

TRƯỜNG ĐẠI HỌC QUỐC TẾ HỒNG BÀNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN

MÔN HỌC
TRÍ TUỆ NHÂN TẠO NÂNG CAO

Đề tài: Hand Tracking

Giảng viên hướng dẫn: Đỗ Văn Nhơn

Sinh viên thực hiện: Nguyễn Minh Vũ

Mã số sinh viên: 2211110063

TP. Hồ Chí Minh, 2024

LỜI CẢM ƠN

Để hoàn thành đề tài này, em đã nhận được sự hướng dẫn, giúp đỡ và góp ý nhiệt tình của quý thầy cô trường Đại Học Quốc Tế Hồng Bàng và thầy Đỗ Văn Nhơn.

Em xin gửi lời biết ơn sâu sắc đến thầy Đỗ Văn Nhơn đã dành nhiều thời gian và tâm huyết hướng dẫn nghiên cứu và giúp em hoàn thành môn học.

Em cũng xin chân thành cảm ơn đến quý thầy cô trường Đại học Quốc Tế Hồng Bàng, đặc biệt là những thầy cô đã tận tình dạy bảo cho em suốt thời gian học tập tại trường.

Em đã có nhiều cố gắng hoàn thiện dự án bằng tất cả năng lực của mình, tuy nhiên không thể tránh khỏi nhiều thiếu sót, rất mong nhận được những đóng góp quý báu của quý thầy cô và các bạn.

TP.HCM, ... Tháng ... Năm 2024

Người thực hiện

Nguyễn Minh Vũ

TRANG CAM KẾT

Tôi xin cam kết báo cáo thường kỳ này được hoàn thành dựa trên các kết quả thực hiện bài thực hành của tôi và các mã nguồn và kết quả này chưa được dùng cho bất cứ báo cáo của sinh viên nào khác.

TP.HCM, ngày ... tháng ... năm 2024

Người thực hiện

Nguyễn Minh Vũ

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

TP.HCM, Ngày ... Tháng ... Năm 2024

Chữ ký giảng viên

MỤC LỤC

LỜI CẢM ƠN 1

TRANG CAM KẾT	2
---------------------	---

NHẬN XÉT CỦA GIẢNG VIÊN	3
-------------------------------	---

1.1. Giới thiệu	6
-----------------------	---

1.1.1. Bối cảnh và lý do chọn đề tài	6
--	---

1.1.2. Mục tiêu	6
-----------------------	---

1.1.3. Phạm vi nghiên cứu	6
---------------------------------	---

1.2. Cơ sở lý thuyết	7
----------------------------	---

1.2.1. Theo dõi bàn tay và nhận dạng cử chỉ	7
---	---

1.2.2. Thư viện MediaPipe	7
---------------------------------	---

1.2.3. Xử lý hình ảnh với OpenCV	7
--	---

1.2.4. Ứng dụng vào phần cứng	7
-------------------------------------	---

1.3. Phương pháp thực hiện	7
----------------------------------	---

1.3.1. Sơ đồ tổng quát của hệ thống	7
---	---

1.3.2. Lựa chọn về cấu hình	8
-----------------------------------	---

1.3.3. Mô hình	9
----------------------	---

1.3.4. Các bước thực hiện chi tiết	9
--	---

1.4. Kết quả	10
--------------------	----

1.4.1. Kết quả nhận diện bàn tay	10
--	----

1.4.2. Nhận diện tay trái và tay phải	10
---	----

1.4.3. Nhận diện cử chỉ tay	11
-----------------------------------	----

1.4.4. Độ chính xác	12
---------------------------	----

1.4.5. Phần cứng	12
------------------------	----

1.5. Đánh giá và kết luận	12
---------------------------------	----

1.5.1. Đánh giá	12
-----------------------	----

1.5.2. Kết luận	12
-----------------------	----

1.6. Hướng phát triển	12
-----------------------------	----

1.6.1. Mở rộng hệ thống nhận diện cử chỉ phức tạp hơn	12
---	----

1.6.2. Tích hợp các hệ thống điều khiển	13
---	----

1.6.3. Cải thiện khả năng hoạt động trong điều kiện thực tế	13
1.6.4. Phát triển thêm sự nhận diện cánh tay	13
KẾT LUẬN	14

Tài liệu tham khảo

1. Python Machine Learning
https://www.w3schools.com/python/python_ml_getting_started.asp
2. AI ROBOT ARM sử dụng Python Arduino OpenCV CVZone - Murtaza's Workshop - Robotics and AI
<https://www.youtube.com/watch?v=7KV5489rL3c>
3. Hand Tracking - Murtaza's Workshop - Robotics and AI
https://www.youtube.com/watch?v=p5Z_GGRCI5s
4. Deep Learning Cơ Bản – Nguyễn Thanh Tuấn
[Sách Deep Learning cơ bản - v2.pdf](#)

1.1. Giới thiệu

1.1.1. Bối cảnh và lý do chọn đề tài

Trong thời đại công nghệ phát triển, việc tương tác giữa người và máy tính đang trở thành một lĩnh vực quan trọng. Một trong những phương pháp tương tác tiên tiến là thông qua các cử chỉ tay thay vì sử dụng chuột, bàn phím hoặc màn hình cảm ứng. Cử chỉ tay là cách tự nhiên để con người giao tiếp với nhau và nếu có thể được nhận dạng, máy tính sẽ mở ra nhiều ứng dụng mới như điều khiển từ xa, hỗ trợ người khuyết tật, hoặc thậm chí trong lĩnh vực thực tế ảo (AR/VR).

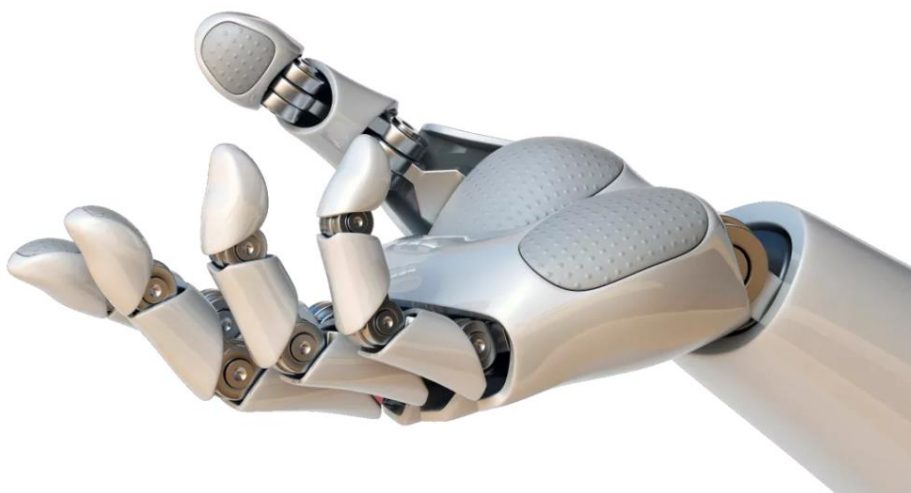
Hiện nay, hầu hết sự tương tác giữa con người với công nghệ thực tế ảo đều thông qua các thiết bị hỗ trợ như găng tay cảm biến, Vậy nên để có thể tương tác với công nghệ ảo mà không cần thiết bị hỗ trợ thì cần phải có một công nghệ thay thế.

Hand tracking là công nghệ được phát triển để theo dõi chuyển động và mô phỏng lại mô hình của bàn tay trong không gian. Việc này được thực hiện bằng cách phân tích các hình ảnh thông qua video. Và sau đó bàn tay được mô phỏng lại bằng mô hình 3D.

1.1.2. Mục tiêu

Mục tiêu của đồ án là xây dựng một hệ thống theo dõi và nhận diện cử chỉ tay trong thời gian thực bằng cách sử dụng camera và các thuật toán xử lý ảnh. Hệ thống này có thể nhận diện vị trí các ngón tay và lòng bàn tay, từ đó xác định các cử chỉ tay khác nhau.

Bên cạnh đó ứng dụng và phát triển phần cứng có tính năng nhận diện và phản hồi thao tác tay.



1.1.3. Phạm vi nghiên cứu

- Nghiên cứu cách phát hiện và theo dõi bàn tay trong video từ camera.
- Ứng dụng thư viện **MediaPipe** để theo dõi cử chỉ tay.

- Xây dựng hệ thống nhận diện cử chỉ tay và phản hồi lại bằng cách hiển thị hình ảnh tương ứng hoặc thực hiện các thao tác điều khiển.
- Xây dựng, sáng tạo, lắp ráp phần cứng tích hợp các tính năng trong ứng dụng phần mềm.

1.2. Cơ sở lý thuyết

1.2.1. Theo dõi bàn tay và nhận dạng cử chỉ

Theo dõi bàn tay (Hand Tracking) là quá trình nhận dạng và theo dõi các điểm chính trên bàn tay (landmarks), bao gồm các khớp ngón tay và lòng bàn tay. Khi đã theo dõi được các điểm này, ta có thể xác định trạng thái của bàn tay (mở, đóng, gập ngón, nắm đấm, v.v.).

1.2.2. Thư viện MediaPipe

- **MediaPipe** là một framework mã nguồn mở do Google phát triển, giúp đơn giản hóa quá trình phát triển các ứng dụng xử lý tín hiệu đa phương tiện, đặc biệt là xử lý ảnh và video.
- Trong trường hợp này, MediaPipe cung cấp một mô hình sẵn có để phát hiện bàn tay và trả về 21 điểm mốc chính (landmarks) trên bàn tay.

1.2.3. Xử lý hình ảnh với OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở được xây dựng để cung cấp một cơ sở hạ tầng chung cho các ứng dụng thị giác máy tính và để đẩy nhanh việc sử dụng nhận thức máy trong các sản phẩm thương mại. Là một sản phẩm được cấp phép BSD, OpenCV giúp các doanh nghiệp dễ dàng sử dụng và sửa đổi mã, cung cấp nhiều công cụ xử lý ảnh mạnh mẽ. OpenCV được sử dụng trong đồ án để thu thập dữ liệu từ camera, hiển thị kết quả và vẽ lên các hình ảnh.

1.2.4. Ứng dụng vào phần cứng

Xử lý và tích hợp chương trình đã thực hiện trước đó và áp dụng vào phần cứng (bàn tay robot). Phần cứng có các ứng dụng và tính năng của chương trình đã thực hiện trước đó.

1.3. Phương pháp thực hiện

1.3.1. Sơ đồ tổng quát của hệ thống

Input: Dữ liệu đầu vào từ camera. Xử lý bao gồm việc xoay hình ảnh, đổi kích thước, chuẩn hoá và chuyển đổi không gian màu.

Phát hiện bàn tay: Sử dụng MediaPipe để phát hiện bàn tay và các điểm landmark.

Nhận diện cử chỉ: Dựa trên vị trí các điểm landmark để xác định cử chỉ tay.

Output: Hiển thị ảnh hoặc thực hiện một hành động dựa trên cử chỉ tay.

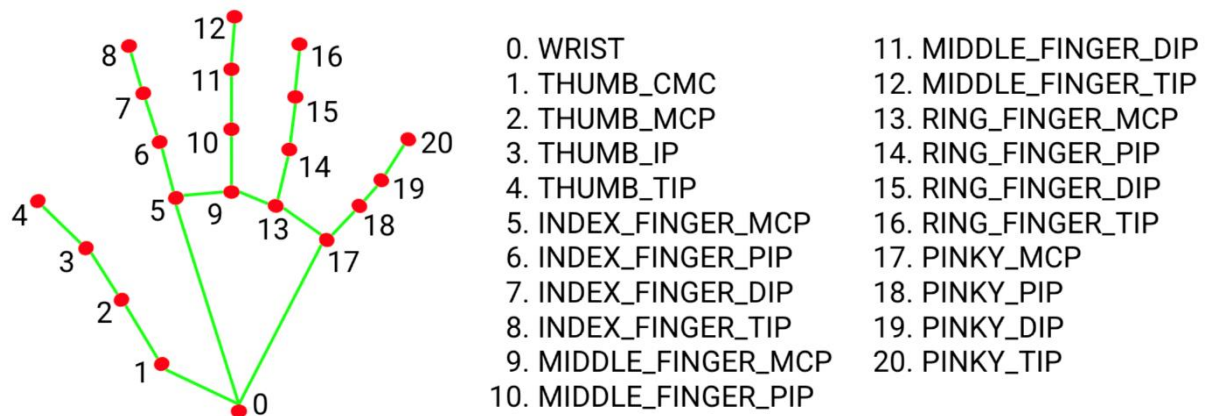
1.3.2. Lựa chọn về cấu hình

Tên lựa chọn	Nội dung mô tả	Khoảng giá trị	Giá trị mặc định
mode	Tham số này xác định liệu bộ phát hiện sẽ xử lý ảnh tĩnh (True) hay video (False). Khi True, nó xử lý từng khung hình độc lập, có thể chậm hơn nhưng chính xác hơn cho ảnh tĩnh.	{ True, False }	False
maxHands	Định nghĩa số lượng tối đa bàn tay có thể phát hiện trong một khung hình. Mặc định là 2, cho phép phát hiện cả hai bàn tay.	> 0	2
detectionCon	Độ tự tin tối thiểu để phát hiện bàn tay. Giá trị càng cao thì mô hình càng đòi hỏi độ chính xác cao hơn để xác định bàn tay.	> 0 && < 1	0.5
trackCon	Độ tự tin tối thiểu để tiếp tục theo dõi bàn tay sau khi phát hiện. Giá trị cao hơn sẽ khiến mô hình cẩn trọng hơn trong việc tiếp tục theo dõi nếu không chắc chắn.	> 0 && < 1	0.5
self.mpHands	Khởi tạo module phát hiện và theo dõi bàn tay của MediaPipe.		
self.hands	Cài đặt mô hình phát hiện bàn tay. Nó sử dụng các tham số được truyền vào		
self.mpDraw	Khởi tạo công cụ vẽ của MediaPipe, giúp vẽ các điểm mốc trên tay và các đường nối giữa các điểm mốc trên hình ảnh.		
img	Ảnh đầu vào (thường là khung hình từ webcam).		
draw	Cờ chỉ định có vẽ điểm mốc lên ảnh hay không.		
handNo	Số thứ tự của bàn tay bạn muốn lấy điểm mốc (nếu phát hiện nhiều bàn tay). Mặc định là 0 (bàn tay đầu tiên).		
lm.x, lm.y	ho tọa độ chuẩn hóa của điểm mốc (từ 0 đến 1), và các giá trị này được nhân với kích thước ảnh (w, h) để chuyển đổi thành tọa độ pixel (cx, cy).		

id	Tọa độ pixel và ID điểm mốc (id) được thêm vào danh sách lmList.
lmList	được trả về, chứa ID và tọa độ pixel của từng điểm mốc. Danh sách này giúp theo dõi vị trí của bàn tay hoặc sử dụng tọa độ cho các bước xử lý tiếp theo.
pTime, cTime	dùng để tính FPS (số khung hình trên giây).

1.3.3. Mô hình

Gói mô hình điểm mốc bàn tay phát hiện bản địa hoá điểm chính của 21 tọa độ khớp tay trong các vùng bàn tay đã phát hiện. Mô hình được huấn luyện về khoảng 30 nghìn hình ảnh thực tế, cũng như một số mô hình bàn tay tổng hợp kết xuất được đặt trên nhiều loại nền.



Gói mô hình điểm mốc kim đồng hồ chứa mô hình phát hiện lòng bàn tay và mô hình phát hiện điểm mốc trên tay. Mô hình phát hiện lòng bàn tay xác định tay trong hình ảnh đầu vào, còn mô hình phát hiện điểm mốc trên bàn tay xác định các dấu vết bàn tay cụ thể trên hình ảnh bàn tay bị cắt do mô hình phát hiện lòng bàn tay xác định.

Vì việc chạy mô hình phát hiện lòng bàn tay tốn nhiều thời gian, nên khi ở chế độ chạy video hoặc phát trực tiếp, Tay Cột mốc sẽ sử dụng hộp giới hạn do mô hình điểm đánh dấu kim đồng hồ xác định trong một khung hình để bản địa hoá vùng tay cho các khung hình tiếp theo. Tay mốc chỉ kích hoạt lại mô hình phát hiện lòng bàn tay nếu mô hình đánh dấu tay không còn xác định sự hiện diện của bàn tay hoặc không theo dõi được tay trong khung hình. Điều này giúp làm giảm số lần Hand Returner kích hoạt mô hình phát hiện lòng bàn tay.

1.3.4. Các bước thực hiện chi tiết

Bước 1: Thu thập hình ảnh từ camera

Sử dụng OpenCV để đọc hình ảnh từ camera. Các khung hình (frames) sẽ được xử lý lần lượt trong thời gian thực.

Bước 2: Phát hiện bàn tay

Dùng thư viện MediaPipe để phát hiện bàn tay trong khung hình. MediaPipe trả về tọa độ của 21 điểm trên bàn tay, giúp xác định chính xác vị trí các ngón tay và khớp nối.

Bước 3: Theo dõi vị trí các ngón tay

Sử dụng các tọa độ landmark từ MediaPipe để theo dõi chuyển động của các ngón tay. Các điểm này được dùng để nhận diện cử chỉ như số ngón tay giơ lên, nắm đấm hoặc mở tay.

Bước 4: Nhận diện tay trái và tay phải

Để phân biệt giữa tay trái và tay phải, so sánh vị trí của ngón cái với ngón trỏ dựa trên tọa độ của các điểm landmark. Điều này cho phép xác định xem tay nào đang xuất hiện trong khung hình.

Bước 5: Phản hồi cử chỉ

Khi nhận diện được một cử chỉ tay, hệ thống sẽ thực hiện các hành động tương ứng, ví dụ như hiển thị một ảnh tương ứng hoặc điều khiển một thiết bị.

Bước 6: Lắp ráp phần cứng

Lắp ráp các phần cứng cho bàn tay robot có các hoạt động cơ bản giống với bàn tay con người.

Bước 7: Cài đặt phần cứng

Sử dụng Arduino để cài đặt cách tính năng và môi trường cơ bản cần thiết. Sau đó nạp chương trình vào main board Arduino Uno.

Bước 8: Cài đặt kết nối với phần cứng

Thiết lập kết nối và đảm bảo phần cứng hoạt động ổn định.

1.4. Kết quả

1.4.1. Kết quả nhận diện bàn tay

Hệ thống nhận diện bàn tay và theo dõi chính xác các điểm landmark trong thời gian thực, với độ trễ thấp.

1.4.2. Nhận diện tay trái và tay phải

Hệ thống có khả năng phân biệt giữa tay trái và tay phải dựa trên vị trí của ngón cái và ngón trỏ.

Kiểm tra điểm đầu của ngón cái((4). THUMB-TIP) nằm bên trái hoặc phải so với điểm đầu của ngón trỏ((8). INDEX_FINGER-TIP). Nếu ngón cái nằm bên trái so với ngón trỏ thì đây là bàn tay phải và ngược lại.

1.4.3. Nhận diện cử chỉ tay

- Hệ thống có thể nhận diện một số cử chỉ cơ bản bằng cách
 - Đối với 4 ngón tay Trỏ, Giữa, Áp út và út
 - + Kiểm tra nếu điểm đầu của mỗi ngón mà nằm dưới (điểm đầu tiên - 2) thì kết luận ngón tay đó đang đóng
 - + Ví dụ: Đối với ngón trỏ
 - Nếu $((8). \text{INDEX_FINGER-TIP}) < ((6). \text{INDEX_FINGER-PIP})$ thì ngón tay đóng
 - Đối với ngón Cái tay phải
 - + Kiểm tra nếu điểm đầu của ngón mà nằm bên trái so với (điểm đầu tiên - 1) thì kết luận ngón tay đang đóng
 - + Ví dụ: Đối với bàn tay Trái
 - Nếu $((4). \text{THUMB_TIP}) < ((3). \text{THUMB_IP})$ tức nằm bên trái thì kết luận ngón tay mở. Hay $((4). \text{THUMB_TIP}) > ((3). \text{THUMB_IP})$ tức nằm bên phải thì kết luận ngón tay đóng.
- Sau khi hệ thống đã nhận diện được trạng thái và hình ảnh bàn tay
 - Tương ứng với kết quả nhận được ngón tay mở tương ứng là true (1), ngón tay đóng tương ứng là false (0). Từ đó nhận được một tập hợp List gồm 5 phần tử chỉ có thể là 0 hoặc 1. Kết quả của List thu được sẽ được so sánh với từng phần tử trong dictionary(dicthand).

```
dicthand = {
    (1, 0, 0, 0, 0): "/1.png", (0, 1, 0, 0, 0): "/2.png", (0, 0, 1, 0, 0):
"/3.png", (0, 0, 0, 1, 0): "/4.png", (0, 0, 0, 0, 1): "/5.png",
    (1, 1, 0, 0, 0): "/6.png", (1, 0, 1, 0, 0): "/7.png", (1, 0, 0, 1, 0):
"/8.png", (1, 0, 0, 0, 1): "/9.png", (0, 1, 1, 0, 0): "/10.png",
    (0, 1, 0, 1, 0): "/11.png", (0, 1, 0, 0, 1): "/12.png", (0, 0, 1, 1, 0):
"/13.png", (0, 0, 1, 0, 1): "/14.png", (0, 0, 0, 1, 1): "/15.png",
    (1, 1, 1, 0, 0): "/16.png", (1, 1, 0, 1, 0): "/17.png", (1, 1, 0, 0, 1):
"/18.png", (1, 0, 1, 1, 0): "/19.png", (1, 0, 1, 0, 1): "/20.png",
    (0, 1, 1, 1, 0): "/21.png", (0, 1, 1, 0, 1): "/22.png", (0, 1, 0, 1, 1):
"/23.png", (0, 0, 1, 1, 1): "/24.png", (1, 0, 0, 1, 1): "/25.png",
    (1, 1, 1, 1, 0): "/26.png", (1, 1, 1, 0, 1): "/27.png", (1, 1, 0, 1, 1):
"/28.png", (1, 0, 1, 1, 1): "/29.png", (0, 1, 1, 1, 1): "/30.png",
    (1, 1, 1, 1, 1): "/31.png", (0, 0, 0, 0, 0): "/32.png"
}
```

- Kết quả thu được sẽ trả về 1 hình ảnh tương ứng.

1.4.4. Độ chính xác

- Hệ thống hoạt động ổn định với độ chính xác cao trong điều kiện ánh sáng tốt.
- Một số trường hợp hệ thống gặp khó khăn trong điều kiện ánh sáng yếu hoặc khi bàn tay bị che khuất một phần.

1.4.5. Phần cứng

- Phần cứng hoạt động tốt các tính năng đã thiết lập.
- Hạn chế:
 - + Sai số kỹ thuật khi lắp ráp các linh kiện.
 - + Phần cứng hoạt động có phần chưa ổn định.



1.5. Đánh giá và kết luận

1.5.1. Đánh giá

- Hệ thống và phần cứng đã đáp ứng được yêu cầu cơ bản về phát hiện và theo dõi cử chỉ tay trong thời gian thực.
- Sử dụng MediaPipe giúp tăng tốc độ phát triển và đạt hiệu suất cao.
- Tuy nhiên, tổng thể cần được cải tiến để hoạt động tốt hơn trong môi trường có ánh sáng kém hoặc khi bàn tay bị xoay nghiêng. Hay phần cứng chưa thực sự ổn định.

1.5.2. Kết luận

Đề tài Hand Tracking là một ứng dụng thú vị và tiềm năng trong lĩnh vực tương tác người-máy. Hệ thống có thể được mở rộng để ứng dụng trong nhiều lĩnh vực như điều khiển thiết bị bằng cử chỉ, tương tác thực tế ảo, hoặc hỗ trợ người khuyết tật.

1.6. Hướng phát triển

1.6.1. Mở rộng hệ thống nhận diện cử chỉ phức tạp hơn

Hệ thống có thể mở rộng để nhận diện thêm nhiều cử chỉ phức tạp hơn, chẳng hạn như cử chỉ trong ngôn ngữ ký hiệu (Sign Language). Xoay hướng bàn tay.

1.6.2. Tích hợp các hệ thống điều khiển

Mở rộng khả năng ứng dụng của hệ thống bằng cách tích hợp nó với các thiết bị điều khiển như TV, máy tính, hoặc hệ thống nhà thông minh.

1.6.3. Cải thiện khả năng hoạt động trong điều kiện thực tế

Cải thiện khả năng nhận diện trong điều kiện ánh sáng yếu hoặc khi bàn tay không hoàn toàn nằm trong khung hình.

1.6.4. Phát triển thêm sự nhận diện cánh tay

Phát triển sự nhận diện không chỉ là bàn tay và các trường hợp cử chỉ toàn bộ cánh tay phức tạp hơn.

KẾT LUẬN

Em xin gửi lời biết ơn sâu sắc đến thầy Đỗ Văn Nhơn đã dành nhiều thời gian và tâm huyết hướng dẫn nghiên cứu và giúp em hoàn thành môn học.

Em đã có nhiều cố gắng hoàn thiện dự án bằng tất cả năng lực của mình, tuy nhiên không thể tránh khỏi nhiều thiếu sót, rất mong nhận được những đóng góp quý báu của quý thầy cô và các bạn.