**Abstract**

Trong bối cảnh dịch bệnh COVID-19, việc tập thể dục đã trải qua những sự biến đổi đáng kể. Khi đó việc tập luyện thể dục tại nhà đã trở thành lựa chọn phổ biến trong khoảng thời gian hạn chế ra khỏi nhà. Tuy nhiên, việc thực hiện các bài tập một cách đúng có thể gặp khó khăn đối với những người mới bắt đầu nếu thiếu sự hướng dẫn từ người huấn luyện chuyên nghiệp, điều này tăng nguy cơ chấn thương không mong muốn. Do vậy, cần phải có các hệ thống hỗ trợ giám sát hiệu suất tập thể dục để giúp ngăn ngừa những chấn thương.

Trong nghiên cứu này, chúng tôi trình bày một phương pháp tiếp cận tiên tiến để phát hiện chính xác các tư thế yoga bằng cách sử dụng kỹ thuật ước tính tư thế với sự hỗ trợ của thư viện OpenCV và mediapipe. Để đảm bảo việc đo và điều chỉnh tư thế cơ thể một cách chính xác trong quá trình tập luyện, giải pháp mà chúng tôi đề xuất sẽ tổ hợp các thuật toán dựa trên mô hình máy học cùng với thị giác máy tính.

Thông qua việc kết hợp sức mạnh của các kỹ thuật máy học và thị giác máy tính, chúng tôi hy vọng rằng phương pháp này sẽ cung cấp cho những người tập luyện một công cụ hữu ích để giúp họ thực hiện các tư thế yoga một cách chính xác và an toàn.

**Chương I: Tổng quan**

* 1. Giới thiệu

Một trong những lĩnh vực được các nhà khoa học nghiên cứu và quan tâm hàng đầu trong nhiều năm gần đây được biết đến như nhận diện vật thể, con người thông qua thị giác máy tính do khả năng ứng dụng linh hoạt. Nhận diện hoạt động, tư thế của con người tác động đến các lĩnh vực khác nhau như tương tác người-máy, robot, trò chơi thực tế ảo, giám sát thông qua video, phân tích hoạt động trong thể thao, theo dõi sức khỏe và một số lĩnh vực liên quan. Gần đây, các hệ thống nhận dạng có những bước tiến nhảy vọt, điển hình nhất là công nghệ video 3D áp dụng trong bóng đá và có thể thay thế vị trí của công nghệ VAR ([**Công nghệ video 3D mới đe dọa sự tồn tại của VAR - VnExpress Số hóa**](https://vnexpress.net/cong-nghe-video-3d-moi-de-doa-su-ton-tai-cua-var-3771689.html)). Mặt khác việc sử dụng Machine Learning nhận dạng các tư thế trong yoga vẫn đang trong quá trình nghiên cứu và phát triển. Dù đã có một số nỗ lực trong việc phát triển nhưng vẫn cần thêm nhiều nghiên cứu và phát triển để cải thiện độ chính xác và độ tin cậy của các mô hình. Đặc biệt, việc xử lý các tư thế yoga phức tạp và có sự thay đổi trong môi trường, ánh sáng, góc chụp và phông nền nơi hình ảnh được chụp là thách thức lớn cho các nhà nghiên cứu và nhà phát triển.

Yoga không chỉ là một hình thức rèn luyện thể chất mà còn là một bộ môn phát triển tâm hồn và tinh thần ([Nguồn gốc của yoga và các bài tập tốt cho sức khỏe – INDIANFOODS - Thực Phẩm Ấn Độ](https://indianfoods.com.vn/blogs/van-hoa-an-do/nguon-goc-cua-yoga-va-cac-bai-tap-tot-cho-suc-khoe#nguon_goc_cua_bo_mon_yoga)). Bộ môn này giúp cải thiện sự tập trung, tâm trí được thư giãn, giảm căng thẳng, và nhờ đó sức khỏe sẽ được cải thiện. Hiện tại yoga đang trở nên phổ biến trên toàn thế giới. Ý nghĩa ngày càng tăng của yoga trong y học là do tác dụng chữa bệnh phi thường trong nhiều bệnh lý ảnh hưởng đến cơ thể con người như các vấn đề về hô hấp, bệnh tim, các vấn đề về cơ xương và ứng dụng học sâu trong chăm sóc sức khỏe. Tuy nhiên, tồn tại khoảng cách giữa sự hiểu biết và nhận thức của một số người về lợi ích của yoga, điều này dẫn đến hàng loạt các vấn đề sức khỏe. Bên cạnh đó có thể hạn chế chúng bằng cách áp dụng yoga như một phần trong thói quen hằng ngày. Áp dụng các công nghệ ngày càng tiến bộ mang đến khả năng giải quyết vấn đề, tăng khả năng tương tác của người dùng đến bài học thông qua công cụ hỗ trợ tự học theo thời gian thực, có thể phát hiện được các động tác yoga khác nhau.

* 1. Mục tiêu

Mục tiêu chính của đề tài là triển khai mô hình máy học để ước tính và nhận diện được các động tác yoga. Tạo các điểm (point) dựa theo các khớp xương của con người (chú thích) trong hình ảnh, video thu thập thông qua camera của thiết bị và phản hồi kết quả động tác của người thực hiện. Trong đề tài này nhóm chúng em nghiên cứu một số động tác yoga tiêu biểu như Child's Pose, Bound Angle Pose, Goddess Pose, Plank Pose, Tree Pose, Warrior II Pose.

* 1. Tình hình nghiên cứu

Về ước tính và xác minh tư thế Yoga đã trở thành một lĩnh vực quan trọng và đang nhận được sự quan tâm rộng rãi từ cộng đồng nghiên cứu và cả người học Yoga. Các nỗ lực nghiên cứu trong lĩnh vực này tập trung vào việc phát triển các phương pháp và công cụ để tự động ước tính và xác minh độ chính xác của các tư thế Yoga, mang lại nhiều lợi ích cho cả giảng dạy và tập luyện tại nhà.

* + 1. Phương pháp nghiên cứu  
       PP thu thập số liệu:
* Sử dụng phương pháp thu thập tài liệu: Nhóm em tìm những tài liệu liên quan đến chủ đề của mình để xây dựng cơ sở lý thuyết

PP thực nghiệm:

* Thiết kế và triển khai hệ thống: Từ những thông tin đã được thu thập, nhóm em tiến hành chọn lọc, tổng hợp và cài đặt phần mềm, thư viện để thực hiện đề tài.
* Tiến hành thử nghiệm: Nhóm em thực hiện các bài kiểm tra, chạy thử nghiệm mô hình và sửa lỗi.

PP phân tích:

* Đánh giá hiệu quả: Tôi tiến hành đánh giá hiệu quả dựa trên độ chính xác của mô hình sau khi được huấn luyện.
* Phân tích dữ liệu: Tôi phân tích dữ liệu dựa vào thời gian thực nghiệm khi thay đổi các động tác và thay đổi đối tượng thực hiện động tác.
  1. Bố cục nội dung

Trong bài báo cáo đồ án 2 bao gồm 5 chương:

* Chương 1: Tổng quan
* Chương 2: Cơ sở lí thuyết
* Chương 3: Thiết kế hệ thống
* Chương 4: Kết quả
* Chương 5: Kết luận và hướng phát triển

**Chương II: Cơ sở lí thuyết**

* 1. Máy học (Machine Learning)
  2. Các mô hình máy học

LogisticRegression  
RidgeClassifier  
RandomForestClassifier  
GradientBoostingClassifier

* 1. Thư viện cần thiết

**Mediapipe**   
Mediapipe là một thư viện phần mềm được phát triển bởi Google, được thiết kế để hỗ trợ xây dựng ứng dụng liên quan đến xử lý hình ảnh và video, thường dựa trên dữ liệu hình ảnh hoặc video từ máy ảnh hoặc nguồn media khác.

Thư viện Mediapipe cung cấp các công cụ và khung làm việc để nhận diện và theo dõi các đối tượng, hình ảnh, cử chỉ và các khía cạnh khác trong thời gian thực. Nó giúp đơn giản hóa việc phát triển các ứng dụng về xử lý hình ảnh, như phát hiện khuôn mặt, theo dõi cử chỉ tay, xác định vị trí khung xương của cơ thể, và nhiều nhiệm vụ xử lý hình ảnh và video khác.

Mediapipe cung cấp các công cụ API, các mô hình mạng neural pre-trained và các thành phần xử lý hình ảnh chất lượng cao, giúp các nhà phát triển tập trung vào việc xây dựng ứng dụng thú vị và độc đáo mà không cần phải đầu tư quá nhiều vào việc triển khai và tinh chỉnh các phương pháp xử lý hình ảnh cơ bản.

**Opencv**OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở được phát triển để hỗ trợ xử lý hình ảnh và thị giác máy tính. Thư viện này được thiết kế để cung cấp các công cụ và chức năng cho việc xử lý hình ảnh, từ việc đọc và ghi ảnh, biến đổi hình ảnh, phát hiện đối tượng, theo dõi vị trí, và nhiều tác vụ xử lý hình ảnh khác.

Các chức năng chính của OpenCV bao gồm:

1. Xử lý hình ảnh cơ bản: Đọc, ghi và biến đổi hình ảnh, điều chỉnh độ tương phản, độ sáng, lọc hình ảnh và các phép biến đổi cơ bản khác.
2. Nhận diện đối tượng: OpenCV cung cấp các thuật toán phổ biến để phát hiện và nhận diện đối tượng trong hình ảnh, như Haar cascades, HOG (Histogram of Oriented Gradients), và Deep Learning-based object detection.
3. Theo dõi vị trí và chuyển động: Theo dõi vị trí của đối tượng trong chuỗi các khung hình, phát hiện chuyển động, và ước tính vận tốc.
4. Thị giác máy tính: Xử lý và phân tích hình ảnh để trích xuất thông tin hữu ích, như biểu đồ, đặc trưng, hoặc các đối tượng quan trọng.
5. Xử lý video: Xử lý và biến đổi video, áp dụng hiệu ứng hình ảnh và các phép biến đổi khác.

**Csv**CSV (Comma-Separated Values) là một định dạng lưu trữ dữ liệu phổ biến được sử dụng để biểu diễn thông tin trong một bảng dữ liệu, trong đó các giá trị được phân tách bằng dấu phẩy (,) hoặc một ký tự phân tách khác. Định dạng CSV thường được sử dụng để trao đổi và lưu trữ dữ liệu tabular (theo dạng bảng) giữa các ứng dụng và hệ thống khác nhau.

Mỗi hàng trong tệp CSV thường tương ứng với một bản ghi hoặc một dòng trong bảng dữ liệu, và các cột tương ứng với các thuộc tính hoặc trường dữ liệu. Các giá trị trong các ô được phân cách bằng dấu phẩy hoặc ký tự phân tách khác, như dấu chấm phẩy (;) hoặc dấu tab (\t).

**Os**

Trong Python, thư viện "os" là một thư viện chuẩn được cung cấp sẵn để tương tác với hệ điều hành và thực hiện các hoạt động liên quan đến hệ thống tệp và thư mục. Thư viện "os" cung cấp các chức năng để thực hiện các tác vụ như quản lý thư mục, tạo và xóa tệp, kiểm tra quyền truy cập, và nhiều hoạt động khác.

**Numpy**

NumPy (Numerical Python) là một thư viện mã nguồn mở trong Python, được sử dụng rộng rãi để thực hiện các phép toán số học và xử lý dữ liệu trong mảng đa chiều. Thư viện này giúp cho việc làm việc với dữ liệu số và phân tích số trở nên dễ dàng và hiệu quả hơn.

Các đặc điểm chính của NumPy:

Mảng Đa Chiều (Arrays): NumPy cung cấp cấu trúc dữ liệu mảng đa chiều, cho phép bạn làm việc với các tập dữ liệu nhiều chiều (ví dụ: ma trận) một cách hiệu quả. Mảng NumPy có thể có kích thước cố định và không thay đổi sau khi khởi tạo.

Phép Toán Số Học: NumPy cung cấp các hàm số học cơ bản (cộng, trừ, nhân, chia, lũy thừa) cũng như các hàm toán học cao cấp (sin, cos, exp, log, v.v.) để thực hiện các phép toán số học trên mảng.

Xử Lý Mảng: Thư viện này cung cấp các hàm cho việc thay đổi kích thước, cắt, chuyển vị và thao tác khác trên mảng.

Lập Trình Vectorized: NumPy khuyến khích việc sử dụng lập trình vectorized để thực hiện các phép toán trên mảng một cách nhanh chóng và hiệu quả, thay vì sử dụng vòng lặp.

Tích Hợp Các Dự Án Khác: NumPy tích hợp tốt với các thư viện và công cụ khác trong hệ sinh thái khoa học dữ liệu của Python như SciPy, pandas, matplotlib, và scikit-learn.

**Pickle**

Pickle là một module trong Python được sử dụng để serialize (chuyển đổi thành dạng bytes) và deserialize (chuyển đổi từ dạng bytes thành đối tượng ban đầu) các đối tượng Python. Quá trình pickle giúp bạn lưu trữ các đối tượng Python vào tệp hoặc truyền qua mạng và sau đó khôi phục chúng thành đối tượng Python ban đầu mà không mất đi thông tin hoặc cấu trúc.

Một số điểm quan trọng về Pickle:

Lưu Trữ Dữ Liệu: Pickle cho phép bạn lưu trữ và truyền các đối tượng Python phức tạp như danh sách, từ điển, lớp, và nhiều loại dữ liệu khác.

Không phải là Định Dạng Giao Tiếp Chung: Pickle thường không phải là định dạng giao tiếp chung vì nó có thể không an toàn đối với dữ liệu không được tin cậy từ nguồn bên ngoài. Điều này có thể dẫn đến các vấn đề về bảo mật hoặc tương thích trong tương lai.

Sử Dụng Pickle:

Để serialize một đối tượng thành dạng bytes: pickle.dump(obj, file)

Để deserialize một đối tượng từ dạng bytes: obj = pickle.load(file)

**Pandas**

Các Chức Năng Xử Lý Dữ Liệu: Pandas cung cấp nhiều chức năng để thực hiện các hoạt động xử lý dữ liệu như lọc, lấy dữ liệu con, sắp xếp, xử lý giá trị bị thiếu, và biến đổi dữ liệu.

Hỗ Trợ Đọc và Ghi Dữ Liệu: Pandas có khả năng đọc và ghi dữ liệu từ và vào nhiều định dạng dữ liệu phổ biến như CSV, Excel, SQL, JSON, và nhiều định dạng khác.

Thống Kê và Tính Toán: Pandas cung cấp các hàm tính toán thống kê, như tính trung bình, tổng, phương sai, và hỗ trợ cho các phép toán thống kê phức tạp.

Tích Hợp Dễ Dàng: Pandas có tích hợp tốt với các thư viện và công cụ khác trong hệ sinh thái khoa học dữ liệu của Python như NumPy, Matplotlib và SciPy.

**Sklearn**

Scikit-learn, thường được viết tắt là "sklearn", là một thư viện mã nguồn mở trong Python chuyên về machine learning (học máy) và data mining (khai phá dữ liệu). Scikit-learn cung cấp nhiều công cụ và thuật toán tiêu chuẩn để xây dựng và huấn luyện các mô hình học máy cho nhiều loại tác vụ khác nhau như phân loại, hồi quy, phân cụm, và nhiều tác vụ khác.

Một số đặc điểm chính của scikit-learn:

1. Thuật Toán Học Máy: Scikit-learn cung cấp một loạt các thuật toán học máy phổ biến như Decision Trees, Random Forests, Support Vector Machines (SVM), Naive Bayes, K-Means, và nhiều thuật toán khác.
2. Tiêu Chuẩn Hóa Giao Diện: Scikit-learn cung cấp một giao diện tiêu chuẩn cho việc xây dựng, huấn luyện và đánh giá mô hình học máy. Điều này giúp cho việc chuyển đổi giữa các thuật toán dễ dàng hơn.
3. Tiền Xử Lý Dữ Liệu: Scikit-learn cung cấp các công cụ để xử lý và tiền xử lý dữ liệu trước khi đưa vào huấn luyện mô hình. Điều này bao gồm chuẩn hoá dữ liệu, xử lý giá trị thiếu, và mã hóa dữ liệu.
4. Đánh Giá và Kiểm Định Mô Hình: Scikit-learn cung cấp các công cụ để đánh giá và kiểm định hiệu suất của mô hình học máy bằng cách sử dụng các phương pháp cross-validation và các chỉ số đánh giá.
5. Hỗ Trợ Đa Loại Tác Vụ: Scikit-learn không chỉ hỗ trợ các tác vụ phân loại và hồi quy, mà còn hỗ trợ phân cụm, giảm chiều dữ liệu, học bán giám sát, và nhiều loại tác vụ khác.

**Chương III: Thiết kế hệ thống**

**3.1 Cơ chế hoạt động**

Người dùng có thể tìm hiểu về các tư thế yoga của hệ thống cung cấp. Dữ liệu được thu thập qua webcam của người dùng. Sử dụng mediapipe để phát hiện các tư thế yoga trong thời gian thực. Các điểm (point) được xác định và lưu trong mảng 3 chiều. Những điểm này dùng để so sánh tư thế người dùng với tư thế yoga mục tiêu. Khi độ trùng khớp cao có thể kết luận tư thế của người dùng là tư thế nào.

A diagram of a computer program

Description automatically generated

Sơ đồ khối cách hoạt động của mô hình

* Thu thập dữ liệu:

Các tư thế được sử dụng trong đề tài là Balasana (Child's Pose), Baddha Konasana (Bound Angle Pose), Utkata Konasana (Goddess Pose), Kumbhakasana (Plank Pose), Vrksasana (Tree Pose), Virabhadrasana II (Warrior II Pose). Tốc độ khung hình 30 FPS (30 khung hình /giây). Khoảng cách để webcam có thể thấy rõ toàn bộ cơ thể trong tầm 4m tính từ vị trí đặt webcam.

* Xử lí dữ liệu:

Sử dụng thư viện mediapipe để trích xuất các điểm (point) của tư thế trong khung hình theo mảng 3 chiều, sau đó làm phẳng thành mảng 1 chiều và lưu trong file csv dưới dạng bảng.

Với tập dữ liệu từ file CSV ta sẽ huấn luyện mô hình máy học phân loại các động tác dựa vào tọa độ các điểm của tư thế và nhãn tên của tư thế đó.

**3.2 Chi tiết hệ thống**

3.2. . Tải các thư viện cần thiết và chạy Mediapipe

A black background with white text

Description automatically generated

Đoạn mã này import thư viện Mediapipe và gán nó cho biến tên là mp. Mediapipe là một thư viện được phát triển bởi Google, cung cấp các công cụ và mô hình để xử lý dữ liệu hình ảnh và video, bao gồm việc nhận diện landmarks trên khuôn mặt, cơ thể, tay, v.v.

Thư viện OpenCV là một thư viện phổ biến cho xử lý dữ liệu hình ảnh và video, giúp ta có thể kết nối vào webcam và xử lý hình ảnh trên đó.

A black background with white text

Description automatically generated

Tạo một biến mp\_drawing để sử dụng các tiện ích liên quan đến việc vẽ landmarks trên hình ảnh.

Tạo một biến mp\_holistic để sử dụng mô hình Holistic của Mediapipe. Mô hình Holistic có khả năng nhận diện khuôn mặt, cơ thể và tay cùng một lúc tuy nhiên đề tài chỉ sử dụng nhận diện cơ thể. Nhóm chọn xài holistic để cho nếu muốn phát triển đề tài xa hơn hoặc muốn sử dụng khuôn mặt và tay để đem lại kết quả chính xác hơn thì sẽ tiện chỉnh sửa cho sau này.

Ta sẽ test 1 đoạn nhỏ để đọc khung hình từ webcam:

A screen shot of a computer program

Description automatically generated

Khởi tạo một đối tượng VideoCapture từ webcam bằng tham số 0 trong hàm VideoCapture. Tham số 0 này đại diện cho webcam đầu tiên được tìm thấy trên hệ thống.

Trong vòng lặp vô hạn, dòng **“ret, frame = cap.read()”** sẽ đọc một khung hình từ webcam. Biến **“ret”** sẽ chứa giá trị True nếu việc đọc thành công, False nếu không thành công và biến **“frame”** chứa khung hình đã đọc.

Sau đó ta hiển thị khung hình đang đọc (frame) trong cửa sổ tên **“Raw Webcam Feed”**.

Câu lệnh if sẽ kiểm tra nếu ta nhấn phím **“q”** thì sẽ kết thúc vòng lặp và đến bước cuối.

2 dòng cuối được dùng để giải phóng tài nguyên của webcam để đảm bảo không để lại tài nguyên mở khi kết thúc chương trình và đóng tất cả các cửa sổ đang hiển thị, làm cho chương trình kết thúc sạch sẽ mà không còn cửa sổ đang chạy.

Sau khi ta đọc được khung hình từ webcam, bước tiếp theo cần làm chính là sử dụng Mediapipe để hiển thị và nối các landmark trên cơ thể:

A screen shot of a computer code

Description automatically generated

Để hiển thị landmark trên cơ thể ta sử dụng mô hình Holistic của Mediapipe. Tham số min\_detection\_confidence và min\_tracking\_confidence xác định ngưỡng tối thiểu để xác định mức độ tin cậy của việc nhận diện và theo dõi.

Vì Mediapipe yêu cầu đầu vào là ảnh RGB nhưng màu của khung hình là dạng BGR nên sau khi lấy khung hình từ webcam về ta cần phải chuyển đổi màu của khung hình từ BGR sang RGB.

Dòng tiếp theo là dòng quan trọng nhất trong đề tài, **“results = holistic.process(image)** vì nó sử dụng mô hình Holistic để xử lý ảnh và nhận diện các điểm đặc trưng (landmark) trong cơ thể và kết quả sẽ được lưu vào biến results.

Sau khi xử lý xong hình ảnh, ta chuyển đổi màu của khung hình từ RGB sang BGR lại để có thể hiển thị trên OpenCV.

Vẽ các landmarks của mô hình Holistic lên khung hình sử dụng hàm **“draw\_landmarks”** của thư viện Mediapipe. Trong trường hợp này, chúng ta vẽ landmarks liên quan đến pose (tư thế cơ thể) và kết nối giữa chúng.

Còn lại thì công dụng như trên.

3.2. Thu thập và lưu trữ điểm đặc trưng

A black screen with white text

Description automatically generated

Thư viện csv cung cấp các chức năng để làm việc với các tệp dữ liệu CSV. Dữ liệu CSV là một loại tệp văn bản dễ đọc được sử dụng để lưu trữ dữ liệu bảng dưới dạng hàng và cột.

Thư viện os cung cấp các chức năng để tương tác với hệ thống tệp và thư mục như tạo và xóa tệp, kiểm tra sự tồn tại của thư mục, …

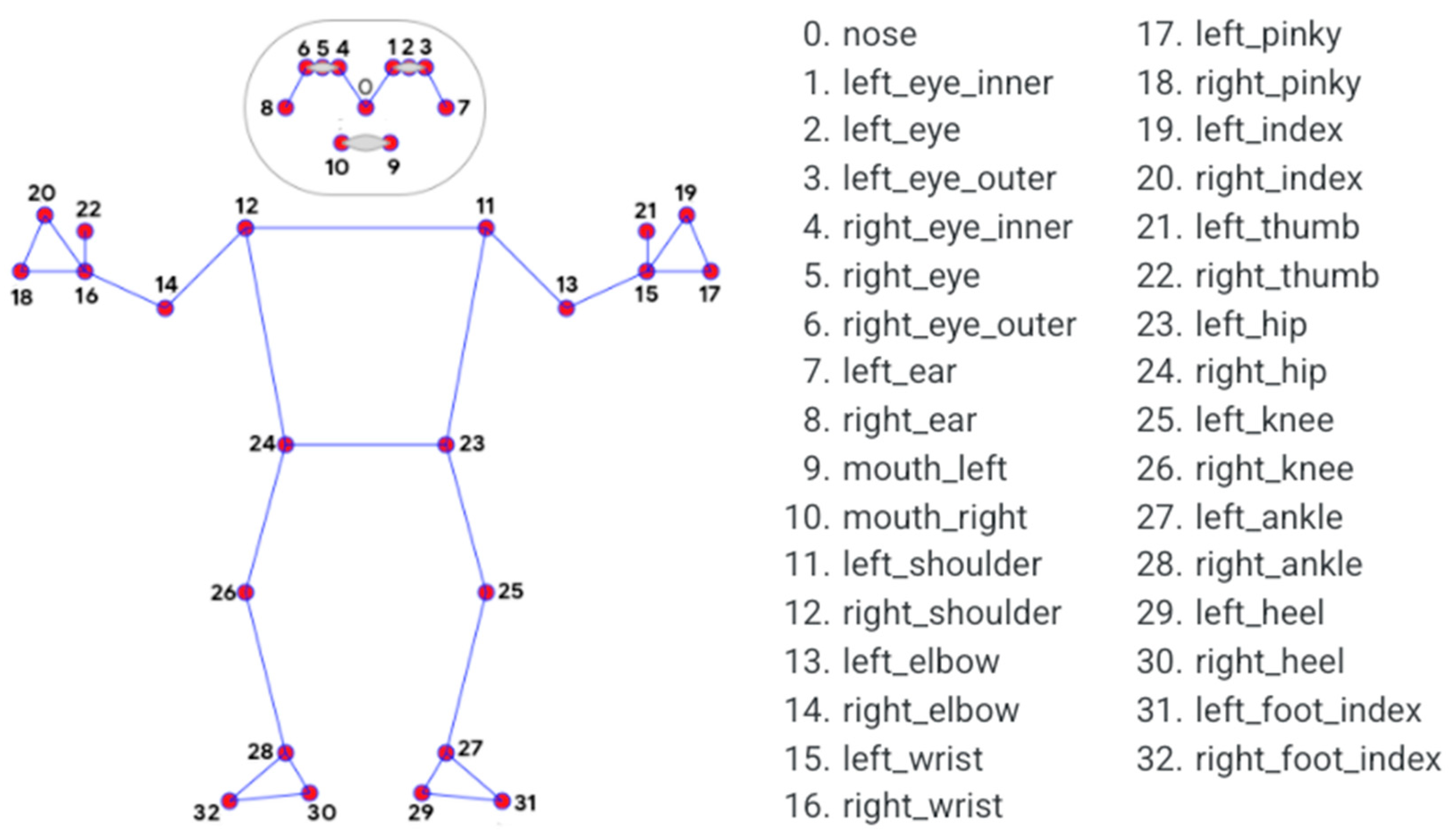
Thư viện numpy cho tính toán số học và xử lý mảng đa chiều. Nó cung cấp nhiều chức năng hữu ích để thao tác và xử lý dữ liệu số.

A screen shot of a computer

Description automatically generated

Ta sử dụng dòng code này để tính toán số lượng điểm đặc trưng được nhận diện trong kết quả của mô hình Holistic. Các điểm này được xác định bằng tọa độ trên khung hình.

Ở đây khi ta chạy **“num\_coords”** thì nó sẽ trả về giá trị 33 điều đó có nghĩa là khung hình có 33 điểm đặc trưng. Để có thể biết rõ 33 điểm đó nằm ở đâu trên cơ thể thì ta có hình sau:



Tiếp theo ta cần chuẩn bị cho việc lưu dữ liệu từ mô hình nhận diện landmarks vào một tệp CSV để sau này có thể sử dụng trong quá trình huấn luyện mô hình machine learning.

A black background with colorful text

Description automatically generated

Ban đầu, ta tạo ra một danh sách landmarks với một phần tử duy nhất **“class”**. Phần từ này sẽ là đại diện cho nhãn tên ví dụ như: Goddess, Tree, …

Tiếp theo đó ta vào vòng lặp để lặp ra từng số cho mỗi đặc trưng và đem vào danh sách.

Mỗi đặc trưng sẽ có 4 thông tin cụ thể sau:

+ x: Tọa độ ngang của điểm đặc trưng.

+ y: Tọa độ dọc của điểm đặc trưng.

+ z: Tọa độ sâu của điểm đặc trưng.

+ v: Mức độ tin cậy (visible) của điểm đặc trưng.

Như vậy, khi vòng lặp kết thúc thì ta sẽ có danh sách **“landmarks”** chứa các tên đặc trưng và thông tin liên quan đến tọa độ và mức tin cậy của từng điểm đặc trưng.

Tiếp theo ta tạo tệp csv

A black screen with text

Description automatically generated

**“with open('coords.csv', mode='w', newline='') as f:”**: Dòng này mở tệp có tên **'coords.csv'** để ghi dữ liệu. Tham số **mode='w'** cho biết chế độ ghi. Tham số **newline=''** giúp tránh vấn đề về xử lý ký tự xuống dòng khác nhau trên các hệ thống khác nhau.

**“csv\_writer = csv.writer (f, delimiter = ’,’, quotechar = ’”’, quoting = csv.QUOTE\_MINIMAL)**:Đoạn mã này tạo một đối tượng csv\_writer để viết dữ liệu vào tệp. Tham số delimiter=',' chỉ định rằng dấu phẩy sẽ được sử dụng làm ký tự phân tách giữa các trường. Tham số quotechar='"' chỉ định rằng dấu ngoặc kép sẽ được sử dụng để bao quanh các trường chứa dấu phẩy. Tham số quoting=csv.QUOTE\_MINIMAL chỉ định rằng chỉ các trường chứa ký tự đặc biệt sẽ được bao quanh bởi dấu ngoặc kép.

**“csv\_writer.writerow(landmarks)”**: Dòng này ghi dòng tiêu đề (header) vào tệp CSV. Dòng này tạo ra một dòng trong tệp CSV chứa tên nhãn dán và các thông tin liên quan đến các đặc trưng đó, tách nhau bởi dấu phẩy. Ta sẽ được kết quả như này:

A black background with white text

Description automatically generated

Sau khi tạo tệp CSV để chứa các thông tin, dữ liệu cần thiết thì đã đến lúc ta thu thập các điểm đặc trưng.

A black background with white text

Description automatically generated

Trước tiên ta cần nêu tên nhãn dán ra trước, đề tài em gồm 6 động tác cơ bản: Bound Angle, Tree, Goddess, Warrior ll, Plank, Child’s Pose. Ban đầu ta đặt class\_name là Bound Angle.

A computer code on a black background

Description automatically generated

Sau khi thu về khung hình và nối các điểm đặc trưng như bước trên, ta làm thêm động tác xuất dữ liệu tọa độ từ các landmark của cơ thể và ghi vào file CSV.

- Khởi đầu, ta gán danh sách các landmarks của tư thế yoga được nhận diện từ kết quả mô hình Holistic cho biến **“pose”**.

Tiếp theo ta tạo một danh sách **“pose\_row”** chứa thông tin về các tọa độ và mức độ tin cậy (visibility) của mỗi landmark trong tư thế. Dòng code này sử dụng một biểu thức danh sách để tạo ra danh sách 2D vì thế nên sau đó ta dùng .flatten() để biến nó thành một danh sách 1D.

Gán biến pose\_row vào row, bước này không cần thiết vì mình chỉ xài mỗi landmark của cơ thể, nếu như trong trường hợp xài cả tay hoặc mặt thì biến row này sẽ là giá trị tổng của cơ thể, mặt, tay sau khi biến đổi sang danh sách 1D.

Thêm tên lớp (class\_name) ở vị trí đầu tiên của danh sách row. Điều này giúp xác định tên của tư thế yoga tương ứng với dữ liệu đặc trưng. Ở ví dụ hiện tại thì tên đặc trưng là Bound Angle.

3.3. Đào tạo mô hình phân loại máy móc

3.3.1. Tải và xử lý dữ liệu thông qua pandas

A black screen with white text

Description automatically generated

Ta sử dụng thư viện pandas vì nó là một thư viện mạnh mẽ cho xử lý và phân tích dữ liệu, chủ yếu dựa trên cấu trúc dữ liệu DataFrame.

Ngoài ra ta còn import hàm **train\_test\_split** từ **model\_selection** của thư viện **scikit-learn**. Hàm này sẽ được sử dụng để chia dữ liệu thành tập huấn luyện và tập kiểm tra khi huấn luyện mô hình máy học.

A black background with white text

Description automatically generated

Sử dụng hàm **read\_csv** từ thư viện **pandas** để đọc dữ liệu từ tệp csv và trả về một DataFrame chứa dữ liệu từ tệp csv này.

Sau khi trả về thì ta gắn cho biến **df** để tiện cho việc sử dụng ở những code sau.

A black background with white text

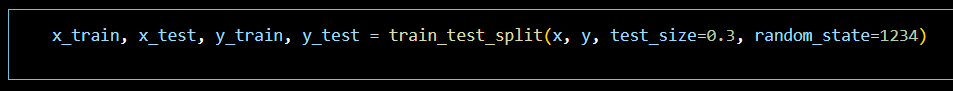
Description automatically generated

Bước tiếp theo ta cần làm là tách tách dữ liệu đặc trưng và nhãn tên.

Để làm được điều đó thì ta sẽ tạo 2 biến x và y. Biến x sẽ chứa các dữ liệu đặc trưng còn biến y sẽ chứa nhãn tên.

Sử dựng **“df.drop”**,ta bỏ lớp **“class”** ở DataFrame, **“axis=1”** là bỏ 1 cột và gán nó vào biến x.

Sau đó ta trích xuất cột **“class”** ở DataFrame và gán nó vào biến y.



Sử dụng hàm **train\_test\_split** từ thư viện **scikit-learn**. Ta sẽ chia dữ liệu thành 2 tập, tập huấn luyện và tập kiểm tra.

Biến x và biến y đều được chia thành 2 tập: x\_train là tập huấn luyện, x\_test là tập kiểm tra và tương tự cho y.

Tham số của **“test\_size”** là 0,3 có nghĩa là trong dữ liệu, sẽ có 30% được chia vào tập kiểm tra, 70% còn lại được chia vào tập huấn luyện.

Tham số của **“random\_state”** là 1234 có nghĩa là tham số này xác định hạt giống (seed) để đảm bảo rằng quá trình chia dữ liệu sẽ được thực hiện cùng một cách mỗi khi chạy lại mã. Điều này giúp đảm bảo kết quả của việc chia dữ liệu là nhất quán và dễ tái sản xuất.

3.3.2. Huấn luyện mô hình phân loại

A screen shot of a computer program

Description automatically generated

Ta import một số module từ thư viện scikit-learn để xây dựng và huấn luyện các mô hình máy học. Thông thường chỉ sử dụng 1 class nhưng để mang lại kết quả tốt nhất, nhóm chọn 4 class để chạy và xem kết quả class nào tốt nhất thì sẽ sử dụng class đó.

Import hàm **make\_pipeline** từ module **pipeline** của thư viện **scikit-learn**. **make\_pipeline** được sử dụng để tạo một chuỗi các bước xử lý dữ liệu và mô hình trong một pipeline (luồng công việc).

Import class **StandardScaler** từ module **preprocessing** của thư viện **scikit-learn**. **StandardScaler** được sử dụng để chuẩn hóa dữ liệu đặc trưng để có giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1.

Import class **LogisticRegression** và **RidgeClassifier** từ module **linear\_model** của thư viện **scikit-learn**. Đây là các mô hình học sử dụng trong phân loại.

Import class **RandomForestClassifier** và **GradientBoostingClassifier** từ module **ensemble** của thư viện **scikit-learn**. Đây là các mô hình học dựa trên phương pháp tập hợp (ensemble).

A screenshot of a computer program

Description automatically generated

Tạo một mảng có tên pipelines, trong đó mỗi phần tử của từ điển đại diện cho một pipeline xử lý dữ liệu và mô hình học máy.

**'lr':make\_pipeline(StandardScaler(), LogisticRegression())**: Đoạn code này tạo một phần tử trong mảng với khóa **'lr'** (tương ứng với **Logistic Regression**). Pipeline này bao gồm hai bước: chuẩn hóa dữ liệu sử dụng **StandardScaler()** và mô hình học máy **LogisticRegression()**.

Tương tự như trên, ta tạo ra phần tử **‘rc’**, **‘rf’**, **‘gb’**, tương ứng với **RidgeClassifier()**, **RandomForestClassifier()**, **GradientBoostingClassifier()**.

A black screen with white text and numbers

Description automatically generated

Tạo một mảng **fit\_models** chứa các mô hình đã được huấn luyện bằng cách lặp qua các pipelines và dùng chúng để huấn luyện trên tập dữ liệu huấn luyện.

**“for algo, pipeline in pipelines.items()”**: Bắt đầu một vòng lặp qua từng phần tử trong mảng pipelines. Biến **algo** sẽ lưu giá trị của khóa (key) trong mảng, tức là tên của mô hình (ví dụ: **'lr'**, **'rc'**, **'rf'**, '**gb'**), và biến **pipeline** sẽ lưu giá trị của giá trị (value), tức là pipeline tương ứng với mô hình.

**“model = pipeline.fit(x\_train, y\_train)”**: Sử dụng pipeline để huấn luyện mô hình trên tập dữ liệu huấn luyện x\_train và y\_train. Biến model sẽ lưu trữ mô hình đã được huấn luyện.

**“fit\_models[algo] = model”**: Gán mô hình đã huấn luyện vào từ điển **fit\_models** với khóa tương ứng là **algo**, tức là tên của mô hình.

Sau khi chạy xong, ta có thể chạy thử dòng code để xem sau quá trình huấn luyện sẽ như nào:

A computer screen shot of white text

Description automatically generated

3.4. Đánh giá và lưu trữ mô hình

A black background with white text

Description automatically generated

**accuracy\_score** được sử dụng để tính độ chính xác của mô hình phân loại bằng cách so sánh dự đoán của mô hình với nhãn thực tế.

Module **pickle** để sử dụng trong việc lưu và nạp (serialize và deserialize) các đối tượng Python, bao gồm cả mô hình máy học.

A screen shot of a computer code

Description automatically generated

Tương tự như code trước, **“for algo, model in fit\_model.items()”** bắt đầu một vòng lặp qua các phần tử trong mảng fit\_models. Biến **algo** lưu tên của mô hình và biến model lưu mô hình tương ứng.

yhat ở đây là y mũ, được dùng để lưu kết quả dự đoán nhãn trên tập dữ liệu kiểm tra.

Dòng cuối được dùng để in ra tên của mô hình và độ chính xác của mô hình bằng cách sử dụng hàm **accuracy\_score** so sánh nhãn thực tế y\_test và nhãn dự đoán yhat.

Ở kết quả, ta thấy lr, rc, rf đều đạt chính xác 100%. Điều này rất ít khi xảy ra, có thể do các thông tin của các điểm đặc trưng quá khác nhau nên dễ phân loại hoặc nguyên nhân gì đó khác. Vì kết quả 3 cái bằng nhau nên ta có thể chọn bất kỳ cái nào, nhóm em chọn rf.

A screen shot of a computer code

Description automatically generated

Sử dụng module **pickle**, ta lưu mô hình RandomForestClassifier đã được huấn luyện vào tệp.

Mở một tệp có tên **“body\_language.pkl”** để ghi dữ liệu. Tham số **wb** cho biết chế độ ghi dưới dạng nhị phân.

Ta sử dụng hàm **pickle.dump** để lưu mô hình RandomForestClassifier từ mảng fit\_models (với **rf**) vào tệp. Đối số đầu tiên là mô hình cần lưu, và đối số thứ hai là tệp mà ta muốn ghi dữ liệu vào. Trường hợp ở đây là ghi mô hình rf đã được huấn luyện vào tệp **“body\_language.pkl”.**

3.5. Phân loại sử dụng mô hình

A black screen with white text

Description automatically generated

Trước khi phân loại, ta cần phải nạp mô hình đã lưu vào trước.

Mở một tệp có tên **“body\_language.pkl”** để đọc dữ liệu. Tham số **rb** cho biết chế độ đọc dưới dạng nhị phân.

Sử dụng hàm **pickle.load** để nạp mô hình từ tệp. Kết quả nạp sẽ được gán cho biến **model**, ta có thể sử dụng mô hình đã huấn luyện mà không cần phải huấn luyện lại từ đầu.

A computer screen with text on it

Description automatically generated

Code vẫn tương tự như lúc trích xuất điểm đặc trưng sang tệp csv nhưng sẽ khác ở phần dưới.

Sau khi lấy được các thông tin của các điểm đặc trưng trên cơ thể, thay vì lưu vào tệp csv thì ta sẽ tạo một DataFrame **x** để chứa dữ liệu điểm đặc trưng.

Ta sử dụng mô hình đã huấn luyện để dự đoán tình trạng cơ thể dựa trên dữ liệu đặc trưng của tình trạng pose trong DataFrame **x**. Kết quả dự đoán được gán cho biến **body\_language\_class**.

Ta có thể tính xác suất dự đoán cho mỗi tình trạng cơ thể sử dụng hàm **predict\_proba** của mô hình. Kết quả xác suất được gán cho biến **body\_language\_prob**.

Và cuối cùng ta sẽ in ra nhãn tên tình trạng cơ thể đã dự đoán (**body\_language\_class**) và xác suất dự đoán cho mỗi lớp (**body\_language\_prob**).

**Chương IV: Kết quả**

**Chương V: Kết luận và hướng phát triển**

5.1. Kết luận

Chương trình thực hiện được tính năng cơ bản là phân loại được động tác yoga người dùng đang thực hiện và chấm điểm.

Quá trình bắt đầu bằng việc thu thập dữ liệu và tiền xử lý hình ảnh để trích xuất các đặc trưng quan trọng. Nhóm em đã sử dụng mô hình Holistic từ thư viện Mediapipe để nhận diện các điểm đặc trưng quan trọng trên cơ thể, bao gồm các điểm trên tay, chân, và cơ thể. Tuy nhiên bây giờ chỉ mới sử dụng cơ thể.

Tiếp theo, chúng tôi đã xây dựng một chuỗi pipeline bao gồm việc chuẩn hóa dữ liệu và đã huấn luyện các mô hình máy học trên tập dữ liệu huấn luyện và đánh giá hiệu suất của chúng trên tập dữ liệu kiểm tra bằng cách sử dụng độ chính xác so sánh giữa y\_test và yhat.

Lưu trữ mô hình bằng cách sử dụng module **pickle**, giúp ta có thể sử dụng lại mô hình đã huấn luyện mà không cần phải huấn luyện lại từ đầu.

5.2. Hướng phát triển

Đề tài này còn nhiều chỗ thiếu xót và có thể được bổ sung, phát triển xa hơn như là:

+ Hiển thị ảnh mẫu của từng động tác và chấm điểm người dùng dựa theo ảnh mẫu đó. Thay đổi động tác mới thông qua nút nhấn.

+ Thiết kế bộ đếm giờ. Nếu thực hiện đúng động tác trên 70% thì bộ đếm giờ sẽ bắt đầu đếm. Mục đích là đo khoảng thời gian mà người dùng có thể giữ nguyên động tác đó.

+ Kết hợp landmarks của mặt và tay để có thể phân loại nhiều động tác hơn và chính xác hơn.

+ Thiết kế 1 trang web để người dùng có thể dễ dàng sử dụng.