE từ ảnh/video, bao gồm các mô hình nổi bật như YoloPose, RTMPose, RTMOPose [11] ... Đồng thời, tìm hiểu các thuật toán học máy kinh điển (KNN, SVM, Decision Tree [12], Random Forest, XGBoost [13], LightGBM [14]) được dùng phổ biến cho bài toán phân loại tư thế dựa trên dữ liệu khung xương. Đồng thời đề án cũng phân tích đặc trưng khung xương, nghiên cứu tiền xử lý dữ liệu khung xương (chuẩn hóa vị trí, tỷ lệ) và phương pháp trích xuất các đặc trưng hình học có ý nghĩa (góc khớp, khoảng cách, tỷ lệ) để tối ưu hóa khả năng phân loại tư thế. Cuối cùng đánh giá định lượng hiệu suất bằng việc xác định các tiêu chí đo lường hiệu suất quan trọng cho hệ thống nhận diện thời gian thực (ví dụ: Độ chính xác, độ triệu hồi, điểm F1, tốc độ khung hình - FPS, độ trễ phản hồi - latency) và xây dựng phương pháp đánh giá khách quan các tổ hợp mô hình HPE và thuật toán học máy.

Về mặt thực tiễn: Đề án xây dựng bộ dữ liệu yoga đa dạng YogaSinglePose-PTIT, thu thập và xây dựng một bộ dữ liệu tư thế yoga cơ bản, chất lượng cao, có chú thích đầy đủ, phù hợp với điều kiện luyện tập cơ bản tại nhà để phục vụ giai đoạn huấn luyện cùng đánh giá mô hình. Ngoài ra, phát triển hệ thống nhận diện thời gian thực, triển khai một hệ thống nhận diện tư thế yoga cho một người dùng, có khả năng xử lý video trực tiếp từ webcam, trích xuất khung xương, xử lý dữ liệu, phân loại tư thế và hiển thị kết quả phản hồi theo thời gian thực. Và tiến hành so sánh và lựa chọn tổ hợp tối ưu mô hình HPE và thuật toán ML khác nhau trên bộ dữ liệu đã xây dựng. Từ đó, đề xuất tổ hợp tối ưu nhất, cân bằng giữa độ chuẩn xác, hiệu năng và tính khả thi vận hành trên thiết bị phần cứng phổ thông.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng: Công trình tập chung tìm hiểu một số mô hình ước lượng tư thế người HPE như MediaPipe Pose, YoloPose, RTMPose,.. và các thuật toán ML được ứng dụng trong bài toán nhận diện tư thế yoga. Bên cạnh đó đề án tập trung và đối tượng nghiên cứu xây dựng bộ dữ liệu là các sinh viên PTIT các tư thế yoga cơ bản, bao gồm 12 tư thế phù hợp cho người mới tập, đặc biệt nhắm đến sinh viên và người đi làm có thói quen ngồi lâu.

Phạm vi nghiên cứu : Công trình chỉ xét đến và giới hạn ở việc nhận dạng 12 tư thế Yoga cơ bản (bao gồm: ngọn núi 1, chó úp mặt, tấm ván, tam giác, cái cây, chiến binh I, chiến binh II, cúi người về phía trước, cây cầu, em bé, ngọn núi 2(leo núi) và tấm ván thấp) được thực hiện bởi người mới tập, chủ yếu là sinh viên và người làm việc văn phòng. Dữ liệu được thu thập từ khoảng 10 người tham gia (có cả nam và nữ), quay trong các điều kiện ánh sáng, phông nền và thiết bị khác nhau nhằm phản ánh môi trường thực tế đa dạng. Nghiên cứu được thực thi trong không gian tại nhà, sử dụng camera điện thoại (độ phân giải tối thiểu 720p) đặt cách người tập 4-5m để ghi hình.

5. Phương pháp nghiên cứu

Nghiên cứu lý thuyết: Đề án tiến hành khảo sát và tổng hợp tài liệu khoa học từ các ấn bản quốc tế chất lượng cao dưới dạng từ hội nghị và tạp chí nghiên cứu liên quan đến ước lượng dáng người (HPE), các thuật toán học máy truyền thống (Machine Learning) cho bài toán phân loại, cũng như các nghiên cứu về nhận diện tư thế yoga.

Nghiên cứu thực nghiệm: Đề án triển khai một chuỗi các thí nghiệm chính là xây dựng và tiền xử lý bộ dữ liệu, thực nghiệm các mô hình ước lượng tư thế người (HPE), thực nghiệm kết hợp với các thuật toán học máy (ML), và đánh giá tổng thể hệ thống thời gian thực.

CHƯƠNG 1: TỔNG QUAN VỀ NHẬN DIỆN TƯ THẾ YOGA

1.1. Khái quát về nhận diện tư thế người

Nhận diện tư thế người (Human Pose Recognition – HPR) thuộc nhóm chủ đề quan trọng trong thị giác máy tính (CV) và tương tác người–máy (Human–Computer Interaction – HCI). Mục tiêu của bài toán này là phân tích từ hình ảnh hoặc video để xác định tư thế từ đó có thể biết được chuyển động và hành động mà con người đang thực hiện. Khác với bài toán ước lượng tư thế người (Human Pose Estimation – HPE), vốn chỉ xoay quanh việc phát hiện cùng định vị các điểm khớp (keypoints) trên cơ thể, HPR là bài toán rộng hơn hướng đến việc phân loại và nhận dạng tư thế/hành động tổng thể của cơ thể người.

Nhờ sự phát triển của học máy và học sâu cùng với thuật toán xử lý hình ảnh hiện đại, HPR đã đạt được nhiều tiến bộ đáng kể trong trong vài năm trở lại đây. Các hệ thống nhận diện tư thế ngày nay có thể nhận biết chính xác các tư thế đơn giản như đứng, ngồi, giơ tay, cúi người, cho đến các chuỗi động tác phức tạp hơn như nhảy, chạy, hoặc thể thao, giám sát nói chung. Nhờ đó, công nghệ HPR trở thành nền tảng cho nhiều ứng dụng thực tiễn. Trong nhận dạng hoạt động HAR [15], hệ thống có thể xác định các hành động cụ thể của con người, mang lại giá trị cao cho các lĩnh vực như giám sát an ninh [16], phân tích thể thao hoặc hỗ trợ chăm sóc sức khỏe. Đồng thời trong tương tác người–máy, công nghệ này giúp nâng cao tính tự nhiên của giao tiếp giữa người và thiết bị, ứng dụng trong môi trường ảo, trò chơi hoặc hệ thống điều khiển từ xa [17]. Ngoài ra, trong phân tích y học, việc nhận diện và theo dõi tư thế, dáng đi hoặc chuyển động của bệnh nhân giúp hỗ trợ chẩn đoán và phục hồi chức năng,... Dẫu vậy, lĩnh vực này cũng có nhiều trở ngại, chẳng hạn như hiện tượng che khuất khiến các bộ phận cơ thể bị ẩn khỏi tầm nhìn, sự thay đổi về trang phục, ánh sáng, góc nhìn hoặc phông nền làm giảm độ chính xác, và yêu cầu xử lý thời gian thực trong các ứng dụng tương tác đòi hỏi hiệu suất tính toán cao [18].

Nhìn chung, nhờ sự tiến bộ nhanh chóng của học sâu và thị giác máy tính, công nghệ nhận diện tư thế người đang ngày càng hoàn thiện, mở ra nhiều triển vọng trong đời sống, y tế, giáo dục, thể thao và các hệ thống thông minh tương lai. Tuy

nhiên, lĩnh vực này vẫn cần được đào sâu thêm về mặt nghiên cứu, nhằm tăng cường độ chuẩn xác, độ ổn định và khả năng ứng biến trong các môi trường thực tế - nơi mà tiềm năng lợi ích thực tế và việc áp dụng của công nghệ này là vô cùng lớn.

1.2. Bài toán nhận diện tư thế yoga

Yoga, một phương pháp rèn luyện cổ xưa kết nối cân bằng giữa thể chất và tinh thần, việc cải thiện sự dẻo dai, sức mạnh cốt lõi, cân bằng cho đến việc giảm thiểu căng thẳng và nâng cao sự minh mẫn [19]. Trong bối cảnh cuộc sống hiện đại, đặc biệt sau các sự kiện xã hội có biến động lớn từ COVID-19 và sự trỗi dậy của AI, xu hướng tập luyện tại nhà đã trở nên phổ biến mạnh mẽ. Hình thức này đặc biệt thu hút học sinh, sinh viên lẫn người làm dân văn phòng – những người hay gặp phải tình trạng một chỗ quá lâu trong ngày và tìm kiếm các cách để rèn luyện linh hoạt, tiết kiệm được thời gian.

Sự bùng nổ của nhu cầu ấy đã thúc đẩy sự ra mắt của rất nhiều ứng dụng và nền tảng tập luyện trực tuyến. Tuy nhiên, việc tự tập luyện mà thiếu đi sự giám sát, hỗ trợ của huấn luyện viên tiềm ẩn nhiều rủi ro, đặc biệt khi người tập thực hiện sai tư thế. Các lỗi nhỏ trong động tác có thể làm giảm hiệu quả bài tập hoặc thậm chí dẫn đến chấn thương như căng cơ hay tổn thương khớp, khiến người tập mất đi lợi ích vốn có của yoga. Trước thực tế này, công nghệ thị giác máy tính đã mở ra hướng đi hướng đi sáng giá — phát triển các hệ thống nhận diện và đánh giá tư thế yoga tự động [20] [21]. Chúng có khả năng phân tích dáng cơ thể người từ video hoặc hình ảnh, sau đó cung cấp phản hồi tức thời giúp người dùng điều chỉnh động tác chính xác hơn. Dù vậy, việc xây dựng một hệ thống như vậy sẽ phải xử lý không ít trở ngại về kỹ thuật phức tạp, từ sự đa dạng về hình thể và tỷ lệ cơ thể của người tập, cho đến các yếu tố môi trường như góc quay camera, điều kiện ánh sáng. Yêu cầu cốt lõi và khó khăn nhất chính là phải đạt được sự cân bằng tối ưu giữa độ chính xác (accuracy) trong nhận dạng và tốc độ xử lý (speed/latency) để đảm bảo trải nghiệm người dùng liền mạch.

Bài toán nhận diện tư thế yoga là một trường hợp đặc thù của nhận diện tư thế người HPR, trong đó mục tiêu là xác định chính xác tư thế mà người tập đang thực

hiện dựa trên dữ liệu hình ảnh. So với các bài toán nhận dạng hành động thông thường, nhận diện tư thế yoga đòi hỏi mức độ chính xác cao hơn, vì nhiều tư thế có hình dáng tương tự nhau nhưng khác biệt tinh tế ở các góc khớp, hướng tay chân hoặc độ nghiêng và cân bằng của cơ thể. Bài toán này có tiềm năng lớn trong thực tế, nhất trong các lĩnh vực như huấn luyện thể chất, đánh giá kỹ thuật rèn luyện thể chất, theo dõi sức khỏe và phục hồi chức năng. Nhờ sự phát triển của các mô hình mới nhẹ và công cụ ước lượng tư thế theo thời gian thực, việc triển khai hệ thống nhận diện tư thế yoga trên thiết bị di động hoặc thiết bị đeo thông minh trở nên khả thi hơn, góp phần hỗ trợ người dùng luyện tập đúng tư thế mà không cần sự hướng dẫn trực tiếp từ huấn luyện viên.

1.3. Các phương pháp và mô hình nhận diện tư thế người

Nhận diện tư thế người HPR được phát triển dựa trên nhiều phương pháp và mô hình khác nhau, tùy theo loại dữ liệu đầu vào và mức độ phức tạp của bài toán. Các hướng tiếp cận và mô hình trong HPR được phát triển đa dạng, nhưng nhìn chung có thể phân loại thành ba nhóm chính:

(1) Phương pháp dựa trên ảnh RGB: Phương pháp dựa trên hình ảnh RGB là hướng tiếp cận phổ biến và cơ bản nhất trong nhận diện tư thế người. Ở cách tiếp cận này, hệ thống sử dụng ảnh hoặc chuỗi khung hình RGB làm đầu vào, sau đó học đặc trưng trực tiếp từ dữ liệu thị giác để phân loại tư thế hoặc hành động tương ứng. Về bản chất, bài toán nhận diện tư thế trong trường hợp này có thể được coi như một bài toán phân loại ảnh thông thường, trong đó mỗi tư thế của con người được xem như một lớp (class) riêng biệt. Các phương pháp cổ điển dựa trên đặc trưng thủ công (như HOG, SIFT) kết hợp với các bộ phân loại (SVM, k-NN, Random Forest) từng được sử dụng nhiều. Song, chúng vẫn bị ràng buộc mạnh bởi điều kiện chiếu sáng, góc nhìn và nền ảnh, nên hiệu quả bị suy giảm trong môi trường thực tế.Sự xuất hiện của học sâu (Deep Learning) đã thay đổi hoàn toàn hướng tiếp cận này. Các mô hình CNN và biến thể như 3D-CNN, Vision Transformer (ViT) [22] ,có thể tự động học đặc trưng từ dữ liệu ảnh thô, không cần bước trích xuất thủ công nên nhận dạng tư thế và hành động với độ chính xác cao hơn.

(2) Phương pháp dựa trên dữ liệu khung xương (Skeleton-based): Các phương pháp này biểu diễn con người thông qua tập hợp các điểm khớp (keypoints) trích xuất từ hình ảnh hoặc video, giúp mô hình tập trung vào cấu trúc hình học và quan hệ vị trí–thời gian của các bộ phận cơ thể. Quá trình nhận diện tư thế người thường gồm ba giai đoạn chính:

(i) Phát hiện điểm khớp (Keypoint Detection) nhằm xác định vị trí các khớp chủ chốt như vai, đầu gối, khuỷu, hoặc cổ tay;

(ii) Trích xuất đặc trưng (Feature Extraction) để học các đặc trưng không gian từ tập điểm khớp, phản ánh cấu trúc và chuyển động cơ thể; và

(iii) Phân loại tư thế (Pose Classification) sử dụng các thuật toán dựa trên hình học, mô hình học máy hoặc học sâu để xác định tư thế tổng thể của con người.

Ba bước này tạo thành chuỗi xử lý chính, giúp hệ thống chuyển đổi dữ liệu hình ảnh thô thành thông tin có ý nghĩa về hành vi và trạng thái của con người.

(3) Mô hình lai: Mô hình lai (hybrid models) kết hợp đồng thời nhiều nguồn dữ liệu — ví dụ ảnh RGB, dữ liệu độ sâu (depth), biểu diễn khung xương (skeleton) và dữ liệu cảm biến (IMU, thiết bị đeo) — nhằm tận dụng ưu điểm của từng kiểu biểu diễn. Việc tích hợp đa phương thức giúp mô hình vừa nắm bắt được thông tin ngữ cảnh và ngoại cảnh từ ảnh RGB, vừa bảo toàn cấu trúc hình học và động học của cơ thể từ skeleton hoặc dữ liệu cảm biến, từ đó ntăng cường độ chính xác và khả năng khái quát trong các tình huống thực tế phức tạp. Cuối cùng, mặc dù mô hình lai đem lại nhiều lợi ích về độ chính xác và tính bền vững khi ứng dụng trong thực tế, chúng cũng đặt ra thách thức về đồng bộ dữ liệu, mức độ phức tạp khi tích hợp và chi phí tính toán.

1.4. Nghiên cứu liên quan đến nhận diện tư thế yoga

Một số các nghiên cứu cũ trước đây đã thiết kế hệ thống hỗ trợ thể dục hoặc yoga dựa trên phân tích ảnh hoặc video mà không sử dụng dữ liệu skeleton. Công trình [23] đã khai thác một mô hình nhận diện tư thế Yoga (asana) dựa trên tập dữ liệu gồm 6 tư thế khác nhau được thu thập bởi 15 tình nguyện viên. Mô hình đề xuất

ứng dụng kiến trúc kết hợp giữa mạng CNN và LSTM, được gọi là Long-Term Recurrent Convolutional Network (LRCN). Trong đó, CNN đảm nhận việc trích xuất đặc trưng không gian từ từng khung hình, còn LSTM chịu trách nhiệm phân tích mối liên hệ theo chuỗi thời gian giữa các khung hình liên tiếp. Kết quả thực nghiệm chỉ ra rằng mô hình đạt chính xác ở mức 81% các tư thế Yoga. Hay trong nghiên cứu [24] , YoNet – một mô hình học sâu mới – được các tác giả giới thiệu nhằm nhận diện năm tư thế yoga khác nhau ngay cả khi dữ liệu huấn luyện tương đối hạn chế. Mô hình này được đánh giá và so sánh hiệu năng với các kiến trúc phân loại ảnh tiên tiến như ResNet, InceptionNet, InceptionResNet và Xception. Các kết quả thu được cho thấy rằng YoNet đạt hiệu suất vượt trội hơn so với các mô hình trên đạt độ chính xác hơn 94% . Nhóm tác giả trong nghiên cứu [25] đã đề xuất một mô hình CNN kết hợp với phép biến đổi Wavelet cho bài toán phân loại tư thế yoga. Ảnh đầu vào được phân rã thành bốn dải con wavelet để sau đó mỗi dải con được đưa vào một mạng CNN riêng biệt, và các kết quả đầu ra được gộp lại (fusion) để xác định tư thế cuối cùng. Nghiên cứu tận dụng bộ dữ liệu yoga gồm năm tư thế có sẵn công khai và thực nghiệm đã cho thấy mô hình được phát triển tốt hơn các mô hình CNN tiêu chuẩn, đạt độ chính xác ở mức 90% trên tập huấn luyện và đạt 80% trên tập kiểm thử. Tổng quan lại có thể thấy các phương pháp này thường khai thác CNN và biến thể của nó hoặc kết hợp với một số cách xử lý ảnh đi kèm để trích xuất đặc trưng thị giác trên ảnh trở nên tốt hơn. Dẫu vậy, việc học các đặc trưng trực tiếp từ ảnh khiến mô hình có thể nhạy cảm với các yếu tố như nhiễu nền, điều kiện ánh sáng thay đổi, trang phục của người tập và góc nhìn của camera, nên độ chính xác đạt được nhìn chung chưa thật sự cao. Bên cạnh đó, việc sử dụng thuần túy CNN thường chưa đủ để phân tích chi tiết tư thế, dẫn đến khả năng đánh giá chất lượng động tác còn hạn chế. Quan trọng hơn là thiếu khả năng phân tích chi tiết tư thế cụ thể của người dùng, khiến cho việc đánh giá chất lượng và mức độ chính xác của từng động tác trở nên khó khăn và khó giải thích.

Để khắc phục những điểm chưa tối ưu của các phương pháp học trực tiếp từ ảnh, nhiều đề tài đã chuyển hướng tập trung vào sử dụng mô hình HPE để trích xuất

biểu diễn trung gian dưới dạng bộ khung xương, sau đó ứng dụng toán học hoặc các mô hình học máy truyền thống để phân loại tư thế dựa trên cấu trúc hình học này. Cụ thể, nghiên cứu [26] đã thực hiện phân loại chuỗi tư thế Chào Mặt Trời bằng bốn mô hình học máy khác nhau, trong đó đặc trưng đầu vào được trích xuất từ các góc khớp thông qua thuật toán ước lượng tư thế thời gian thực. Kết quả cho thấy là mô hình KNN có sự chính xác cao nhất, lên tới 96%. Trong khi đó, nghiên cứu [27] lại giới thiệu một phát hiện và đánh giá tư thế dựa trên PoseNet của Google, có khả năng tạo ảnh khung xương người tập và so sánh với tập huấn luyện nhằm đưa ra dự đoán, qua đó hỗ trợ người dùng duy trì tư thế chính xác và nâng cao hiệu quả luyện tập mà không cần huấn luyện viên trực tiếp. Nghiên cứu khác [21] tập trung phát triển một hệ thống phát hiện và hiệu chỉnh tư thế yoga theo thời gian thực sử dụng MediaPipe để trích xuất các điểm khớp cơ thể (pose landmarks), sau đó đưa dữ liệu vào một bộ phân loại được huấn luyện theo hướng học có giám sát nhằm nhận dạng tư thế. Nguồn dữ liệu liệu huấn luyện được lấy từ Mendeley Yoga Dataset, tập trung vào bốn tư thế cơ bản gồm Anantasana, Bhujangasana, Tadasana và Vajrasana. Kết quả ghi nhận độ chính xác vượt trội của mô hình ở tập huấn luyện và kiểm định đều trên 90% với số lượng tham số nhỏ, thể hiện tính hiệu quả, gọn nhẹ và khả năng ứng dụng cao trong việc nhận diện và hiệu chỉnh tư thế yoga theo thời gian thực. Nghiên cứu [28] tập trung vào phân loại tư thế yoga bằng cách kết hợp YOLOv8 và MediaPipe để phát hiện các điểm khớp (keypoints) trên cơ thể người tập. Quá trình phân loại được thực hiện với nhiều mô hình học máy cơ sở và mô hình học tổ hợp (ensemble learning) trên một bộ dữ liệu tùy chỉnh gồm nhiều tư thế yoga khác nhau. Kết quả cho thấy MediaPipe cho độ chính xác cao hơn YOLOv8 (đặc biệt là LightGBM sử dụng keypoints từ MediaPipe đạt độ chính xác cao nhất), khẳng định sự cần thiết của việc phát hiện sớm keypoints chính xác trong nâng cao hiệu suất phân loại. Nghiên cứu [29] xây dựng một bộ dữ liệu lớn gồm hơn 5.500 hình ảnh của 10 tư thế yoga, sử dụng thuật toán tf-pose estimation để trích xuất khung xương và góc khớp cơ thể theo thời gian thực. Những đặc trưng này đóng vai trò làm đầu vào cho nhiều mô hình ML nhằm phân loại tư thế. Qua thực nghiệm, mô hình Random Forest đạt độ chính xác

99,04% cũng đã cho hé lộ triển vọng của phương pháp trong nhận dạng và phát hiện tư thế yoga. Nghiên cứu [30] có phạm vi tổng quát hơn khi kết hợp nhiều thuật toán ước lượng tư thế như MediaPipe, PoseNet và OpenPose để trích xuất các điểm khớp cơ thể (keypoints) từ ảnh hoặc video, qua đó xây dựng mô hình khung xương 2 chiều 2D và 3 chiều 3D phục vụ nhận dạng tư thế yoga. Nhóm tác giả xem xét năm tư thế đặc trưng gồm Downdog, Goddess, Plank, Tree và Warrior II, đồng thời áp dụng nhiều bộ phân loại ML như SVM, Random Forest, KNN,… để so sánh hiệu quả nhận dạng. Bộ dữ liệu được chia theo nhiều tỷ lệ khác nhau trong huấn luyện–kiểm thử nhằm đánh giá độ tin cậy của mô hình. Kết quả cho thấy SVM và KNN đạt độ chính xác cao ổn định, trong khi Random Forest đạt đến 100% ở một số tư thế, chứng minh khả năng kết hợp linh hoạt giữa ước lượng tư thế và học máy trong bài toán phát hiện và phân loại tư thế yoga. Mặc dù vậy, một hạn chế chung trong các nghiên cứu cũ trước là phần lớn chỉ làm vào một hoặc một vài tổ hợp cụ thể giữa các mô hình ước lượng dáng người (HPE) và thuật toán học máy (ML). Do đó, vẫn còn sự thiếu hụt đáng kể các nghiên cứu so sánh chéo có phạm vi đầy đủ và được triển khai theo phương pháp tiếp cận có hệ thống, khiến việc lựa chọn tổ hợp tối ưu cho từng mục tiêu ứng dụng cụ thể trở nên khó khăn. Đề án này hướng đến việc khắc phục hạn chế đó thông qua phân tích định lượng và so sánh có hệ thống hiệu suất của nhiều tổ hợp khác nhau giữa các phương pháp trích xuất khung xương và thuật toán học máy — từ cách tiếp cận cổ điển đến các mô hình tiên tiến. Bên cạnh đó, đề án cũng đánh giá sự cân đối giữa chất lượng kết quả và tốc độ xử lý, nhằm đề xuất giải pháp phù hợp cho các giải pháp yoga thông minh hoạt động trong môi trường thực tế.

1.5. Kết luận chương 1

Chương 1 đã trình bày tổng quan về lĩnh vực nhận diện tư thế yoga, làm rõ các khái niệm căn bản về nhận diện tư thế người đến những đặc thù riêng của bài toán trong yoga. Các phương pháp và mô hình, kỹ thuật phổ biến từ trước đến những năm gần đây đã được giới thiệu trong việc trích xuất và phân loại tư thế, cả đối với bài toán phân loại dáng người nói chung lẫn bài toán nhận diện và phân loại dáng yoga cụ thể. Qua thực tiễn, nhiều các công trình hiện nay chỉ tập trung vào một số tổ hợp

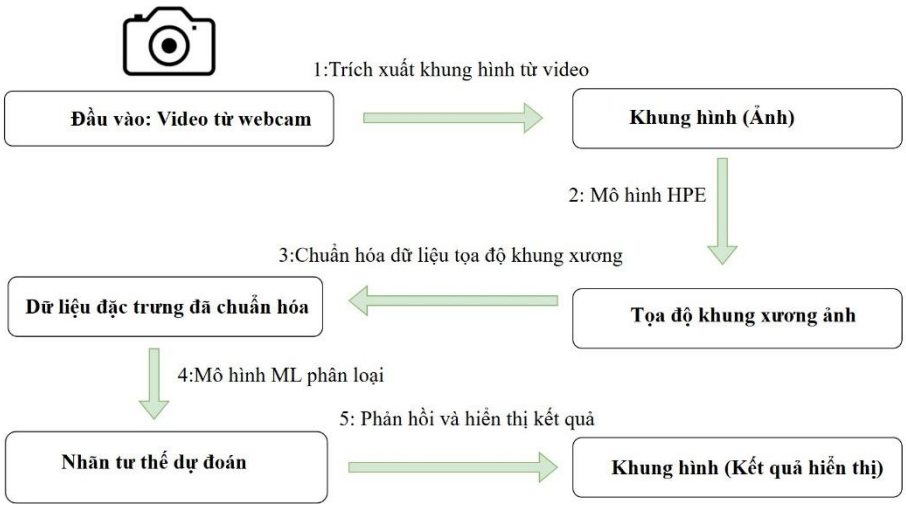
nhất định giữa mô hình trích xuất đặc trưng và thuật toán học máy, chưa có một nghiên cứu tổng quát, so sánh toàn diện để xác định tổ hợp tối ưu cho bài toán nhận diện tư thế yoga.

Trên cơ sở đó, chương 2 sẽ tập trung phân tích chi tiết các kỹ thuật gắn với bài toán này. Đồng thời, chương sẽ đề xuất một pipeline nghiên cứu cụ thể, giới thiệu chi tiết hơn về các mô hình HPE và các thuật toán ML được sử dụng, phục vụ cho việc benchmark và xác định tổ hợp phương pháp tối ưu cho các bước nghiên cứu kế tiếp.

CHƯƠNG 2: NGHIÊN CỨU ĐỀ XUẤT HỆ THỐNG NHẬN DIỆN TƯ THẾ YOGA THỜI GIAN THỰC

2.1. Mô hình đề xuất cho bài toán

Kiến trúc đề xuất cho bài toán nhận diện tư thế Yoga theo thời gian thực được xây dựng dựa trên chuỗi các bước xử lý từ dữ liệu đầu vào là video đến đầu ra là nhãn tư thế dự đoán. Pipeline của hệ thống được khái quát trong Hình 2.1, mô tả toàn bộ quy trình xử lý dữ liệu từ thu nhận hình ảnh đến phân loại tư thế. Hệ thống gồm bốn mô-đun chính, mỗi mô-đun thực thi một chức năng riêng biệt nhưng liên kết chặt với nhau để quá trình nhận dạng diễn ra chính xác và thời gian thực.



Hình 2.1: Pipeline của giải pháp đề xuất cho bài toán nhận diện tư thế Yoga.

Đầu tiên, dữ liệu đầu vào là chuỗi khung hình thu được từ webcam (Bước 1). Các khung hình này được đưa vào mô đun trích xuất khung xương, sử dụng mô hình ước lượng tư thế người (HPE) để phát hiện các điểm khớp (keypoints) trên cơ thể người. Tiếp theo, dữ liệu khung xương được chuẩn hóa và xử lý trong mô đun xử lý và trích xuất đặc trưng nhằm cân bằng tọa độ và trích xuất các đặc trưng hình học quan trọng (như góc, khoảng cách, tỷ lệ) phản ánh rõ ràng tư thế của người tập. Các đặc trưng đầu ra sau đó được đưa vào mô đun phân loại tư thế, nơi một số mô hình học máy (ML) được huấn luyện để dự đoán nhãn tư thế Yoga tương ứng. Cuối cùng, kết quả nhận dạng được chuyển đến mô đun phản hồi và hiển thị, nơi hệ thống trực quan hóa tư thế trên giao diện người dùng, hiển thị nhãn tư thế nhận dạng được và

cung cấp phản hồi cho người tập. Nhờ sự kết hợp của các mô-đun trên, hệ thống có thể nhận diện và phân loại tư thế Yoga một cách ổn định trong thời gian thực.

2.2. Mô đun trích xuất khung xương

2.2.1. Các mô hình được áp dụng

Nhìn chung, các hướng tiếp cận trong bài toán Human Pose Estimation [31] có thể được chia nhóm dựa trên bốn tiêu chí chính.

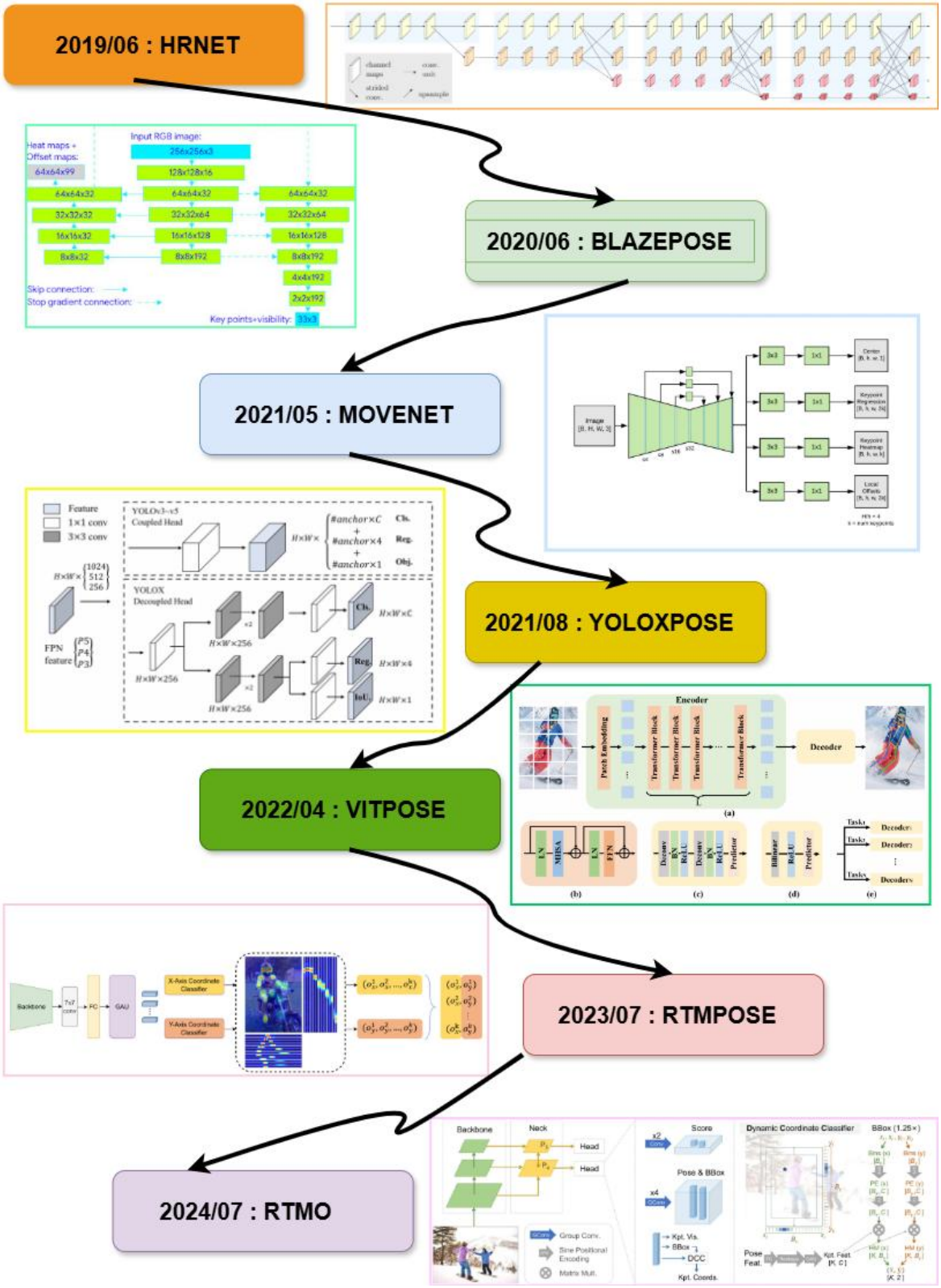
(1) Chiều dữ liệu: Phân biệt giữa ước lượng tư thế trong không gian hai chiều và ba chiều. Trong khi 2D pose estimation nghiên cứu trong việc xác định vị trí các khớp trên mặt phẳng ảnh, 3D pose estimation hướng tới việc tái tạo cấu trúc không gian của cơ thể, mang lại mô hình hình học chính xác hơn nhưng cần dữ liệu huấn luyện quy mô lớn và hạ tầng tính toán mạnh mẽ.

(2) Đối tượng xử lý: Gồm hai hướng chính là đơn nhân (single-person) và đa nhân (multi-person). Bài toán đơn nhân giả định chỉ có một người trong khung hình, trong khi bài toán đa nhân là phát hiện tọa độ điểm khớp cho từng người nên dẫn đến độ phức tạp cao hơn đáng kể.

(3) Cấu trúc mô hình: Gồm ba hướng tiếp cận phổ biến là top-down, bottom- up và one-stage. Cách top-down thường phát hiện người trước rồi mới ước lượng tư thế từng cá thể; ngược lại, bottom-up xác định toàn bộ các điểm khớp trong ảnh trước khi nhóm chúng lại thành từng cá thể; còn one-stage cố gắng kết hợp cả hai bước trong một mô hình duy nhất, hướng đến tối ưu hiệu năng thời gian thực.

(4) Dữ liệu đầu vào: Chấp nhận ảnh tĩnh đơn lẻ (image-based) hoặc chuỗi video (video-based). Trong khi ảnh tĩnh chỉ phản ánh thông tin không gian tại một thời điểm, dữ liệu video cho phép khai thác thêm thông tin về chuyển động và sự liên tục theo thời gian, giúp cải thiện độ ổn định của dự đoán.

Hệ thống nghiên cứu này được xây dựng ưu tiên hướng tới ứng dụng các mô hình ước lượng tư thế 2D để phân tích tư thế của một người/ nhiều người qua hình ảnh, nhằm đáp ứng yêu cầu nhận dạng tư thế Yoga theo thời gian thực trong ứng dụng cá nhân, đặc biệt trong các môi trường tập luyện tại nhà. Các mô hình HPE được sử dụng trong đề án này được thể hiện qua Hình 2.2.



Hình 2.2: Các mô hình HPE được sử dụng trong đề án

HRNet [32] là một kiến trúc mạng được đề xuất cho bài toán ước lượng tư thế người, với mục tiêu học được các biểu diễn có độ phân giải cao và đáng tin cậy. Thay vì sử dụng kiến trúc thông thường theo hướng từ cao xuống thấp rồi khôi phục lại độ phân giải như các mô hình trước đây, HRNet duy trì trực tiếp biểu diễn độ phân giải cao trong suốt quá trình xử lý. Cấu trúc của HRNet bắt đầu với một nhánh có độ phân giải cao, sau đó dần bổ sung các nhánh có độ phân giải thấp hơn và kết nối song song các nhánh này. Giữa các nhánh diễn ra quá trình hợp nhất đa tỉ lệ lặp đi lặp lại, cho phép các biểu diễn ở nhiều mức độ phân giải khác nhau trao đổi và bổ sung thông tin cho nhau. Nhờ đó, HRNet có khả năng biểu diễn đặc trưng không gian phong phú và chính xác hơn, giúp dự đoán bản đồ nhiệt của các điểm khớp (keypoints) đạt độ chính xác cao. Mạng đã chứng minh hiệu quả vượt trội trên các bộ dữ liệu chuẩn như COCO, MPII, và PoseTrack, khẳng định ưu thế trong việc duy trì độ phân giải cao và khai thác thông tin đa tỉ lệ một cách hiệu quả.

BlazePose [7] là một mô hình được Google Research phát triển, hướng tới mục tiêu nhẹ, nhanh và chính xác, đặc biệt tối ưu cho thiết bị di động. Mô hình được xây dựng dựa trên kiến trúc mạng CNN gọn nhẹ, cho phép dự đoán 33 điểm khớp (keypoints) trên cơ thể người, bao gồm cả các khớp chính như đầu, vai, tay, chân và cột sống. Điểm nổi bật của BlazePose nằm ở thiết kế kết hợp giữa hai thành phần chính: Bộ phát hiện (detector) dùng để xác định vùng chứa cơ thể người trong khung hình, và Bộ theo dõi tư thế (pose tracker) dùng để dự đoán chính xác tọa độ các điểm khớp, đồng thời cập nhật vùng có sự hiện diện của người và tinh chỉnh vùng quan tâm ở khung hình hiện tại. Kiến trúc này không chỉ giúp giảm tải tính toán mà còn đảm bảo hiệu năng thời gian thực, đạt hơn 30 khung hình/giây trên thiết bị Pixel 2, phù hợp cho các ứng dụng sử dụng được ngay trên thiết bị (on-device) mà không cần đến máy chủ. BlazePose được sử dụng phổ quát trong các lĩnh vực như theo dõi bài tập thể hình (fitness tracking), phân tích động tác yoga, nhận dạng ngôn ngữ ký hiệu và thực tế tăng cường (AR). Bên cạnh đó, với cấu trúc 33 điểm khớp nhất quán với các mô hình BlazeFace và BlazePalm, BlazePose còn có thể mở rộng làm nền tảng

cho các mô hình nhận dạng khuôn mặt và bàn tay, hướng tới hệ thống nhận diện chuyển động toàn thân 3D.

MoveNet [6] là mô hình phát hiện tư thế người (pose detection) mới được Google giới thiệu cùng API phát hiện tư thế trong TensorFlow.js, nổi bật với tốc độ siêu nhanh và độ chính xác cao khi nhận diện 17 điểm khớp (keypoints) trên cơ thể người. Mô hình có hai phiên bản: Lightning – tối ưu hóa cho ứng dụng yêu cầu độ trễ cực thấp, và Thunder – hướng đến độ chính xác cao hơn, đều đạt tốc độ hơn 30 FPS trên hầu hết các thiết bị hiện đại, rất hợp cho các ứng dụng thể thao, thể hình và sức khỏe thời gian thực. Về kiến trúc, MoveNet sử dụng phương pháp ước lượng từ dưới lên (bottom-up estimation) với MobileNetV2 kết hợp Feature Pyramid Network (FPN) làm bộ trích xuất đặc trưng, cùng bốn đầu dự đoán (prediction heads) để xác định tâm người, bản đồ điểm khớp và tinh chỉnh vị trí sub-pixel. Thiết kế này giúp mô hình duy trì độ phân giải cao, phát hiện chuẩn xác ngay cả khi di chuyển nhanh hoặc tư thế phức tạp. Trong ứng dụng thực tế, MoveNet đã được IncludeHealth – công ty công nghệ sức khỏe – triển khai trong hệ thống tập luyện và điều trị vật lý trị liệu từ xa, chứng minh khả năng cân bằng giữa tốc độ và độ chính xác, mở ra tiềm năng lớn cho các ứng dụng chăm sóc sức khỏe và theo dõi chuyển động người dùng trong thời gian thực.

YOLOX [5] là thế hệ mô hình phát hiện đối tượng hiệu năng cao được xây dựng dựa trên chuỗi mô hình YOLO, trong đó nhóm tác giả đã thực hiện nhiều cải tiến thực tiễn quan trọng. Mô hình này chuyển đổi cấu trúc truyền thống của YOLO sang cơ chế anchor-free (không sử dụng neo) và tích hợp các kỹ thuật phát hiện hiện đại như đầu tách biệt (decoupled head) và chiến lược gán nhãn tiên tiến SimOTA, giúp nâng cao đáng kể độ chính xác và tốc độ xử lý. Nhờ những cải tiến này, YOLOX đạt được kết quả vượt trội (state- of-the-art) trên nhiều quy mô mô hình khác nhau đặc biệt YOLOX-L còn đạt giải Nhất tại cuộc thi Streaming Perception Challenge, chứng minh hiệu quả thực tế của mô hình. YOLOX đồng thời cung cấp các phiên bản triển khai tương thích với ONNX, TensorRT, NCNN và OpenVINO, phục vụ hiệu quả cho cả mục đích nghiên cứu và ứng dụng thực tế.

ViTPose [33] là một kiến trúc ước tính tư thế người dựa hoàn toàn trên bộ biến đổi thị giác (ViT), thể hiện cách tiếp cận đơn giản nhưng hiệu quả vượt trội cho các nhiệm vụ thị giác. Khác với nhiều phương pháp trước đây thường tích hợp các thành phần chuyên biệt cho nhận dạng hình học hay đặc trưng không gian, ViTPose không dựa trên kiến thức miền cụ thể trong thiết kế mà sử dụng kiến trúc transformer thuần túy, không phân cấp làm xương sống để trích xuất đặc trưng hình ảnh cho từng đối tượng người. Phía trên đó, một bộ giải mã nhẹ được sử dụng để dự đoán các điểm khớp (keypoints) tư thế. Kiến trúc này có tính mở rộng rất cao, có độ linh hoạt cao trong việc lựa chọn cơ chế chú ý, độ phân giải đầu vào, chiến lược tiền huấn luyện và tinh chỉnh, giúp thích ứng với nhiều loại nhiệm vụ ước tính tư thế khác nhau. Các kết quả thu được trên MS COCO Keypoint Detection cho thấy ViTPose cơ bản đã vượt qua nhiều phương pháp tiên tiến hiện hành, trong khi phiên bản lớn nhất của mô hình — ViTPose với backbone ViTAE-G — đạt 80.9 AP trên tập COCO test-dev, thiết lập một chuẩn hiệu suất mới mở ra hướng đi mới trong khai thác tiềm năng của các bộ biến đổi thị giác thuần túy cho nhiều nhiệm vụ thị giác máy tính khác.

RTMPose [34] áp dụng phương pháp top-down, trong đó sử dụng bộ phát hiện để xác định hộp giới hạn trước khi ước tính tư thế từng người. Dù từng bị xem là chậm do phụ thuộc vào bước phát hiện, phương pháp này nay trở nên mạnh mẽ hơn nhờ những đột phá trong những bộ phát hiện thời gian thực. Mô hình áp dụng CSPNeXt làm xương sống (backbone), đạt được điểm cân bằng tốt giữa độ chuẩn xác và mức hiệu quả triển khai. Phần đầu ra được thiết kế dựa trên thuật toán SimCC, xử lý việc dự đoán điểm mốc dưới dạng bài toán phân loại, giúp giảm chi phí tính toán hơn so với phương pháp dựa trên heatmap. Ngoài ra, framework còn tích hợp các chiến lược huấn luyện tối ưu và các kỹ thuật suy luận như phát hiện bỏ qua khung hình, kết hợp loại bỏ tư thế trùng lặp (Pose NMS) và lọc làm mịn nhằm tăng độ ổn định và tính thời gian thực.

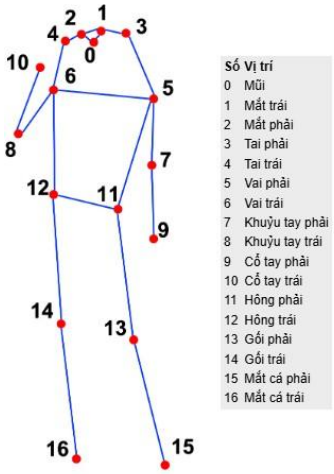
RTMO [11] là một framework ước tính tư thế đa người một giai đoạn thời gian thực, được thiết kế nhằm cân bằng hiệu quả giữa tốc độ và độ chính xác. Không giống với kiểu top-down hai giai đoạn thường chậm khi số lượng người tăng, RTMO tích

hợp phân loại tọa độ thông qua bản đồ nhiệt 1-D kép (dual 1-D heatmaps) trực tiếp trong kiến trúc YOLO, giúp duy trì tốc độ cao mà vẫn đạt độ chính xác cạnh tranh. Framework sử dụng bộ phân loại tọa độ động (dynamic coordinate classifier) kết hợp với hàm mất mát chuyên biệt để giải quyết việc không tương thích giữa phân loại tọa độ và các mô hình dự đoán dày đặc (dense prediction models). RTMO cũng khai thác thùng tọa độ (coordinate bins) để định vị điểm mốc chính xác cho các trường hợp có kích thước và độ phức tạp khác nhau, đồng thời mang lại một phương pháp ước tính tư thế đa người (MPPE) tích hợp liền mạch, tối ưu hóa cả tốc độ và hiệu suất.

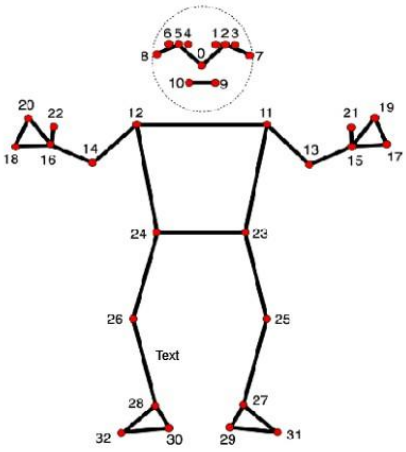
2.2.2. Nhận diện và xử lý keypoints

Trong các mô hình ước lượng tư thế, số lượng và loại keypoints mà mô hình trả về phụ thuộc vào dataset mà mô hình được huấn luyện. Các mô hình có thể trả về số lượng keypoint khác nhau, tuy nhiên, để thuận tiện cho việc xử lý và so sánh, kết quả thường được quy về chuẩn keypoint COCO, vốn được sử dụng rộng rãi trong nhiều bài toán nhận diện tư thế. Việc này giúp tích hợp dữ liệu từ nhiều mô hình khác nhau, thống nhất pipeline xử lý và dễ dàng triển khai các thuật toán phân tích chuyển động hoặc nhận diện hành vi.

Keypoints chuẩn COCO bao gồm 17 điểm chính của cơ thể người, đại diện cho các vị trí như mũi, mắt, tai, vai, khuỷu tay, cổ tay, hông, gối và mắt cá chân. Khi sử dụng các mô hình như MoveNet, dữ liệu trả về thường ở dạng mảng [số người, 17, 3], trong đó 17 là số keypoint theo chuẩn COCO, mỗi điểm gồm tọa độ (x, y) và giá trị độ tin cậy (confidence). Skeleton COCO cũng định nghĩa các kết nối giữa các keypoints để tạo thành khung xương, phục vụ cho việc hiển thị hoặc phân tích chuyển động. Ngược lại, các mô hình như BlazePose của MediaPipe cung cấp 33 keypoints chi tiết hơn, bao gồm cả các chi tiết nhỏ trên tay, chân, mặt và các ngón tay. Dữ liệu trả về cho mỗi điểm gồm tọa độ (x, y, z) và độ tin cậy. Việc này giúp khai thác chi tiết hơn trong các ứng dụng cần độ chính xác cao hoặc phân tích chuyển động chi tiết. “Danh sách các điểm khớp chuẩn theo COCO (17 điểm) và các điểm khớp của BlazePose (33 điểm) bằng tiếng Việt được mô tả trong Hình 2.3.



Mô tả 17 keypoints – tập dữ liệu COCO





Mô tả 33 keypoints – mô hình BlazePose

Hình 2.3: Danh sách các điểm khớp của COCO dataset (17 điểm) và BlazePose (33 điểm) bằng tiếng Việt

Việc quy chuẩn keypoints về chuẩn COCO giúp dễ dàng so sánh và kết hợp dữ liệu từ nhiều mô hình, trong khi các bộ keypoints mở rộng như BlazePose vẫn có thể được sử dụng khi cần chi tiết cao hơn. Điều này rất hữu ích cho các pipeline nhận diện tư thế, theo dõi chuyển động, trong nghiên cứu thể thao và theo dõi sức khỏe. Trong quá trình ước lượng dữ liệu keypoint, các mô hình ước lượng tư thế người (HPE) có thể trả về những điểm với độ tin cậy thấp do nhiều nguyên nhân, bao gồm chất lượng hình ảnh kém, góc nhìn khó hoặc các tư thế phức tạp. Để đảm bảo tính nhất quán và công bằng khi so sánh hiệu năng giữa các mô hình, đề án áp dụng một ngưỡng trung bình 0.3 cho điểm tin cậy của các khớp xương (keypoint confidence score). Cụ thể, với mỗi ảnh, khi nhận được kết quả của mô hình HPE cho ảnh ta sẽ tính giá trị trung bình của tổng hợp các điểm tin cậy trong vector keypoint. Nếu điểm trung bình này thấp hơn 0.3, toàn bộ vector đặc trưng của các keypoint trong ảnh đó sẽ được gán bằng 0, coi như mô hình không phát hiện được tư thế người trong ảnh. Tiếp cận này không những giúp loại bỏ các kết quả không đáng tin cậy mà còn đảm bảo rằng tất cả các mô hình HPE được xem xét theo cùng một thước đo trước khi đưa vào mô đun tiếp theo.

2.3. Mô đun xử lý và trích xuất đặc trưng

Để tập trung vào tư thế và cấu trúc tương đối, thay vì bị ảnh hưởng bởi chiều cao hay vị trí của người trong khung hình, đề án áp dụng một phương pháp chuẩn hóa dựa trên vị trí trung tâm hông. Cụ thể, tọa độ của hông trái và hông phải được sử dụng để xác định tâm hông (hip center), trong khi vai trái và vai phải được dùng để tính tâm vai (shoulder center) và ước lượng kích thước thân (torso size). Khoảng cách lớn nhất giữa các điểm khớp và tâm hông, kết hợp với một hệ số tỷ lệ 2.5 từ kích thước thân, được chọn làm thang đo chuẩn hóa. Mỗi điểm khớp sau đó được dịch chuyển sao cho tâm hông trở thành gốc tọa độ và được chia cho thang đo này để đảm bảo kích thước cơ thể nằm trong một phạm vi chuẩn. Quá trình chuẩn hóa này giúp dữ liệu khung xương trở nên bất biến với vị trí, tỷ lệ cơ thể và khoảng cách tới camera, từ đó nâng cao độ ổn định và thích nghi tốt với ảnh Yoga lạ bất thường. Việc kết hợp chuẩn hóa tọa độ với đánh giá độ tin cậy của các keypoint đảm bảo rằng các vector đặc trưng đầu ra đều có chất lượng cao, loại bỏ các dữ liệu nhiễu, đồng thời duy trì tiêu chuẩn thống nhất cho các bước phân tích tiếp theo. Dưới đây là các bước minh họa quy trình chuẩn hóa các keypoints:

- Bước 1: Nhận dữ liệu đầu vào: Danh sách keypoints [N × 3] gồm tọa độ (x, y) và độ tin cậy confidence.
- Bước 2: Xác định tâm hông: Lấy tọa độ hông trái và hông phải, sau đó tính trung điểm → hip center.
- Bước 3: Xác định tâm vai và kích thước thân: Lấy tọa độ vai trái và vai phải, tính trung điểm → shoulder center; Torso size = khoảng cách từ shoulder center đến hip center.
- Bước 4: Xác định thang đo chuẩn hóa: Tính khoảng cách lớn nhất từ tất cả keypoints tới hip center; scale = max(torso size×2.5, khoảng cách lớn nhất)

• Bước 5: Chuẩn hóa từng keypoint: Với mỗi keypoint (x, y, confidence), dịch chuyển (x − hip_center.x, y – hip_center.y), chia cho scale, giữ nguyên confidence, và thêm vào normalized_keypoints.

• Bước 6: Xuất dữ liệu đầu ra: Danh sách normalized_keypoints đã chuẩn hóa, có thể tiến hành cho bước kế tiếp là nhận diện tư thế.

2.4. Mô đun phân loại tư thế

Mô đun phân loại tư thế là giai đoạn không thể thiếu chịu trách nhiệm chuyển đổi các vector đặc trưng trích xuất từ hình ảnh hoặc video thành nhãn tư thế cụ thể. Các vector đặc trưng này tổng hợp thông tin về vị trí các điểm khớp, hình dạng cơ thể và các mối quan hệ không gian giữa các khớp, từ đó giúp mô đun phân loại có thể nhận diện tư thế một cách chính xác và ổn định. Trong nghiên cứu, nhiều mô hình ML truyền thống và hiện đại đã được vận dụng nhằm so sánh hiệu năng và khả năng thích ứng của từng mô hình trên cùng bộ dữ liệu vector đặc trưng. Tuy nhiên, các mô hình quá nặng hoặc quá hiện đại không được sử dụng trong hệ thống này, bởi chúng sử dụng tài nguyên tính toán của chúng lớn nên thời gian suy luận lâu hơn, làm giảm khả năng triển khai trên các ứng dụng nền tảng di động hoặc môi trường trực tuyến. Việc ứng dụng các mô hình ML phù hợp không chỉ giúp đào tạo nhanh và dễ triển khai, mà còn mang lại tốc độ suy luận cao, rất lý tưởng với ứng dụng nhận diện tư thế Yoga theo thời gian thực. Đồng thời, việc đối chiếu hiệu năng giữa những mô hình còn giúp phân tích ưu nhược điểm, từ đó lựa chọn mô hình cân bằng giữa độ chuẩn xác, tốc độ và tính mở rộng của hệ thống.

Việc đánh giá được thí nghiệm trên các mô hình ML chủ yếu là có giám sát, cụ thể bao gồm:

K-Nearest Neighbors [8] là một thuật toán học máy đơn giản dựa trên khoảng cách giữa các điểm dữ liệu. Mỗi mẫu mới được quyết định dựa trên nhãn của K điểm lân cận nhất về mặt khoảng cách trong tập huấn luyện (thường là Euclidean, Minkowski hoặc Manhattan). KNN không yêu cầu huấn luyện cầu kỳ, không cần giả định gì về phân phối dữ liệu, tuy nhiên với bộ dữ liệu lớn, tốc độ giảm và nhạy cảm với nhiễu nên chỉ hợp cho các bài toán từ nhỏ đến trung bình, hoặc làm baseline để

so sánh với các mô hình phức tạp hơn, chứ ít khi được chọn làm giải pháp chính cho hệ thống lớn hoặc yêu cầu tốc độ cực cao.

Decision Tree [12] là một mô hình phân loại dựa trên cấu trúc cây, sử dụng các câu hỏi điều kiện để chia dữ liệu theo nhãn. ID3 và các phiên bản cải tiến như C4.5, CART là thuật toán điển hình để có thể tạo ra cây quyết định. Mô hình dễ hiểu, trực quan và giải thích được quyết định điều mà nhiều mô hình như CNN không làm được. Tuy nhiên, cây đơn lẻ dễ bị overfitting và tổng quát kém trên dữ liệu mới nên thường càn phải giới hạn độ sâu cây, áp dụng cắt tỉa (pruning).

Support Vector Machine [9] là một thuật toán mạnh mẽ, tối ưu hóa biên phân tách giữa các lớp dữ liệu. SVM vượt trội với dữ liệu có ranh giới phân loại rõ ràng và có thể mở rộng sang không gian chiều cao bằng kernel. Nó nổi bật khi có thể tổng quát hóa tốt ngay cả với tập dữ liệu nhỏ. Tuy nhiên, SVM có thể chậm khi xử lý tập dữ liệu lớn và yêu cầu lựa chọn hàm kernel phù hợp.

Multi-Layer Perceptron [35] là một thuật toán học sâu thuộc họ mạng nơ-ron truyền thẳng có thể học các mối quan hệ phi tuyến giữa các đặc trưng. Khác với những mô hình tuyến tính, MLP có cấu trúc gồm ba lớp: lớp đầu vào nhận keypoints từ mô hình HPE, một hoặc nhiều lớp ẩn với các hàm kích hoạt như ReLU, sigmoid, hay tanh … giúp tạo ra biểu diễn phức tạp hơn, và cuối cùng là lớp đầu ra đưa ra xác suất thuộc về từng lớp hành động. MLP sử dụng hàm kích hoạt phi tuyến và được huấn luyện bằng thuật toán backpropagation có thể giải quyết các bài toán phân loại phức tạp và thích hợp với dữ liệu không tuyến tính tuy nhiên, MLP lại yêu cầu tính toán cao hơn.

Random Forest [10] là một kỹ thuật học tập tổ hợp (hay còn gọi là ensemble learning) gồm một bộ các cây quyết định, trong đó từng cây được xây dựng trên một tập con dữ liệu và tập con đặc trưng khác nhau. Để phân loại một mẫu mới, nó kết hợp kết quả (bằng cách lấy đa số phiếu) từ tất cả các cây, giúp kiểm soát hiện tượng overfitting và tăng cường độ chuẩn xác. RF cũng cung cấp đánh giá của các đặc trưng cực kỳ hữu ích để phân tích xem keypoint nào đóng góp nhiều nhất vào việc nhận

diện dáng Yoga. Tuy nhiên, số lượng cây lớn có thể làm tăng chi phí tính toán và khó diễn giải.

Gaussian Naive Bayes [36] là một thuật toán học có giám sát dựa trên giả định của lý thuyết Bayes, rằng từng đặc trưng độc lập nhau và tuân thủ phân phối chuẩn. Mô hình đơn giản, siêu nhanh và hiệu quả với dữ liệu có phân phối gần chuẩn. GaussianNB thích hợp cho các bài toán phân loại thời gian thực. Điểm hạn chế là hiệu năng giảm nếu các đặc trưng có mối quan hệ mạnh với nhau.

XGBoost [13] là một thuật toán boosting mạnh mẽ, thông qua việc sử dụng tập hợp cây quyết định để tăng cường độ chuẩn xác dự đoán. Nó tối ưu hóa hiệu năng và khả năng tổng quát hóa, đồng thời hỗ trợ dữ liệu lớn và phân tán. XGBoost cũng tích hợp cơ chế điều chỉnh trọng số, giảm overfitting và tăng tốc tính toán để trở thành một trong số các thuật toán ưa chuộng trong các cuộc thi học máy nhưng yêu cầu nhiều tài nguyên và cần tinh chỉnh siêu tham số.

LightGBM [14] là một phiên bản cải tiến của gradient boosting, tối ưu cho tốc độ huấn luyện và hiệu quả mở rộng trên dữ liệu lớn. LightGBM sử dụng chiến lược phát triển cây dựa trên lá (leaf-wise) với mức độ tổn thất tối đa để giảm bộ nhớ giúp tăng tốc độ huấn luyện. LightGBM đặc biệt hiệu quả cao khi mà dữ liệu có số lượng đặc trưng lớn và tập dữ liệu khổng lồ.

2.5. Mô đun phản hồi và hiển thị

Mô đun phản hồi và hiển thị đảm bảo có thể quan sát kết quả nhận diện tư thế một cách trực quan, bằng cách hiển thị các nhãn tư thế và điểm khớp lên giao diện người dùng, giúp người tập điều chỉnh tư thế kịp thời và chuẩn xác. Trong quá trình hiển thị, hình ảnh được resize về kích thước ban đầu để đảm bảo tỷ lệ các điểm khớp đúng với thực tế, sau đó vẽ các điểm khớp với độ chính xác cao và nối các khớp theo cấu trúc xương (skeleton) bằng các đường nối. Nhãn tư thế được hiển thị trực tiếp trên hình ảnh bằng văn bản, giúp ta dễ dàng nhận biết tư thế hiện tại. Cách tiếp cận này không chỉ tạo thông tin phản hồi trực quan mà còn duy trì hiệu suất thời gian thực, phù hợp với các ứng dụng luyện tập Yoga hoặc thể thao trên các thiết bị di động hoặc là ứng dụng web.

Mô đun phản hồi và hiển thị thực hiện các bước tính toán tuần tự như sau:

• Bước 1: Chuẩn bị hình ảnh và vector điểm khớp. Nhận đầu vào là hình ảnh gốc và danh sách các điểm khớp (landmarks) được mô hình ước lượng tư thế trả về. Kiểm tra tính hợp lệ của dữ liệu: nếu hình ảnh hoặc landmarks trống, trả về None.

• Bước 2: Resize hình ảnh về kích thước ban đầu. Lấy chiều cao h và chiều rộng w của hình ảnh. Nhân tọa độ chuẩn hóa của mỗi điểm khớp với w và h để đưa về tọa độ pixel thực tế trên ảnh.

• Bước 3: Vẽ các điểm khớp. Duyệt qua từng điểm khớp và vẽ hình tròn tại vị trí (x, y) tính được, đảm bảo sử dụng độ chính xác float và làm tròn để tăng độ chính xác hiển thị.

• Bước 4: Vẽ xương (skeleton). Duyệt qua danh sách các cặp điểm khớp được kết nối và vẽ đường thẳng nối từ điểm đầu đến điểm cuối, tạo thành khung xương của cơ thể.

• Bước 5: Hiển thị nhãn tư thế. Nếu kết quả phân loại hợp lệ, lấy nhãn tương ứng từ danh sách lớp và vẽ nhãn lên hình ảnh tại vị trí cố định (ví dụ: góc trên bên trái).

• Bước 6: Trả về hình ảnh đã hiển thị. Hình ảnh cuối cùng có đầy đủ các điểm khớp, đường nối skeleton và nhãn tư thế. Có thể đưa hình ảnh này lên giao diện người dùng để cung cấp phản hồi trực quan và thời gian thực.

2.6. Kết luận chương 2

Chương 2 đã trình bày chi tiết nghiên cứu và đề xuất hệ thống nhận diện tư thế Yoga thời gian thực, được thiết kế theo kiến trúc mô đun. Mô đun trích xuất khung xương thực hiện việc ước lượng các điểm khớp (keypoints) chính xác, là cơ sở cho các bước xử lý tiếp theo. Mô đun xử lý và trích xuất đặc trưng chuyển đổi các điểm khớp thành vector đặc trưng được chuẩn hóa, phục vụ đầu vào của việc phân loại tư thế. Mô đun phân loại tư thế áp dụng các mô hình ML lựa chọn những mô hình cân

bằng mượt mà giữa tốc độ và độ chuẩn xác, lý tưởng với các ứng dụng thời gian thực. Cuối cùng, mô đun phản hồi và hiển thị đảm bảo trực quan hóa kết quả bằng cách vẽ skeleton và hiển thị nhãn tư thế lên hình ảnh, giúp người có thể xem và tự điều chỉnh tư thế kịp thời và chính xác.

Từ những nghiên cứu lý thuyết và thiết kế mô đun trong Chương 2, Chương 3 sẽ triển khai chi tiết về các thực nghiệm và đánh giá hệ thống nhận diện tư thế Yoga thời gian thực, bao gồm bộ dữ liệu sử dụng, cấu hình môi trường thực nghiệm, và kết quả thu được.

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1. Bộ dữ liệu thực nghiệm

Đề án đã khảo sát một số bộ dữ liệu từ các nghiên cứu trước [3] [24] [37] [38], dựa trên cơ sở các tiêu chí như thiết bị ghi hình sử dụng, số lượng tư thế và người tham gia, tổng số mẫu, sự đa dạng trong cách thực hiện tư thế, cũng như các điều kiện môi trường như ánh sáng, không gian và trang phục. Nhằm đáp ứng nhu cầu thực tế cho việc nhận diện tư thế Yoga tại nhà, đề án đã thiết kế một bộ dữ liệu mới với 12 tư thế cơ bản, phù hợp với người mới tập, đặc biệt hướng tới sinh viên và những người đi làm thường xuyên ngồi lâu. Các tư thế bao gồm: Cái cây, Chó úp mặt, Chiến binh 1, Chiến binh 2, Cúi người về phía trước, Em bé, Ngọn núi 1, Ngọn núi 2 (Leo núi), Tấm ván, Tấm ván thấp, Tam giác và Cây cầu. Bộ dữ liệu được xây dựng từ 10 tình nguyện viên, bao gồm 5 nam và 5 nữ. Mỗi người tham gia thực hiện đầy đủ 12 tư thế Yoga khác nhau trong quá trình ghi hình. Các tư thế được thực hành dưới nhiều tình huống khác nhau, sử dụng các thiết bị quay khác nhau, kết hợp với các cảnh nền và mức độ nhiễu đa dạng, nhằm đảm bảo tính phong phú và độ khó cho việc huấn luyện và đánh giá mô hình, chi tiết tạo dáng của 12 nhãn được minh họa trong Hình 3.1.

Để thu thập dữ liệu cho từng tư thế, đề án tiến hành quay video riêng biệt, mỗi video ghi lại một người thực hiện một tư thế cụ thể. Camera được đặt cách người tập khoảng 4–5 mét nhằm đảm bảo chất lượng hình ảnh, với chuẩn HD (720p), tốc độ hiển thị 30 khung hình/giây, và mỗi video có độ dài trên 10 giây. Mỗi nhãn tư thế gồm 10 video, trong đó 2 video được quay theo chiều ngang và 8 video theo chiều dọc. Trong bước tiền xử lý, các khung hình được trích xuất từ video, cắt thành hình vuông để loại bỏ các yếu tố thừa và duy trì tỷ lệ phù hợp. Tiếp đó, các ảnh được chuẩn hóa về kích thước 640 × 640 × 3 pixel, đảm bảo tính nhất quán và tương thích với các mô hình HPE. Để tăng độ phong phú của dữ liệu, đề án áp dụng các kỹ thuật tăng cường như lật ngang, xoay nhẹ từ -10° đến 10°, dịch chuyển và thực hiện các biến dạng nhẹ để tái hiện sự linh hoạt của cơ thể, mô phỏng các điều kiện tập luyện khác nhau. Sau quá trình này, bộ dữ liệu cuối cùng gồm 13.928 ảnh . Bảng 3.1 trình bày

chi tiết số lượng ảnh của từng tư thế Yoga.



Hình 3.1: Minh họa 12 tư thế Yoga trong bộ dữ liệu

Bảng 3.1: Số lượng ảnh cho từng tư thế Yoga

Tư thế Yoga	**Số lượng**	**Tư thế Yoga**	**Số lượng**
Cái cây	1125	Cây cầu	1160
Chó úp mặt	1170	Cúi người về phía trước	1140
Chiến binh 1	1140	Chiến binh 2	1170
Em bé	1170	Leo núi	1165
Ngọn núi	1215	Tam giác	1145
Tấm ván	1140	Tấm ván thấp	1188

3.2. Cài đặt thực nghiệm

3.2.1. Môi trường phần cứng và phần mềm

Toàn bộ thực nghiệm được triển khai trên một hệ thống với phần cứng gồm CPU Intel Core i5-13400F thế hệ thứ 13, 10 nhân, 16 luồng, xung nhịp cơ bản 2,50 GHz và bộ nhớ RAM DDR4 dung lượng 32 GB.

Các thí nghiệm được triển khai sử dụng môi trường ảo Anaconda, cho phép quản lý các gói thư viện và phiên bản Python một cách dễ dàng, linh hoạt, đồng thời đảm bảo môi trường cô lập, tránh xung đột giữa các thư viện. Bảng 3.2 tổng hợp các thư viện chính và chức năng của chúng:

Bảng 3.2: Các gói thư viện được sử dụng trong quá trình thực nghiệm

Thư viện	**Ý nghĩa / Chức năng**
matplotlib	Thư viện vẽ đồ thị cơ bản, trực quan hóa kết quả, biểu đồ.
mediapipe [39]	Thư viện nhẹ, hỗ trợ trích xuất keypoints cơ thể người dùng mô hình BlazePose.
mmcv	Thư viện hỗ trợ cơ sở hạ tầng, bao gồm xử lý ảnh, tensor operations, logging, cấu hình mô hình.
mmdet	Thư viện phát hiện đối tượng (object detection).
mmengine	Engine cốt lõi của OpenMMLab, cung cấp framework huấn luyện, logging và checkpoint.
mmpose [40]	Thư viện OpenMMLab hỗ trợ nhiều mô hình HPE có thể chạy CPU/GPU.
mmpretrain	Bộ mô hình tiền huấn luyện từ OpenMMLab, hỗ trợ các mô hình như ViTPose.
numpy	Thao tác ma trận và tính toán số học hiệu quả.
opencv-python	Thư viện xử lý ảnh/video, hỗ trợ đọc, ghi, resize, hiển thị và thao tác ảnh.
openmim	Công cụ quản lý các gói của OpenMMLab, dùng

	để cài đặt mmengine, mmcv, mmpose, mmdet.
pandas	Quản lý dữ liệu dạng bảng (DataFrame), đọc/ghi CSV, tiền xử lý dữ liệu.
scikit-learn	Bộ công cụ ML đa dạng, hỗ trợ KNN, SVM, Decision Tree, Random Forest, train/test split, metrics.
tensorflow	Framework học sâu phổ biến, dùng để tải mô hình Movenet.
tensorflow_hub	Kho mô hình đã huấn luyện sẵn trên TensorFlow, hỗ trợ tải và sử dụng nhanh mô hình Movenet.
torch	Framework học sâu (PyTorch), được yêu cầu để chạy MMPose.
torchvision	Bộ công cụ đi kèm PyTorch phục vụ cho xử lý hình ảnh và mô hình học sâu.

3.2.2. Thiết lập mô hình HPE

Trong nghiên cứu này, tám mô hình ước lượng tư thế người (Human Pose Estimation – HPE) được lựa chọn để tiến hành so sánh hiệu năng, tập trung vào các yếu tố quan trọng như tốc độ suy luận, độ chính xác nhận dạng khớp, và khả năng triển khai trong môi trường thực tế. Tất cả các mô hình đều được lựa chọn ở phiên bản nhẹ nhất, nhằm đảm bảo tính khả thi khi áp dụng cho các hệ thống nhận diện tư thế Yoga thời gian thực :

Các mô hình từ thư viện Mediapipe:

- BlazePose33 (MediaPipe): Phiên bản nhẹ với 33 điểm khớp, được tối ưu hóa cho thiết bị không có GPU, có khả năng chạy trực tiếp trên CPU hoặc webcam thời gian thực.
- BlazePose17 (MediaPipe): Vẫn sử dụng mô hình BlazePose33 nhưng rút gọn kết quả đầu ra xử lý chỉ còn 17 điểm khớp theo chuẩn COCO.

Các mô hình từ thư viện TensorFlow:

- MoveNetTF2: Bản *single-pose-lightning*, sử dụng TensorFlow 2, tối ưu

cho độ chính xác và tốc độ.

- MoveNetTFLite: Bản *single-pose-lightning* đã được lượng tử hóa sang float16, giúp giảm kích thước mô hình và tăng tốc trên thiết bị di động.

Các mô hình tiên tiến từ thư viện MMPose[40]:

- YOLOXPose[5]: Sử dụng cấu hình *yoloxpose_tiny_4xb64-300e_coco-416*, phiên bản YOLOXPose-tiny, huấn luyện 300 epoch trên tập MS COCO, batch size 4 × 64, ảnh đầu vào 416 × 416, tối ưu cho tốc độ thời gian thực.
- RTMO[11]: Sử dụng cấu hình *rtmo-s_8xb32-600e_coco-640x640*, phiên bản RTMO-small, batch size 8 × 32, huấn luyện 600 epoch, đầu vào 640 × 640, kết hợp phát hiện và ước lượng tư thế trong một mạng duy nhất.
- HRNet[32]: Sử dụng cấu hình *td-hm_hrnet-w32_fp16-8xb64-210e_coco-256x1* phiên bản HRNet-W32, huấn luyện 210 epoch trên tập COCO, kích thước đầu vào 256 × 192, batch size 8 × 64, sử dụng FP16 để tăng tốc tính toán.
- ViTPose[33]: Sử dụng cấu hình *td-hm_ViTPose-small_8xb64-210e_coco-256x1* phiên bản ViTPose-small, batch size 8 × 64, huấn luyện 210 epoch, đầu vào 256 × 192, tận dụng Transformer để học đặc trưng khớp.
- RTMPose[34]: Sử dụng cấu hình *rtmpose-t_8xb256-420e_coco-256x192*, phiên bản RTMPose-tiny, batch size 8 × 256, huấn luyện 420 epoch, kích thước đầu vào 256 × 192, thiết kế gọn nhẹ và nhanh.

3.2.3. Thiết lập đào tạo mô hình ML

Trong quá trình đào tạo mô hình ML, 3.000 mẫu được tách riêng làm tập kiểm thử (test set), mẫu còn lại được dùng cho huấn luyện và xác thực mô hình. Việc chia dữ liệu được thực hiện theo chiến lược stratified sampling nhằm đảm bảo tỉ lệ phân bố nhãn giữa các tập dữ liệu được giữ nguyên như trong toàn bộ tập gốc, đồng thời cố định hạt giống ngẫu nhiên là 42 để đảm dễ dàng năng tái lập kết quả. Để đánh giá hiệu quả của các đặc trưng khung xương (pose features) được trích xuất từ những mô hình ước lượng tư thế người (HPE), tám thuật toán học máy khác nhau được triển khai từ thư viện scikit-learn [41] (Decision Tree [12], GaussianNB [36], KNN [8], LightGBM [14], MLPClassifier [35], Random Forest [10], SVM [9], và XGBoost

[13]). Quy trình tối ưu được cấu hình với 10 vòng lặp tìm kiếm và đánh giá chéo 5 lần (5-fold cross-validation), trong đó chỉ số Accuracy được xem như là thang đo cho đánh giá chính và là hàm mục tiêu để tái huấn luyện mô hình tốt nhất. Toàn bộ công đoạn huấn luyện và đánh giá được thực thi song song trên CPU/GPU với cài đặt không gian tìm kiếm các siêu tham số trình này chi tiết trong Bảng 3.3. Sau khi tối ưu, mô hình tốt nhất (best estimator) được dùng để đánh giá hiệu năng tổng thể trên tập kiểm thử độc lập.

Bảng 3.3: Không gian tìm kiếm siêu tham số cho các mô hình ML

Mô hình	**Không gian siêu tham số**
KNN	Số láng giềng (n_neighbors) ∈ [3, 19]; trọng số (weights) ∈ {uniform, distance}; khoảng cách (metric) ∈ {euclidean, manhattan}
SVM	Hằng số điều chỉnh C ∈ [10^{-4}, 10^{3}] (log-uniform); kiểu hạt nhân (kernel) ∈ {linear, rbf, poly}; γ ∈ [10^{-5}, 10] (log-uniform)
DT	Độ sâu tối đa (max_depth) ∈ [2, 50]; số mẫu tách nút tối thiểu (min_samples_split) ∈ [2, 20]; số mẫu lá tối thiểu (min_samples_leaf) ∈ [1, 10]; tiêu chí (criterion) ∈ {gini, entropy}
RF	Số cây (n_estimators) ∈ [100, 500]; độ sâu tối đa (max_depth) ∈ [3, 20]; số mẫu tách nút tối thiểu (min_samples_split) ∈ [2, 20]; số đặc trưng tối đa (max_features) ∈ {sqrt, log2, None}; tiêu chí (criterion) ∈ {gini, entropy}
XGBoost	Số cây (n_estimators) ∈ [100, 1000]; tốc độ học (learning_rate) ∈ [0.005, 0.3] (log-uniform); độ sâu tối đa (max_depth) ∈ [3, 15]; tỉ lệ mẫu con (subsample) ∈ [0.5, 1.0]; tỉ lệ đặc trưng theo cây (colsample_bytree) ∈ [0.5, 1.0]; trọng số tối thiểu nút con (min_child_weight) ∈ [1, 10]
LightGBM	Số cây (n_estimators) ∈ [100, 1000]; tốc độ học (learning_rate) ∈

	[0.005, 0.3] (log-uniform); độ sâu tối đa (max_depth) ∈ [–1, 15] (–1 = không giới hạn); số lá (num_leaves) ∈ [15, 255]; tỉ lệ mẫu con (subsample) ∈ [0.5, 1.0]; tỉ lệ đặc trưng (feature_fraction) ∈ [0.5, 1.0]
GaussianNB	Độ trơn biến (var_smoothing) ∈ [10^{-10}, 10^{-6}] (log-uniform)
MLPClassifier	Kích thước lớp ẩn (hidden_layer_sizes) ∈ {50, 100, 150, 200, 300}; hàm kích hoạt (activation) ∈ {relu, tanh, logistic}; thuật toán tối ưu (solver) ∈ {adam, sgd}; hệ số α ∈ [10^{-5}, 0.1] (log-uniform); tốc độ học khởi tạo (learning_rate_init) ∈ [10^{-4}, 0.1] (log-uniform); số vòng lặp tối đa (max_iter) ∈ [200, 500]

3.2.4. Tiêu chí đánh giá hiệu năng

Dựa trên cấu hình thực nghiệm, đề án này đánh giá hiệu năng của các mô hình ước lượng tư thế người (HPE) thông qua hai chỉ số chính:

Tỷ lệ phát hiện (Detection Rate): Thể hiện mức độ mà mô hình có thể nhận dạng được các ảnh có điểm tin cậy tổng hợp

$$\text{Detection Rate} = \frac{Số\ ảnh\ có\ score_i > \tau}{N} \qquad (1)$$

Trong đó:

- $score_i$: điểm tin cậy của ảnh thứ i;
- τ: ngưỡng tin cậy (confidence threshold);
- N: tổng số ảnh trong bộ dữ liệu.

Trong đề án này, giá trị ngưỡng được đặt là $\tau = 0.3$.

Thời gian suy luận trung bình (Inference Time): Thời gian suy luận thể hiện tốc độ xử lý của mô hình trên từng ảnh. Giả sử tổng thời gian mô hình cần để xử lý toàn bộ N ảnh là T , thì thời gian suy luận trung bình được tính bằng tổng thời gian chia cho số lượng ảnh:

$$\text{Inference Time} = \frac{T}{N} \qquad (2)$$

- T : tổng thời gian (giây hoặc mili-giây) mà mô hình cần để dự đoán các điểm khớp (keypoints) cho toàn bộ tập ảnh;
- N: tổng số ảnh trong bộ dữ liệu.

Sau khi đánh giá hiệu năng HPE, mỗi mô hình HPE được kết hợp với từng mô hình học máy (ML) để tạo thành các tổ hợp HPE–ML. Hiệu năng của các tổ hợp này được đánh giá dựa trên:

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3)$$

Precision (Độ chính xác) **và Recall** (Độ nhạy / Độ thu hồi):

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (4) \qquad \text{Recall} = \frac{TP}{TP + FN} \qquad (5)$$

F1-score là trung bình điều hòa giữa Precision và Recall:

$$\text{F1} = 2 . \frac{Precision . Recall}{Precision + Recall} \qquad (6)$$

Trong đó:

- TP và TN lần lượt thể hiện số mẫu dương và âm được mô hình dự đoán chính xác.
- FP và FN lần lượt thể hiện số mẫu âm bị mô hình dự đoán nhầm thành dương và số mẫu dương được dự đoán nhầm thành âm.

Việc kết hợp bốn chỉ số nhằm phản ánh hiệu quả nhận dạng của mô hình, nhất là trong các dữ liệu có sự mất cân bằng giữa các lớp.

3.3. Kết quả và đánh giá thực nghiệm

3.3.1. Hiệu suất ước lượng tư thế người

Bảng 3.4 trình bày kết quả phân tích hiệu năng của chín mô hình ước lượng tư thế người (HPE) trên cùng tập dữ liệu thử nghiệm. Có thể thấy, hầu hết các mô hình đều đạt tỷ lệ phát hiện cao trên 97%, cho thấy khả năng nhận diện tư thế ổn định trong

nhiều điều kiện khác nhau.

Bảng 3.4: So sánh kết quả thực nghiệm của các mô hình HPE

Mô hình HPE	Detection Rate (%)	Inference Time (ms)
BLAZEPOSE_33	97.83	36.34
BLAZEPOSE_17	97.83	32.15
MOVENET_TFLITE	97.49	10.56
MOVENET_TF2	97.41	80.36
RTMO	99.99	268.28
RTMPOSE	75.53	54.07
HRNET	99.98	389.03
VITPOSE	99.84	189.27
YOLOXPOSE	99.94	131.58

Trong nhóm các mô hình có độ chính xác vượt trội, RTMO và HRNet đạt tỷ lệ phát hiện 99.99% và 99.98% gần như tuyệt đối, phản ánh năng lực biểu diễn đặc trưng mạnh mẽ của các mô hình có kiến trúc sâu. Tuy nhiên, thời gian suy luận của hai mô hình này khá lớn (268.28 ms và 389.03 ms), khiến chúng kém phù hợp cho hệ thống thời gian thực. ViTPose và YOLOXPose cũng đạt độ chính xác rất cao (99.84% và 99.94%), song thời gian suy luận vẫn còn tương đối cao (189.27 ms và 131.58 ms), cho thấy chi phí tính toán đáng kể do sử dụng các khối Transformer và backbone phức tạp.

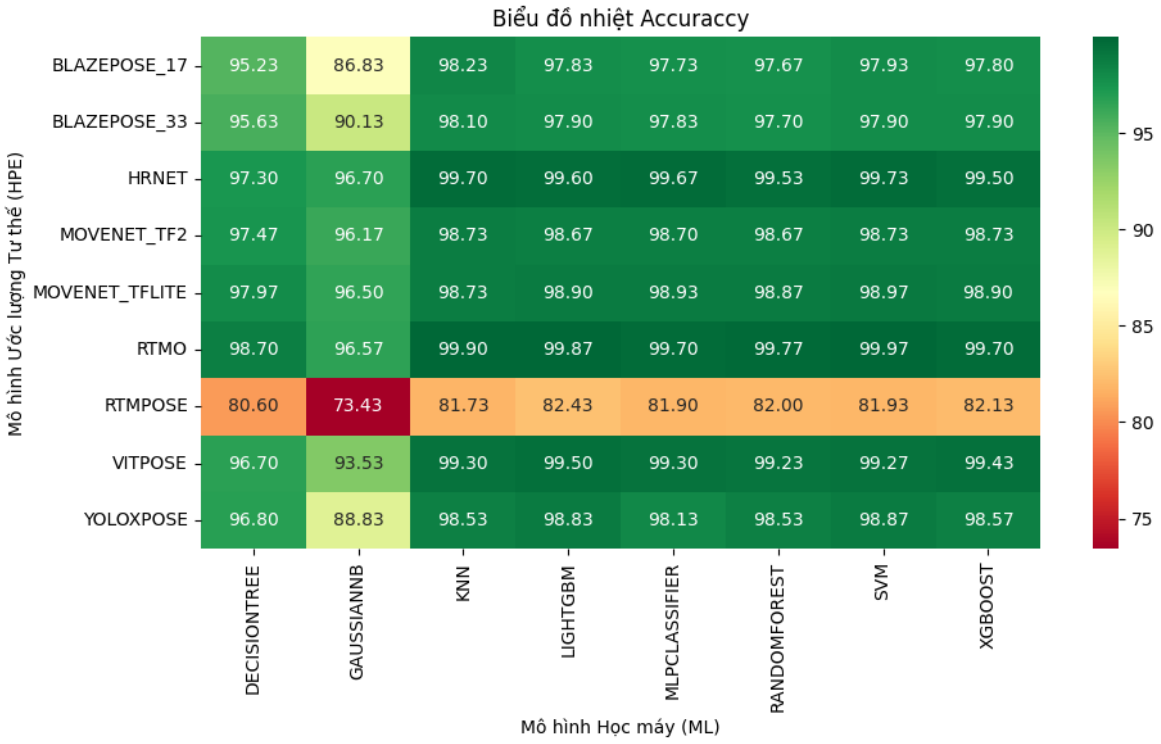
Hai biến thể BlazePose_33 và BlazePose_17 đạt độ chính xác 97.83% với thời gian suy luận lần lượt là 36.34 ms và 32.15 ms, thể hiện sự cân bằng khá tốt giữa hiệu năng và tốc độ, phù hợp với các ứng dụng trên thiết bị di động hoặc hệ thống có tài nguyên hạn chế. Ngược lại, RTMPose có kết quả thấp nhất trong nhóm với tỷ lệ phát hiện chỉ đạt 75.53%, thậm chí thời gian suy luận trung bình là 54.07 ms – cao hơn BlazePose. Điều này phản ánh mô hình này chưa đạt được hiệu quả kỳ vọng xử lý bài toán theo đúng yêu cầu và đặc trưng dữ liệu.

Với MoveNet_TF2 cho thấy khả năng chấp nhận được chỉ 80.36 ms, trong khi vẫn đạt độ chính xác 97.41%. Đáng chú ý, MoveNet_TFLite, phiên bản lượng tử

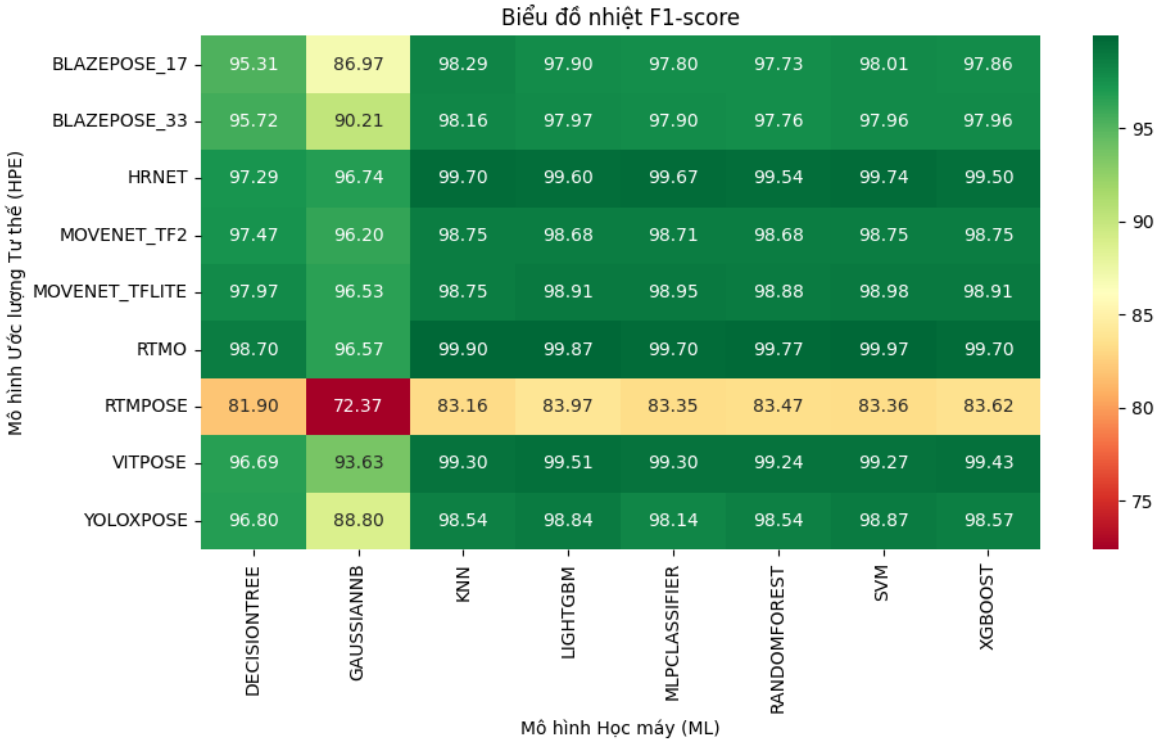
hóa, cho kết quả tốt hơn một chút về tỷ lệ phát hiện (97.49%) nhưng tốt hơn rất nhiều về tốc độ ước lượng tư thế (chỉ 10.56 ms), cho thấy tính năng tối ưu hóa vượt trội khi triển khai trên các thiết bị biên. Nhờ sự kết hợp giữa tốc độ xử lý nhanh, độ chuẩn xác ổn định và dung lượng mô hình nhỏ, MoveNet_TFLite được xem là lựa chọn phù hợp nhất cho các ứng dụng yêu cầu phản hồi thời gian thực hoặc hoạt động trên phần cứng giới hạn, đồng thời vẫn đảm bảo chất lượng nhận diện tư thế ở mức cao.

Tổng thể, các phát hiện cho thấy MoveNet_TF2 là phương án ưu việt nhất cho các nền tảng thời gian thực do có thể cân bằng tốt giữa tốc độ và độ chuẩn xác, trong khi RTMO, HRNet và ViTPose phù hợp hơn cho các hệ thống yêu cầu độ chính xác cực cao nhưng không bị giới hạn về tài nguyên tính toán.

3.3.2. So sánh hiệu suất kết hợp HPE và thuật toán học máy



Hình 3.2: Kết quả độ chính xác (Accuracy) của các tổ hợp mô hình HPE và ML



Hình 3.3: Kết quả điểm F1-Score của các tổ hợp mô hình HPE và ML

Hình 3.2 và 3.3 minh họa hai biểu đồ nhiệt thể hiện kết quả trung bình của các mô hình HPE khi kết hợp với các mô hình ML khác nhau, được đánh giá qua hai chỉ số chính là Accuracy và F1-score. Nhìn chung, các biểu đồ cho thấy xu hướng tương đồng, phản ánh tính ổn định của hệ thống khi đánh giá theo cả độ chính xác lẫn độ cân bằng giữa precision và recall. Trước hết, có thể nhận thấy rằng RTMO, HRNet, và VitPose là những mô hình HPE đạt hiệu suất nổi bật nhất. Các mô hình này đều đạt Accuracy và F1-score trên 98–99% với hầu hết các bộ phân loại, đặc biệt khi kết hợp với KNN hay là SVM kết quả gần đạt mức tối đa. Điều này chứng tỏ rằng những mô hình HPE hiện đại có khả năng trích xuất đặc trưng tư thế người một cách giàu thông tin và ổn định, giúp cho các mô hình ML phía sau dễ dàng phân loại chính xác hơn. Ngược lại, RTMPose thể hiện hiệu suất thấp nhất trong cả hai thước đo, chỉ đạt 80–83%, nghiêm trọng nhất khi kết hợp với GaussianNB chỉ đạt từ 72-73%. Kết quả này cho thấy đặc trưng tư thế được trích xuất từ RTMPose có thể chưa đủ ổn định hoặc không phù hợp với những mô hình phân loại tuyến tính, khiến cho độ chính xác tổng thể bị suy giảm. Nhóm mô hình trung bình như BlazePose_17, BlazePose_33,

và YOLOXPose cũng đạt kết quả khá tốt (dao động 95–98%), nhưng vẫn thấp hơn đáng kể so với nhóm RTMO và HRNet. Sự cải thiện nhẹ từ BlazePose_17 lên BlazePose_33 cho thấy việc tăng số lượng keypoints có thể giúp mô hình nhận diện tư thế chi tiết hơn và cải thiện kết quả phân loại.

Tổng thể, các kết quả cho thấy:

- Các mô hình HPE hiện đại (như RTMO, HRNet, ViTPose) mang lại hiệu suất vượt trội và ổn định nhất.
- Các thuật toán ML (như RandomForest, SVM, XGBoost) có khả năng khai thác đặc trưng tốt hơn so với các phương pháp như GaussianNB hay DecisionTree.
- Độ sai khác giữa các mô hình HPE lớn hơn nhiều so với giữa các mô hình ML, cho thấy hiệu suất tổng thể phụ thuộc vào chất lượng mô hình ước lượng tư thế hơn là mô hình học máy được sử dụng (xem thêm tại Bảng Phụ lục PL.1).

Trong đề án này, MoveNetTFLite chính thức được chọn làm mô hình HPE chính nhờ sự cân bằng vượt trội giữa độ chính xác, tốc độ và kích thước. Điều quan trọng là kết hợp MoveNetTFLite với mô hình ML nào phù hợp nhất để triển khai. Có thể thấy, khi kết hợp với 8 mô hình ML thì tất cả đều đạt F1-score trên 96.5%, trong đó 6/8 mô hình đạt ≥ 98.87% và cao nhất là tổ hợp MoveNet_TFLite + SVM với F1-score 98.98%. Về điểm accuracy, tổ hợp này cũng dẫn đầu ở mức cao 98.97%. Đặc biệt, khi quan sát bảng kết quả chi tiết Bảng PL.1, sai số giữa các fold trong cross-validation của F1-score luôn dưới 1%, chứng tỏ đây là sự kết hợp ổn định, ít nhiễu và phù hợp hoàn hảo cho cả công đoạn thử nghiệm lẫn ứng dụng thực tế. Vì vậy, đề án quyết định lựa chọn chính thức tổ hợp MoveNetTFLite kết hợp với SVM làm giải pháp cuối cùng để thực thi thực tế.

3.3.3. Đánh giá hiệu suất thời gian thực

Hình 3.4 và 3.5 minh họa kết quả nhận diện tư thế Yoga thời gian thực trên hai thiết bị khác nhau. Cụ thể, Hình 3.4 thể hiện kết quả trên máy bàn Gigabyte sở hữu bộ xử lý CPU Intel Core i5-13400F (RAM 32 GB), trong khi Hình 3.5 cho thấy kết quả trên laptop Acer Swift SF314-512, dùng bộ xử lý Intel Core i7-1260P (12

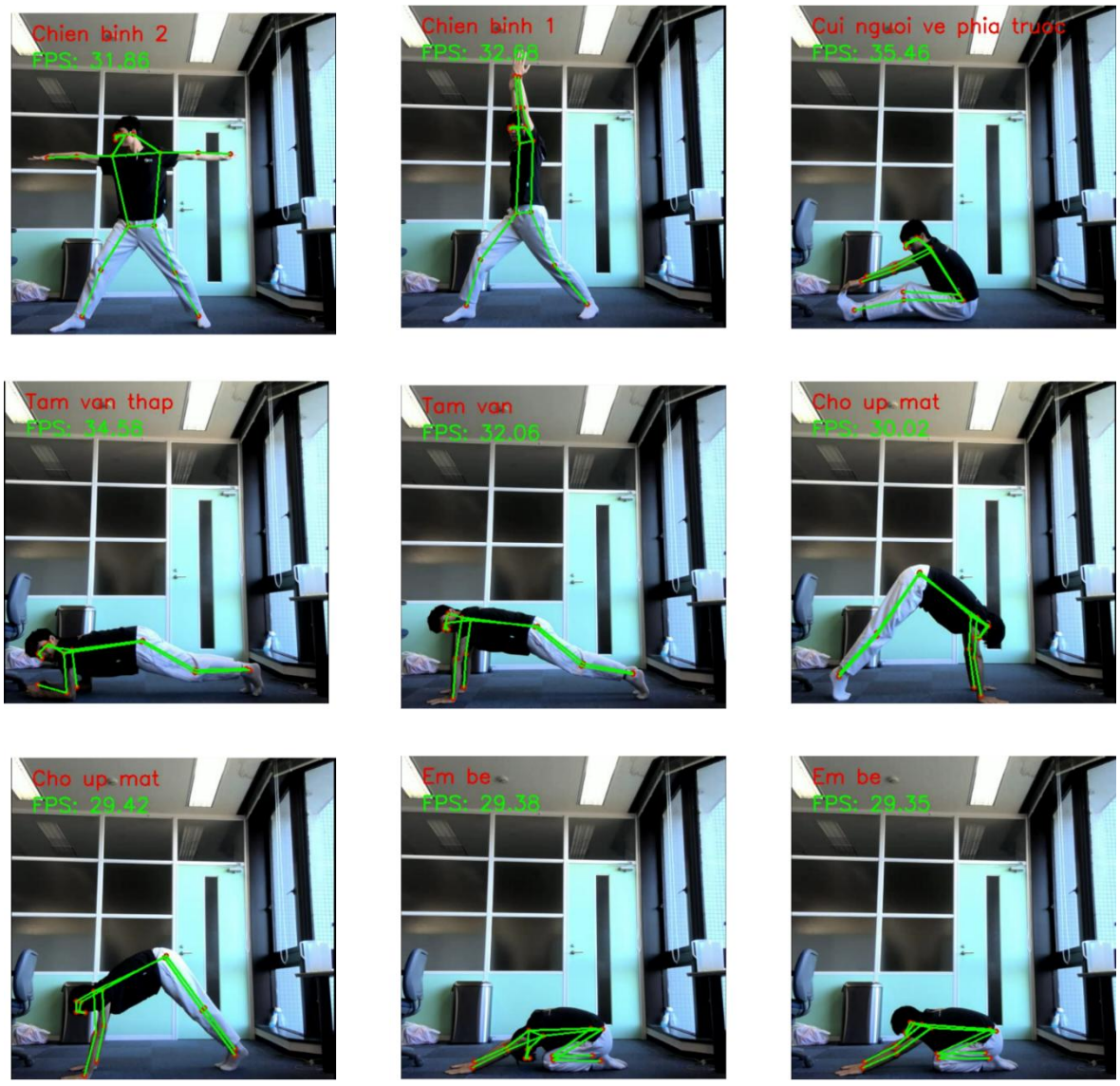
nhân, 16 luồng, xung nhịp cơ bản 2.1 GHz), RAM 16 GB Kết quả chỉ ra hệ thống vẫn duy trì mức chính xác tốt và tốc độ xử lý ổn định trong các phần cứng khác nhau. Kết quả cho thấy MoveNet TFLite khi được triển khai kết hợp với bộ phân loại SVM trong nhận diện tư thế Yoga, hệ thống đạt tốc độ xử lý trung bình 30–31 FPS trên các cấu hình phần cứng thử nghiệm. Hiệu suất này chứng minh khả năng xử lý thời gian thực mượt mà, đồng thời vẫn duy trì độ chính xác nhận diện cao, nhờ sự kết hợp hiệu quả giữa mô hình trích xuất đặc trưng gọn nhẹ và bộ phân loại có độ tổng quát tốt.



Hình 3.4: Kết quả nhận diện và phân loại tư thế Yoga thời gian thực trên máy cây Gigabyte (32 GB RAM) sử dụng MoveNet TFLite và SVM.

Quan sát chi tiết hơn trong các hình minh họa, có thể nhận thấy rằng hệ thống vẫn duy trì khả năng phân loại chính xác ngay cả khi một số điểm khớp (keypoint) bị thiếu hoặc không được phát hiện đầy đủ, người dùng thực hiện tư thế hơi lệch hướng so với vị trí trực diện, hoặc khoảng cách giữa người tập và camera thay đổi đáng kể (từ rất gần đến khá xa). Điều này cho thấy tính ổn định, khả năng thích ứng và chịu lỗi tốt của mô hình trong các điều kiện nhận dạng không lý tưởng, giúp đảm bảo hiệu quả nhận diện tư thế ngay cả trong môi trường luyện tập thực tế đa dạng. Bên cạnh đó, các điểm khớp và nhãn tư thế được hiển thị rõ nét, sinh động, thuận tiện người tập dễ quan sát, đối chiếu và điều chỉnh động tác trong khi luyện tập. Nhờ vậy, hệ

thống không chỉ hỗ trợ nhận diện tư thế tự động mà còn đóng vai trò công cụ phản hồi trực quan hiệu quả trong các ứng dụng huấn luyện Yoga tương tác thời gian thực.



Hình 3.5: Kết quả nhận diện và phân loại tư thế Yoga thời gian thực trên laptop Acer (16 GB RAM) sử dụng MoveNet TFLite và SVM.

3.4. Kết luận chương 3

Trong phạm vi chương 3, toàn bộ quá trình thực nghiệm và kết quả đánh giá hiệu năng của hệ thống nhận diện tư thế Yoga được phân tích, bao gồm các mô hình HPE và ML được áp dụng trong giai đoạn phân loại. Trước hết, bộ dữ liệu và môi trường thực nghiệm đã được thiết lập nhằm giữ vững khách quan và khả năng tái lập. Các mô hình HPE được triển khai và đánh giá dựa trên hai tiêu chí chính: tỷ lệ phát

hiện (Detection Rate) và thời gian suy luận trung bình (Inference Time). Kết quả cho thấy mỗi mô hình có đặc trưng và ưu thế riêng: HRNet và RTMO đạt độ chính xác cao nhất nhưng tiêu tốn thời gian suy luận lớn, trong khi MoveNetTFLite thể hiện tốc độ vượt trội (12.61 ms) cùng độ chính xác cao (97.49%). Khi kết hợp các mô hình HPE với các bộ phân loại học máy, kết quả thực nghiệm chứng minh rằng hiệu suất tổng thể phụ thuộc đồng thời vào cả hai thành phần. Trong đó, cặp mô hình MoveNetTFLite–SVM đạt hiệu năng cao và ổn định với độ chính xác 98.97%, điểm F1 đạt 98.98%, và tốc độ xử lý trung bình 30–31 FPS trên phần cứng thực nghiệm, đủ khả năng phục vụ các tác vụ cần tốc độ xử lý tức thì.

Tổng kết lại, các kết quả của chương 3 đã chứng tỏ tính khả thi và cách tiếp cận kết hợp mô hình HPE nhẹ với bộ phân loại học máy là hiệu quả. Kết quả này là cơ sở cho việc xây dựng, triển khai và đánh giá hệ thống nhận dạng tư thế Yoga thời gian thực trong thực tế.

KẾT LUẬN

Bài toán nhận diện tư thế Yoga là một vấn đề giàu tính ứng dụng trong bối cảnh tập luyện và theo dõi sức khỏe ngay tại nhà đang dần trở nên xu hướng. Việc xây dựng hệ thống có thể nhận biết và đánh giá chuẩn xác tư thế người tập không chỉ hỗ trợ phòng tránh chấn thương mà còn góp phần tăng cường hiệu quả luyện tập, đồng thời tạo nên nhiều tiềm năng trên các nền tảng sức khỏe thông minh và huấn luyện cá nhân hóa.

Đề án đã tiến hành đánh giá toàn diện nhiều mô hình trích xuất khung xương (HPE) kết hợp với các thuật toán ML nhằm tìm được giải pháp tối ưu cho nhận diện tư thế Yoga theo thời gian thực. Qua thực nghiệm trên tập dữ liệu YogaSinglePose-PTIT, được thu thập trong môi trường sinh hoạt thực tế, cho thấy tổ hợp MoveNet TFLite và SVM đạt hiệu suất ấn tượng cả về độ chính xác lẫn tốc độ xử lý, thỏa mãn tốt các tiêu chuẩn khắt khe của hệ thống thời gian thực. Thành quả này chứng minh rằng nghiên cứu đề xuất hoàn toàn có thể được ứng dụng vào các hệ thống giám sát tư thế luyện tập thông minh, hỗ trợ người dùng tự tập Yoga tại nhà.

Các bước phát triển tiếp theo sẽ hướng đến việc mở rộng khả năng nhận diện đa người dùng, cải thiện độ ổn định trong các bối cảnh khác nhau, đồng thời tích hợp cơ chế phản hồi thông minh cho phép người tập nhận hướng dẫn chỉnh sửa tư thế trực tiếp. Đặc biệt, nhóm hướng tới việc áp dụng các kỹ thuật Explainable AI nhằm giúp hệ thống không chỉ đưa ra kết quả nhận dạng, mà còn có tính năng giải thích dễ hiểu,trực quan các vùng khung xương hoặc tư thế khiến mô hình đưa ra quyết định đó. Điều này sẽ tăng tính minh bạch, giúp người dùng hiểu rõ hơn về lý do hệ thống đánh giá một tư thế là “đúng” hay “chưa chuẩn”, qua đó nâng cao độ tin cậy và trải nghiệm sử dụng. Cuối cùng, việc đóng gói mô hình vận hành trên các thiết bị biên (edge devices) hoặc ứng dụng web/app di động sẽ được xem xét, nhằm tối ưu hiệu năng và mở rộng khả năng áp dụng thực tế cho các thiết bị cấu hình thấp, hướng tới một hệ thống nhận diện tư thế Yoga thông minh, nhẹ, nhanh và thân thiện với người dùng.

CÔNG BỐ KHOA HỌC

Tên bài báo: Đánh giá hiệu suất kết hợp mô hình trích xuất khung xương và học máy trong nhận diện tư thế Yoga thời gian thực.

Tác giả: Nguyễn Quang Huy, Nguyễn Tuấn Phát, Đinh Hoàng Anh, Triệu Ngọc Tâm, Nguyễn Duy Phương.

Hội nghị: Hội thảo Quốc gia lần thứ XXVIII – VNICT 2025: Một số vấn đề chọn lọc của Công nghệ thông tin và Truyền thông, Ninh Bình, 2025.

Ngày tổ chức hội nghị: 15–16/11/2025.

ISBN: 978-604-45-1878-7.

Tóm tắt: Sự gia tăng nhanh chóng của việc tập luyện Yoga tại nhà kéo theo sự quan tâm ngày càng cao đối với các hệ thống có khả năng phát hiện và phân loại tư thế kịp thời để tránh chấn thương và cải thiện hiệu quả luyện tập. Việc phát triển các hệ thống này đặt ra thách thức lớn khi chúng không những cần đạt độ chính xác nhận diện cao mà còn phải vận hành trong bối cảnh thời gian thực trên các thiết bị giới hạn về phần cứng. Nghiên cứu này tiến hành so sánh có hệ thống giữa kết hợp 8 mô hình ước lượng tư thế người (HPE) và 8 thuật toán học máy (ML) khác nhau trong bài toán nhận diện tư thế Yoga, nhằm tìm ra tổ hợp tối ưu nhất xét trên cả độ chính xác và hiệu quả tính toán. Kết quả thực nghiệm chỉ ra các kết quả đều đạt được độ chính xác tốt. Trong đó MoveNetTFLite kết hợp SVM đạt độ chính xác 98.96% và duy trì tốc độ xử lý trung bình 30–31 FPS, đảm bảo cân bằng nhất về hiệu suất phân loại và đáp ứng yêu cầu phản hồi ngay lập tức. Từ đó kết quả nghiên cứu mang lại định hướng khả thi cho các hệ thống nhận diện tư thế Yoga thời gian thực có thể triển khai trên nền tảng di động và nhúng.

Từ khóa: Nhận diện tư thế Yoga, ước lượng tư thế người, học máy, hệ thống thời gian thực.

TÀI LIỆU

[1] E. D. Michelis, “A preliminary survey of modern yoga studies,” Asian Medicine, vol. 3, no. 1, pp. 1 – 19, 2007. [Online]. Available: https://brill.com/view/journals/asme/3/1/article-p1_3.xml

[2] K. Pilkington, G. Kirkwood, H. Rampes, and J. Richardson, “Yoga for depression: The research evidence,” Journal of Affective Disorders, vol. 89, no. 1, pp. 13–24, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165032705002570

[3] M. Verma, S. Kumawat, Y. Nakashima, and S. Raman, “Yoga-82: A new dataset for fine-grained classification of human poses,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 4472–4479.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[5] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “Yolox: Exceeding yolo series in 2021,” arXiv preprint arXiv:2107.08430, 2021.

[6] TensorFlow. (2021) Movenet: Ultra fast and accurate pose detection model. Accessed: 2025. [Online]. Available: https://www.tensorflow. org/hub/tutorials/movenet

[7] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, “Blazepose: On-device real-time body pose tracking,” 2020. [Online]. Available: https://arxiv.org/abs/2006.10204

[8] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification,” IEEE Transactions on Information Theory, vol. 13, pp. 21–27, 1967.

[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “Training algorithm for optimal margin classifiers,” in Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, 1992, pp. 144–152.

[10] L. Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5–32, 2001.

[11] P. Lu, T. Jiang, Y. Li, X. Li, K. Chen, and W. Yang, “Rtmo: Towards high-performance one-stage real-time multi-person pose estimation,” in 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 1491–1500.

[12] J. R. Quinlan, “Induction of decision trees,” Machine Learning, vol. 1, pp. 81–106, 1986.

[13] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, vol. 13-17-August-2016, 2016,pp. 785–794.

[14] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in Advances in Neural Information Processing Systems, vol. 2017- December, 2017, pp. 3146–3154.

[15] M. Dhiman, A. Sharma, and S. Singh, “Recognition and classification of human actions using 2d pose estimation and machine learning,” Journal of Computing Science and Engineering, vol. 16, no. 4, p. 199 – 210, 2022.

[16] D. Srivastav, A. Bajpai, and A. Singhal, “A temporal convolutional neural network based activity recognition model using a real-time two-dimensional single pose estimation framework,” in 2022 12th International Conference on Cloud Computing, Data Science and Engineering (Confluence), 2022, pp. 434–438.

[17] M. Xu and L. Qian, “Research on human pose recognition based on skeletal information,” 2022, Conference paper, p. 856 – 859.

[18] P. Verma, R. Srivastava, and S. K. Tripathy, “An assessment towards 2d and 3d human pose estimation and its applications to activity recognition: A review,” SN Computer Science, vol. 6, no. 2, 2025.

[19] P. Sengupta, “Health impacts of yoga and pranayama: A state-of-the-art review,” International Journal of Preventive Medicine, vol. 3, no. 7, p. 444 – 458, 2012.

[20] F. E. Fadzli, M. I. S. Adanan, A. W. Ismail, N. M. Suaib, N. A. A. Halim, and M. A. Ahmad, “Designing human pose recognition for yoga posture in web-based augmented reality,” 2024, Conference paper, p. 1543 – 1547.

[21] S. Dhanasekar and S. Sindhu, “Asana ai - yoga pose detection using pose estimation and real-time feedback system,” 2025, Conference paper.

[22] P. Shruti and R. Rekha, “A review of convolutional neural networks, its variants and applications,” 2023, Conference paper, p. 31 – 36.

[23] U. Kulkarni, Y. Diwan, P. Hegde, P. Mutnale, B. Jain, M. S M, and S. Gurlahosur, “Yoga pose detection using long-term recurrent convolutional network,” in 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), 2023, pp. 1–7.

[24] F. B. Ashraf, M. U. Islam, M. R. Kabir, and J. Uddin, “Yonet: A neural network for yoga pose classification,” SN Computer Science, vol. 4, 02 2023.

[25] V. AYDIN, “Comparison of cnn-based methods for yoga pose classification,” Turkish Journal of Engineering, vol. 8, 05 2023.

[26] J. Palanimeera and K. Ponmozhi, “Classification of yoga pose using machine learning techniques,” Materials Today: Proceedings, vol. 37, pp. 2930–2933, 2021, international Conference on Newer Trends and Innovation in Mechanical Engineering: Materials Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S2214785320365822

[27] K. G, K. S, G. K. R, and R. S, “Yoga posture detection using machine learning,” in 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2023, pp. 1–6.

[28] D. Debalaxmi, D. K. Vishwakarma, and V. Ranga, “Analyzing yoga pose recognition: A comparison of mediapipe and yolo keypoint detection with ensemble techniques,” in 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2024, pp. 1011–1017.

[29] Y. Agrawal, Y. Shah, and A. Sharma, “Implementation of machine learning technique for identification of yoga poses,” in 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), April 2020, pp. 40–43.

[30] M. J.H., C. H.K., K. Santhosh Kumar, and S. S. Prakash, “Comprehensive analysis of pose estimation and machine learning classifiers for precise yoga pose detection and classification,” Procedia Computer Science, vol. 258, pp. 3345–3356, 2025, international Conference on Machine Learning and Data Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050925016965

[31] L. Zhou, X. Meng, Z. Liu, M. Wu, Z. Gao, and P. Wang, “Human pose- based estimation, tracking and action recognition with deep learning: A survey,” 2023. [Online]. Available: https://arxiv.org/abs/2310.13039

[32] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5686–5696.

[33] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, “ViTPose: Simple vision transformer baselines for human pose estimation,” in Advances in Neural Information Processing Systems, 2022.

[34] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li, and K. Chen, “Rtmpose: Real-time multi-person pose estimation based on mmpose,” 2023. [Online]. Available: https://arxiv.org/abs/2303.07399

[35] R. Tadeusiewicz, “Neural networks: A comprehensive foundation,” Control Engineering Practice, vol. 3, 1995.

[36] G. H. John and P. Langley, “Estimating continuous distributions in bayesian classifiers,” 2013. [Online]. Available: https://arxiv.org/abs/1302.4964

[37] Y. Suryawanshi, N. Gunjal, B. Kanorewala, and K. Patil, “Yoga dataset: A resource for computer vision-based analysis of yoga asanas,” Data in Brief, vol. 48,

p. 109257, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340923003761

[38] J. Li, H. Hu, J. Li, and X. Zhao, “3d-yoga: A 3d yoga dataset for visual- based hierarchical sports action analysis,” in Computer Vision – ACCV 2022, L. Wang, J. Gall, T.-J. Chin, I. Sato, and R. Chellappa, Eds. Cham: Springer Nature Switzerland, 2023, pp. 55–71.

[39] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, “Mediapipe: A framework for building perception pipelines,” 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[40] M. Contributors, “Openmmlab pose estimation toolbox and benchmark,” https://github.com/open-mmlab/mmpose, 2020.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Mu¨ller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, “Scikit-learn: Machine learning in python,” 2018. [Online]. Available: https://arxiv.org/abs/1201.0490

PHỤ LỤC

Để thể hiện đầy đủ kết quả đánh giá hiệu suất của tổ hợp HPE+ML, đề án tiến hành đo lường các chỉ số quan trọng bao gồm CV F1-Score trên tập xác thực, Accuracy, Precision, Recall và F1-Score trên tập kiểm tra. Các kết quả chi tiết được trình bày trong Bảng PL.1 cho thấy mô hình đạt độ ổn định tốt trong quá trình cross-validation và giữ được hiệu năng nhất quán trên dữ liệu kiểm thử. Bảng được tô màu trong kết quả, trong đó xanh lá đậm và xanh nhạt lần lượt biểu thị kết quả tốt nhất và tốt nhì, còn đỏ đậm và đỏ nhạt biểu thị kết quả tệ nhất và tệ nhì. Những chỉ số này cung cấp căn cứ để nhận định chất lượng tổng thể của mô hình và khả năng khái quát hóa trong bối cảnh bài toán đặt ra.

Bảng PL.0.1: Phụ lục tổng hợp kết quả Cross-Validation và Test của mô hình HPE+ML

HPE	ML	cv_ f1score	Test acc	Test pre	Test recall	Test f1score
BLAZEPOSE_33	KNN	97.56 ± 0.32	98.1	98.33	98.11	98.16
BLAZEPOSE_33	SVM	97.88 ± 0.41	97.9	98.2	97.91	97.96
BLAZEPOSE_33	DECISIONTREE	95.87 ± 0.35	95.63	96.05	95.64	95.72
BLAZEPOSE_33	RANDOMFOREST	97.86 ± 0.25	97.7	98.02	97.71	97.76
BLAZEPOSE_33	XGBOOST	97.81 ± 0.14	97.9	98.18	97.91	97.96
BLAZEPOSE_33	LIGHTGBM	97.88 ± 0.21	97.9	98.2	97.91	97.97
BLAZEPOSE_33	GAUSSIANNB	90.61 ± 0.83	90.13	90.81	90.16	90.21
BLAZEPOSE_33	MLPCLASSIFIER	97.78 ± 0.42	97.83	98.14	97.84	97.9
BLAZEPOSE_17	KNN	97.75 ± 0.27	98.23	98.48	98.24	98.29
BLAZEPOSE_17	SVM	97.82 ± 0.30	97.93	98.26	97.94	98.01
BLAZEPOSE_17	DECISIONTREE	95.51 ± 0.23	95.23	95.65	95.24	95.31
BLAZEPOSE_17	RANDOMFOREST	97.76 ± 0.28	97.67	97.98	97.68	97.73

BLAZEPOSE_17	XGBOOST	97.76 ± 0.22	97.8	98.09	97.81	97.86
BLAZEPOSE_17	LIGHTGBM	97.85 ± 0.20	97.83	98.15	97.84	97.9
BLAZEPOSE_17	GAUSSIANNB	88.03 ± 0.67	86.83	88.93	86.83	86.97
BLAZEPOSE_17	MLPCLASSIFIER	97.63 ± 0.22	97.73	98.06	97.74	97.8
MOVENET_TFLITE	KNN	98.80 ± 0.22	98.73	98.81	98.74	98.75
MOVENET_TFLITE	SVM	98.86 ± 0.19	98.97	99.03	98.98	98.98
MOVENET_TFLITE	DECISIONTREE	97.39 ± 0.53	97.97	98.01	97.98	97.97
MOVENET_TFLITE	RANDOMFOREST	98.82 ± 0.16	98.87	98.94	98.88	98.88
MOVENET_TFLITE	XGBOOST	98.77 ± 0.15	98.9	98.98	98.91	98.91
MOVENET_TFLITE	LIGHTGBM	98.85 ± 0.18	98.9	98.97	98.91	98.91
MOVENET_TFLITE	GAUSSIANNB	96.48 ± 0.44	96.5	96.69	96.51	96.53
MOVENET_TFLITE	MLPCLASSIFIER	98.78 ± 0.19	98.93	99	98.94	98.95
MOVENET_TF2	KNN	98.72 ± 0.19	98.73	98.83	98.74	98.75
MOVENET_TF2	SVM	98.80 ± 0.22	98.73	98.82	98.74	98.75
MOVENET_TF2	DECISIONTREE	97.21 ± 0.39	97.47	97.54	97.47	97.47
MOVENET_TF2	RANDOMFOREST	98.75 ± 0.15	98.67	98.77	98.68	98.68
MOVENET_TF2	XGBOOST	98.71 ± 0.16	98.73	98.84	98.75	98.75
MOVENET_TF2	LIGHTGBM	98.77 ± 0.16	98.67	98.76	98.68	98.68
MOVENET_TF2	GAUSSIANNB	96.45 ± 0.34	96.17	96.4	96.18	96.2
MOVENET_TF2	MLPCLASSIFIER	98.72 ± 0.16	98.7	98.78	98.71	98.71
RTMO	KNN	99.67 ± 0.13	99.9	99.9	99.9	99.9
RTMO	SVM	99.85 ± 0.09	99.97	99.97	99.97	99.97
RTMO	DECISIONTREE	98.58 ± 0.18	98.7	98.72	98.7	98.7

RTMO	RANDOMFOREST	99.71 ± 0.06	99.77	99.77	99.77	99.77
RTMO	XGBOOST	99.67 ± 0.04	99.7	99.71	99.7	99.7
RTMO	LIGHTGBM	99.80 ± 0.07	99.87	99.87	99.87	99.87
RTMO	GAUSSIANNB	96.84 ± 0.09	96.57	96.61	96.59	96.57
RTMO	MLPCLASSIFIER	99.68 ± 0.12	99.7	99.71	99.7	99.7
RTMPOSE	KNN	83.18 ± 2.07	81.73	93.52	81.74	83.16
RTMPOSE	SVM	84.53 ± 0.72	81.93	93.57	81.94	83.36
RTMPOSE	DECISIONTREE	82.35 ± 0.76	80.6	91.33	80.61	81.9
RTMPOSE	RANDOMFOREST	84.43 ± 0.62	82	93.75	82.01	83.47
RTMPOSE	XGBOOST	84.54 ± 0.71	82.13	94.01	82.14	83.62
RTMPOSE	LIGHTGBM	84.69 ± 0.61	82.43	94.33	82.44	83.97
RTMPOSE	GAUSSIANNB	73.35 ± 0.37	73.43	84.19	73.45	72.37
RTMPOSE	MLPCLASSIFIER	84.36 ± 0.62	81.9	93.58	81.91	83.35
HRNET	KNN	99.72 ± 0.10	99.7	99.72	99.69	99.7
HRNET	SVM	99.70 ± 0.08	99.73	99.75	99.73	99.74
HRNET	DECISIONTREE	97.51 ± 0.40	97.3	97.33	97.27	97.29
HRNET	RANDOMFOREST	99.54 ± 0.10	99.53	99.55	99.53	99.54
HRNET	XGBOOST	99.68 ± 0.13	99.5	99.51	99.5	99.5
HRNET	LIGHTGBM	99.78 ± 0.09	99.6	99.61	99.6	99.6
HRNET	GAUSSIANNB	97.04 ± 0.39	96.7	96.85	96.69	96.74
HRNET	MLPCLASSIFIER	99.61 ± 0.10	99.67	99.68	99.67	99.67
VITPOSE	KNN	99.54 ± 0.18	99.3	99.31	99.3	99.3
VITPOSE	SVM	99.55 ± 0.17	99.27	99.28	99.26	99.27

VITPOSE	DECISIONTREE	96.38 ± 0.10	96.7	96.7	96.68	96.69
VITPOSE	RANDOMFOREST	99.31 ± 0.24	99.23	99.25	99.23	99.24
VITPOSE	XGBOOST	99.41 ± 0.10	99.43	99.45	99.43	99.43
VITPOSE	LIGHTGBM	99.64 ± 0.09	99.5	99.52	99.5	99.51
VITPOSE	GAUSSIANNB	93.56 ± 0.45	93.53	93.95	93.52	93.63
VITPOSE	MLPCLASSIFIER	99.28 ± 0.27	99.3	99.31	99.3	99.3
YOLOXPOSE	KNN	98.73 ± 0.21	98.53	98.57	98.53	98.54
YOLOXPOSE	SVM	98.58 ± 0.24	98.87	98.88	98.86	98.87
YOLOXPOSE	DECISIONTREE	96.12 ± 0.44	96.8	96.84	96.79	96.8
YOLOXPOSE	RANDOMFOREST	98.90 ± 0.25	98.53	98.56	98.53	98.54
YOLOXPOSE	XGBOOST	98.87 ± 0.29	98.57	98.59	98.57	98.57
YOLOXPOSE	LIGHTGBM	99.05 ± 0.23	98.83	98.85	98.83	98.84
YOLOXPOSE	GAUSSIANNB	89.13 ± 0.64	88.83	89.41	88.84	88.8
YOLOXPOSE	MLPCLASSIFIER	98.50 ± 0.32	98.13	98.16	98.13	98.14

BẢN CAM ĐOAN

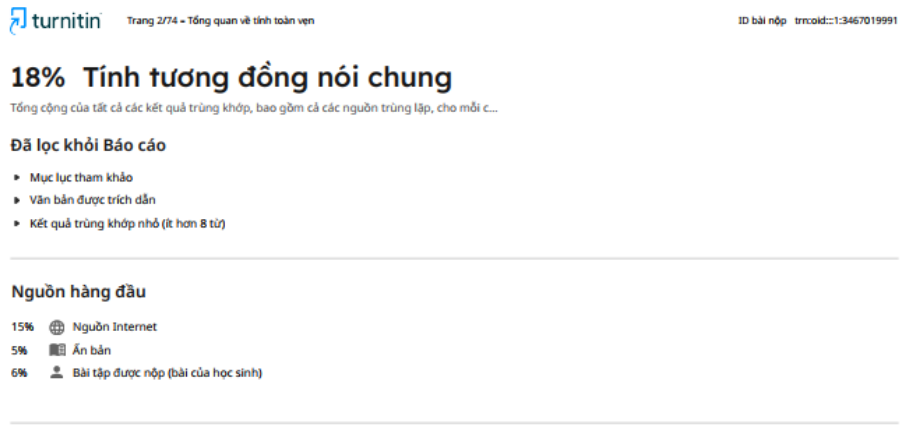
Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án qua phần mềm Turnitin một cách trung thực và đạt kết quả mức độ tương đồng 18% toàn bộ nội dung đề án. Bản đề án kiểm tra qua phần mềm là bản cứng đề án đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

Hà Nội, ngày 28 tháng 01 năm 2026

Người làm đơn

Nguyễn Quang Huy

KẾT QUẢ KIỂM TRA TRÙNG LẶP TÀI LIỆU



turnitin Trang 2/74 - Tổng quan về tính toàn vẹn ID bài nộp trn:oid:::1:3467019991

18% 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

15% Nguồn Internet
5% Ấn bản
6% Bài tập được nộp (bài của học sinh)

turnitin Trang 2/74 - Tổng quan về tính toàn vẹn ID bài nộp trn:oid:::1:3467019991

XÁC NHẬN CỦA

NGƯỜI HƯỚNG DẪN KHOA HỌC

TS. Nguyễn Duy Phương

Hà Nội, ngày 28 tháng 01 năm 2026

Người làm đơn

Nguyễn Quang Huy