Đầu tiên chúng ta cần cài đặt

Đặc biệt là thư viện medipife để có thể sử dụng đến **HandsLandmarks để có thể vẽ ra khung xương và so sánh các điểm len mắc**

**Bước 2 thiết lập các web cam để có thể đọc được các cử chỉ hành động vì dùng pc nên để là 0 còn các thiết bị khác thì để là 1**

**đọc các dữ liệu ảnh thông qua khung hình của cam**

**Sử dụng hàm HandsLandmarks Nó phát hiện các mốc bàn tay trong khung được cung cấp bằng mô hình Bàn tay MediaPipe.**

**đếm số ngón tay qua hàm Fingrs**

### **Đến với hàm detectHandsLandmarks dùng để đọc ảnh thông qua video của thiết bị và xây dựng các khung xương tương ứng và các điểm tương ứng với khung bàn tay**

* **Lập luận** :
  + **image**: Hình ảnh đầu vào nơi cần phát hiện các mốc bằng tay.
  + **hands**: Đối tượng MediaPipe Hands để xử lý.
  + **draw**(tùy chọn, mặc định=True): Cờ Boolean để cho biết có nên vẽ các mốc trên hình ảnh hay không.
  + **display**(tùy chọn, mặc định=True): Cờ Boolean để cho biết có hiển thị kết quả đầu ra bằng Matplotlib hay không.
* **Các bước chức năng** :
  + **Sao chép hình ảnh** : Sao chép hình ảnh đầu vào để tránh sửa đổi hình ảnh gốc.
  + **Convert Image to RGB** : Chuyển đổi hình ảnh từ BGR (định dạng màu mặc định của OpenCV) sang RGB, điều này cần thiết cho MediaPipe.
  + **Xử lý bàn tay** : Sử dụng **hands**đối tượng để xử lý hình ảnh RGB đã chuyển đổi và lấy các mốc bàn tay thông qua **hands.process(imgRGB)**.
  + **Vẽ cột mốc** : Nếu **draw**là Đúng và có các mốc bằng tay được phát hiện, nó sẽ vẽ các mốc trên hình ảnh đầu ra bằng cách sử dụng **mp\_drawing.draw\_landmarks**.
  + **Đầu ra hiển thị** : Sử dụng Matplotlib để hiển thị ảnh gốc và ảnh có dấu tay được phủ nếu **display**là True.