**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO**

**ĐỒ ÁN MÔN HỌC**

**THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI: TÌM HIỂU VÀ MÔ PHỎNG BÀI TOÁN NHẬN DIỆN THAO TÁC CHO TRÒ CHƠI DINOSAUR RUNNER CAT RUNNER VÀ CAR DRIVE**

Thành Viên : : **PHẠM DƯƠNG MINH NHẬT -19IT182**

Lớp : **19IT3**

Giảng viên hướng dẫn: **TS. VƯƠNG CÔNG ĐẠT**

***Da nang, June, 2023***

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO**

**ĐỒ ÁN MÔN HỌC**

**THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI: TÌM HIỂU VÀ MÔ PHỎNG BÀI TOÁN NHẬN DIỆN THAO TÁC CHO TRÒ CHƠI DINOSAUR RUNNER CAT RUNNER VÀ CAR DRIVE**

Giảng viên hướng dẫn: **TS. VƯƠNG CÔNG ĐẠT**

***Da nang, June, 2023***

# LỜI MỞ ĐẦU

Với kiến thức đã được học hỏi và trao dồi trong kì vừa qua với môn học Thị Giác Máy Tính do thầy Vương Công Đạt giảng dạy và khả năng tìm tòi nghiên cứu của chúng em.

Để học vận dụng kiến thức đã học vừa qua về Thị Giác Máy Tính. Cho nên chúng em muốn tìm hiểu và mô phỏng lại bài toán nhận diện thao tác bàn tay cho các trò chơi đơn giản như Dinosaur Runner và Cat Runner....

# LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến thầy TS. Vương Công Đạt đã dành dụm thời gian của mình hướng dẫn bọn em trong thời gian hoàn thành đồ án môn học. Thầy đã tận tình truyền đạt cho chúng em những kiến thức thầy có và giúp đỡ bọn em trong suốt quá trình hoàn thành đồ án môn học này.

Và xin chân thành cảm ơn thầy cô trong Trường Đại học Công nghệ thông tin và truyền thông Việt Hàn – ĐHĐN cũng đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức quý báu và tạo điều kiện giúp đỡ chúng em trong suốt quá trình học tập.

Xin chân thành cảm ơn !

# NHẬN XÉT CỦA GIẢNG VIÊN

**……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**Đà Nẵng, ngày ... tháng ... năm 2023**

**Giảng viên hướng dẫn**

# MỤC LỤC

Trang

[LỜI MỞ ĐẦU 3](#_Toc137485204)

[LỜI CẢM ƠN 4](#_Toc137485205)

[NHẬN XÉT CỦA GIẢNG VIÊN 5](#_Toc137485206)

[MỤC LỤC 1](#_Toc137485207)

[MỤC LỤC HÌNH ẢNH 2](#_Toc137485208)

[CHƯƠNG I. GIỚI THIỆU 1](#_Toc137485209)

[1.1. Hand Landmarks Detection trong Mediapipe 1](#_Toc137485210)

[1.2. Mô hình Convolutional Neural Network (CNN) 2](#_Toc137485211)

[1.3. Mô hình Convolutional Neural Network (CNN) trong Hand Landmark Detection 2](#_Toc137485212)

[1.4 Các thuật toán khác 4](#_Toc137485213)

[1.4.1 Kalman Filter. 4](#_Toc137485214)

[1.4.2 Geodesic Regression. 4](#_Toc137485215)

[1.4.3 RANSAC (RANdom SAmple Consensus). 4](#_Toc137485216)

[1.4.4 One Euro Filter. 4](#_Toc137485217)

[CHƯƠNG II. Mô hình hóa thuật toán 6](#_Toc137485218)

[2.1 Lưu đồ thuật toán Convolutional Neural Network (CNN) 6](#_Toc137485219)

[2.1.1 Lưu đồ thuật toán 6](#_Toc137485220)

[2.1.2 Giải thích lưu đồ thuật toán 7](#_Toc137485221)

[2.2 Công thức toán học được áp dụng 7](#_Toc137485222)

[2.2.1 Convolutional Layers 7](#_Toc137485223)

[2.2.2 Pooling Layers 8](#_Toc137485224)

[2.2.3 Hand Gesture Recognition (Nhận dạng cử chỉ tay) 9](#_Toc137485225)

[CHƯƠNG III. TRIỂN KHAI XÂY DỰNG 11](#_Toc137485226)

[3.1 Triển khai bài toán 11](#_Toc137485227)

[3.2 Xây dựng bài toán 14](#_Toc137485228)

[3.3 Dữ liệu training 17](#_Toc137485229)

[CHƯƠNG IV. KẾT QUẢ 19](#_Toc137485230)

[4.1 Kết quả thực hiện 19](#_Toc137485231)

[CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 21](#_Toc137485232)

[5.1 Kết luận 21](#_Toc137485233)

[5.2 Hướng phát triển 21](#_Toc137485234)

[TÀI LIỆU THAM KHẢO 22](#_Toc137485235)

# MỤC LỤC HÌNH ẢNH

[Hình 1: Hand Landmarks Detection trong Mediapipe 1](#_Toc137485253)

[Hình 2 Mô hình CNN cho Hand Landmark Detection 3](#_Toc137485254)

[Hình 3: Lưu đồ thuật toán CNN 6](#_Toc137485255)

[Hình 4: Cấu trúc của đơn vị dư với nút cổ chai được sử dụng trong phương pháp đề xuất. 8](#_Toc137485256)

[Hình 5: Cấu trúc module ASPP với 5 thao tác song song. 9](#_Toc137485257)

[Hình 6: Kiến trúc của CNN được sử dụng trong giai đoạn nhận dạng Hình 1(b). 10](#_Toc137485258)

[Hình 7: 21 tọa độ được phát hiện trong bàn tay 11](#_Toc137485259)

[Hình 8: Gốc tọa độ của hình ảnh 11](#_Toc137485260)

[Hình 9: Thao tác bàn tay của hành động nhảy 12](#_Toc137485261)

[Hình 10: Thao tác bàn tay của hành động sang phải 12](#_Toc137485262)

[Hình 11: Thao tác bàn tay của hành động sang trái 12](#_Toc137485263)

[Hình 12: Thao tác bàn tay của hành động cúi xuống 13](#_Toc137485264)

[Hình 13: Thao tác bàn tay gốc 13](#_Toc137485265)

[Hình 14: Ví dụ về Data Training 18](#_Toc137485266)

[Hình 15: Sử dụng thao tác để chơi game Cat Runner 19](#_Toc137485267)

[Hình 16: Sử dụng thao tác để chơi game Dinosaur T Rex 19](#_Toc137485268)

[Hình 17: Sử dụng thao tác để chơi game đua xe 20](#_Toc137485269)

# CHƯƠNG I. GIỚI THIỆU

## Hand Landmarks Detection trong Mediapipe

Hand Landmarks Detection là một chức năng trong Mediapipe, một framework mã nguồn mở của Google được sử dụng để xây dựng ứng dụng thị giác máy tính thời gian thực. Hand Landmarks Detection trong Mediapipe cho phép nhận dạng và xác định vị trí các điểm landmarks trên tay trong hình ảnh hoặc video.

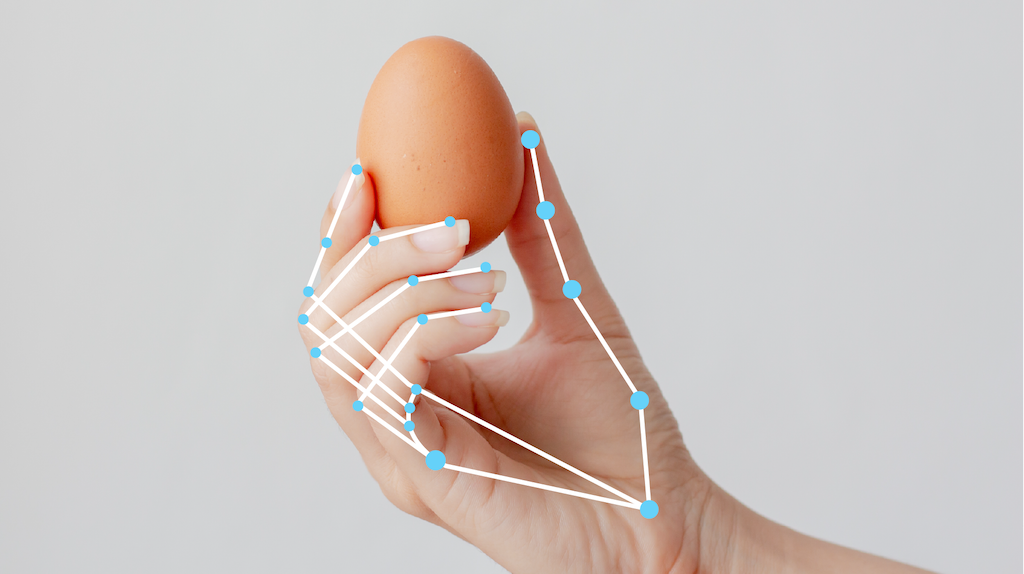
Với Hand Landmarks Detection, Mediapipe có khả năng phát hiện các điểm landmarks quan trọng trên tay như ngón tay, khớp cổ tay, nút bàn tay, và các điểm nằm giữa các khớp. Điều này cho phép phát hiện và theo dõi vị trí và chuyển động của tay trong thời gian thực.

Hand Landmarks Detection trong Mediapipe sử dụng một mô hình máy học (thường là mô hình CNN) được huấn luyện trên một lượng lớn dữ liệu tay có các điểm đánh dấu (annotated landmarks). Mô hình này được tối ưu để dự đoán vị trí chính xác của các landmarks trên tay dựa trên các đặc trưng của hình ảnh hoặc video đầu vào.

Khi được triển khai trong Mediapipe, Hand Landmarks Detection có thể được sử dụng trong nhiều ứng dụng thực tế như:

* Điều khiển tương tác người-máy (human-machine interaction): Phát hiện và theo dõi tay cho phép người dùng tương tác với máy tính hoặc thiết bị di động thông qua cử chỉ và chuyển động tay.
* Thị giác máy tính: Hand Landmarks Detection có thể được sử dụng để nhận dạng và phân loại các cử chỉ tay, như cử chỉ bấm, vuốt, vẽ hoặc chỉ vào các vật thể trong hình ảnh hoặc video.
* Ứng dụng thời trang: Các ứng dụng liên quan đến thời trang có thể sử dụng Hand Landmarks Detection để phát hiện vị trí của tay và áp dụng các mô hình 3D hoặc các phép biến đổi để thực hiện mô phỏng hoặc thử đồ ảo.

Hand Landmarks Detection trong Mediapipe cung cấp một giải pháp mạnh mẽ và linh hoạt để nhận dạng và theo dõi vị trí của tay trong thời gian thực, mở ra nhiều tiềm năng ứng dụng sáng tạo trong lĩnh vực thị giác máy tính.



Hình : Hand Landmarks Detection trong Mediapipe

## Mô hình Convolutional Neural Network (CNN)

Mô hình Convolutional Neural Network (CNN) là một loại mô hình học sâu (deep learning) được sử dụng rộng rãi trong lĩnh vực thị giác máy tính và xử lý ảnh. CNN được thiết kế đặc biệt để hiệu quả trong việc xử lý dữ liệu có cấu trúc lưới như hình ảnh.

CNN được lấy cảm hứng từ cách mà hệ thần kinh thị giác của con người hoạt động. Nó sử dụng các lớp tích chập (convolutional layers) để học các đặc trưng cục bộ trong dữ liệu đầu vào, như các đường viền, góc, hay mẫu màu sắc. Các đặc trưng này được học tự động từ dữ liệu trong quá trình huấn luyện.

Một mô hình CNN thường bao gồm các thành phần chính sau:

* Lớp tích chập (Convolutional layer): Lớp tích chập là thành phần quan trọng của CNN. Nó áp dụng một tập hợp các bộ lọc (filters) trượt qua toàn bộ hoặc một phần của đầu vào. Bộ lọc này thực hiện phép tích chập để tạo ra các feature map, nơi mỗi phần tử đại diện cho mức độ kích hoạt của bộ lọc tại một vị trí cụ thể trong đầu vào.
* Hàm kích hoạt (Activation function): Sau mỗi lớp tích chập, một hàm kích hoạt được áp dụng lên đầu ra để tạo ra đầu ra kích hoạt của lớp. Hàm kích hoạt thông thường sử dụng là hàm ReLU (Rectified Linear Unit) để tạo ra đầu ra không âm và giúp tăng tính phi tuyến của mô hình.
* Lớp pooling (Pooling layer): Lớp pooling được sử dụng để giảm kích thước của feature map bằng cách lấy giá trị tối đa (max pooling) hoặc giá trị trung bình (average pooling) trong một vùng nhất định. Việc pooling giúp giảm số lượng tham số và chi phí tính toán.
* Lớp kết nối đầy đủ (Fully connected layer): Sau khi thông qua các lớp tích chập và lớp pooling, các feature map được "làm phẳng" thành một vectơ 1 chiều. Vectơ này được đưa vào lớp kết nối đầy đủ, trong đó mỗi nút tương ứng với một trọng số được học để tạo ra đầu ra cuối cùng.
* Hàm mất mát (Loss function): Cuối cùng, một hàm mất mát được sử dụng để đo lường sự sai khác giữa đầu ra dự đoán và đầu ra thực tế. Hàm mất mát được tối ưu trong quá trình huấn luyện để đạt được dự đoán chính xác.

Mô hình CNN được huấn luyện thông qua quá trình lan truyền ngược (backpropagation) và giảm gradient (gradient descent) để điều chỉnh các trọng số và cập nhật mô hình. Với sự phát triển của CNN, nhiều kiến trúc nâng cao đã được đề xuất như VGGNet, ResNet, và InceptionNet, giúp cải thiện hiệu suất của mô hình trong việc nhận dạng và phân loại hình ảnh.

Với khả năng học các đặc trưng cục bộ và khả năng nhận biết thông qua hình ảnh, mô hình CNN đã chứng minh sự mạnh mẽ trong nhiều nhiệm vụ như phân loại hình ảnh, nhận dạng đối tượng, và xử lý ảnh y tế.

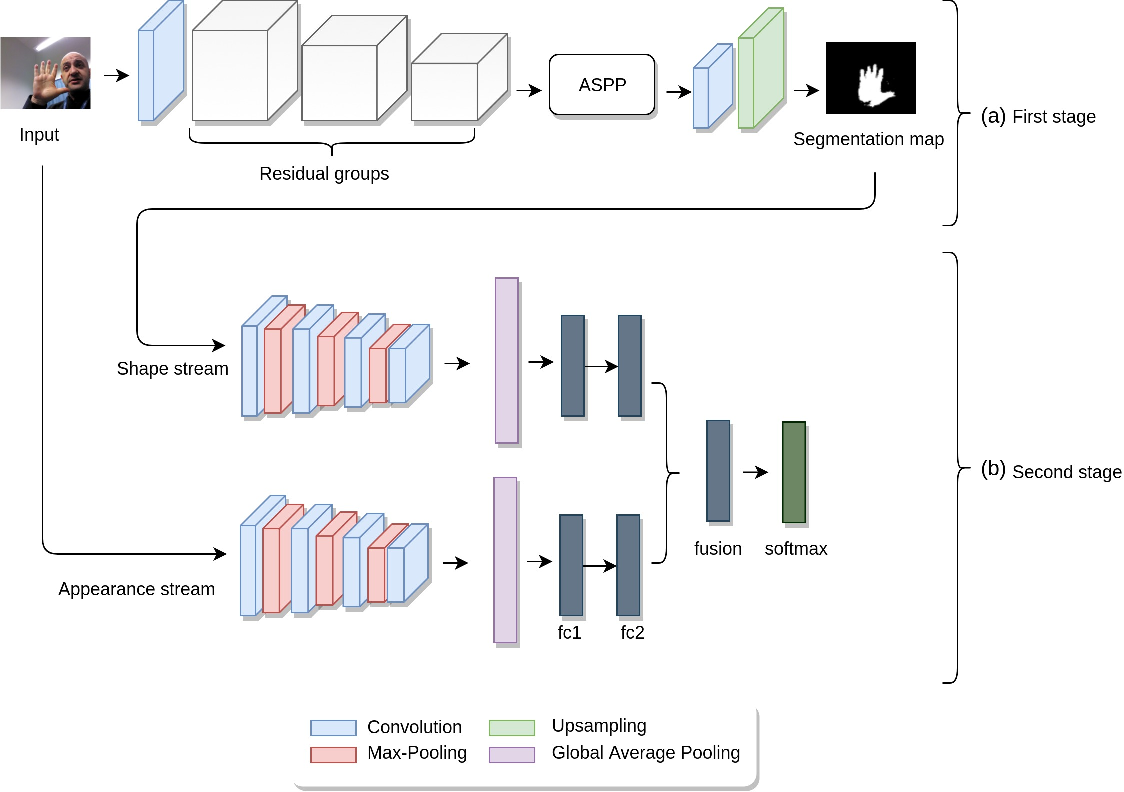
## Mô hình Convolutional Neural Network (CNN) trong Hand Landmark Detection

Mô hình CNN (Convolutional Neural Network) được sử dụng trong Hand Landmarks Detection là một kiến trúc mạng neural được thiết kế đặc biệt để xử lý dữ liệu hình ảnh. Nó là một trong những mô hình phổ biến và mạnh mẽ trong lĩnh vực thị giác máy tính.

Một mô hình CNN thường bao gồm nhiều lớp khác nhau, bao gồm các lớp tích chập, lớp pooling và lớp kết nối đầy đủ. Dưới đây là một giải thích chi tiết về các thành phần chính của một mô hình CNN trong ngữ cảnh của Hand Landmarks Detection:

* Lớp tích chập (Convolutional layer): Lớp này sử dụng các bộ lọc (filters) nhỏ để áp dụng phép tích chập trên hình ảnh đầu vào. Mỗi bộ lọc nhằm tìm kiếm các đặc trưng nhỏ trong hình ảnh, chẳng hạn như cạnh, góc, hoặc vết mờ. Các bộ lọc này được di chuyển qua toàn bộ hình ảnh để tạo ra các đặc trưng được kích hoạt (feature maps) tương ứng.
* Lớp pooling: Lớp này được sử dụng để giảm kích thước của đặc trưng và giảm thiểu việc tính toán. Lớp pooling thường sử dụng phép gộp (max pooling hoặc average pooling) để lấy giá trị lớn nhất hoặc trung bình trong một vùng cụ thể của đặc trưng. Điều này giúp giữ lại thông tin quan trọng và làm giảm độ phân giải.
* Lớp kết nối đầy đủ (Fully Connected layer): Sau khi đặc trưng đã được rút trích thông qua các lớp tích chập và pooling, chúng được đưa vào một hoặc nhiều lớp kết nối đầy đủ. Lớp này có nhiệm vụ học các mối quan hệ phức tạp giữa các đặc trưng và thực hiện phân loại hoặc dự đoán các kết quả mong muốn.

Trong quá trình huấn luyện, mô hình CNN được điều chỉnh bằng cách tối ưu hóa hàm mất mát (loss function) để đạt được đầu ra dự đoán gần với các điểm định vị thực tế của các landmarks trên tay. Quá trình này thường sử dụng phương pháp lan truyền ngược (backpropagation) để cập nhật.



Hình Mô hình CNN cho Hand Landmark Detection

(a): mạng phân đoạn tay trước tiên được huấn luyện để phân đoạn các vùng tay khỏi nền,

(b): luồng hình dạng và luồng ngoại hình sau đó được huấn luyện riêng biệt để thu thập thông tin tương ứng, sau đó được kết hợp trong lớp kết hợp trước bộ phân loại softmax.

## 1.4 Các thuật toán khác

### 1.4.1 Kalman Filter.

Kalman Filter là một thuật toán ước lượng trạng thái được sử dụng để làm mịn và dự đoán các vị trí của các landmarks trên tay.

Nó hoạt động bằng cách kết hợp mô hình dự đoán và thông tin đo lường thực tế để ước lượng trạng thái tốt hơn.

Mô hình dự đoán dự đoán trạng thái tiếp theo dựa trên trạng thái hiện tại và dự đoán chuyển động của đối tượng.

Thông tin đo lường thực tế bao gồm các đo lường từ cảm biến như máy ảnh hoặc cảm biến khác để cung cấp dữ liệu về vị trí thực tế của đối tượng.

Kalman Filter sử dụng các bước dự đoán và cập nhật để cải thiện ước lượng vị trí của landmarks trên tay.

### 1.4.2 Geodesic Regression.

Geodesic Regression là một thuật toán được sử dụng để xác định các điểm landmark dọc theo các ngón tay dựa trên thông tin toàn cục của tay.

Nó xác định quỹ đạo dựa trên sự phụ thuộc không gian của các điểm landmark trên các khung hình liền kề.

Quá trình geodesic regression bao gồm tìm ra đường cong tương đối nhỏ nhất (geodesic curve) trên không gian đa chiều của các điểm landmark.

Thuật toán sử dụng các phương pháp tối ưu hóa để xác định quỹ đạo tốt nhất dựa trên sự phụ thuộc không gian của các điểm landmark.

### 1.4.3 RANSAC (RANdom SAmple Consensus).

RANSAC là một thuật toán sử dụng để tìm ra các điểm landmark trên tay từ một tập hợp lớn các điểm ảnh ban đầu.

Thuật toán RANSAC được sử dụng để loại bỏ nhiễu và các điểm không đáng tin cậy trong quá trình phát hiện landmark.

Nó hoạt động bằng cách chọn một số lượng nhỏ các điểm ngẫu nhiên từ tập dữ liệu ban đầu và tạo ra một mô hình dự đoán từ các điểm này.

Sau đó, thuật toán đánh giá sự phù hợp của các điểm còn lại với mô hình dự đoán và loại bỏ các điểm không phù hợp.

Quá trình này được lặp lại nhiều lần để tìm ra mô hình tốt nhất và các điểm landmark cuối cùng.

### 1.4.4 One Euro Filter.

One Euro Filter là một thuật toán được sử dụng để loại bỏ nhiễu và làm mịn các thông tin về vận tốc và gia tốc của các landmarks trên tay.

Nó giúp giảm độ rung và làm cho dữ liệu được xử lý mượt hơn.

One Euro Filter sử dụng một bộ lọc tuyến tính và non-linear để ước lượng vận tốc và gia tốc của các landmarks.

Nó điều chỉnh mức độ lọc dựa trên độ nhanh chóng thay đổi của dữ liệu đầu vào, cho phép thông tin chính xác đi qua trong các tình huống nhanh chóng thay đổi và giảm nhiễu trong các tình huống ổn định.

# CHƯƠNG II. Mô hình hóa thuật toán

## Lưu đồ thuật toán Convolutional Neural Network (CNN)

### Lưu đồ thuật toán

A picture containing sketch, diagram, drawing, design

Description automatically generated

Hình : Lưu đồ thuật toán CNN

### Giải thích lưu đồ thuật toán

* Start: Bắt đầu thuật toán
* Input: Một hình ảnh chứa một tay hoặc một khung video chứa các khung hình của tay.
* Convolutional Layers:
* Áp dụng một số lớp tích chập để tìm kiếm các đặc trưng nhỏ trong hình ảnh.
* Mỗi lớp tích chập có một số lượng bộ lọc (filters) được huấn luyện để phát hiện các đặc trưng cụ thể.
* Các lớp tích chập tạo ra các đặc trưng được kích hoạt (feature maps) tương ứng.
* Pooling Layers:
* Áp dụng lớp pooling (thường là max pooling) để giảm kích thước của đặc trưng và giảm thiểu việc tính toán.
* Lớp pooling lấy giá trị lớn nhất trong mỗi vùng pooling để bảo toàn thông tin quan trọng.
* Flatten Layer:
* Dữ liệu đặc trưng sau khi được rút trích qua lớp pooling được "làm phẳng" thành một vectơ 1 chiều.
* Fully Connected Layers:
* Đưa vectơ đặc trưng vào các lớp kết nối đầy đủ.
* Các lớp kết nối đầy đủ học các mối quan hệ phức tạp giữa các đặc trưng để phân loại hoặc dự đoán các điểm định vị của các landmarks trên tay.
* Output Landmarks: Các điểm định vị dự đoán của các landmarks trên tay.
* End: Kết thúc thuật toán.

## 2.2 Công thức toán học được áp dụng

### 2.2.1 Convolutional Layers

Tại Lớp Tích Chập (Convolutional Layers) đã sử dụng kỹ thuật cho phép các gradient chảy qua nhiều lớp trong quá trình đào tạo bằng cách sử dụng ánh xạ nhận dạng như các kết nối bỏ qua. Tận dụng lợi thế của việc học còn lại để xây dựng một CNN sâu với 28 lớp tích chập.

A diagram of a identity mapping

Description automatically generated with low confidence

Hình : Cấu trúc của đơn vị dư với nút cổ chai được sử dụng trong phương pháp đề xuất.

BN có nghĩa là bình thường hóa hàng loạt (Batch Normalization).

Mỗi đơn vị có thể được định nghĩa là:

*yk* = *h*(*xk*) + *F* (*xk*; *Wk*)*,*

trong đó *xk* và *yk* đại diện cho đầu vào và đầu ra của đơn vị thứ k, *h*(*xk*) là ánh xạ đồng nhất và *F* là hàm thặng dư được xác định bởi trọng số *Wk*.

### 2.2.2 Pooling Layers

Atrous Spatial Pyramid Pooling - Thông tin theo ngữ cảnh đã được chứng minh là cực kỳ quan trọng đối với các nhiệm vụ phân đoạn ngữ nghĩa. Trong CNN, kích thước của trường tiếp nhận có thể xác định đại khái mức độ chúng ta cần để nắm bắt ngữ cảnh. Mặc dù phạm vi tiếp nhận rộng hơn cho phép chúng tôi thu thập nhiều ngữ cảnh hơn. Nhưng kích thước thực tế của các trường tiếp nhận trong CNN nhỏ hơn nhiều so với kích thước lý thuyết, đặc biệt là ở các lớp cấp cao hơn. Để giải quyết vấn đề này, đề xuất tích chập atrous có thể mở rộng trường tiếp nhận theo cấp số nhân để nắm bắt bối cảnh tầm xa mà không làm mất độ phân giải không gian và tăng số lượng tham số. Kết hợp các thang thông tin theo ngữ cảnh khác nhau bằng cách sử dụng mô-đun ASPP như trong *Hình 5*. Loại mô-đun này đã được sử dụng thành công trong DeepLabv3 tiên tiến nhất để phân đoạn ngữ nghĩa. Mô-đun ASPP được sử dụng trong công việc này có năm cấp độ, một tích chập 1×1 và bốn tích chập 3×3 với tốc độ cực kỳ tương ứng là 1, 3, 6, 12 và 18 (tất cả đều có 32 bộ lọc). Năm bản đồ tính năng kết quả được nối với nhau. Sau đó, đầu ra từ đơn vị nối được đưa vào tích chập 1 × 1 và được lấy mẫu ngược bằng phép nội suy song tuyến tính 4x tiêu chuẩn để cung cấp thuật toán học từ đầu đến cuối cho phân đoạn tay.

**A diagram of a flowchart

Description automatically generated with low confidence**

Hình : Cấu trúc module ASPP với 5 thao tác song song.

### 2.2.3 Hand Gesture Recognition (Nhận dạng cử chỉ tay)

Trong giai đoạn này của phương pháp được đề xuất, dự đoán nhãn cử chỉ tay bằng cách sử dụng kết hợp hai CNN. Như được hiển thị trong *Hình 1*(b), giai đoạn này bao gồm hai CNN riêng biệt có cùng kiến trúc, trong đó mỗi mạng khai thác thông tin dựa trên hình dạng và giao diện tương ứng, được truyền tải bằng bản đồ phân đoạn thủ công từ giai đoạn trước và hình ảnh RGB để nhận dạng cử chỉ tay mạnh mẽ. Mỗi CNN bao gồm một số hoạt động bao gồm tích chập, tổng hợp và ReLU, theo sau là hai lớp được kết nối đầy đủ ở cuối mạng. Cấu trúc chi tiết của các CNN này được trình bày trong *Hình 6*.

Sau đó, chức năng hợp nhất được triển khai để hợp nhất các điểm đầu ra của lớp được kết nối đầy đủ cuối cùng của mỗi mạng theo cách tổng kết từng phần tử. Chính thức hơn cho f S ∈ Rd và f A ∈ Rd lần lượt là đầu ra của các lớp được kết nối đầy đủ cuối cùng (fc2) từ CNN thứ nhất (luồng hình dạng) và thứ hai (luồng hình thức). Đầu ra của hàm hợp f sum là:

A picture containing font, text, white, typography

Description automatically generated

Trong đó 1 ≤ i ≤ d, f sum  ∈ Rd, và d là số nơ-ron (trong thí nghiệm là 64 cho fc2). Sau đó, véc tơ đặc trưng f sum được đưa vào một bộ phân loại softmax để học tập theo chủ đề chung. Cần lưu ý rằng, vì mô hình của chúng tôi được hưởng lợi từ việc phân đoạn nên nó lấy hình ảnh RGB làm đầu vào duy nhất tại thời điểm thử nghiệm.

A picture containing text, screenshot, number, font

Description automatically generated

Hình : Kiến trúc của CNN được sử dụng trong giai đoạn nhận dạng Hình 1(b).

Lưu ý rằng cả hai CNN đều có cùng kiến trúc, ngoại trừ luồng hình dạng CNN sử dụng hình ảnh đầu vào một kênh.

# CHƯƠNG III. TRIỂN KHAI XÂY DỰNG

## 3.1 Triển khai bài toán

Mediapipe có khả năng phát hiện các điểm landmarks quan trọng trên tay như ngón tay, khớp cổ tay, nút bàn tay, và các điểm nằm giữa các khớp, 21 tọa độ trong bàn tay sẽ được phát hiện tương ứng theo *Hình 7*

A picture containing text, font, diagram, screenshot

Description automatically generated

Hình : 21 tọa độ được phát hiện trong bàn tay

Từ 21 tọa độ Landmarks ta sẽ phân tích và xây dựng bài toán.

Gốc tọa độ x,y của ảnh đầu vào sẽ là góc trên cùng bên trái và đạt cực đại tại góc dưới cùng bên phải như *Hình 8*.

A picture containing text, diagram, line, screenshot

Description automatically generated

Hình : Gốc tọa độ của hình ảnh

Từ đó góc trái dưới cùng có tọa độ x,y đạt giới hạn là (1,1). Là nơi phát hiện Landmarks trên bàn tay để xử lí

**Thống nhất thao tác cho game Dinosaur và Cat Runner:**

* Nút tiến (đi lên, nhảy): Bàn tay sẽ đưa duỗi ra hết như *Hình 9*

A picture containing sketch, line art, drawing, linedrawing

Description automatically generated

Hình : Thao tác bàn tay của hành động nhảy

* Nút sang phải: Bàn tay sẽ nắm vào hết, ngón cái đưa ra như *Hình 10*

A picture containing sketch, line art, linedrawing, drawing

Description automatically generated

Hình : Thao tác bàn tay của hành động sang phải

* Nút sang trái: Bàn tay sẽ nắm vào hết, ngón út đưa ra như *Hình 11*

A hand making a thumbs up gesture

Description automatically generated with low confidence

Hình : Thao tác bàn tay của hành động sang trái

* Nút cúi xuống: Bàn tay sẽ nắm vào hết, ngón trỏ đưa ra như *Hình 12*

A picture containing sketch, linedrawing, line art, drawing

Description automatically generated

Hình : Thao tác bàn tay của hành động cúi xuống

* Không làm gì: Bàn tay sẽ nắm vào hết như *Hình 13*

A picture containing sketch, line art, drawing, linedrawing

Description automatically generated

Hình : Thao tác bàn tay gốc

**Thống nhất thao tác cho game đua xe:**

* Tay nắm vô lăng: Bàn tay sẽ nắm vào hết như *Hình 13*

**Phân tích thao tác trong bài toán:**

Theo như 21 điểm tọa độ trên bàn tay được phát hiện như trong *Hình 7* kết hợp với gốc tọa độ ở *Hình 8*.

* Nút tiến (đi lên, nhảy): Bàn tay 5 ngón duỗi ra hết tương ứng với tọa độ y của Landmark[8] < tọa độ y Landmark[7], tọa độ y của Landmark[12] < tọa độ y Landmark[11], tọa độ y của Landmark[16] < tọa độ y Landmark[15], tọa độ y của Landmark[20] < tọa độ y Landmark[19] và tọa độ y của Landmark[4] < tọa độ y Landmark[3]
* Nút sang phải: Bàn tay nắm vào hết, ngón cái duỗi thằng ra tương ứng với tọa độ y của Landmark[8] > tọa độ y Landmark[7], tọa độ y của Landmark[12] > tọa độ y Landmark[11], tọa độ y của Landmark[16] > tọa độ y Landmark[15], tọa độ y của Landmark[20] > tọa độ y Landmark[19] và tọa độ x của Landmark[4] < tọa độ x Landmark[3]
* Nút sang trái: Bàn tay nắm vào hết, ngón út duỗi thằng ra tương ứng với tọa độ y của Landmark[8] > tọa độ y Landmark[7], tọa độ y của Landmark[12] > tọa độ y Landmark[11], tọa độ y của Landmark[16] > tọa độ y Landmark[16], tọa độ y của Landmark[20] < tọa độ y Landmark[19] và tọa độ x của Landmark[4] < tọa độ x Landmark[3]
* Nút cúi xuống: Bàn tay nắm vào hết, ngón út duỗi thằng ra tương ứng với tọa độ y của Landmark[8] < tọa độ y Landmark[7], tọa độ y của Landmark[12] > tọa độ y Landmark[11], tọa độ y của Landmark[16] > tọa độ y Landmark[15], tọa độ y của Landmark[20] > tọa độ y Landmark[19] và tọa độ x của Landmark[4] < tọa độ x Landmark[3]
* Bàn tay gốc (không ấn nút gì): Bàn tay nắm vào hết tương ứng với tọa độ y của Landmark[8] > tọa độ y Landmark[7], tọa độ y của Landmark[12] > tọa độ y Landmark[11], tọa độ y của Landmark[16] > tọa độ y Landmark[15], tọa độ y của Landmark[20] > tọa độ y Landmark[19] và tọa độ x của Landmark[4] < tọa độ x Landmark[3]
* Đua xe rẽ trái: Điều kiện là cả 2 bàn tay xuất hiện trong khung ảnh. Lấy Landmark[9] của tay trái so sánh với Landmark[9] của tay phải, nếu tọa độ y Landmark[9] của tay trái bé hơn tọa độ Landmark[9] của tay phải, sẽ xử lí rẻ trái
* Đua xe rẽ phải: Điều kiện là cả 2 bàn tay xuất hiện trong khung ảnh. Lấy Landmark[9] của tay trái so sánh với Landmark[9] của tay phải, nếu tọa độ y Landmark[9] của tay trái lớn hơn tọa độ y Landmark[9] của tay phải, sẽ xử lí rẻ phải

## 3.2 Xây dựng bài toán

* Import các thư viện cần thiết:

**A screen shot of a computer screen

Description automatically generated with low confidence**

* Khởi tạo biến và thư viện Mediapipe:

A screen shot of a computer program

Description automatically generated with low confidence

* Khởi tạo Camera:

A picture containing text, font, screenshot, graphics

Description automatically generated

Đoạn mã này khởi tạo camera và gán nó cho biến "cap". Tham số "0" chỉ định rằng camera mặc định của hệ thống sẽ được sử dụng.

* Xử lí khung hình từ camera

A screen shot of a computer program

Description automatically generated with medium confidence

Chuyển đổi không gian màu của khung hình từ BGR sang RGB và gán cho ảnh.

Lật ngược hình ảnh theo chiều ngang bằng cách sử dụng cv2.flip().

Đặt cờ cho hình ảnh có thể được ghi.

Sử dụng mediapipe để xử lý khung hình và phát hiện các landmarks trên tay.

Chuyển đổi không gian màu của ảnh từ RGB sang BGR.

* Kiểm tra các Landmarks và xử lí dữ liệu

A picture containing text, screenshot, font

Description automatically generated

* Xủ lí nút tiến (nhảy):

A screen shot of a computer code

Description automatically generated with low confidence

* Xử lí nút sang phải:

A screen shot of a computer code

Description automatically generated with medium confidence

* Xử lí nút sang trái:

A screen shot of a computer code

Description automatically generated with medium confidence

* Xử lí nút cúi xuống:

A screen shot of a computer program

Description automatically generated with low confidence

* Bắt sự kiện không làm gì:

A screen shot of a computer code

Description automatically generated with low confidence

* Vẽ 21 điểm Landmarks trên bàn tay:

A picture containing text, screenshot, font

Description automatically generated

* Hàm vẽ vô lăng

A picture containing screenshot, text, font

Description automatically generated

* Hàm xử lí rẽ trái của game đua xe

A screen shot of a computer code

Description automatically generated with low confidence

* Hàm xử lí rẽ phải của game đua xe

A screen shot of a computer code

Description automatically generated with low confidence

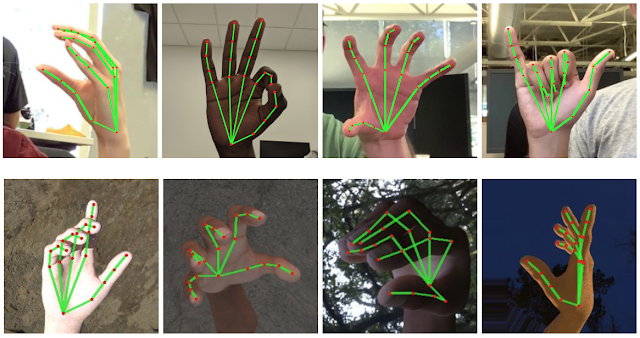
## 3.3 Dữ liệu training

Thông tin về dữ liệu huấn luyện (training data) mà Hand Landmarks Detection của Mediapipe đã sử dụng không được công bố rõ ràng. Tuy nhiên, thông thường dữ liệu huấn luyện cho mô hình Hand Landmarks Detection bao gồm hàng ngàn hình ảnh hoặc video chứa các hình ảnh của tay người trong nhiều góc độ và tình huống khác nhau.

Dữ liệu huấn luyện thường được gắn nhãn (annotated) để chỉ định vị trí chính xác của các điểm landmark trên tay trong mỗi hình ảnh hoặc khung hình video. Nhãn này thông thường được xác định bằng cách sử dụng các công cụ đánh dấu tay (hand labeling tools) để đánh dấu các điểm landmark trên tay trong từng hình ảnh.

Mediapipe có thể đã sử dụng nhiều nguồn dữ liệu khác nhau để huấn luyện mô hình Hand Landmarks Detection của họ, bao gồm cả dữ liệu công khai có sẵn và dữ liệu thu thập từ nguồn tư nhân. Dữ liệu có thể được thu thập từ nhiều nguồn, bao gồm các cơ sở dữ liệu công khai, bộ dữ liệu tư nhân và dữ liệu tổ chức hoặc thu thập bởi Mediapipe hoặc các đối tác của họ.

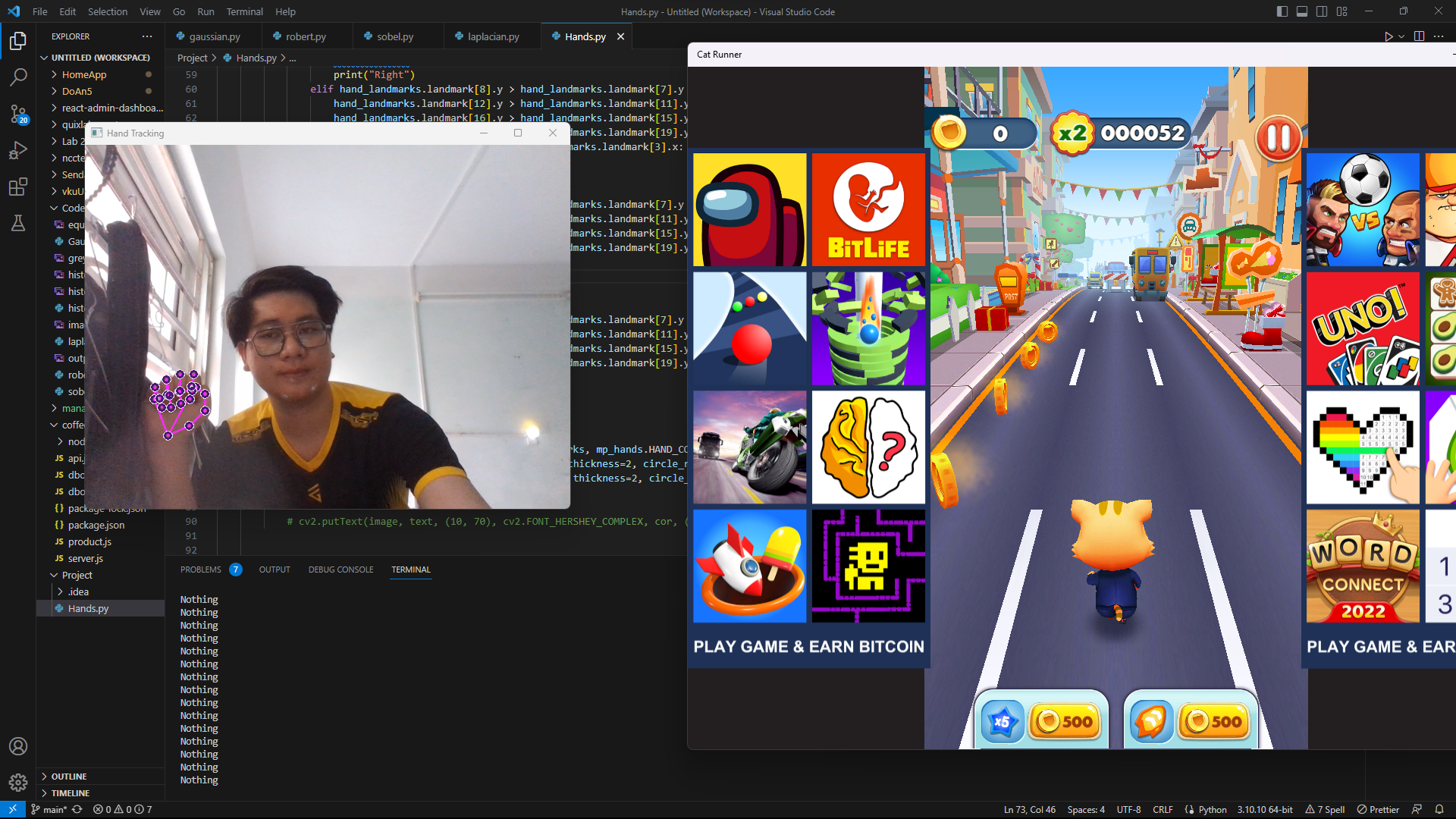
Thông qua việc sử dụng dữ liệu huấn luyện đa dạng và phong phú, mô hình Hand Landmarks Detection trong Mediapipe có khả năng nhận diện và dự đoán vị trí các điểm landmark trên tay một cách chính xác và ổn định trong nhiều tình huống và môi trường khác nhau.



Hình : Ví dụ về Data Training

# CHƯƠNG IV. KẾT QUẢ

## 4.1 Kết quả thực hiện



Hình : Sử dụng thao tác để chơi game Cat Runner

A screenshot of a computer

Description automatically generated

Hình : Sử dụng thao tác để chơi game Dinosaur T Rex

A screenshot of a video game

Description automatically generated

Hình : Sử dụng thao tác để chơi game đua xe

# CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1 Kết luận

* Sau thời gian khoảng vừa học vừa triển khai xây dựng thì cuối cùng cơ bản hệ thống cũng hoàn thành.
* Hệ thống có thể sử dụng để chơi cho một vài game sử dụng hướng di chuyển đơn giản
* Ứng dụng vẫn còn vài hạn chế như: sử dụng presskey liên tục có đôi lúc giật lag, giao diện chưa được bắt mắt

## 5.2 Hướng phát triển

* Hoàn thiện hơn về phần giao diện
* Cố gắng cho ứng dụng trở nên mượt mà, dễ sử dụng
* Cố gắng phân loại nhiều hơn cho nhiều tựa game khác

# TÀI LIỆU THAM KHẢO

[1] [Hand landmarks detection guide  |  MediaPipe  |  Google for Developers](https://developers.google.com/mediapipe/solutions/vision/hand_landmarker)

[2] [HGR-Net: A Two-stage Convolutional Neural Network for Hand Gesture Segmentation and Recognition: Paper and Code - CatalyzeX](https://www.catalyzex.com/paper/arxiv:1806.05653)