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

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Hà Nội – Năm 2025

Đề án tốt nghiệp được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: TS. Nguyễn Duy Phương

Phản biện 1: TS. Đỗ Thị Bích Ngọc

Phản biện 2: TS. Nguyễn Xuân Thắng

Đề án tốt nghiệp sẽ được bảo vệ trước Hội đồng chấm đề án tốt nghiệp thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: 8 giờ 15 ngày 24 tháng 01 năm 2026

Có thể tìm hiểu đề án tốt nghiệp tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

MỞ ĐẦU

1. Lý do chọn đề tài

Yoga ngày càng phổ biến nhờ lợi ích sức khỏe [1][2], nhưng thiếu hướng dẫn trực tiếp khiến người mới dễ sai tư thế, dẫn đến chấn thương. Các phương pháp tự kiểm tra qua hình ảnh hoặc video kém hiệu quả. Vì vậy, đề tài "Ứng dụng mô hình trích xuất khung xương và học máy trong nhận diện tư thế Yoga thời gian thực" nhằm xây dựng giải pháp nhận diện tư thế chính xác, phù hợp cho tập luyện tại nhà, triển khai dễ dàng trên thiết bị phổ thông, tiết kiệm chi phí và tập trung vào tư thế cơ bản cho một người dùng.

2. Tổng quan về vấn đề nghiên cứu

Các nghiên cứu DL trên ảnh RGB (ResNet [4]) dễ bị ảnh hưởng bởi phông nền, ánh sáng, trang phục, trong khi phương pháp skeleton-based sử dụng HPE như YOLOPose [5], MoveNet [6], BlazePose [7] trích xuất khung xương và đặc trưng hình học, giảm nhiễu, giữ cấu trúc tư thế. Kết hợp skeleton với ML (KNN [8], SVM [9], Random Forest [10]) học trực tiếp từ dữ liệu khung xương, nhẹ và ổn định hơn. Dẫu vậy, vẫn tồn tại ba hạn chế chính: (1) thiếu bộ dữ liệu phù hợp cho người mới, đặc biệt trong bối cảnh Việt Nam; (2) thiếu đánh giá, so sánh toàn diện giữa các mô hình HPE và thuật toán ML để chọn ra tổ hợp tối ưu; (3) chưa có phân tích đầy đủ về tốc độ nhận diện trong điều kiện triển khai thực tế.

3. Mục đích nghiên cứu

Đề án xây dựng và đánh giá hệ thống nhận diện tư thế Yoga thời gian thực kết hợp HPE (YoloPose, RTMPose, RTMOPose [11], MMPose [16]) và ML (KNN, SVM, Decision Tree [12], Random Forest, XGBoost [13], LightGBM [14]). Nghiên cứu tối ưu hóa đặc trưng khung xương (chuẩn hóa, góc, khoảng cách) và đánh giá bằng Accuracy, Recall, F1, FPS, latency. Thực tiễn gồm tạo bộ dữ liệu tư thế cơ bản có chú thích, phát triển hệ thống cho một người dùng từ webcam, so sánh các tổ hợp HPE–ML để chọn giải pháp tối ưu, cân bằng độ chính xác, tốc độ và khả năng triển khai.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Đề án tập trung khảo sát các mô hình ước lượng tư thế người như MediaPipe Pose, YOLOXPose, RTMPose và các thuật toán ML dùng trong phân loại tư thế yoga. Đồng thời, công trình xây dựng bộ dữ liệu gồm 12 tư thế yoga cơ bản, được thực hiện bởi sinh viên PTIT – nhóm người mới tập, phù hợp với đối tượng sinh viên và người đi làm ngồi lâu.

Phạm vi nghiên cứu: Nghiên cứu giới hạn ở 12 tư thế yoga cơ bản. Dữ liệu thu thập từ khoảng 10 người (nam và nữ) trong điều kiện ánh sáng, phông nền và thiết bị khác nhau, phản ánh môi trường tập luyện tại nhà. Việc ghi hình được thực hiện bằng camera điện thoại tối thiểu 720p, đặt cách người tập 4–5m.

5. Phương pháp nghiên cứu

Nghiên cứu lý thuyết: Đề án khảo sát và tổng hợp các công trình khoa học chất lượng cao liên quan đến ước lượng dáng người (HPE), các thuật toán ML truyền thống cho bài toán phân loại, và các nghiên cứu nhận diện tư thế yoga.

Nghiên cứu thực nghiệm: Đề án thực hiện chuỗi thí nghiệm gồm xây dựng – tiền xử lý bộ dữ liệu, đánh giá các mô hình HPE, kết hợp HPE với các thuật toán ML, và kiểm thử hiệu năng toàn hệ thống trong bối cảnh 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 (HPR) là chủ đề quan trọng trong thị giác máy tính và tương tác người–máy, hướng đến việc phân loại tư thế hoặc hành động từ ảnh/video. Khác với ước lượng tư thế người (HPE), vốn chỉ xác định vị trí keypoints, HPR bao quát hơn khi tập trung vào nhận dạng tư thế/hành động hoàn chỉnh của cơ thể. Nhờ sự phát triển của học sâu và xử lý ảnh, HPR đã đạt nhiều tiến bộ, cho phép nhận diện từ các tư thế đơn giản đến chuỗi hành động phức tạp trong các ứng dụng như nhận dạng hoạt động HAR [15], giám sát an ninh [16], phân tích thể thao, chăm sóc sức khỏe và tương tác người–máy [17].

Tuy vậy, HPR vẫn gặp thách thức như che khuất bộ phận cơ thể, thay đổi ánh sáng, trang phục, góc nhìn, phông nền và yêu cầu xử lý thời gian thực đòi hỏi khả năng tính toán cao [18]. Nhìn chung, dù công nghệ đã có bước tiến mạnh mẽ và mở ra nhiều cơ hội ứng dụng trong y tế, giáo dục, thể thao và hệ thống thông minh, lĩnh vực này vẫn cần nghiên cứu sâu hơn để cải thiện độ chính xác và tính ổn định trong môi trường thực tế.

1.2. Bài toán nhận diện tư thế yoga

Yoga mang lại nhiều lợi ích về thể chất và tinh thần như tăng dẻo dai, sức mạnh cốt lõi, cân bằng và giảm căng thẳng [19]. Trong bối cảnh hiện đại sau COVID-19 và sự phát triển của AI, nhu cầu tập luyện tại nhà tăng mạnh, đặc biệt ở học sinh, sinh viên và người làm văn phòng. Tuy vậy, việc tập không có huấn luyện viên dễ dẫn đến sai tư thế, làm giảm hiệu quả hoặc gây chấn thương. Điều này thúc đẩy sự xuất hiện của các hệ thống nhận diện tư thế yoga dựa trên thị giác máy tính [20][21], giúp phân tích dáng người và đưa ra phản hồi tức thời. Tuy nhiên, bài toán gặp nhiều thách thức như sự đa dạng cơ thể, góc quay, ánh sáng, và đặc biệt là yêu cầu cân bằng giữa độ chính xác và tốc độ xử lý.

Nhận diện tư thế yoga là trường hợp đặc thù của HPR, đòi hỏi độ chính xác cao do nhiều tư thế rất giống nhau nhưng khác biệt ở các góc và hướng tay chân. Bài toán này có giá trị lớn trong huấn luyện thể chất, theo dõi sức khỏe và phục hồi chức năng. Nhờ các mô hình HPE mới nhẹ và khả năng xử lý thời gian thực, việc triển khai trên thiết bị di động hoặc thiết bị đeo trở nên khả thi, hỗ trợ người tập chỉnh sửa tư thế mà không cần huấn luyện viên trực tiếp.

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 hướng tiếp cận khác nhau, chủ yếu gồm ba nhóm chính:

(1) Phương pháp dựa trên ảnh RGB: Sử dụng ảnh hoặc chuỗi khung hình RGB làm đầu vào và xem nhận diện tư thế như một bài toán phân loại ảnh. Các phương pháp truyền thống dựa trên đặc trưng thủ công (HOG, SIFT) và bộ phân loại (SVM, k-NN) bị hạn chế bởi ánh sáng, góc nhìn và nền ảnh. Sự ra đời của học

sâu đã cải thiện đáng kể hiệu quả, khi CNN, 3D-CNN và Vision Transformer (ViT) [22] có thể tự học đặc trưng trực tiếp từ ảnh thô và đạt độ chính xác cao hơn trong bối cảnh phức tạp.

(2) Phương pháp dựa trên dữ liệu khung xương (Skeleton-based): Biểu diễn con người bằng các điểm khớp giúp mô hình tập trung vào hình học và quan hệ không gian–thời gian. Quy trình gồm ba bước: (i) phát hiện keypoints, (ii) trích xuất đặc trưng để học cấu trúc và chuyển động cơ thể, và (iii) phân loại tư thế bằng thuật toán hình học, học máy hoặc học sâu. Chuỗi xử lý này chuyển đổi dữ liệu hình ảnh thành thông tin tư thế có ý nghĩa.

(3) Mô hình lai: Kết hợp nhiều nguồn dữ liệu như RGB, depth, skeleton hoặc cảm biến (IMU) để tận dụng ưu điểm của từng dạng biểu diễn. Nhờ tích hợp đa phương thức, mô hình vừa nắm bắt ngữ cảnh từ ảnh, vừa duy trì cấu trúc động học từ skeleton, giúp tăng độ chính xác và khả năng khái quát. Tuy nhiên, mô hình lai đòi hỏi xử lý đồng bộ dữ liệu và chi phí tính toán cao hơn.

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

Một số nghiên cứu trước đây triển khai hệ thống hỗ trợ thể dục hoặc yoga dựa trên phân tích ảnh/video mà không dùng dữ liệu khung xương. Nghiên cứu [23] phát triển mô hình LRCN (CNN + LSTM) để nhận diện 6 tư thế yoga từ dữ liệu 15 người, đạt độ chính xác 81%. Công trình [24] giới thiệu YoNet cho 5 tư thế với dữ liệu huấn luyện hạn chế, so sánh với ResNet, InceptionNet, InceptionResNet và Xception, cho thấy YoNet vượt trội với độ chính xác hơn 94%. Nghiên cứu [25] kết hợp CNN và phép biến đổi Wavelet trên bộ dữ liệu 5 tư thế, đạt hơn 90% trên tập huấn luyện và 80% tập kiểm thử. Nhìn chung, các phương pháp dựa trên ảnh khai thác CNN và tiền xử lý để cải thiện chất lượng đặc trưng, nhưng vẫn nhạy với ánh sáng, nền, trang phục và góc nhìn; đồng thời khó phân tích chi tiết từng khớp, dẫn tới hạn chế trong đánh giá chất lượng động tác.

Để khắc phục các hạn chế trên, nhiều nghiên cứu chuyển sang sử dụng ước lượng tư thế (HPE) để trích xuất khung xương rồi phân loại dựa trên hình học. Nghiên cứu [26] sử dụng đặc trưng góc khớp để phân loại chuỗi Chào Mặt Trời bằng bốn mô

hình ML, trong đó KNN đạt 96%. Nghiên cứu [27] khai thác PoseNet để tạo skeleton và so sánh với mẫu huấn luyện nhằm hỗ trợ giữ tư thế đúng. Nghiên cứu [21] dùng MediaPipe để trích xuất keypoints và phân loại bốn tư thế từ Mendeley Yoga Dataset, đạt độ chính xác trên 90% và hoạt động thời gian thực. Công trình [28] kết hợp YOLOv8 và MediaPipe để lấy keypoints, sau đó thử nghiệm nhiều mô hình ML và ensemble, trong đó MediaPipe cho kết quả tốt hơn; LightGBM đạt chính xác cao nhất. Nghiên cứu [29] xây dựng bộ dữ liệu 5.500 ảnh với 10 tư thế, dùng tf-pose estimation để trích xuất skeleton, và Random Forest đạt 99,04%. Nghiên cứu [30] so sánh nhiều thuật toán HPE (MediaPipe, PoseNet, OpenPose) và nhiều bộ phân loại ML trên năm tư thế phổ biến, ghi nhận SVM, KNN và Random Forest đạt độ chính xác cao, thậm chí tới 100% ở một số tư thế.

Tuy vậy, đa số các nghiên cứu cũ chỉ thử nghiệm một vài tổ hợp giữa HPE và ML, thiếu các phân tích so sánh chéo toàn diện. Đề án này hướng tới khắc phục hạn chế đó bằng cách đánh giá định lượng nhiều tổ hợp giữa HPE và ML, đồng thời xem xét sự cân bằng giữa độ chính xác và tốc độ xử lý nhằm đề xuất giải pháp phù hợp cho các hệ thống yoga thông minh trong thực tế.

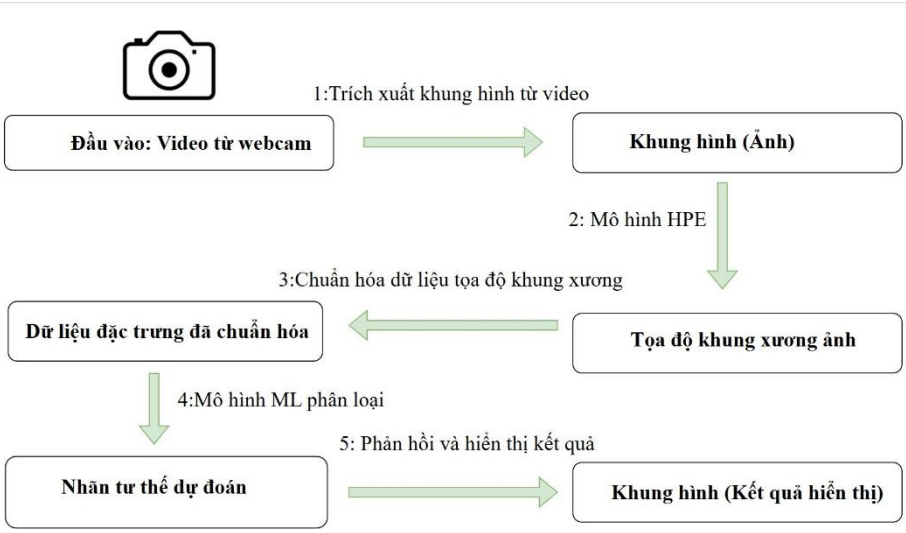
1.5. Kết luận chương 1

Chương 1 đã cung cấp tổng quan về nhận diện tư thế yoga, từ các khái niệm cơ bản về HPR đến những đặc thù của bài toán yoga. Các phương pháp và mô hình phổ biến trong trích xuất và phân loại tư thế, cả chung và riêng cho yoga, đã được giới thiệu. Thực tế cho thấy nhiều nghiên cứu chỉ tập trung vào một số tổ hợp HPE và ML cụ thể, chưa có đánh giá toàn diện để xác định tổ hợp tối ưu. Trên cơ sở đó, chương 2 sẽ phân tích chi tiết các kỹ thuật liên quan, đề xuất pipeline nghiên cứu cụ thể, giới thiệu HPE và các thuật toán ML dùng để benchmark, phục vụ xác định tổ hợp phương pháp tối ưu cho các nghiên cứu tiếp theo.

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 nhận diện tư thế Yoga thời gian thực được xây dựng theo chuỗi xử lý từ video đầu vào đến nhãn tư thế dự đoán (Hình 2.1) với bốn mô-đun chính. Đầu tiên, webcam cung cấp chuỗi khung hình (Bước 1) được đưa vào mô-đun trích xuất khung xương (HPE) để phát hiện keypoints. Tiếp theo, mô-đun xử lý và trích xuất đặc trưng chuẩn hóa tọa độ và tính toán các đặc trưng hình học quan trọng như góc, khoảng cách, tỷ lệ. Các đặc trưng này được phân loại bằng mô hình ML để dự đoán nhãn tư thế. Cuối cùng, kết quả được trực quan hóa và phản hồi trên giao diện người dùng, giúp hệ thống nhận diện tư thế ổn định trong thời gian thực.



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

2.2. Mô đun trích xuất khung xương

2.2.1. Các mô hình được áp dụng

Các hướng tiếp cận trong Human Pose Estimation [31] có thể phân nhóm theo bốn tiêu chí: (1) Chiều dữ liệu: 2D pose estimation xác định vị trí khớp trên mặt phẳng ảnh, 3D pose estimation tái tạo cấu trúc không gian chính xác hơn nhưng yêu cầu dữ liệu và tính toán lớn; (2) Đối tượng xử lý: single-person hay multi-person, với độ

phức tạp cao hơn cho bài toán đa người; (3) Cấu trúc mô hình: top-down (phát hiện người trước ước lượng tư thế), bottom-up (ước lượng toàn bộ keypoints trước khi nhóm) và one-stage (kết hợp cả hai bước); (4) Dữ liệu đầu vào: ảnh tĩnh (image-based) hoặc chuỗi video (video-based), trong đó video cung cấp thêm thông tin chuyển động giúp tăng ổn định dự đoán.

Đề án ưu tiên sử dụng các mô hình ước lượng tư thế 2D cho nhận dạng Yoga thời gian thực tại nhà :

- HRNet [32] duy trì biểu diễn độ phân giải cao xuyên suốt, kết hợp đa nhánh và hợp nhất đa tỉ lệ, đạt hiệu quả trên COCO, MPII, PoseTrack.
- BlazePose [7] nhẹ, nhanh, dự đoán 33 keypoints, bao gồm detector và pose tracker, tối ưu cho thiết bị di động, ứng dụng theo dõi thể hình, yoga, AR.
- MoveNet [6] dự đoán 17 keypoints, phiên bản Lightning cho tốc độ, Thunder cho chính xác, sử dụng bottom-up với MobileNetV2 + FPN, đạt >30 FPS.
- YOLOX [5] sử dụng cơ chế anchor-free, decoupled head, SimOTA, hiệu năng cao, hỗ trợ ONNX/TensorRT/NCNN/OpenVINO.
- ViTPose [33] dựa trên transformer thuần túy, bộ giải mã nhẹ, linh hoạt, đạt 80.9 AP trên COCO test-dev.
- RTMPose [34] top-down, sử dụng CSPNeXt backbone, SimCC output, tích hợp Pose NMS và lọc mượt, tăng tính ổn định thời gian thực.
- RTMO [11] one-stage đa người, dual 1-D heatmaps, dynamic coordinate classifier, coordinate bins, tối ưu tốc độ và độ chính xác cho MPPE.

Những mô hình này cung cấp nền tảng mạnh mẽ để triển khai nhận diện tư thế Yoga thời gian thực, cân bằng giữa độ chính xác và hiệu năng.

2.2.2. Nhận diện và xử lý keypoints

Trong các mô hình ước lượng tư thế, số lượng và loại keypoints phụ thuộc vào dataset huấn luyện. Để thuận tiện cho xử lý và so sánh, kết quả thường được quy về chuẩn COCO với 17 điểm chính như mũi, mắt, tai, vai, khuỷu tay, cổ tay, hông, gối và mắt cá chân. Ví dụ, MoveNet trả về mảng [số người, 17, 3] gồm tọa độ (x, y) và độ tin cậy, cùng với skeleton COCO dùng cho hiển thị hoặc phân tích chuyển động.

Ngược lại, BlazePose cung cấp 33 keypoints chi tiết hơn, gồm tọa độ (x, y, z) và độ tin cậy, giúp phân tích chuyển động chính xác hơn.

Việc chuẩn hóa về COCO giúp so sánh và kết hợp dữ liệu từ nhiều mô hình, trong khi bộ mở rộng như BlazePose vẫn có thể dùng khi cần chi tiết cao. Trong quá trình ước lượng, các điểm keypoint có thể có độ tin cậy thấp do hình ảnh kém, góc nhìn khó hoặc tư thế phức tạp. Đề án áp dụng ngưỡng trung bình 0.3 cho điểm tin cậy: nếu trung bình keypoints < 0.3, toàn bộ vector đặc trưng của ảnh được gán 0, coi như không phát hiện được tư thế. Cách này đảm bảo tính nhất quán và công bằng khi so sánh hiệu năng giữa các mô hình HPE trước khi đưa dữ liệu vào mô-đun tiếp theo.

2.3. Mô đun xử lý và trích xuất đặc trưng

Đề án áp dụng chuẩn hóa dựa trên **tâm hông (hip center)** để dữ liệu keypoints bất biến với vị trí, tỷ lệ cơ thể và khoảng cách tới camera. Tâm vai (shoulder center) và kích thước thân (torso size) được sử dụng để xác định thang đo chuẩn hóa, kết hợp hệ số 2.5. Mỗi keypoint sau đó được dịch chuyển sao cho tâm hông là gốc tọa độ và chia cho thang đo, giữ nguyên độ tin cậy, giúp nâng cao độ ổn định và khả năng tổng quát của mô hình.

Các bước chuẩn hóa keypoints:

1. Nhập dữ liệu: Keypoints [N × 3] gồm (x, y, confidence).
2. Tâm hông: Trung điểm hông trái và phải → hip center.
3. Tâm vai & kích thước thân: Trung điểm vai trái và phải → shoulder center; Torso size = khoảng cách shoulder center – hip center.
4. Thang đo chuẩn hóa: scale = max(torso size × 2.5, khoảng cách lớn nhất từ keypoints tới hip center).
5. Chuẩn hóa keypoints: Dịch chuyển theo hip center, chia cho scale, giữ confidence.
6. Xuất dữ liệu: Danh sách normalized_keypoints sẵn sàng cho phân loại tư thế và phân tích chuyển động.

2.4. Mô đun phân loại tư thế

Mô đun phân loại tư thế chịu trách nhiệm chuyển đổi vector đặc trưng từ keypoints thành nhãn tư thế. Các vector tổng hợp thông tin về vị trí khớp, hình dạng cơ thể và mối quan hệ không gian, giúp phân loại chính xác và ổn định. Trong nghiên cứu, các mô hình học máy phù hợp được lựa chọn để cân bằng giữa độ chính xác, tốc độ và khả năng triển khai thời gian thực, đặc biệt trên thiết bị di động. Việc so sánh hiệu năng giữa các mô hình cũng giúp đánh giá ưu nhược điểm và lựa chọn mô hình tối ưu.

Các mô hình có giám sát được đánh giá gồm:

- K-Nearest Neighbors (KNN) [8]: Phân loại dựa trên nhãn của K điểm gần nhất; đơn giản, phù hợp dữ liệu nhỏ–vừa nhưng chậm với tập lớn và nhạy nhiễu.
- Decision Tree (DT) [12]: Dữ liệu được chia theo điều kiện, dễ hiểu, nhưng cây đơn dễ overfitting.
- Support Vector Machine (SVM) [9]: Tối ưu biên phân tách, tổng quát tốt với dữ liệu nhỏ, nhưng chậm với tập lớn và cần chọn kernel phù hợp.
- Multi-Layer Perceptron (MLP) [35]: Mạng nơ-ron nhiều lớp, giải quyết dữ liệu phi tuyến, tính toán cao hơn.
- Random Forest (RF) [10]: Tập hợp nhiều cây quyết định, kiểm soát overfitting, tăng độ chính xác, nhưng chi phí tính toán cao.
- Gaussian Naive Bayes (GaussianNB) [36]: Dựa trên Bayes, nhanh, hiệu quả với dữ liệu chuẩn, giảm hiệu năng nếu đặc trưng liên quan.
- XGBoost [13]: Boosting từ cây quyết định, độ chính xác cao, hỗ trợ dữ liệu lớn, nhưng cần nhiều tài nguyên và tinh chỉnh tham số.
- LightGBM [14]: Gradient boosting tối ưu tốc độ và bộ nhớ, hiệu quả với dữ liệu lớn và nhiều đặc trưng.

Các mô hình này giúp so sánh hiệu năng, chọn ra giải pháp cân bằng giữa độ chính xác, tốc độ và khả năng mở rộng cho nhận diện tư thế Yoga thời gian thực.

2.5. Mô đun phản hồi và hiển thị

Mô đun phản hồi và hiển thị trực quan hóa kết quả nhận diện tư thế bằng cách hiển thị nhãn tư thế và các điểm khớp lên hình ảnh, giúp người tập điều chỉnh kịp thời. Hình ảnh được resize về kích thước gốc để đảm bảo tỷ lệ đúng, sau đó vẽ các điểm khớp và nối thành skeleton. Nhãn tư thế được hiển thị trực tiếp trên hình, tạo phản hồi trực quan mà vẫn duy trì hiệu suất thời gian thực, phù hợp cho các ứng dụng Yoga, thể thao trên thiết bị di động hoặc web.

Các bước thực hiện:

1. Chuẩn bị dữ liệu: Nhận ảnh gốc và danh sách keypoints; kiểm tra hợp lệ, nếu trống trả về None.
2. Resize ảnh: Lấy chiều cao và rộng, nhân tọa độ keypoints chuẩn hóa để chuyển về tọa độ pixel thực tế.
3. Vẽ keypoints: Duyệt từng điểm, vẽ hình tròn tại tọa độ (x, y) với độ chính xác cao.
4. Vẽ skeleton: Duyệt danh sách cặp keypoints, vẽ đường nối tạo khung xương.
5. Hiển thị nhãn tư thế: Lấy nhãn từ kết quả phân loại và vẽ lên ảnh tại vị trí cố định.
6. Trả về hình ảnh: Hình ảnh cuối cùng có keypoints, skeleton và nhãn, sẵn sàng đưa lên giao diện người dùng cho phản hồi trực quan và thời gian thực.

2.6. Kết luận chương 2

Chương 2 đã trình bày hệ thống nhận diện tư thế Yoga thời gian thực theo kiến trúc mô đun. Mô đun trích xuất khung xương ước lượng keypoints chính xác, làm cơ sở cho mô đun xử lý và trích xuất đặc trưng, chuyển đổi thành vector chuẩn hóa phục vụ phân loại tư thế. Mô đun phân loại sử dụng các mô hình ML cân bằng giữa tốc độ và độ chính xác, phù hợp thời gian thực. Cuối cùng, mô đun phản hồi và hiển thị trực quan hóa skeleton và nhãn tư thế, giúp người tập điều chỉnh kịp thời. Từ thiết kế này, Chương 3 sẽ trình bày các thực nghiệm và đánh giá hệ thống, bao gồm bộ dữ liệu, môi trường cài đặt và kết quả đạt được.

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1. Bộ dữ liệu thực nghiệm

Bộ dữ liệu đóng vai trò then chốt trong phát triển và đánh giá mô hình nhận diện tư thế Yoga. Đề án khảo sát các bộ dữ liệu trước [3][24][37][38] dựa trên thiết bị ghi hình, số lượng tư thế và người tham gia, tổng mẫu, sự đa dạng tư thế và điều kiện môi trường. Nhằm phục vụ nhu cầu tập luyện tại nhà, đề án xây dựng bộ dữ liệu mới gồm 12 tư thế cơ bản (Cái cây, Chó úp mặt, Chiến binh 1 & 2, Cúi người về trước, Em bé, Ngọn núi 1 & 2, Tấm ván, Tấm ván thấp, Tam giác, Cây cầu) từ 10 tình nguyện viên (5 nam, 5 nữ). Mỗi người thực hiện đầy đủ 12 tư thế dưới nhiều điều kiện thiết bị, cảnh nền và mức độ nhiễu khác nhau, đảm bảo tính phong phú và độ khó cho huấn luyện và đánh giá. Hình 3.1 minh họa chi tiết 12 nhãn tư thế.



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

Để thu thập dữ liệu, mỗi video ghi lại một người thực hiện một tư thế riêng biệt, đặt camera cách 4–5 mét, chuẩn HD 720p, 30 FPS, dài trên 10 giây. Mỗi nhãn gồm 10 video (2 ngang, 8 dọc). Trong tiền xử lý, khung hình được trích xuất, cắt thành hình vuông, chuẩn hóa kích thước 640 × 640 × 3 để phù hợp với các mô hình

HPE. Để tăng độ phong phú, áp dụng các kỹ thuật tăng cường như lật ngang, xoay ±10°, dịch chuyển và biến dạng nhẹ. Sau cùng, bộ dữ liệu gồm **13.928 ảnh**, chi tiết từng tư thế được trình bày trong Bảng 3.1.

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

Thực nghiệm được triển khai trên hệ thống với CPU Intel Core i5-13400F (10 nhân, 16 luồng, xung nhịp 2,5 GHz) và RAM 32 GB, sử dụng môi trường ảo Anaconda để quản lý các gói thư viện và phiên bản Python. Trong quá trình thực nghiệm, các thư viện chính được sử dụng bao gồm matplotlib, mediapipe, mmcv, mmdet, mmengine, mmpose[40], mmpretrain, numpy, opencv-python, openmim, pandas, scikit-learn, tensorflow, tensorflow_hub, torch và torchvision.

3.2.2. Thiết lập mô hình HPE

Trong nghiên cứu này, tám mô hình ước lượng tư thế người (HPE) phiên bản nhẹ nhất được lựa chọn để so sánh hiệu năng về tốc độ và độ chính xác, phù hợp với ứng dụng nhận diện tư thế Yoga thời gian thực. Nhóm Mediapipe gồm BlazePose33 và BlazePose17; nhóm TensorFlow gồm MoveNetTF2 (single-pose-lightning) và MoveNetTFLite (single-pose-lightning float16); nhóm MMPose gồm HRNet-W32 (td-hm_hrnet-w32_fp16-8xb64-210e_coco-256x1), YOLOXPose-tiny (yoloxpose_tiny_4xb64-300e_coco-416), RTMPose-tiny (rtmpose-t_8xb256-420e_coco-256x192), ViTPose-small (td-hm_ViTPose-small_8xb64-210e_coco-256x1) và RTMO-small (rtmo-s_8xb32-600e_coco-640x640). Việc lựa chọn đa dạng

mô hình giúp đánh giá toàn diện từ các mô hình nhẹ cho thiết bị cá nhân đến các kiến trúc tiên tiến cho hiệu năng cao.

3.2.3. Thiết lập đào tạo mô hình ML

Trong quá trình đào tạo mô hình học máy, 3.000 mẫu được tách riêng làm tập kiểm thử (test set), còn lại được sử dụng cho huấn luyện và xác thực. Việc chia dữ liệu áp dụng stratified sampling để giữ nguyên tỉ lệ nhãn giữa các tập, với hạt giống ngẫu nhiên cố định là 42 để đảm bảo kết quả có thể tái lập. Tám thuật toán học máy từ scikit-learn [41](Decision Tree [12], GaussianNB [36], KNN [8], LightGBM [14], MLPClassifier [35], Random Forest [10], SVM [9], XGBoost [13]) được triển khai để đánh giá hiệu quả của các đặc trưng khung xương (pose features) từ các mô hình HPE. Quá trình tối ưu siêu tham số thực hiện với 10 vòng lặp tìm kiếm và 5-fold cross-validation, sử dụng Accuracy làm thang đo chính. Mô hình tốt nhất (best estimator) sau tối ưu được dùng để đánh giá hiệu năng tổng thể trên tập kiểm thử độc lập.

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) 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 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.

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

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 (Độ chính xác tổng thể):

$$\text{Inference Time} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

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

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

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

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

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

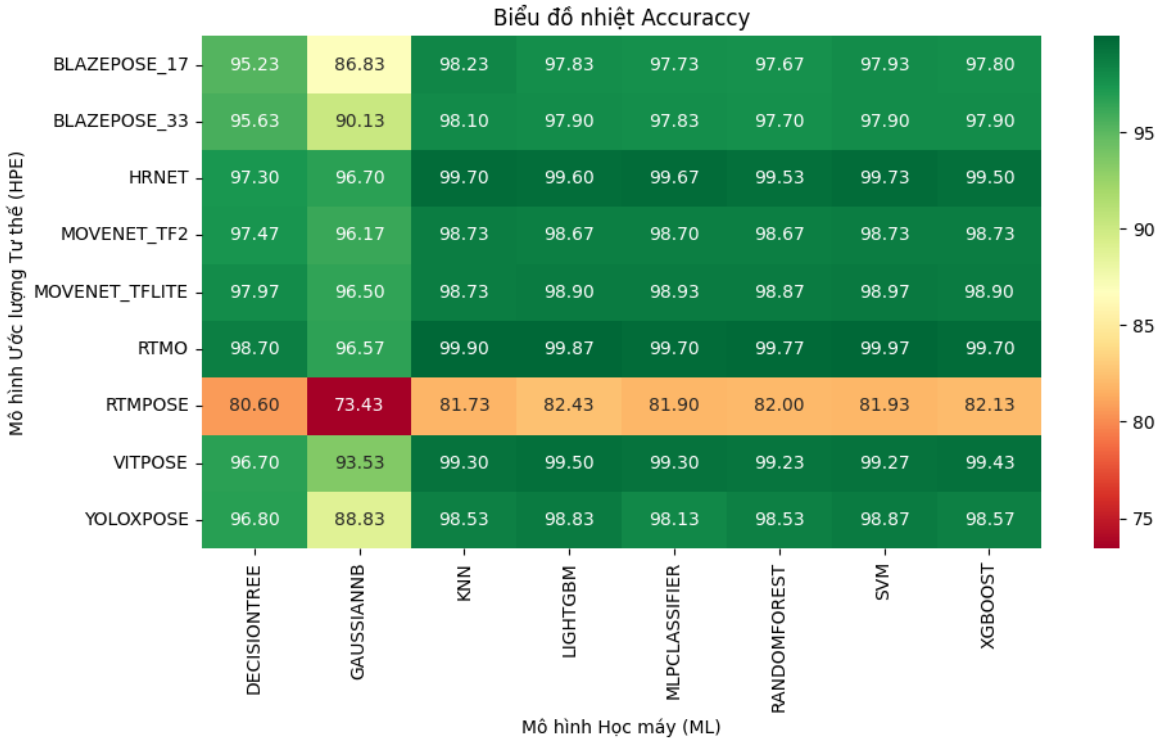
Bảng 3.2 trình bày kết quả phân tích hiệu năng của chín mô hình ước lượng tư thế người (HPE) trên cùng tập dữ liệu thử nghiệm. Có thể thấy, hầu hết các mô hình đều đạt tỷ lệ phát hiện cao trên 97%, cho thấy khả năng nhận diện tư thế ổn định trong nhiều điều kiện khác nhau.

Bảng 3.2: So sánh kết quả thực nghiệm của các mô hình HPE

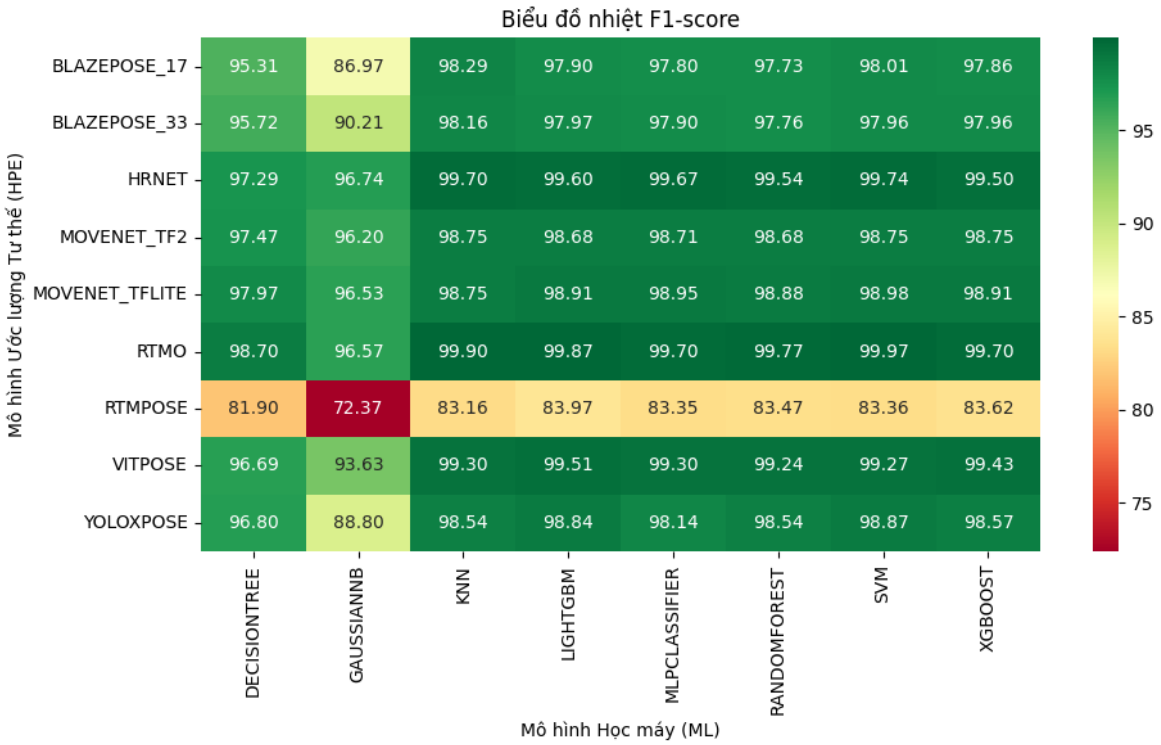
Mô hình HPE	Detection Rate (%)	Inference Time (ms)
BLAZEPOSE_33	97.83	36.34
BLAZEPOSE_17	97.83	32.15
MOVENET_TFLITE	97.49	10.56
MOVENET_TF2	97.41	80.36
RTMO	99.99	268.28
RTMPOSE	75.53	54.07
HRNET	99.98	389.03
VITPOSE	99.84	189.27
YOLOXPOSE	99.94	131.58

Kết quả cũng cho thấy MoveNet_TFLite cân bằng tốt giữa tốc độ (10.56 ms) và độ chính xác (97.49%), phù hợp ứng dụng thời gian thực, trong khi RTMO, HRNet và ViTPose đạt độ chính xác cao nhưng chậm hơn, thích hợp cho hệ thống không giới hạn tài nguyên.

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. Kết quả cho thấy RTMO, HRNet và ViTPose là những mô hình HPE hiện đại hiệu suất cao, trong khi nhóm BlazePose và YOLOXPose đạt mức trung bình. MoveNet_TFLite được chọn làm mô hình chính nhờ cân bằng giữa độ chính xác, tốc độ và kích thước, kết hợp với SVM đạt F1-score 98.98% và Accuracy 98.97%, ổn định và phù hợp cho ứng dụng thực tế.

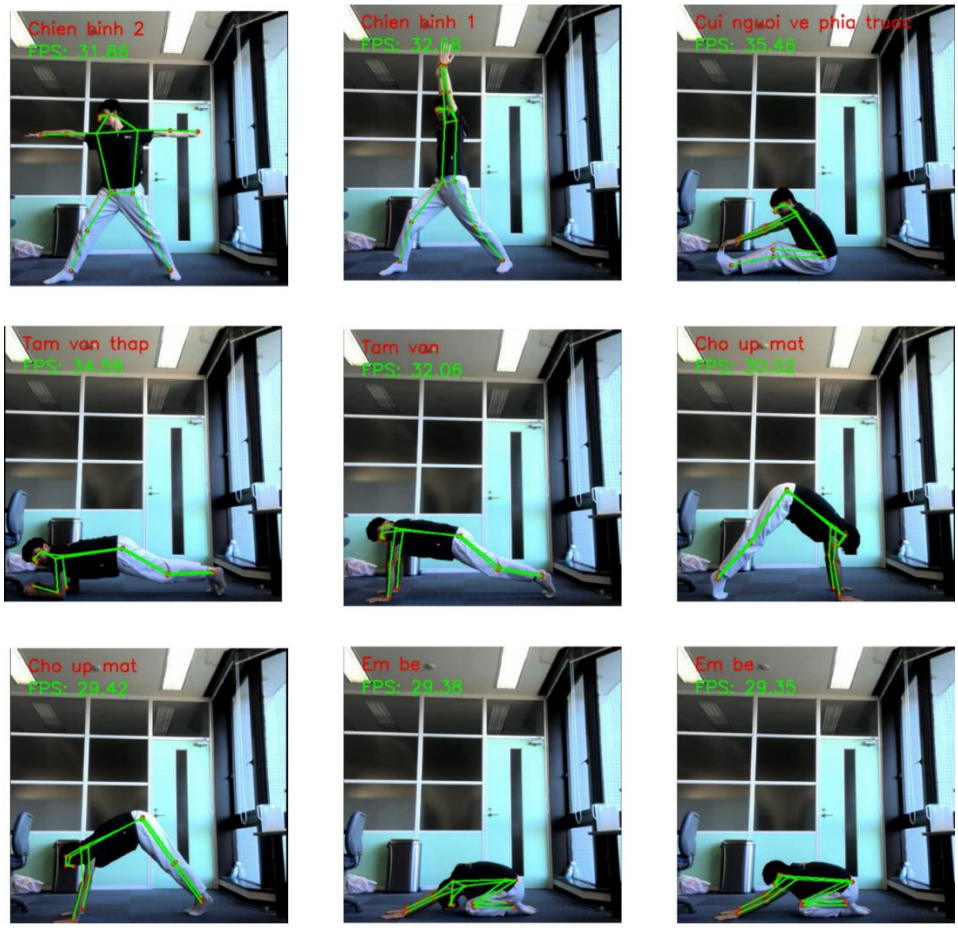
3.3.3. Đánh giá hiệu suất thời gian thực

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.



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.

Kết quả thực nghiệm cho thấy khi sử dụng tổ hợp MoveNet TFLite + SVM, hệ thống nhận diện tư thế Yoga đạt 30–31 FPS, đảm bảo xử lý thời gian thực mượt mà cùng độ chính xác cao. Hệ thống duy trì khả năng phân loại ổn định ngay cả khi một số keypoint bị thiếu, tư thế lệch hướng hoặc khoảng cách với camera thay đổi, cho thấy tính ổn định, thích ứng và chịu lỗi tốt. Các điểm khớp và nhãn tư thế được hiển thị rõ ràng, thuận tiện cho người tập quan sát và điều chỉnh động tác, đồng thời cung cấp phản hồi trực quan hiệu quả trong luyện tập Yoga 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

Chương 3 trình bày quá trình thực nghiệm và đánh giá hiệu năng hệ thống nhận diện tư thế Yoga, bao gồm các mô hình HPE và ML. Bộ dữ liệu và môi trường thực nghiệm được thiết lập để đảm bảo khách quan và tái lập kết quả. Các mô hình HPE được đánh giá theo tỷ lệ phát hiện và thời gian suy luận: HRNet và RTMO đạt độ chính xác cao nhất nhưng chậm, trong khi MoveNetTFLite nổi bật với tốc độ 12.61 ms và độ chính xác 97.49%. Khi kết hợp với các bộ phân loại ML, hiệu suất tổng thể phụ thuộc vào cả hai thành phần, trong đó tổ hợp MoveNetTFLite + SVM đạt độ chính xác 98.97%, F1-score 98.98% và tốc độ 30–31 FPS, đáp ứng yêu cầu thời gian thực. Kết quả chứng minh tính khả thi của việc kết hợp HPE nhẹ với học máy để triển khai hệ thống nhận diện tư thế Yoga thực tế.

KẾT LUẬN

Bài toán nhận diện tư thế Yoga ứng dụng trong tập luyện tại nhà giúp phòng tránh chấn thương và nâng cao hiệu quả luyện tập. Đề án đánh giá các mô hình HPE kết hợp ML và xác định MoveNet TFLite + SVM là tổ hợp tối ưu, vừa chính xác vừa xử lý nhanh trên tập dữ liệu YogaSinglePose-PTIT, đáp ứng yêu cầu thời gian thực. Hướng phát triển tiếp theo bao gồm mở rộng nhận diện đa người, tăng độ ổn định, tích hợp phản hồi thông minh và áp dụng Explainable AI để minh bạch quyết định của mô hình, đồng thời triển khai trên thiết bị biên và ứng dụng web/app di động, tạo hệ thống nhận diện tư thế Yoga thông minh, nhanh và thân thiện.

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.

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