**PHÂN TÍCH VÀ Ý TƯỞNG PHÁT TRIỂN THUẬT TOÁN**

**Đề tài: Thiết kế hệ thống nhận diện bàn tay và thao tác với bàn tay**

**I- Phân tích đề tài**

**1. Lý do chọn đề tài**

Trong khoảng mấy năm trở lại đây,"Thị giác máy tính" đã nhanh chóng trở thành một động lực mạnh mẽ đằng sau sự phát triển của nhiều lĩnh vực và mang lại nhiều lợi ích đáng kể.Trong thực tế, khả năng nhận diện và xử lý hình ảnh của thị giác máy tính đã mở ra một loạt các ứng dụng quan trọng.Nhận biết được lợi ích to lớn của "Thị giác máy tính" nhóm chúng em đã lựa chọn đề tài "Thiết kế hệ thống nhận diện bàn tay và thao tác với bàn tay".Việc tích hợp thị giác máy tính vào hệ thống nhận diện bàn tay không chỉ giúp nâng cao độ chính xác trong xác thực người dùng mà còn tạo ra một cơ sở vững chắc cho các ứng dụng như an ninh thông tin, quản lý truy cập, và kiểm soát tự động.

Sự linh hoạt và hiệu suất của thị giác máy tính đã thể hiện rõ trong việc tương tác người-máy, điều này không chỉ tạo ra các ứng dụng giải trí mới mà còn mở ra những cánh cửa mới trong lĩnh vực y tế, nơi mà khả năng chẩn đoán và theo dõi bệnh tình được cải thiện đáng kể.

**2. Giới thiệu chung về đề tài**

Đề tài "Thiết kế hệ thống nhận diện bàn tay và thao tác với bàn tay" tập trung vào việc phát triển một hệ thống sử dụng thư viện mã nguồn mở OpenCV để theo dõi và nhận diện các cử chỉ,hình ảnh, video của bàn tay của con người và thực hiện những yêu cầu mà người thiết kế đã cài đặt. OpenCV là một thư viện phổ biến trong lĩnh vực xử lý ảnh và thị giác máy tính, cung cấp các công cụ mạnh mẽ cho việc xử lý, phân tích và nhận diện đối tượng trong ảnh và video.

**3. Phân tích thuật toán**

- Nhóm chúng em phát triển đề tài sau khi tham khảo và phát triển module mã nguồn mở.

Link module mã nguồn mở: <https://developers.google.com/mediapipe/solutions/vision/hand_landmarker>

- Đoạn module phát triển sử dụng thư viện Mediapipe cụ thể là MediaPipe Hands.MediaPipe Hands sử dụng một mô hình học sâu (Deep Neural Network - DNN) để phát hiện và theo dõi lòng bàn tay.Và trong thư viện này , mô hình chính được sử dụng để nhận diện bàn tay là mô hình Hand Landmarker và Hand Tracking.Và cả 2 mô hình này đều sử dụng DNN để phát hiện và theo dõi lòng bàn tay.

- Mô hình DNN trong MediaPipe Hands hoạt động theo hai giai đoạn:

+ Phát hiện lòng bàn tay (Palm detection): Mô hình này hoạt động trên toàn bộ hình ảnh đầu vào và cung cấp một hình ảnh đã được cắt của bàn tay.Trong mã nguồn mở, hàm "findHands()" được sử dụng để thực hiện công việc này.Hàm này xử lý hình ảnh đầu vào và phát hiện lòng bàn tay trong hình ảnh.Nếu lòng bàn tay được phát hiện, nó sẽ vẽ các điểm đánh dấu và kết nối chúng.

+ Xác định vị trí các điểm đánh dấu trên tay (Hand landmarks identification): Sau khi phát hiện lòng bàn tay,MediaPipe thực hiện việc xác định vị trí chính xác của 21 tọa độ lòng bàn tay 3D trong vùng tay được phát hiện.Trong mã nguồn mở, hàm "findPosition()" được sử dụng để thực hiện công việc này. Hàm này xác định vị trí của các điểm đánh dấu trên tay và vẽ chúng lên hình ảnh.

+ Trong mã nguồn mở này, "mpHands.Hands()" được sử dụng, nó tạo ra một đối tượng để phát hiện lòng bàn tay và xác định vị trí các điểm đánh dấu trên tay. Điều này bao gồm cả việc phát hiện lòng bàn tay (tương đương với Hand Landmarker) và theo dõi lòng bàn tay qua thời gian (tương đương với Hand Tracking).

- Cụ thể là:

+ Hand Landmarker: Phần này được thực hiện bởi hàm "findPosition()". Hàm này xác định vị trí của các điểm đánh dấu trên tay và vẽ chúng lên hình ảnh.

+ Hand Tracking: Phần này được thực hiện bởi hàm "findHands()".Hàm này xử lý hình ảnh đầu vào và phát hiện lòng bàn tay trong hình ảnh.Nếu lòng bàn tay được phát hiện, nó sẽ vẽ các điểm đánh dấu và kết nối chúng.

- Hai hàm "findPosition()" và "findHands()": Cả hai hàm này đều sử dụng kết quả từ "mpHands.Hands().process(imgRGB)" nơi imgRGB là hình ảnh đầu vào.Hàm "process()" này thực hiện cả hai công việc: phát hiện lòng bàn tay và xác định vị trí các điểm đánh dấu trên tay.

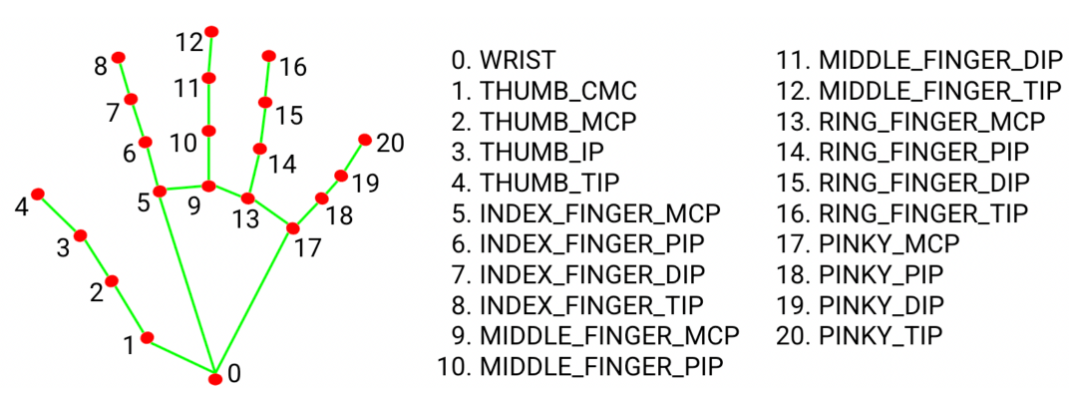
- Quá trình thực hiện:

+ Sử dụng thư viện mở OpenCV để xử lý video.Cụ thể được sử dụng để đọc video từ camera và chuyển hình ảnh từ không gian màu BGR sang RGB.

+ Sử dụng thư viện Mediapipe: là một framework được sử dụng để xây dựng pipeline cho dữ liệu chuỗi thời gian như âm thanh, video.Cụ thể đối với đoạn mã nguồn mở trên thì dùng để xây dựng mô hình theo dõi bàn tay.

+ Phát hiện lòng bàn tay: Mô hình này sử dụng một mô hình phát hiện lòng bàn tay đơn lẻ.Mô hình này hoạt động trên toàn bộ hình ảnh đầu vào và cung cấp một hình ảnh đã được cắt của bàn tay.

+ Xác định vị trí các điểm đánh dấu trên tay: Sau khi phát hiện lòng bàn tay, MediaPipe thực hiện việc xác định vị trí chính xác của 21 tọa độ lòng bàn tay 3D trong vùng tay được phát hiện.



*Hình 1.1 21 điểm trên bàn tay sau khi đã phát hiện bàn tay*

**4. Phát triển thuật toán**

- Dựa vào phân tích thuật toán và chức năng module mở ở trên, nhóm chúng em đã phát triển thành đề tài "Thiết kế hệ thống nhận diện bàn tay và thao tác với bàn tay".

- Nhóm chúng em đã phát triển thêm 3 chức năng từ module mở trên thành đề tài này.Ba chức năng đó là: Nhận diện số đếm từ 0 đến 5 thông qua 5 ngón tay, Điều chỉnh thanh âm lượng , Tạo phím tắt thông qua các ngón tay.

**4.1 Nhận diện số đếm từ 0 đến 5**

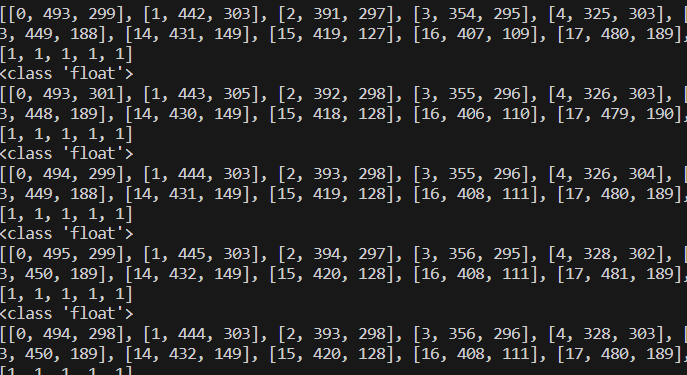
- Module trên đã nhận diện được bàn tay và đánh dấu thành 21 điểm trên bàn tay.Dựa vào điều này nhóm chúng em đã thiết kế chức năng "Nhận diện số đếm từ 0 đến 5 thông qua năm ngón trên bàn tay".

- Sau khi đã nhận diện được bàn tay và có được 21 điểm trên bàn tay.Lấy các điểm đầu trên các ngón tay và đưa vào một list. List đó chứa 5 điểm [4,8,12,16,20].

- Mỗi một điểm chứa 3 trạng thái đó là: vị trí điểm, chiều rộng , chiều cao.Ví dụ:

điểm 4 có là [4,width,height].Cái này rất quan trọng trong việc xác định ngón mở hay là không mở.

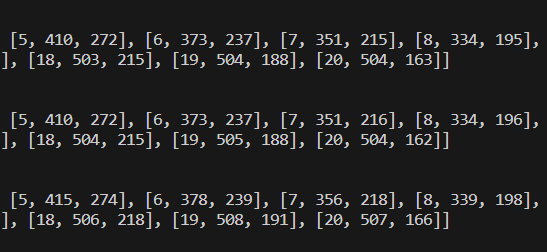
- Nhóm chúng em tiến hành viết cho ngón cái trước do ngón cái không thể cụp xuống được nên sẽ để ngón cái cụp khi hướng sang bên phải , ngón cái duỗi khi hướng sang bên trái.Để biết được điều đó nhóm chúng em dựa vào trạng thái của mỗi điểm.Đây là trạng thái '4' điểm trên ngón cái khi ngón cái đang duỗi:



*Hình 1.2 Trạng thái các điểm của ngón cái*

chúng ta nhận thấy khi đang duỗi thì trạng thái width của '4' điểm đang giảm dần từ điểm '4' xuống điểm '1'.Dựa vào điều đó ta so sánh điểm '4' với điểm '3'.Nếu trạng thái width của điểm '3' lớn hơn trạng thái width của điểm '4' thì ngón cái mở ngược lại thì ngón cái quặp.

- Đó là trường hợp với ngón cái, còn đối với 4 ngón còn lại giống nhau do đều có thể co duỗi dễ dàng nên sẽ giống nhau.Dựa vào trạng thái của mỗi điểm để nhận biết ngón duỗi hay quặp.Vì trạng thái cảu 4 ngón là giống nhau nên lấy đại diện ngón trỏ làm ví dụ.Dưới đây là trạng thái của ngón trỏ khi đang duỗi:



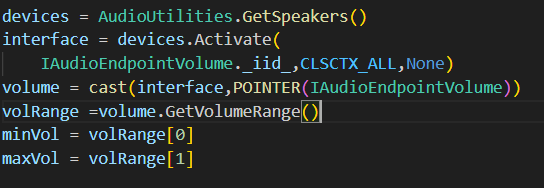
*Hình 1.3 Trạng thái các điểm của ngón trỏ*

Nhìn vào hình ảnh , nhận thấy từ điểm đầu ngón tay đến điểm cuối ngón tay trạng thái height tăng dần.Để nhận biết ngón tay duỗi thì so sánh trạng thái height của điểm '8' với trạng thái height của điểm '6' nếu điểm '8' nhỏ hơn điểm '6' thì đang duỗi, ngược lại thì đang quặp.Chúng em dùng vòng lặp để nhận diện với 4 ngón giống này.

- Tạo một list có chức năng là chứa trạng thái của mỗi ngón tay.Quy định nếu ngón duỗi thì thêm vào list số '1' , ngược lại thêm vào list số '0'.Sau đó đếm số số '1' thì đó chính là số ngón tay đang mở ứng với số đếm.

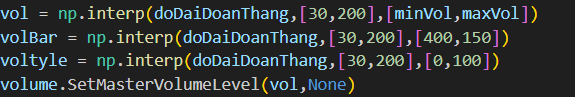
**4.2 Điều chỉnh thanh âm lượng**

- Tương tác với thiết bị âm thanh trên hệ thống.Lấy thông tin về thiết bị loa và giao diện điều khiển âm lượng của nó trên hệ thống Windows.



*Hình 1.2 Lệnh tương tác với thiết bị loa và âm lượng*

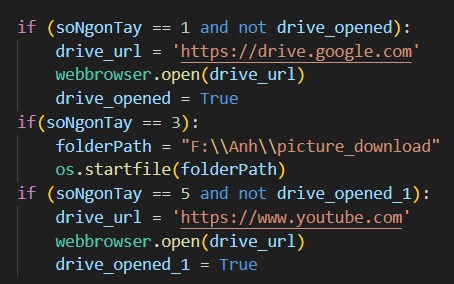
- Chúng em sẽ dùng khoảng cách giữa ngón cái và ngón trỏ để điều chỉnh thanh âm lượng.Để tính khoảng cách của 2 ngón này thì lấy trạng thái with và height để tính toán.Và cuối cùng là ứng nó vào với dải âm thanh máy chạy.Sử dụng thư viện NumPy để thực hiện việc chuyển đổi giá trị của biến doDaiDoanThang thành giá trị âm lượng tương ứng và sau đó áp dụng giá trị đó vào thiết bị âm thanh.:



*Hình 1.2 Lệnh chuyển đổi giá trị khoảng cách thành giá trị âm lượng*

**4.3 Tạo phím tắt thông qua các ngón tay**

- Từ kết quả nhận diện số đếm từ chức năng thứ nhất, vận dùng vào chức năng này.Tạo phím tắt ứng với mỗi ngón tay.Mỗi ngón tay sẽ chứa 1 phím tắt.Trong quá trình nhận diện bày tay mở hay đóng, nếu ngón mở, sẽ dẫn đến một liên kết hay một mục nào đó mà ta đã tạo phím tắt trước.Dưới đây là mô phỏng với 2 ngón tay:



*Hình 1.2 Lệnh tạo phím tắt*