

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

****

**BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**

**HỌC PHẦN: TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI:**

**NGHIÊN CỨU NHẬN DẠNG CỬ CHỈ BÀN TAY**

|  |  |
| --- | --- |
| Sinh viên thực hiện: | Dương Đức Tâm - 2022605730  Vũ Trọng Tấn- 2022605506  Nguyễn Công Thành - 2022600390 |
| Nhóm: | 14 |
| Lớp học phần: | 20241IT6094004 |
| Khoá: | 17 |
| Giảng viên hướng dẫn: | Trần Thanh Huân |

**Hà Nội, 2024**

|  |  |
| --- | --- |
| BỘ CÔNG THƯƠNG  TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập – tự do – hạnh phúc  -------o0o------- |

**PHIẾU GIAO ĐỀ TÀI**

**BÁO CÁO THỰC NGHIỆM/THÍ NGHIỆM**

Nhóm thực hiện: 1) Dương Đức Tâm

2) Vũ Trọng Tấn

3) Nguyễn Công Thành

Lớp học phần: 20241IT6094004 Khóa:K17 Khoa: Khoa Công nghệ thông tin

Tên đề tài: Nghiên cứu định dạng cử chỉ bàn tay

Mục đích: Công nghệ nhận diện cử chỉ bàn tay không chỉ mở ra cơ hội đổi mới trong các lĩnh vực hiện tại mà còn tạo tiền đề phát triển cho các ứng dụng mới trong tương lai. Việc kết hợp công nghệ này vào đời sống hàng ngày sẽ mang lại sự tiện ích, cải thiện năng suất và nâng cao trải nghiệm người dùng.

Yêu cầu:- Hoàn thành các nhiệm vụ của đề tài

-Sử dụng kỹ năng làm việc nhóm trong quá trình thực hiện báo cáo

-Trình bày báo cáo đúng yêu cầu, sử dụng kỹ năng “Viết báo cáo” khi thực hiện đề tài

-Sử dụng kỹ năng viết tài liệu kỹ thuật và phi kỹ thuật trong phần mở đầu của báo cáo

Kết quả thu được: Bản báo cáo đề tài (bản cứng và bản mềm); sản phẩm đề tài

Ngày giao đề tài: 20/11/2024

Ngày hoàn thành: 20/12/2024

Giảng viên hướng dẫn: Trần Thanh Huân

Hà nội, Ngày 06 Tháng 11 Năm 2024

GIẢNG VIÊN

Trần Thanh Huân

**PHIẾU PHÂN CÔNG NHIỆM VỤ**

Nhóm 14, gồm 3 thành viên

1. Dương Đức Tâm
2. Vũ Trọng Tấn

3) Nguyễn Công Thành

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Người thực hiện** | **Công việc** | **Kết quả đạt được** | **Nhận xét của GV** |
| **Tuần 1** | | | | |
| 1 | Dương Đức Tâm | Tìm hiểu về python, thư viện trên internet, giáo trình | Hoàn thành |  |
| 2 | Vũ Trọng Tấn | Tìm hiểu về python, thư viện trên internet, giáo trình | Hoàn thành |  |
| 3 | Nguyễn Công Thành | Tìm hiểu về python, thư viện trên internet, giáo trình | Hoàn thành |  |
| **Tuần 2** | | | | |
| 1 | Dương Đức Tâm | Tìm hiểu về CNN | Hoàn thành |  |
| 2 | Vũ Trọng Tấn | Viết chương trình, kiểm thử | Hoàn thành |  |
| 3 | Nguyễn Công Thành | Tối ưu hóa chương trình | Hoàn thành |  |
| **Tuần 3** | | | | |
| 1 | Vũ Trọng Tấn | 1. Tổng hợp báo cáo 2. Hoàn thành chương trình demo | 1. Hoàn thành báo cáo 2. Hoàn thành chuong trình demo |  |
| 2 | Dương Đức Tâm | Hỗ trợ làm nội dung báo cáo | Hoàn thành nội dung được phân công |  |
| 3 | Nguyễn Công Thành | Hỗ trợ làm nội dung báo cáo | Hoàn thành nội dung được phân công |  |

# **MỤC LỤC**

[LỜI CẢM ƠN 1](#_heading=h.2s8eyo1)

[MỤC LỤC 2](#_heading=h.17dp8vu)

[DANH MỤC HÌNH ẢNH 4](#_heading=h.26in1rg)

[LỜI NÓI ĐẦU 5](#_heading=h.1ksv4uv)

[MỞ ĐẦU 7](#_heading=h.44sinio)

[1. Lý do lựa chọn đề tài 7](#_heading=h.2jxsxqh)

[2. Mục tiêu đề tài 7](#_heading=h.z337ya)

[3. Phương pháp nghiên cứu 8](#_heading=h.3j2qqm3)

[4. Đối tượng và phạm vi nghiên cứu 8](#_heading=h.1y810tw)

[CHƯƠNG I: TỔNG QUAN 9](#_heading=h.2xcytpi)

[1. Khái niệm trí tuệ nhân tạo 9](#_heading=h.1ci93xb)

[2. Vai trò của trí tuệ nhân tạo 10](#_heading=h.3whwml4)

[3. Ứng dụng của trí tuệ nhân tạo 11](#_heading=h.2bn6wsx)

[CHƯƠNG II: TÌM HIỂU VỀ CNN VÀ CÁC THƯ VIỆN SỬ DỤNG 13](#_heading=h.qsh70q)

[1. Mạng CNN (Convolutional Neural Network) 13](#_heading=h.3as4poj)

[1.1. Giới thiệu chung 13](#_heading=h.1pxezwc)

[1.2. Convolutional (Tích chập) 13](#_heading=h.49x2ik5)

[1.3. Cấu trúc mạng CNN 16](#_heading=h.ihv636)

[1.3.1. Trường tiếp nhận cục bộ (local receptive field) 18](#_heading=h.1hmsyys)

[1.3.2. Trọng số chia sẻ (Shared Weight and Bias) 21](#_heading=h.1v1yuxt)

[1.3.3. Lớp tổng hợp (pooling layer) 21](#_heading=h.4f1mdlm)

[1.3.4. Cách chọn tham số cho CNN 23](#_heading=h.nmf14n)

[2. Các thư viện được sử dụng trong chương trình 23](#_heading=h.37m2jsg)

[2.1. OpenCV 23](#_heading=h.1mrcu09)

[2.2. MediaPipe 25](#_heading=h.46r0co2)

[2.3. Autopy 26](#_heading=h.2lwamvv)

[2.4. Numpy 27](#_heading=h.111kx3o)

2.5. Time 29

[CHƯƠNG III: PHÂN TÍCH ÁP DỤNG THUẬT TOÁN ĐỂ XỬ LÝ YÊU CẦU 30](#_heading=h.3l18frh)

[3.1. Đầu vào, đầu ra bài toán và hướng giải quyết 30](#_heading=h.206ipza)

[3.1.1. Đầu vào và đầu ra 30](#_heading=h.4k668n3)

[3.1.2. Hướng giải quyết bài toán 30](#_heading=h.2zbgiuw)

[3.2. Xây dựng các hàm 31](#_heading=h.1egqt2p)

[3.3. Kết quả đạt được 38](#_heading=h.25b2l0r)

[3.3.1 Di chuột 38](#_heading=h.kgcv8k)

[3.3.2 Click chuột trái 39](#_heading=h.1jlao46)

[3.4. Đánh giá hiệu năng 40](#_heading=h.3hv69ve)

[3.4.1. Phương Pháp Đánh Giá 40](#_heading=h.1x0gk37)

[3.4.2. Đánh Giá Hiệu Năng 40](#_heading=h.4h042r0)

# 

# **DANH MỤC HÌNH ẢNH**

# 

[Hình 2.1. Tích chập 14](#_heading=h.2p2csry)

[Hình 2.2. Giá trị tại ô 1x1 15](#_heading=h.147n2zr)

[Hình 2.3. Giá trị tại ô 2x1 15](#_heading=h.3o7alnk)

[Hình 2.4. Giá trị tại ô 1x1 16](#_heading=h.23ckvvd)

[Hình 2.5. Cấu trúc mạng CNN 17](#_heading=h.32hioqz)

[Hình 2.6. Trường tiếp nhận cục bộ 19](#_heading=h.41mghml)

[Hình 2.7. Trường tiếp nhận cục bộ (tiếp) 19](#_heading=h.2grqrue)

[Hình 2.8. Trường tiếp nhận cục bộ (tiếp) 20](#_heading=h.vx1227)

[Hình 2.9. Trường tiếp nhận cục bộ (tiếp) 21](#_heading=h.3fwokq0)

[Hình 2.10. Lớp tổng hợp 22](#_heading=h.19c6y18)

[Hình 2.11. Mô tả max pooling 22](#_heading=h.3tbugp1)

[Hình 2.12. Mô tả các lớp 23](#_heading=h.28h4qwu)

[Hình 3.1. Hàm khởi tạo ‘init’ 31](#_heading=h.3ygebqi)

[Hình 3.2. Hàm ‘findHands’ 32](#_heading=h.2dlolyb)

[Hình 3.3. Hàm findPosition 33](#_heading=h.sqyw64)

[Hình 3.4. Hàm fingersUp 33](#_heading=h.3cqmetx)

[Hình 3.5. Thiết lập tham số và đối tượng 34](#_heading=h.1rvwp1q)

[Hình 3.6. Hàm index\_finger\_flexed 35](#_heading=h.4bvk7pj)

[Hình 3.7. Hàm rightClick 35](#_heading=h.2r0uhxc)

[Hình 3.8. Hàm moving 36](#_heading=h.1664s55)

[Hình 3.9. Vòng lặp chính 37](#_heading=h.3q5sasy)

[Hình 3.10. Di chuột 38](#_heading=h.34g0dwd)

[Hình 3.11. Click chuột trái 39](#_heading=h.43ky6rz)

[Hình 3.12. Click chuột phải 39](#_heading=h.xvir7l)

# 

# **LỜI NÓI ĐẦU**

Ngày nay Trí tuệ nhân tạo (AI) đang có những tác động sâu sắc đến mọi khía cạnh của cuộc sống, từ cách chúng ta tương tác với thế giới xung quanh đến cách chúng ta làm việc và học tập. Một trong những ứng dụng nổi bật của AI là nhận dạng cử chỉ bàn tay.

Nhận dạng cử chỉ bàn tay là một công nghệ giúp máy tính nhận dạng các hành động của bàn tay từ đó điều khiển máy tính. Công nghệ này có nhiều ưu điểm so với các phương pháp điều khiển truyền thống, chẳng hạn như chuột và bàn phím, bao gồm:

● Tăng tính tự nhiên và linh hoạt trong tương tác với máy tính.

● Tăng khả năng tiếp cận cho người khuyết tật.

● Tăng khả năng tương tác trong môi trường thực tế ảo và tăng cường.

Trong báo cáo này, chúng tôi sẽ đề xuất một phương pháp nhận dạng cử chỉ bàn tay từ đó có thể khiển máy tính dựa trên công nghệ nhận dạng hình ảnh và học máy.

Giới thiệu về đề tài

Đề tài “Nghiên cứu nhận dạng cử chỉ bàn tay" nhằm mục đích xây dựng một ứng dụng cho phép người dùng điều khiển máy tính bằng cử chỉ của bàn tay. Ứng dụng sẽ sử dụng công nghệ nhận dạng hình ảnh và học máy để nhận dạng các cử chỉ của bàn tay và chuyển đổi chúng thành các thao tác trên máy tính.

Ứng dụng này có thể được ứng dụng trong nhiều lĩnh vực khác nhau, chẳng hạn như:

● Tương tác với máy tính trong môi trường thực tế ảo.

● Điều khiển các thiết bị điện tử gia dụng.

● Hỗ trợ người khuyết tật.

Cấu trúc của báo cáo

Báo cáo gồm 3 chương:

+ Chương 1: Tổng quan

Chương này sẽ giới thiệu tổng quan về đề tài, bao gồm: Lý do chọn đề tài, mục tiêu của đề tài, phạm vi của đề tài, các phương pháp nghiên cứu, …

+ Chương 2: Tìm hiểu về CNN và các thư viện được sử dụng

Chương này sẽ tìm hiểu về CNN, một loại mạng nơ-ron nhân tạo được sử dụng rộng rãi trong nhận dạng hình ảnh. Chương này cũng sẽ giới thiệu một số thư viện Python được sử dụng để xây dựng ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay.

+ Chương 3: Phân tích các áp dụng trong các thuật toán của chương II để xử lý yêu cầu của đề tài

Chương này sẽ phân tích các thuật toán CNN được sử dụng trong ứng dụng nghiên cứu nhận dạng cử chỉ bàn tay. Chương này cũng sẽ mô tả cách xây dựng ứng dụng dựa trên các thuật toán này.

- Kết thúc báo cáo là phần kết luận, tổng kết lại những nội dung chính của báo cáo.

# 

# **PHẦN MỞ ĐẦU**

## **1. Lý do lựa chọn đề tài**

Trong những năm gần đây, công nghệ nhận dạng cử chỉ đang phát triển mạnh mẽ, với nhiều ứng dụng trong thực tế. Trong đó, ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay là một trong những ứng dụng có tiềm năng để phát triển lớn.

Sử dụng cử chỉ bàn tay để điều khiển máy tính mang lại nhiều ưu điểm so với các phương pháp truyền thống như sử dụng chuột và bàn phím. Cụ thể, cử chỉ bàn tay mang lại sự tự nhiên, thoải mái, linh hoạt hơn cho người dùng ví dụ như chúng ta có thể điều khiển nhiều thiết bị điện tử cùng 1 lúc chỉ thông qua cử chỉ bàn tay mà không bị hạn chế bởi những thiết bị khác. Điều này đặc biệt phù hợp với những người khuyết tật, hạn chế khả năng vận động, người cao tuổi hoặc những người có nhu cầu sử dụng thiết bị điện tử linh hoạt hơn trong nhiều môi trường khác nhau.

Ngoài ra, ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay cũng có thể mở ra nhiều khả năng mới cho tương lai, chẳng hạn như:

● Tạo ra các giao diện người dùng tự nhiên và thân thiện hơn.

● Mang lại khả năng tương tác giữa người với thiết bị dễ dàng, linh hoạt hơn, thuận tiện hơn, nhiều chức năng hơn.

● Cho phép người dùng tương tác với máy tính một cách nhanh chóng và chính xác hơn.

● Mở rộng khả năng ứng dụng của máy tính trong các lĩnh vực như giải trí, giáo dục, chăm sóc sức khỏe,...

## **2. Mục tiêu đề tài**

Mục tiêu của đề tài là xây dựng một ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay. Ứng dụng này phải đáp ứng các yêu cầu sau:

Cho phép người dùng điều khiển các thao tác cơ bản trên máy tính như: đếm số ngón tay, di chuyển con trỏ, click chuột, mở ứng dụng,...

Có độ chính xác cao, không bị ảnh hưởng bởi các yếu tố môi trường như ánh sáng, âm thanh, vật cản, …

Tối ưu chức năng trên hầu hết các thiết bị.

## **3. Phương pháp nghiên cứu**

Để đạt được mục tiêu đề tài, các phương pháp nghiên cứu sau sẽ được sử dụng:

● Nghiên cứu tài liệu: Tìm hiểu các nghiên cứu trước đây về lĩnh vực nhận dạng cử chỉ và điều khiển máy tính thông qua cử chỉ bàn tay.

● Xử lý dữ liệu: Sử dụng các thư model đã được xây dựng sẵn để phân tích và nhận dạng cử chỉ bàn tay.

● Phát triển ứng dụng: Xây dựng ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay dựa trên kết quả phân tích và nhận dạng cử chỉ.

## **4. Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu của đề tài là các cử chỉ bàn tay của người dùng. Phạm vi nghiên cứu của đề tài tập trung vào việc xây dựng ứng dụng điều khiển máy tính thông qua các cử chỉ cơ bản như: đếm số ngón tay, di chuyển con trỏ, click chuột, mở ứng dụng,...

Tổng kết, đề tài "Xây dựng ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay" là một đề tài có tính ứng dụng cao và có tiềm năng phát triển trong tương lai. Đề tài này sẽ góp phần nâng cao khả năng tương tác giữa người dùng và máy tính, mang lại trải nghiệm sử dụng máy tính tự nhiên và thoải mái hơn.

# **CHƯƠNG I: TỔNG QUAN**

## **1. Khái niệm trí tuệ nhân tạo**

Trong lĩnh vực Công nghệ thông tin, Trí tuệ nhân tạo (AI) cũng có thể hiểu là “thông minh nhân tạo”, tức là sự thông minh của máy móc do con người tạo ra, đặc biệt tạo ra cho máy tính, robot, hay các máy móc có các thành phần tính toán điện tử. AI là một ngành mới, nhưng phát triển rất mạnh mẽ và đem lại nhiều kết quả to lớn. Mùa hè 1956, tại hội thảo ở Darmouth John McCarthy đã đưa ra thuật ngữ trí tuệ nhân tạo (Artificial Intelligence - AI). Mốc thời gian này được xem là thời điểm ra đời thực sự của lĩnh vực nghiên cứu AI.

AI là một lĩnh vực nghiên cứu của khoa học máy tính và khoa học tính toán nói chung. Có nhiều quan điểm khác nhau về AI. Do đó có nhiều định nghĩa khác nhau về lĩnh vực này. Sau đây là một số định nghĩa:

- “Sự nghiên cứu các năng lực trí tuệ thông qua việc sử dụng các mô hình tính toán” (Charniak và McDormott, 1985).

- “Nghệ thuật tạo ra các máy thực hiện các chức năng đòi hỏi sự thông minh khi được thực hiện bởi con người” (Kurzweil, 1990).

- “Lĩnh vực nghiên cứu tìm cách giải thích và mô phỏng các hành vi thông minh trong thuật ngữ các quá trình tính toán” (Schalkoff, 1990).

- “Sự nghiên cứu các tính toán để có thể nhận thức, lập luận và hành động” (Winston, 1992).

- “Một nhánh của khoa học máy tính liên quan đến sự tự động hóa các hành vi thông minh” (Luger and Stubblefield, 1993).

- “AI là sự nghiên cứu thiết kế các tác nhân thông minh” (Poole, Mackworth and Goebel, 1998).

**Trí tuệ nhân tạo (AI)** là một nhánh của khoa học máy tính tập trung vào việc tạo ra các hệ thống hoặc chương trình máy tính có khả năng mô phỏng các hành vi và suy nghĩ thông minh như con người.

#### **Nguồn gốc của AI:**

Thuật ngữ "Artificial Intelligence" được giới thiệu lần đầu vào năm 1956 tại Hội nghị Dartmouth do John McCarthy khởi xướng. Mục tiêu ban đầu của AI là nghiên cứu và xây dựng các hệ thống có khả năng tự học và tự giải quyết các vấn đề.

#### **Định nghĩa AI:**

AI có thể được hiểu qua hai khía cạnh:

1. **Khía cạnh chức năng:** Là khả năng của máy móc hoặc chương trình thực hiện các nhiệm vụ đòi hỏi trí thông minh, chẳng hạn như học tập, nhận thức, lập luận, và ra quyết định.
2. **Khía cạnh kỹ thuật:** Là các công nghệ, thuật toán và mô hình toán học được sử dụng để phát triển hệ thống thông minh.

#### **Các nhánh chính trong AI:**

1. **Machine Learning (Học máy):**
   * Tập trung vào việc phát triển các thuật toán giúp máy móc tự học từ dữ liệu mà không cần lập trình rõ ràng.
   * Các dạng phổ biến:
     + **Supervised Learning (Học có giám sát):** Dựa trên dữ liệu đầu vào và kết quả mong muốn.
     + **Unsupervised Learning (Học không giám sát):** Tìm kiếm cấu trúc ẩn trong dữ liệu không có nhãn.
     + **Reinforcement Learning (Học tăng cường):** Học thông qua thử nghiệm và phản hồi từ môi trường.
2. **Deep Learning (Học sâu):**
   * Là một nhánh của Machine Learning, sử dụng các mạng nơ-ron sâu để xử lý các dữ liệu phức tạp như hình ảnh, âm thanh, và video.
3. **Computer Vision (Thị giác máy tính):**
   * Phân tích và hiểu dữ liệu hình ảnh và video.
   * Ứng dụng: Nhận diện khuôn mặt, phân đoạn hình ảnh, theo dõi đối tượng.
4. **Natural Language Processing (Xử lý ngôn ngữ tự nhiên - NLP):**
   * Hiểu và tạo ra ngôn ngữ tự nhiên.
   * Ứng dụng: Chatbot, dịch tự động, phân tích cảm xúc.
5. **Robotics (Ngành robot):**
   * Phát triển các robot thông minh có thể cảm nhận và tương tác với môi trường xung quanh.
6. **Expert Systems (Hệ thống chuyên gia):**
   * Hệ thống sử dụng tri thức chuyên ngành để đưa ra quyết định hoặc giải quyết vấn đề cụ thể.

## **2. Vai trò của trí tuệ nhân tạo**

Trí tuệ nhân tạo nghiên cứu kỹ thuật làm cho máy tính có thể “suy nghĩ một cách thông minh” và mô phỏng quá trình suy nghĩ của con người khi đưa ra những quyết định, lời giải. Trên cơ sở đó, ta có thể thiết kế các chương trình cho máy tính để giải quyết bài toán.

Sự ra đời và phát triển của AI đã tạo ra một bước nhảy vọt về chất trong kỹ thuật và kỹ nghệ xử lý thông tin. Trí tuệ nhân tạo chính là cơ sở của công nghệ xử lý thông tin mới, độc lập với công nghệ xử lý thông tin truyền thống dựa trên văn bản giấy tờ. Điều này được thể hiện qua các mặt sau:

Nhờ những công cụ hình thức hoá (các mô hinh logic ngôn ngữ, logic mờ,...), các tri thức thủ tục và tri thức mô tả có thể biểu diễn được trong máy. Do vậy quá trình giải bài toán được thực hiện hiệu quả hơn.

Mô hình logic ngôn ngữ đã mở rộng khả năng ứng dụng của máy tính trong lĩnh vực đòi hỏi tri thức chuyên gia ở trình độ cao, rất khó như: y học, sinh học, địa lý, tự động hóa.

Một số phần mềm trí tuệ nhân tạo thể hiện tính thích nghi và tính mềm dẻo đối với các lớp bài toán thuộc nhiều lĩnh vực khác nhau.

Khi máy tính được trang bị các phần mềm trí tuệ nhân tạo, việc sử dụng mạng sẽ cho phép giải quyết những bài toán cỡ lớn và phân tán.

## **3. Ứng dụng của trí tuệ nhân tạo**

Trí tuệ nhân tạo (AI) đang có những tác động sâu sắc đến mọi khía cạnh của cuộc sống, từ cách chúng ta tương tác với thế giới xung quanh đến cách chúng ta làm việc và học tập. Dưới đây là một số ví dụ về các ứng dụng của AI trong thực tế:

* Quảng cáo và tiếp thị: AI được sử dụng để nhắm mục tiêu quảng cáo đến những người có khả năng quan tâm nhất, dựa trên dữ liệu về hành vi duyệt web, lịch sử mua sắm và sở thích của họ. AI cũng được sử dụng để tạo ra nội dung quảng cáo hấp dẫn hơn, chẳng hạn như video và quảng cáo trực tiếp.

Trợ lý ảo: AI giúp các trợ lý ảo hiểu và phản hồi các yêu cầu của người dùng một cách tự nhiên và thông minh hơn. Ví dụ, Siri của Apple có thể hiểu các câu hỏi và yêu cầu được nói bằng giọng nói tự nhiên, và Alexa của Amazon có thể điều khiển các thiết bị gia dụng thông minh.

Tìm kiếm và khám phá: AI được sử dụng để cải thiện hiệu quả của các công cụ tìm kiếm, chẳng hạn như Google Search và Bing. AI có thể hiểu ý định tìm kiếm của người dùng tốt hơn và đưa ra các kết quả phù hợp hơn.

Lọc thư rác: AI giúp lọc thư rác và lừa đảo ra khỏi hộp thư đến. AI có thể phát hiện các dấu hiệu của thư rác, chẳng hạn như từ ngữ và cụm từ cụ thể, cũng như các hành vi đáng ngờ, chẳng hạn như nhiều email được gửi từ cùng một địa chỉ IP.

Dịch ngôn ngữ: AI được sử dụng để dịch ngôn ngữ tự nhiên một cách tự động. Ví dụ, Microsoft Translator có thể dịch văn bản và giọng nói giữa hơn 100 ngôn ngữ.

Nhận dạng khuôn mặt và phân loại hình ảnh: AI được sử dụng để nhận dạng khuôn mặt và phân loại hình ảnh. Ví dụ, Face ID của Apple có thể mở khóa iPhone bằng cách nhận dạng khuôn mặt của người dùng.

Trò chơi: AI được sử dụng để tạo ra các trò chơi điện tử phức tạp và hấp dẫn hơn. Ví dụ, AlphaGo của Google đã đánh bại một kỳ thủ cờ vây chuyên nghiệp.

Tài chính: AI được sử dụng trong lĩnh vực tài chính để đưa ra các quyết định giao dịch thông minh, quản lý rủi ro và phát hiện gian lận.

Nông nghiệp: AI được sử dụng trong nông nghiệp để tối ưu hóa quy trình sản xuất. Ví dụ, AI có thể được sử dụng để dự đoán sản lượng cây trồng và tối ưu hóa việc sử dụng nước và phân bón.

An ninh mạng: AI được sử dụng để phát hiện và ngăn chặn các cuộc tấn công mạng. Ví dụ, AI có thể được sử dụng để phân tích lưu lượng mạng và phát hiện các hoạt động đáng ngờ.

Giáo dục: AI được sử dụng trong giáo dục để cá nhân hóa việc học tập và cung cấp các hỗ trợ bổ sung cho học sinh. Ví dụ, các gia sư ảo có thể cung cấp các bài tập và hướng dẫn được cá nhân hóa cho từng học sinh.

Những ứng dụng này chỉ là một số ví dụ điển hình về cách AI đang thay đổi thế giới của chúng ta. AI có tiềm năng to lớn để cải thiện cuộc sống của chúng ta trong nhiều lĩnh vực khác nhau, từ cách chúng ta làm việc đến cách chúng ta học tập và giải trí.

# **CHƯƠNG II: TÌM HIỂU VỀ CNN VÀ CÁC THƯ VIỆN SỬ DỤNG**

## **1. Mạng CNN (Convolutional Neural Network)**

### **1.1. Giới thiệu chung**

Mạng nơ-ron tích chập (CNN - Convolutional Neural Network) là một loại mạng nơ-ron nhân tạo được thiết kế đặc biệt để xử lý và phân tích dữ liệu có cấu trúc dạng lưới, chẳng hạn như hình ảnh và chuỗi thời gian. CNN được phát triển dựa trên ý tưởng mô phỏng cách hoạt động của thị giác con người, đặc biệt là khả năng nhận diện các đặc trưng quan trọng trong hình ảnh.

Nó đã đạt được thành công đáng kinh ngạc trong lĩnh vực thị giác máy tính. CNN có khả năng tự động học và rút trích các đặc trưng ý nghĩa từ dữ liệu đầu vào, giúp nó rất hiệu quả trong các nhiệm vụ như phân loại ảnh, phát hiện đối tượng và phân đoạn ảnh.

### **1.2. Convolutional (Tích chập)**

**Tích chập** (convolution) là một phép toán toán học được sử dụng để kết hợp hai hàm nhằm tạo ra một hàm thứ ba thể hiện cách một hàm biến đổi khi được áp dụng bởi hàm kia. Trong bối cảnh mạng nơ-ron tích chập (CNN), phép toán này được sử dụng để trích xuất các đặc trưng (features) từ dữ liệu đầu vào, chẳng hạn như hình ảnh.

Trong xử lý hình ảnh, tích chập được thực hiện bằng cách trượt một **kernel** (hoặc filter - một ma trận kích thước nhỏ) qua toàn bộ ma trận dữ liệu đầu vào (chẳng hạn ma trận pixel của hình ảnh). Tại mỗi vị trí, kernel thực hiện phép nhân từng phần tử với dữ liệu tương ứng trong vùng đó, sau đó tổng hợp kết quả.

Convolution là quá trình tích chập diễn ra bằng cách nhân các giá trị tại các vị trí tương ứng trong ma trận đầu vào và ma trận bộ lọc, sau đó cộng tất cả các kết quả lại. Kết quả của phép tích chập là một đầu ra mới, thường được gọi là Feature Map hoặc Convolved Feature.

Công thức tổng quát của tích chập 2D:

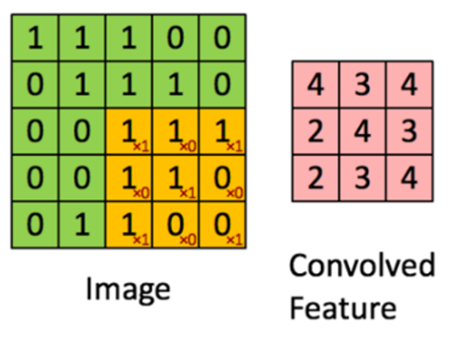


##### *Hinh 2.1 Công thức tổng quát tích châp 2D*

Trong đó:

* S(i,j): Giá trị của điểm đầu ra tại vị trí (i,j).
* K(m,n): Giá trị của kernel tại vị trí (m,n)
* I(i+m,j+n)): Giá trị của đầu vào tại vị trí tương ứng khi kernel trượt đến.
* ∗\*∗: Phép tích chập.

**Ví dụ minh họa :**

****

##### *Hình 2.2. Tích chập*

Trong ví dụ trên hình 2.1, ta có:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ma trận đầu vào (Image):  5x5   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 1 | 1 | 1 | 0 | 0 | | 0 | 1 | 1 | 1 | 0 | | 0 | 0 | 1 | 1 | 1 | | 0 | 0 | 1 | 1 | 0 | | 0 | 1 | 1 | 0 | 0 | | \* | Ma trận lọc (Filter):  3x3     |  |  |  | | --- | --- | --- | | 1 | 0 | 1 | | 0 | 1 | 0 | | 1 | 0 | 1 | | => | Convoled Feature  3x3     |  |  |  | | --- | --- | --- | | 4 | 3 | 4 | | 2 | 4 | 3 | | 2 | 3 | 4 | |

# 

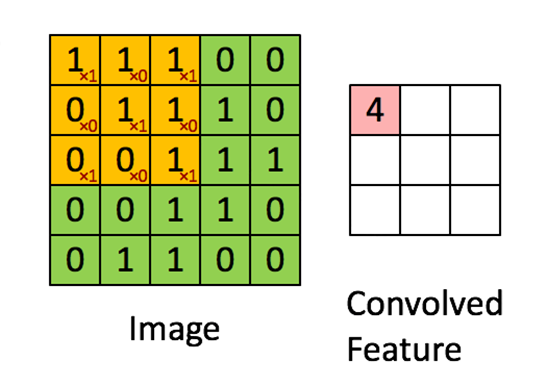
Trong đó :

· Ma trận đầu vào dạng 5x5 có giá trị pixel là 0 và 1

· Ma trận bộ lọc dạng 3x3

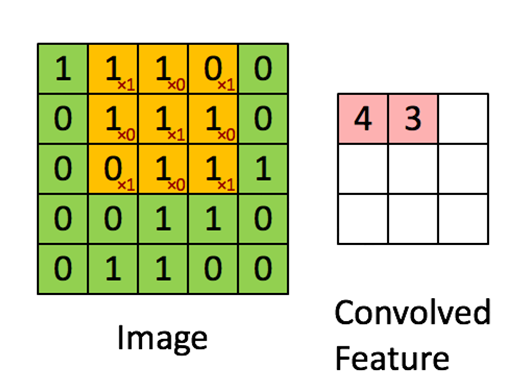
· Và lớp tích chập của ma trận hình ảnh 5x5 nhân với ma trận bộ lọc 3x3 gọi là Convoled Feature hay Feature Map.

Thực hiện tính toán ta được :

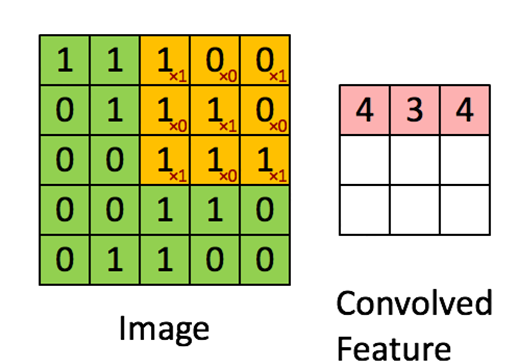


##### *Hình 2.3. Giá trị tại ô 1x1*

=> 1\*1 + 1\*0 + 1\*1 + 0\*0 + 1\*1 + 1\*0 + 0\*1 + 0\*0 + 1\*1 = 4

*Hình 2.3. Giá trị tại ô 2x1*

=> 1\*1 + 1\*0 + 0\*1 + 1\*0 + 1\*1 + 1\*0 + 0\*1 + 1\*0 +1\*1 = 3



##### *Hình 2.4. Giá trị tại ô 1x1*

=> 1\*1 + 0\*0 + 0\*1 + 1\*0 + 1\*1 + 0\*0 + 1\*1 + 1\*0 + 1\*1 = 4

Tương tự ta tiếp tục tính để ra được kết quả của Convolved Feature.

Phép tích chập có thể được thực hiện với nhiều bộ lọc khác nhau để trích xuất các đặc trưng khác nhau từ dữ liệu đầu vào. Nhờ tính chất chia sẻ tham số, phép tích chập giúp giảm số lượng tham số cần học trong mô hình và tạo ra các Feature Map có tính chất không gian (Spatial) và cấu trúc.

Trong lĩnh vực xử lý ảnh, tích chập được sử dụng trong các mạng Neural Convolution (CNN) để phân loại ảnh, nhận dạng vật thể, phân đoạn ảnh và nhiều ứng dụng khác. Ngoài ra, tích chập cũng có thể được áp dụng trong các lĩnh vực khác như xử lý âm thanh và xử lý ngôn ngữ tự nhiên.

### **1.3. Cấu trúc mạng CNN**

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (Feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các Layer liên kết được với nhau thông qua cơ chế Convolution.

Layer tiếp theo là kết quả Convolution từ Layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi Neuron ở lớp kế tiếp sinh ra từ kết quả của Filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các Filter khác nhau thông thường có hàng trăm hàng nghìn Filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số Layer khác như Pooling/Subsampling Layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (Training) CNN tự động học các giá trị qua các lớp Filter dựa vào cách thức mà bạn thực hiện. Ví dụ : trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các Filter tương ứng theo thứ tự Raw Pixel > Edges > Shapes > Facial > High-level Features. Layer cuối cùng được dùng để phân lớp ảnh.

##### *Hình 2.5. Cấu trúc mạng CNN*

Trong mô hình CNN có 2 khía cạnh cần quan tâm là **Tính bất biến** (Location Invariance) và **Tính kết hợp** (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (Translation, Rotation, Scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (Translation), phép quay (Rotation) và phép co giãn (Scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua Convolution từ các Filter.

Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

Mạng CNN sử dụng 3 ý tưởng cơ bản:

● Các trường tiếp nhận cục bộ (Local Receptive Field)

● Trọng số chia sẻ (Shared Weights)

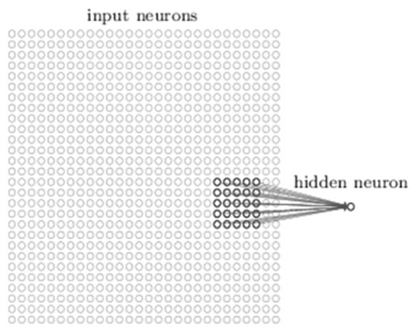
● Tổng hợp (Pooling).

# 

#### **1.3.1. Trường tiếp nhận cục bộ (local receptive field)**

Đầu vào của mạng CNN là một ảnh. Ví dụ như ảnh có kích thước 28×28 thì tương ứng đầu vào là một ma trận có 28×28 và giá trị mỗi điểm ảnh là một ô trong ma trận. Trong mô hình mạng ANN truyền thống thì chúng ta sẽ kết nối các neuron đầu vào vào tầng ảnh.

Tuy nhiên trong CNN chúng ta không làm như vậy mà chúng ta chỉ kết nối trong một vùng nhỏ của các Neuron đầu vào như một Filter có kích thước 5×5 tương ứng (28 - 5 + 1) 24 điểm ảnh đầu vào. Mỗi một kết nối sẽ học một trọng số và mỗi Neuron ẩn sẽ học một Bias. Mỗi một vùng 5×5 đấy gọi là một trường tiếp nhận cục bộ.



##### *Hình 2.6. Trường tiếp nhận cục bộ*

Một cách tổng quan, ta có thể tóm tắt các bước tạo ra 1 Hidden layer bằng các cách sau:

1. Tạo ra Neuron ẩn đầu tiên trong lớp ẩn 1

##### *Hình 2.7. Trường tiếp nhận cục bộ (tiếp)*

2. Dịch Filter qua bên phải một cột sẽ tạo được Neuron ẩn thứ 2.

##### *Hình 2.8. Trường tiếp nhận cục bộ (tiếp)*

Với bài toán nhận dạng ảnh người ta thường gọi ma trận lớp đầu vào là Input tensor (Image tensor), trọng số xác định các đặc trưng là Shared Weight và độ lệch xác định một Feature Map là Shared Bias. Như vậy đơn giản nhất là qua các bước trên chúng ta chỉ có 1 Feature Map. Tuy nhiên trong nhận dạng ảnh chúng ta cần nhiều hơn một Feature Map.

##### *Hình 2.9. Trường tiếp nhận cục bộ (tiếp)*

Như vậy, Local Receptive Field thích hợp cho việc phân tách dữ liệu ảnh, giúp chọn ra những vùng ảnh có giá trị nhất cho việc đánh giá phân lớp.

#### **1.3.2.** **Trọng số chia sẻ (Shared Weight and Bias)**

Đầu tiên, trong lớp tích chập, mỗi Filter (Kernel) có một tập hợp trọng số riêng. Khi Filter quét qua từng vùng nhỏ (patch) của hình ảnh, nó tính toán bằng cách nhân từng phần tử của Filter với giá trị tương ứng của vùng đó rồi cộng tổng lại.. Các trọng số của Filter là cố định trong suốt quá trình quét toàn bộ hình ảnh, nghĩa là cùng một bộ trọng số được áp dụng cho tất cả các vùng của ảnh

Trong một lớp tích chập, mỗi Filter cũng có một giá trị độ lệch riêng (Bias). Đây là giá trị này được cộng vào kết quả của phép tích chập để điều chỉnh đầu ra. Bias giúp điều chỉnh giá trị đầu ra, cung cấp cho mô hình khả năng thể hiện linh hoạt hơn, nhất là khi giá trị đầu ra cần dịch chuyển để phù hợp với dữ liệu thực tế.

Tóm lại, trọng số chia sẻ và độ lệch chia sẻ là những yếu tố quan trọng trong cấu trúc của CNN, giúp mô hình trở nên hiệu quả và mạnh mẽ hơn trong việc nhận diện và phân loại hình ảnh, đồng thời giảm thiểu số lượng tham số cần thiết để huấn luyện.

#### **1.3.3.** **Lớp tổng hợp (pooling layer)**

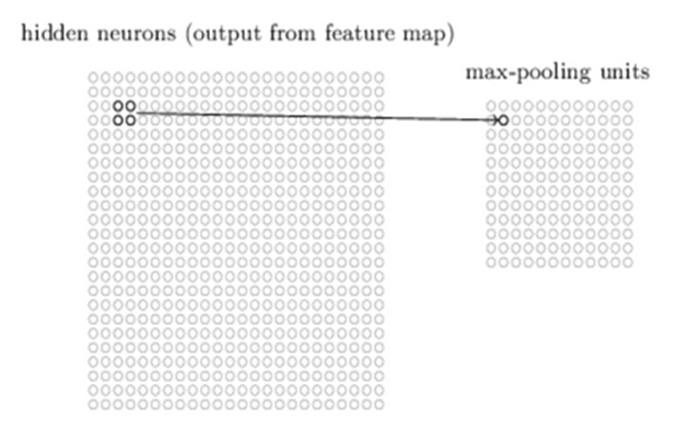
Lớp tổng hợp (Pooling layer) trong mạng nơ-ron tích chập (CNN) là một thành phần quan trọng, giúp giảm kích thước của đầu ra từ lớp tích chập trong khi vẫn giữ lại các đặc trưng quan trọng. Điều này giúp giảm số lượng tham số, tăng hiệu quả tính toán, và hạn chế hiện tượng quá khớp.

# 

##### *Hình 2.10. Lớp tổng hợp*

Loại lớp Pooling phổ biến là Max-pooling, thủ tục này lấy giá trị lớn nhất trong mỗi vùng con (patch) của ma trận đầu vào.

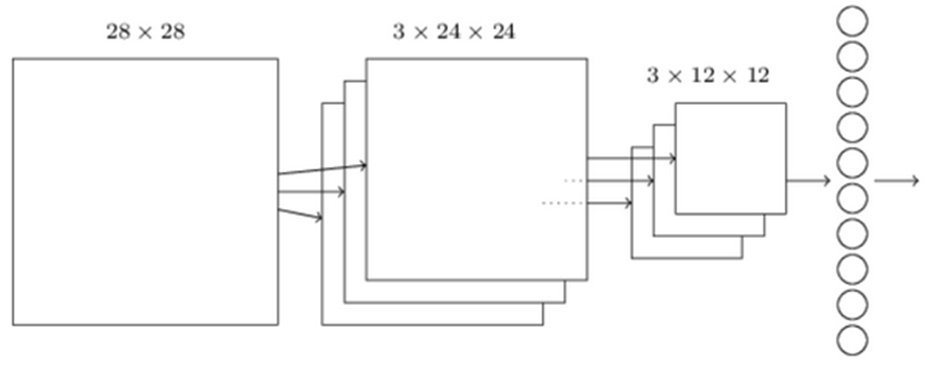
. Ví dụ, trong một vùng 2x2, max pooling sẽ trả về giá trị lớn nhất.



##### *Hình 2.11. Mô tả max pooling*

Như vậy qua lớp Max-Pooling thì số lượng neuron giảm đi phân nửa. Trong một mạng CNN có nhiều Feature Map nên mỗi Feature Map chúng ta sẽ cho mỗi Max-Pooling khác nhau. Chúng ta có thể thấy rằng Max-Pooling giữ lại các đặc trưng quan trọng nhất (như cạnh, góc, v.v.). Ngoài Max-Pooling còn có L2 Pooling.

Cuối cùng ta đặt tất cả các lớp lại với nhau thành một CNN với đầu ra gồm các neuron với số lượng tùy bài toán.



##### *Hình 2.12. Mô tả các lớp*

2 lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối (Fully Connected Layer) . Lớp này nối mọi nơron từ lớp Max pooled tới mọi nơron của tầng ra.

#### **1.3.4.** **Cách chọn tham số cho CNN**

● Số lớp nên được quyết định dựa trên độ phức tạp của dữ liệu. Với các bài toán đơn giản (ví dụ: nhận diện chữ số MNIST), 2-3 lớp là đủ. Đối với bài toán phức tạp hơn (như nhận diện hình ảnh trong ImageNet), số lớp có thể tăng lên 10-50 lớp.

● Kích thước Kernel: Phổ biến là 3x3 hoặc 5x5. Kernel nhỏ giữ thông tin chi tiết tốt và giảm số tham số, kernel lớn tóm tắt nhiều thông tin hơn trong mỗi bước, nhưng có thể làm mất thông tin chi tiết.

● Số lượng bộ lọc: Số lượng bộ lọc thường tăng theo từng lớp tích chập, ví dụ: 32, 64, 128. Bộ lọc càng nhiều, khả năng học được nhiều đặc trưng càng cao, nhưng tăng số bộ lọc đồng nghĩa với tăng chi phí tính toán.

● Bước trượt : Thường dùng 1 hoặc 2

● Hàm kích hoạt: ReLU là lựa chọn phổ biến

● Pooling size: thường là 2×2 hoặc 4×4 cho ảnh đầu vào lớn

● Optimizer : Adam Optimizer hoặc SGD với Momentum

● Cách cuối cùng là thực hiện nhiều lần việc train test để chọn ra được param tốt nhất.

## **2. Các thư viện được sử dụng trong chương trình**

### **2.1. OpenCV**

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở phổ biến trong lĩnh vực thị giác máy tính và xử lý hình ảnh. Nó được phát triển ban đầu bởi Intel vào năm 1999 và hiện được duy trì bởi một nhóm các nhà phát triển độc lập. OpenCV cung cấp một bộ công cụ và thuật toán để xử lý, phân tích và trích xuất thông tin từ hình ảnh và video.Dưới đây là một số điểm chính về OpenCV:

● Đặc điểm chính:

- Đa nền tảng: OpenCV hỗ trợ nhiều hệ điều hành như Windows, Linux, macOS và Android, cũng như các nền tảng nhúng.

- Ngôn ngữ lập trình: OpenCV cung cấp API cho các ngôn ngữ lập trình phổ biến như C++, Python, Java và MATLAB, cho phép lập trình viên sử dụng ngôn ngữ ưa thích của mình để phát triển ứng dụng.

- Công cụ và thuật toán: OpenCV cung cấp nhiều công cụ và thuật toán để xử lý hình ảnh và video, bao gồm lọc ảnh, biến đổi màu sắc, phát hiện đối tượng, nhận dạng khuôn mặt, theo dõi chuyển động, phân đoạn hình ảnh và nhiều tác vụ khác.

● Các chức năng chính của OpenCV:

- Xử lý hình ảnh: OpenCV cung cấp các chức năng để xử lý hình ảnh như chuyển đổi màu sắc, cắt, xoay, co giãn và thu phóng ảnh. Nó cũng hỗ trợ trích xuất đặc trưng và phát hiện biên cạnh.

- Xử lý video: OpenCV hỗ trợ xử lý video, bao gồm đọc và ghi video, trích xuất khung hình, thực hiện các phép biến đổi trên video và áp dụng các hiệu ứng video.

- Nhận dạng đối tượng: OpenCV cung cấp các thuật toán nhận dạng đối tượng như phát hiện khuôn mặt, phát hiện vật thể và nhận dạng biểu đồ. Nó cũng hỗ trợ phân loại và nhận dạng ký tự.

- Thị giác máy tính: OpenCV hỗ trợ các công cụ và thuật toán để xử lý và phân tích hình ảnh, bao gồm trích xuất đặc trưng, phân đoạn hình ảnh và phân loại hình ảnh.

● Sử dụng OpenCV:

- Cài đặt: Bạn có thể cài đặt OpenCV bằng cách tải xuống và cài đặt từ trang web chính thức của OpenCV hoặc sử dụng các công cụ quản lý gói như pip (Python) hoặc conda.

- Tài liệu và ví dụ: OpenCV cung cấp tài liệu chi tiết và ví dụ trên trang web chính thức của nó, bao gồm cả hướng dẫn sử dụng các chức năng và thuật toán khác nhau.

- Cộng đồng: OpenCV có một cộng đồng lớn và nhiều nguồn tài nguyên trên mạng như diễn đàn, blog và khoá học trực tuyến. Bạn có thể tham gia vào cộng đồng để tìm hiểu thêm về OpenCV, đặt câu hỏi và chia sẻ kiến thức với những người khác.

Tóm lại, OpenCV là một thư viện mã nguồn mở mạnh mẽ cho xử lý hình ảnh và thị giác máy tính. Nó cung cấp nhiều công cụ và thuật toán để thực hiện các tác vụ như xử lý hình ảnh, xử lý video, nhận dạng đối tượng và thị giác máy tính. Bạn có thể cài đặt OpenCV và tìm hiểu thêm thông qua tài liệu, ví dụ và cộng đồng sử dụng OpenCV trên trang web chính thức và các nguồn tài nguyên trực tuyến.

### **2.2. MediaPipe**

MediaPipe là một framework mã nguồn mở được phát triển bởi Google Research, hướng tới việc xây dựng các ứng dụng xử lý hình ảnh và video trực tiếp. Nó cung cấp các công cụ và thư viện để phân tích, nhận dạng và theo dõi các đối tượng trong ảnh và video. Dưới đây là một số thông tin chi tiết về MediaPipe

● Đặc điểm chính:

- Modular và linh hoạt: MediaPipe cho phép bạn xây dựng các ứng dụng thị giác máy tính theo cách modul và linh hoạt. Bạn có thể kết hợp các thành phần sẵn có để tạo ra các ứng dụng tùy chỉnh hoặc tạo các mô-đun mới.

- Đa nền tảng: MediaPipe hỗ trợ chạy trên nhiều nền tảng, bao gồm cả điện thoại di động, máy tính cá nhân và các hệ thống nhúng.

- Hiệu suất cao: Framework này được tối ưu hóa để đạt hiệu suất cao, cho phép xử lý hình ảnh và video thời gian thực.

- Hỗ trợ nhiều tác vụ: MediaPipe hỗ trợ nhiều loạt tác vụ như nhận dạng khuôn mặt, theo dõi chuyển động và động lực, nhận dạng điểm đặt tay, nhận dạng động tác, nhận dạng vị trí xương và nhiều tác vụ khác.

● Các thành phần chính:

- Calculator: Là một thành phần xử lý chính trong MediaPipe. Calculator có thể nhận đầu vào từ nguồn dữ liệu như camera hoặc video và thực hiện các phép tính xử lý để trích xuất thông tin hoặc thực hiện các tác vụ như nhận dạng và theo dõi.

- Packet: Là đơn vị dữ liệu cơ bản trong MediaPipe. Nó chứa thông tin được chuyển đổi và truyền giữa các thành phần trong pipeline.

- Graph: Là một biểu đồ biểu diễn luồng dữ liệu và quy trình xử lý thông qua các thành phần của MediaPipe. Một đồ thị bao gồm các Calculator và kết nối giữa chúng để xác định cách dữ liệu được xử lý và truyền qua các bước.

● Cách sử dụng:

- Ngôn ngữ lập trình: MediaPipe hỗ trợ nhiều ngôn ngữ lập trình như C++, Python và Java. Bạn có thể sử dụng ngôn ngữ lập trình ưu thích của mình để phát triển ứng dụng sử dụng MediaPipe.

- Xây dựng pipeline: Bạn có thể xây dựng pipeline xử lý dữ liệu trong MediaPipe bằng cách kết hợp các Calculator và thiết lập kết nối giữa chúng trong một đồ thị. Điều này cho phép bạn tùy chỉnh và xây dựng các ứng dụng thị giác máy tính theo nhu cầu của bạn.

- Triển khai: MediaPipe cung cấp công cụ và tài liệu để giúp bạn triển khai ứng dụng sử dụng MediaPipe trên các nền tảng khác nhau, bao gồm cả điện thoại di động và máy tính cá nhân.

MediaPipe đã được sử dụng rộng rãi trong nhiều ứng dụng thực tế như nhận dạng khuôn mặt trong thời gian thực, theo dõi chuyển động và xử lý hình ảnh động. Nó cung cấp một cách tiếp cận linh hoạt và mạnh mẽ để xây dựng các ứng dụng thị giác máy tính hiệu quả và đa nền tảng. Để tìm hiểu thêm về MediaPipe, bạn có thể truy cập trang web chính thức của dự án hoặc tìm kiếm các tài liệu, ví dụ và hướng dẫn sử dụng trên mạng.

### **2.3. Autopy**

AutoPy là một thư viện mã nguồn mở trong Python, được thiết kế để tự động hóa các thao tác trên máy tính, bao gồm di chuyển chuột, nhấp chuột, gõ phím và chụp ảnh màn hình. Thư viện này hỗ trợ các lập trình viên xây dựng các ứng dụng tự động hóa hoặc robot máy tính nhằm thực hiện các công việc lặp lại hoặc tương tác với giao diện người dùng một cách hiệu quả. Dưới đây là một số điểm chính về Autopy:

● **Tính năng chính**:

- **Điều khiển chuột:** Autopy cho phép bạn di chuyển chuột đến các tọa độ xác định trên màn hình, thay đổi tốc độ di chuyển và thực hiện các thao tác nhấp chuột như nhấn và giữ, nhấn nút trái/phải chuột.

- **Nhập dữ liệu bàn phím:** Bằng cách sử dụng Autopy, bạn có thể gửi các lệnh gõ phím đến máy tính như gõ các ký tự, nhấn phím mũi tên, phím chức năng và các phím tùy chỉnh.

- **Chụp và lưu màn hình:** Thư viện cung cấp khả năng chụp ảnh màn hình và lưu lại dưới dạng file hình ảnh, hỗ trợ chụp toàn màn hình hoặc một khu vực cụ thể

● **Hỗ trợ đa nền tảng:** AutoPy tương thích với nhiều hệ điều hành như Windows, macOS, và Linux, giúp bạn xây dựng các ứng dụng tự động hóa hoạt động trên nhiều môi trường khác nhau.

● **Sử dụng Autopy:**

- **Cài đặt:** Bạn có thể cài đặt Autopy bằng cách sử dụng công cụ quản lý gói cho từng hệ điều hành hoặc tải xuống mã nguồn và biên dịch thủ công.

- **Tài liệu và ví dụ:** Autopy cung cấp tài liệu chi tiết và ví dụ trên trang web chính thức của nó, giúp bạn nắm vững cách sử dụng và triển khai các chức năng của thư viện .

- **Cộng đồng:** Mặc dù cộng đồng của AutoPy không lớn như OpenCV, nhưng bạn vẫn có thể tìm thấy các tài liệu tham khảo, blog, hoặc hướng dẫn từ các diễn đàn lập trình và khóa học trực tuyến.

Tóm lại, AutoPy là một công cụ hữu ích cho các tác vụ tự động hóa máy tính, cung cấp những tính năng cơ bản nhưng thiết yếu, chẳng hạn như điều khiển chuột, gõ phím và chụp màn hình. Với khả năng hỗ trợ đa nền tảng và tài liệu chi tiết, thư viện này là một lựa chọn tuyệt vời cho những người mới bắt đầu hoặc đang cần một giải pháp tự động hóa nhẹ nhàng. Nếu bạn muốn tìm hiểu thêm, hãy tham khảo tài liệu chính thức và các nguồn tài nguyên trực tuyến để mở rộng kiến thức

### **2.4. Numpy**

NumPy (viết tắt của **Numerical Python**) là một thư viện mã nguồn mở trong Python, được thiết kế để xử lý dữ liệu số một cách hiệu quả. Thư viện này cung cấp cấu trúc dữ liệu mảng đa chiều mạnh mẽ cùng các công cụ tính toán khoa học, giúp nó trở thành nền tảng của hệ sinh thái khoa học dữ liệu trong Python. NumPy được ứng dụng rộng rãi trong các lĩnh vực như tính toán khoa học, học máy, xử lý hình ảnh, và xử lý tín hiệu. Dưới đây là một số điểm chính về NumPy:

- **Mảng đa chiều:** NumPy cung cấp một cấu trúc dữ liệu mảng đa chiều, được gọi là ndarray. Mảng NumPy có thể chứa các phần tử cùng kiểu dữ liệu và có kích thước cố định. Mảng này được tối ưu hóa để thực hiện các phép toán số học và logic nhanh chóng trên dữ liệu lớn.

- **Các chức năng và phép toán số học**: NumPy cung cấp một loạt các phép toán số học và logic trên mảng, bao gồm cộng, trừ, nhân, chia, lũy thừa, truy xuất phần tử, phép so sánh, phép logic và nhiều hơn nữa. Các phép toán này được thực hiện một cách hiệu quả và nhanh chóng trên toàn bộ mảng.

- **Hỗ trợ cho ma trận và phép toán ma trận:** NumPy cung cấp các chức năng và phép toán cho việc làm việc với ma trận, bao gồm khởi tạo ma trận, tính toán ma trận nghịch đảo, tính toán ma trận chuyển vị, nhân ma trận, tính toán giá trị riêng và vector riêng, và nhiều tác vụ khác liên quan đến ma trận.

- **Tích hợp với các thư viện khác:** NumPy tích hợp chặt chẽ với các thư viện khoa học dữ liệu khác như SciPy (thư viện toán học và khoa học), Matplotlib (thư viện vẽ đồ thị) và Pandas (thư viện xử lý dữ liệu). Sự tích hợp này giúp xử lý dữ liệu và thực hiện phân tích số liệu một cách hiệu quả.

- **Tốc độ và hiệu suất:** NumPy được viết bằng ngôn ngữ C và C++, nhờ đó các phép toán trên mảng trong NumPy có hiệu suất vượt trội so với các vòng lặp trong Python thuần. Điều này đặc biệt quan trọng khi làm việc với dữ liệu lớn.

● Sử dụng NumPy:

- **Cài đặt:** NumPy được cài đặt thông qua công cụ quản lý gói như pip (Python Package Index) hoặc conda.

- **Tài liệu và ví dụ:** NumPy cung cấp tài liệu chi tiết và ví dụ trên trang web chính thức của nó, giúp lập trình viên nắm vững cách sử dụng các chức năng và phép toán của thư viện.

- **Cộng đồng:** NumPy có một cộng đồng lớn và sôi động, với nhiều nguồn tài nguyên trực tuyến như diễn đàn, blog và tài liệu hướng dẫn. Bạn có thể tìm kiếm thông tin và giải đáp các câu hỏi cụ thể từ cộng đồng này.

Tóm lại, NumPy là một thư viện cốt lõi trong hệ sinh thái khoa học dữ liệu Python, cung cấp các công cụ mạnh mẽ để làm việc với mảng và thực hiện các phép toán nhanh chóng, hiệu quả. Với sự hỗ trợ rộng rãi từ cộng đồng và khả năng tích hợp với các thư viện khác, NumPy là lựa chọn lý tưởng cho các dự án liên quan đến xử lý và phân tích dữ liệu. Bạn có thể bắt đầu khám phá NumPy thông qua tài liệu chính thức và các ví dụ minh họa có sẵn.

### **2.5.Time**

  Thư viện time trong Python là một module chuẩn, cung cấp một số lượng lớn các hàm liên quan đến việc xử lý thời gian. Nó cho phép lập trình viên đo thời gian, trì hoãn thực thi chương trình, chuyển đổi giữa các định dạng thời gian khác nhau và làm việc với thời gian địa phương cũng như thời gian chuẩn quốc tế (UTC). Dưới đây là một số ứng dụng chính về time:

- **Lấy thời gian hiện tại:** Thư viện time cung cấp hàm time() để lấy thời gian hiện tại dưới dạng Unix timestamp (số giây tính từ ngày 1 tháng 1 năm 1970).

- **Chuyển đổi thời gian:** Các hàm như localtime(), gmtime(), và strftime() cho phép chuyển đổi giữa Unix timestamp và các định dạng thời gian khác nhau, bao gồm cả thời gian địa phương và thời gian UTC.

- **Tạo độ trễ:** Hàm sleep() được sử dụng để tạo độ trễ trong chương trình, tức là dừng chương trình lại trong một khoảng thời gian nhất định.

- **Đo thời gian thực thi:** Thư viện time cung cấp các công cụ để đo thời gian thực thi của một đoạn mã. Bằng cách lấy thời gian trước và sau khi thực hiện đoạn mã và tính toán chênh lệch, ta có thể xác định được thời gian cần thiết để thực hiện đoạn mã đó.

· Ứng dụng của thư viện time:

o **Quản lý thời gian thực thi**: Đo lường hiệu suất của các đoạn mã.

o **Đồng bộ hóa**: Tạo độ trễ để đồng bộ hóa các sự kiện hoặc tác vụ trong chương trình.

o **Chuyển đổi thời gian**: Làm việc với thời gian địa phương và thời gian quốc tế (UTC).

o **Định dạng thời gian**: Hiển thị thời gian theo các định dạng khác nhau, đáp ứng yêu cầu của người dùng hoặc hệ thống.

Thư viện time đóng vai trò quan trọng trong việc xử lý thời gian trong lập trình Python, từ việc đo thời gian thực thi, tạo độ trễ, đến chuyển đổi và định dạng thời gian theo yêu cầu của ứng dụng

# 

# **CHƯƠNG III: PHÂN TÍCH ÁP DỤNG THUẬT TOÁN ĐỂ XỬ LÝ YÊU CẦU**

## **3.1. Đầu vào, đầu ra bài toán và hướng giải quyết**

### **3.1.1. Đầu vào và đầu ra**

**Đầu vào:**

● Dữ liệu đầu vào chính là video từ camera máy tính, hình ảnh từ camera được xử lý bởi Mediapipe Hand Tracking Module.

● OpenCV được sử dụng để đọc video và xử lý các frame.

● Module này nhận diện và trích xuất các điểm đặc trưng trên bàn tay, chẳng hạn như các đầu ngón tay.

**Đầu ra:** Sử dụng vị trí đầu ngón tay trỏ để điều khiển vị trí chuột, ngón giữa để click chuột trái và ngón áp út để click chuột phải

### **3.1.2. Hướng giải quyết bài toán**

**Nhận diện bàn tay:** Sử dụng Mediapipe Hand Tracking Module để nhận diện và trích xuất các điểm đặc trưng trên bàn tay từ video camera.

**Xác định hành động:** Dựa vào vị trí của các điểm đặc trưng, xác định các hành động như di chuyển chuột, click trái và click phải.

**Sử dụng Autopy:** Autopy được sử dụng để thực hiện các hành động về chuột trực tiếp trên máy tính.

**Tương tác dựa trên ngón tay:** Sử dụng trạng thái của các ngón tay để kích thích các hành động chuột. Ví dụ ấn ngón áp út để click chuột phải.

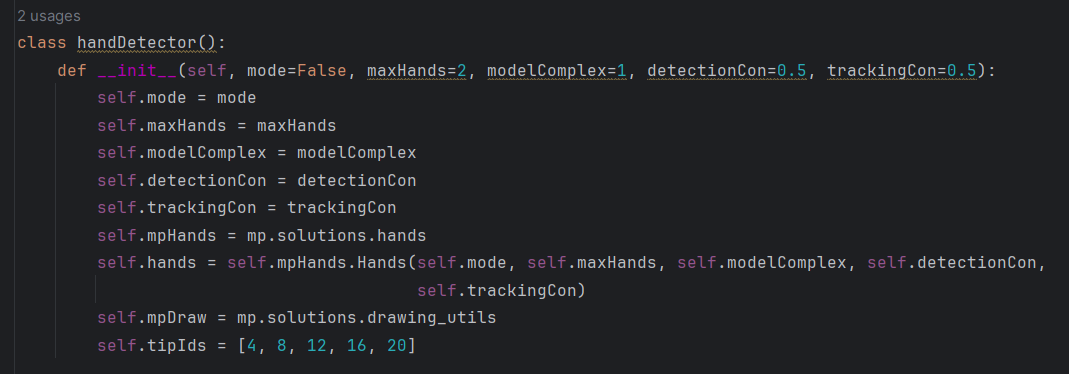
**Điều khiển mượt mà:** Sử dụng hệ số làm mịn (smoothening) để làm mịn chuyển động của chuột và giảm độ nhảy.

**Chạy liên tục:** Sử dụng một vòng lặp để liên tục đọc và xử lý video từ camera, cung cấp trải nghiệm điều khiển chuột liên tục.

Giải pháp này sử dụng việc nhận diện và theo dõi bàn tay để tương tác với máy tính. Hành động di chuyển chuột và click chuột được thực hiện dựa trên trạng thái của các ngón tay, và Autopy được sử dụng để thực hiện các hành động này trực tiếp trên máy tính.

## **3.2. Xây dựng các hàm**

**Hàm khởi tạo ‘\_\_init\_\_’**



##### *Hình 3.1. Hàm khởi tạo ‘init’*

● Hàm khởi tạo của class, thiết lập các tham số như chế độ (mode), số lượng tay tối đa (maxHands), độ phức tạp của mô hình (modelComplex), ngưỡng nhận diện (detectionCon), và ngưỡng theo dõi (trackingCon).

● Khởi tạo đối tượng Hands từ thư viện mediapipe để sử dụng trong việc nhận diện bàn tay.

● mpDraw được sử dụng để vẽ các landmark và đường kết nối trên bàn tay.

**Hàm ‘findHands’**



##### *Hình 3.2. Hàm ‘findHands’*

● Hàm này nhận một ảnh và thực hiện việc nhận diện bàn tay trên ảnh đó.

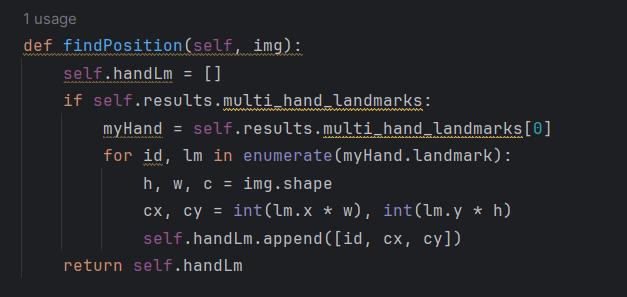
● Chuyển đổi ảnh sang định dạng RGB.

● Sử dụng self.hands.multi\_hand\_landmarks để xử lý ảnh và nhận diện bàn tay.

● Nếu draw là True, hàm sẽ vẽ các landmarks và đường kết nối trên bàn tay trên ảnh.

● Trả về ảnh sau khi xử lý.

**Hàm findPosition**

****

##### *Hình 3.3. Hàm findPosition*

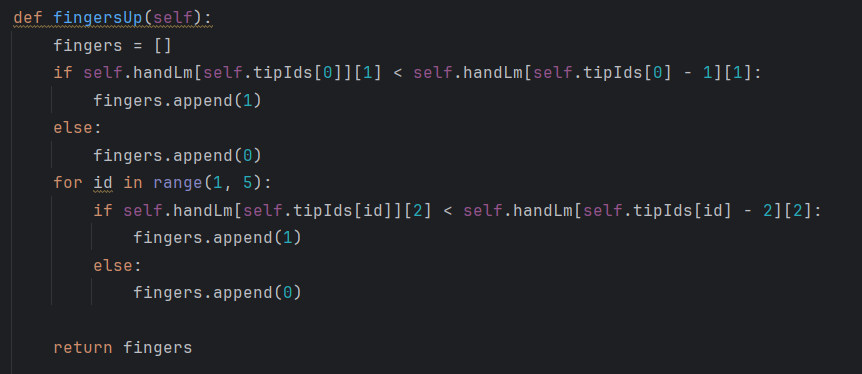
● Hàm này trả về vị trí của các điểm landmark trên bàn tay.

● Lấy tất cả các landmarks của tay đầu tiên (được giả định là chỉ có một tay được nhận diện).

● Chuyển đổi tọa độ từ tỷ lệ (0-1) sang tọa độ pixel trên ảnh.

● Trả về danh sách các điểm landmark.

**Hàm fingersUp**



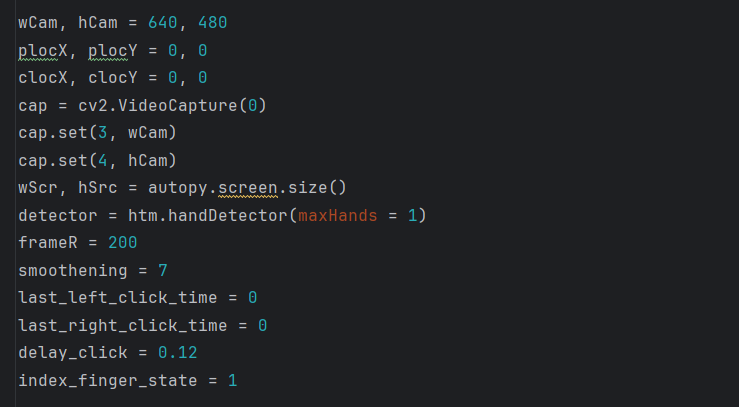
##### *Hình 3.4. Hàm fingersUp*

● Hàm này xác định xem các ngón tay có đang nằm lên không dựa trên vị trí của các điểm landmark.

● Kiểm tra vị trí của các điểm cuối cùng của ngón tay so với các điểm trước đó để xác định ngón tay có đang nằm lên không.

● Trả về một danh sách, mỗi phần tử tương ứng với một ngón tay và có giá trị 1 nếu ngón tay nằm lên và 0 nếu không.

**Thiết lập các tham số và đối tượng**



##### *Hình 3.5. Thiết lập tham số và đối tượng*

● wCam, hCam: Chiều rộng và chiều cao của webcam.

● plocX, plocY: Vị trí trước đó của chuột.

● clocX, clocY: Vị trí hiện tại của chuột.

● cap: Đối tượng VideoCapture để kết nối với webcam.

● wScr, hScr: Kích thước của màn hình.

● detector: Đối tượng handDetector từ module HandTrackingModule (htm).

● frameR: Bán kính để xác định khu vực nhận diện tay.

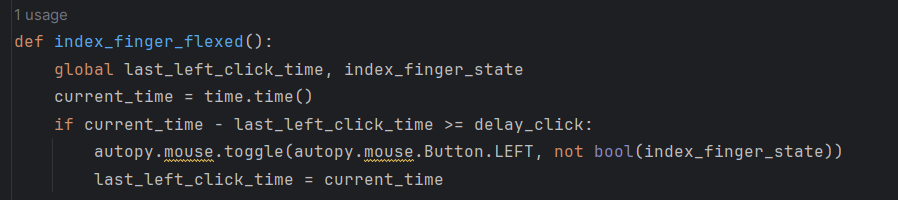
● smoothening: Hệ số làm mịn cho việc chuyển động của chuột.

● last\_left\_click\_time, last\_right\_click\_time: Thời điểm cuối cùng khi nhấn nút trái và phải của chuột.

● delay\_click: Thời gian chờ giữa các lần nhấn chuột.

● index\_finger\_state: Trạng thái của ngón tay áp út.

**Hàm index\_finger\_flexed**

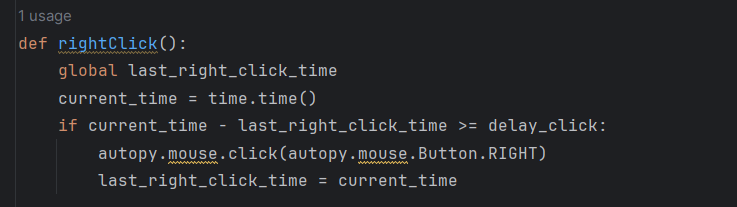


##### *Hình 3.6. Hàm index\_finger\_flexed*

● Hàm này kiểm tra xem ngón áp út có đang nằm lên hay không.

● Nếu có và đã đủ thời gian chờ, thì thực hiện toggle (chuyển đổi trạng thái) của nút chuột trái.

**Hàm rightClick**

****

##### *Hình 3.7. Hàm rightClick*

● Hàm rightClick có chức năng thực hiện một click chuột nếu đã đủ thời gian chờ giữa các lần click.

● global last\_right\_click\_time: Khi sử dụng global, biến last\_right\_click\_time được coi là biến toàn cục, tức là nó không chỉ tồn tại trong phạm vi của hàm rightClick mà còn được sử dụng và duy trì giá trị ở phạm vi bên ngoài của hàm.

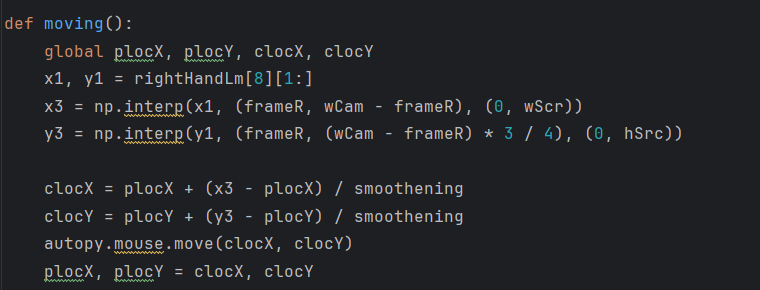
● current\_time = time.time(): Lấy thời điểm hiện tại sử dụng hàm time.time() và gán cho biến current\_time. Điều này sẽ đại diện cho thời điểm hiện tại khi hàm rightClick được gọi.

● if current\_time - last\_right\_click\_time >= delay\_click:: Kiểm tra xem đã đủ thời gian chờ giữa các lần click chuột phải chưa. delay\_click là thời gian tối thiểu giữa hai lần click.

● autopy.mouse.click(autopy.mouse.Button.RIGHT): Nếu đã đủ thời gian chờ, sử dụng autopy.mouse.click để thực hiện một click chuột phải.

● last\_right\_click\_time = current\_time: Gán giá trị của current\_time cho last\_right\_click\_time, cập nhật thời điểm cuối cùng khi click chuột phải.

**Hàm moving**

****

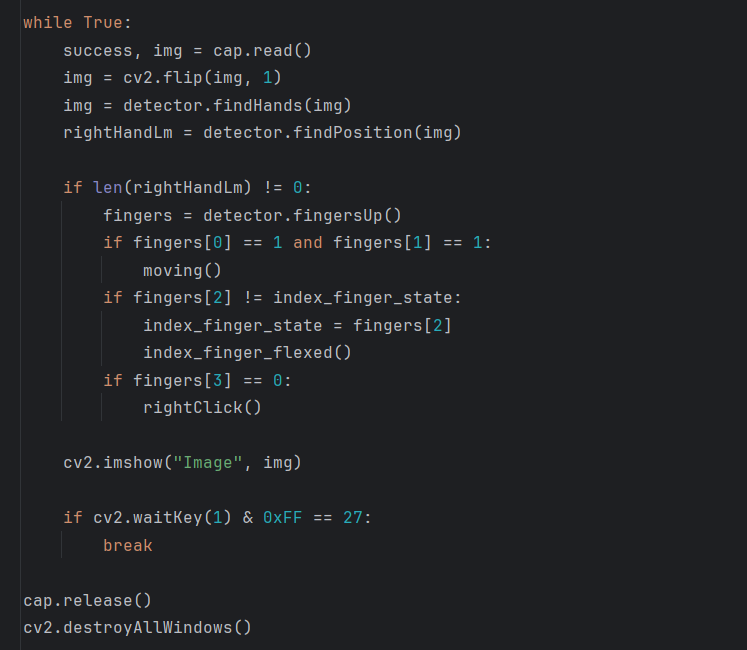
##### *Hình 3.8. Hàm moving*

● Hàm này tính toán vị trí mới của chuột dựa trên vị trí của ngón tay cái.

● Sử dụng np.interp để chuyển đổi vị trí tương đối của ngón tay thành vị trí trên màn hình.

● Áp dụng hệ số smoothening để giảm độ nhảy của chuột và di chuyển chuột đến vị trí mới.

**Vòng lặp chính**

****

##### *Hình 3.9. Vòng lặp chính*

● Trong vòng lặp, ảnh từ webcam được đọc và lật ngược.

● detector.findHands được sử dụng để nhận diện và vẽ các landmarks của bàn tay.

● Nếu có ít nhất một landmarks của bàn tay được nhận diện, hàm detector.fingersUp được gọi để xác định trạng thái của các ngón tay.

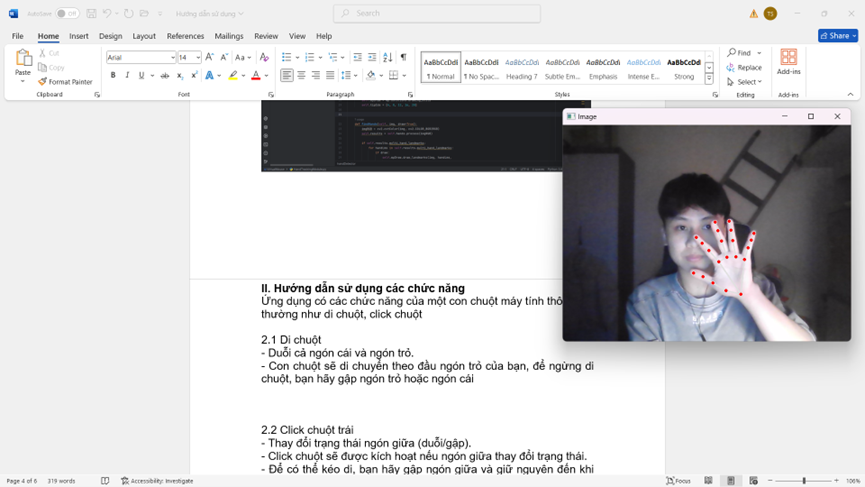
● Dựa vào trạng thái của ngón tay, có thể thực hiện di chuyển chuột (moving), thực hiện click chuột trái (index\_finger\_flexed), và thực hiện click chuột phải (rightClick).

● Ảnh với các landmarks và chuột được hiển thị, và chương trình kết thúc khi nhấn phím Esc.

## **3.3. Kết quả đạt được**

Ứng dụng có các chức năng của một con chuột máy tính thông thường như di chuột, click chuột

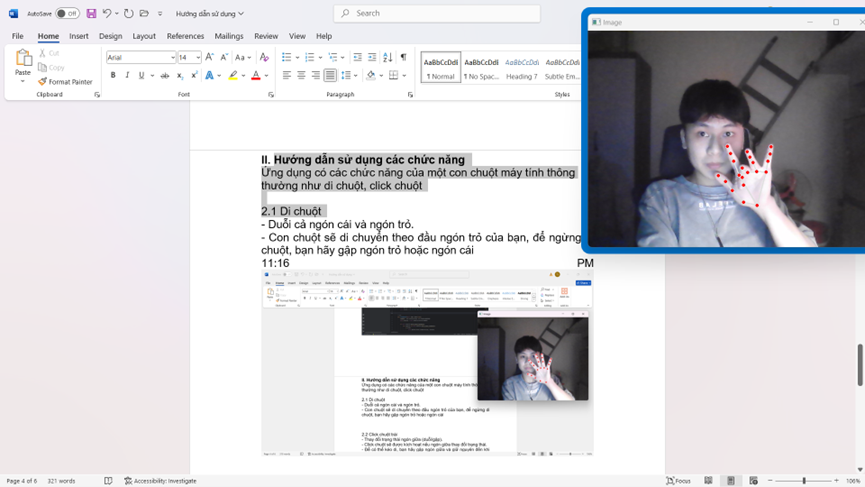
### **3.3.1 Di chuột**



##### *Hình 3.10. Di chuột*

Con chuột sẽ di chuyển theo đầu ngón trỏ của bạn, để ngừng di chuột, bạn hãy gập ngón trỏ hoặc ngón cái

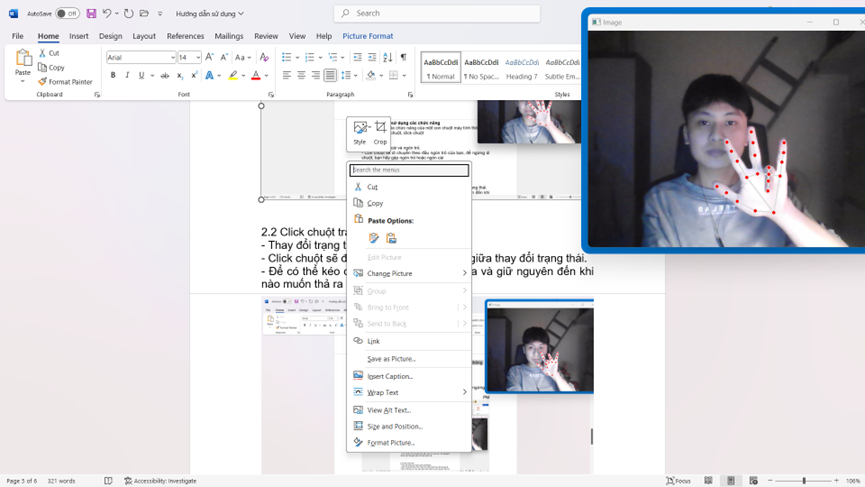
### **3.3.2 Click chuột trái**



##### *Hình 3.11. Click chuột trái*

Để click chuột trái, bạn chỉ cần gập ngón giữa lại, đề có thể kéo di, bạn hãy gập ngón giữa và giữ nguyên đến khi nào muốn thả ra

**3.3.3 Click chuột phải**

**

*Hình 3.12. Click chuột phải*

Tương tự như click chuột trái, khi muốn click chuột phải, bạn chỉ cần gập ngón áp út

## **3.4. Đánh giá hiệu năng**

### **3.4.1. Phương Pháp Đánh Giá**

**Tính Chính Xác của Nhận Diện Cử Chỉ**

- Ưu Điểm: Các cử chỉ của bàn tay được nhận diện một cách chính xác, đặc biệt là trong việc theo dõi ngón tay.

- Khuyết Điểm: Đôi khi có thể xảy ra việc nhận diện không chính xác khi ngón tay chạm vào nhau hoặc khi có đối tượng khác xuất hiện trong khung hình.

**Tính Chính Xác của Việc Điều Khiển Chuột**

- Ưu Điểm: Chuột di chuyển mượt mà và đáp ứng nhanh chóng theo cử chỉ của người dùng.

- Khuyết Điểm: Có thể có độ trễ nhỏ khiến cho việc theo kịp ngón tay có thể không luôn được thực hiện một cách chính xác.

**Tính Ổn Định và Tương Thích**

- Ưu Điểm: Chương trình ổn định, không gặp sự cố lớn trong quá trình thực hiện.

- Khuyết Điểm: Có thể gặp khó khăn khi sử dụng trong điều kiện ánh sáng yếu hoặc nền phức tạp.

### **3.4.2. Đánh Giá Hiệu Năng**

**Tốc Độ Xử Lý**

- Ưu Điểm: Xử lý hình ảnh nhanh chóng, đảm bảo tốc độ chấp nhận được cho việc theo dõi và nhận diện cử chỉ.

- Khuyết Điểm: Có thể cần tối ưu hóa mã nguồn để cải thiện tốc độ xử lý trong các trường hợp đặc biệt.

**Tính Thực Tế và Tiện Ích Sử Dụng**

- Ưu Điểm: Cung cấp một phương tiện tương tác sáng tạo, thay thế cho chuột và bàn phím truyền thống.

- Khuyết Điểm: Có thể gặp khó khăn khi sử dụng trong các điều kiện môi trường khác nhau, đặc biệt là với nền và ánh sáng biến động.

# **KẾT LUẬN VÀ KIẾN NGHỊ**

Qua quá trình nghiên cứu, đề tài đã đạt được những kết quả sau:

● Đã xây dựng được ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay dựa trên công nghệ nhận dạng hình ảnh và học máy.

● Ứng dụng có thể điều khiển các thao tác trên máy tính, bao gồm: di chuyển con trỏ, nhấn chuột, nhập liệu,...

Kết quả nghiên cứu của đề tài có những đóng góp mới sau:

● Đề tài đã xây dựng được một ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay có độ chính xác cao, khả năng đáp ứng tốt các yêu cầu của người dùng.

● Khả năng ứng dụng của kết quả nghiên cứu của đề tài là rất lớn, bao gồm:

● Ứng dụng có thể được sử dụng trong các thiết bị điện tử gia dụng, như TV, máy tính bảng,...

● Ứng dụng có thể được sử dụng trong các thiết bị công nghiệp, như robot, máy CNC,...

● Ứng dụng có thể được sử dụng trong các thiết bị y tế, như máy móc hỗ trợ người khuyết tật,...

Dựa trên kết quả nghiên cứu của đề tài, có thể đưa ra một số kiến nghị sau:

● Tiếp tục nghiên cứu và phát triển ứng dụng để nâng cao độ chính xác, khả năng đáp ứng và tính linh hoạt của ứng dụng.

● Nghiên cứu áp dụng ứng dụng vào thực tiễn đời sống và sản xuất.

Hi vọng rằng, với những đóng góp mới của đề tài, ứng dụng điều khiển máy tính thông qua cử chỉ bàn tay có thể được ứng dụng rộng rãi trong thực tiễn đời sống và sản xuất, mang lại nhiều lợi ích cho con người.

# **TÀI LIỆU THAM KHẢO**

**[1] Antonio Aceves-Fernandez, Marco - 2018 - Artificial Intelligence: Emerging trends and Applications - IntechOpen**

**[2] Nguyễn Phương Nga (Chủ biên), Trần Hùng Cường - 2021- Giáo trình : Trí tuệ nhân tạo - Trường Đại Học Công Nghiệp Hà Nội - NXB Thống Kê**

**[3] Practeek Joshi - 2017 - Artificial intelligence with python - Packt Publishing**