**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**---------------------------------------**

****

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH KHOA HỌC MÁY TÍNH**

**ĐỀ TÀI: NGHIÊN CỨU VÀ ỨNG DỤNG THUẬT TOÁN AI HỖ TRỢ NGƯỜI KHIẾM THÍNH**

|  |  |
| --- | --- |
| GVHD: | ThS. Chu Thị Quyên |
| Mã lớp độc lập: | 2020DHKHMT02 |
| Người thực hiện:  Mã sinh viên: | Nguyễn Minh Hiếu  2020605559 – K15 |
|  | |

Hà Nội ─ Năm 2024

# MỤC LỤC

[DANH MỤC NHỮNG TỪ VIẾT TẮT iii](#_Toc167034069)

[DANH MỤC BẢNG BIỂU iv](#_Toc167034070)

[DANH MỤC HÌNH ẢNH v](#_Toc167034071)

[LỜI CẢM ƠN vii](#_Toc167034072)

[LỜI NÓI ĐẦU 1](#_Toc167034073)

[CHƯƠNG 1. TỔNG QUAN BÀI TOÁN NHẬN DẠNG NGÔN NGỮ KÝ HIỆU 4](#_Toc167034074)

[1.1. Giới thiệu về thị giác máy tính 4](#_Toc167034075)

[1.2. Giới thiệu bài toán 7](#_Toc167034076)

[1.2.1. Giới thiệu chung 7](#_Toc167034077)

[1.2.2. Các khó khăn thách thức của bài toán 8](#_Toc167034078)

[1.2.3. Mô tả 8](#_Toc167034079)

[1.2.4. Ứng dụng 9](#_Toc167034080)

[CHƯƠNG 2. CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN 11](#_Toc167034081)

[2.1. Một số kỹ thuật giải quyết bài toán 11](#_Toc167034082)

[2.1.1. Các kỹ thuật trích chọn đặc trưng 11](#_Toc167034083)

[2.1.2. Các kỹ thuật phân lớp 14](#_Toc167034084)

[2.1.3. Mạng nơ ron tích chập (Convolutional Neural Networks) 16](#_Toc167034085)

[2.1.4. Mô hình học kết hợp (Hybrid Learning Model) 24](#_Toc167034086)

[2.1.5. Phương pháp Hand Tracking 25](#_Toc167034087)

[2.2. Mô hình đề xuất cho bài toán 28](#_Toc167034088)

[2.2.1. Mô hình VGG16 28](#_Toc167034089)

[2.2.2. Mô hình mạng nơ ron tích chập (CNN) 29](#_Toc167034090)

[2.2.3. Mô hình học kết hợp CNN-SVM 31](#_Toc167034091)

[CHƯƠNG 3. THỰC NGHIỆM 34](#_Toc167034092)

[3.1. Bộ dữ liệu 34](#_Toc167034093)

[3.2. Huấn luyện mô hình 36](#_Toc167034094)

[3.2.1. Huấn luyện mô hình VGG16 37](#_Toc167034095)

[3.2.2. Huấn luyện mô hình CNN 39](#_Toc167034096)

[3.2.3. Huấn luyện mô hình kết hợp CNN-SVM 40](#_Toc167034097)

[3.3. Các kết quả thực nghiệm 40](#_Toc167034098)

[CHƯƠNG 4. XÂY DỰNG SẢN PHẨM DEMO 43](#_Toc167034099)

[4.1. Giới thiệu về công cụ Tkinter 43](#_Toc167034100)

[4.2. Phân tích thiết kế hệ thống 44](#_Toc167034101)

[4.2.1. Biểu đồ use case 44](#_Toc167034102)

[4.2.2. Mô tả chi tiết các use case 45](#_Toc167034103)

[4.3. Giao diện ứng dụng 48](#_Toc167034104)

[4.4. Chức năng chính của hệ thống 50](#_Toc167034105)

[KẾT LUẬN 52](#_Toc167034106)

[TÀI LIỆU THAM KHẢO 53](#_Toc167034107)

# DANH MỤC NHỮNG TỪ VIẾT TẮT

|  |  |
| --- | --- |
| CNN | Convolutional Neural Networks |
| FC | Fully Connected |
| GPU | Graphics Processing Unit |
| K-NN | K-Nearest Neighbors |
| ReLU | Rectified Linear Unit |
| SVM | Support Vector Machine |
| SSD | Single Shot Detector |

# DANH MỤC BẢNG BIỂU

[Bảng 3. 1 Dữ liệu thực nghiệm 36](#_Toc166369496)

[Bảng 3. 2 Độ chính xác phân lớp của các mô hình trên độ đo Accuracy 41](#_Toc166369497)

[Bảng 3. 3 Kết quả thử nghiệm các mô hình trên một số độ đo khác 41](#_Toc166369498)

# DANH MỤC HÌNH ẢNH

[Hình 1. 1 So sánh giữa thị giác máy tính và con người 5](#_Toc166980661)

[Hình 1. 2 Một số tác vụ của thị giác máy tính 5](#_Toc166980662)

[Hình 1. 3 Ví dụ về thị giác máy tính trong chẩn đoán 6](#_Toc166980663)

[Hình 2. 1 Minh họa thuật toán HOG 12](#_Toc167034035)

[Hình 2. 2 Minh họa thuật toán SIFT 13](#_Toc167034036)

[Hình 2. 3 Minh họa SVM 15](#_Toc167034037)

[Hình 2. 4 Minh họa KNN 16](#_Toc167034038)

[Hình 2. 5 Minh họa mô hình CNN 17](#_Toc167034039)

[Hình 2. 6 Minh họa mô hình Alexnet 19](#_Toc167034040)

[Hình 2. 7 Minh họa mô hình VGG 20](#_Toc167034041)

[Hình 2. 8 Minh họa kiến trúc mạng Resnet 21](#_Toc167034042)

[Hình 2. 9 Kiến trúc Residual 22](#_Toc167034043)

[Hình 2. 10 Mô hình Palm Detector 26](#_Toc167034044)

[Hình 2. 11 Mô hình SSD 26](#_Toc167034045)

[Hình 2. 12 Mô hình Hand Landmark Model 28](#_Toc167034046)

[Hình 2. 13. Minh hoa kiến trúc mạng VGG16 28](#_Toc167034047)

[Hình 2. 14 Mô tả kiến trúc mô hình CNN 30](#_Toc167034048)

[Hình 2. 15 Minh họa mô hình học kết hợp CNN-SVM 33](#_Toc167034049)

[Hình 3. 1 Minh họa dataset 34](#_Toc167034050)

[Hình 3. 2 Một số ký hiệu hay nhầm 35](#_Toc167034051)

[Hình 3. 3 Một số ký hiệu hay nhầm khác 36](#_Toc167034052)

[Hình 3. 4 Minh họa mô hình CNN 39](#_Toc167034053)

[Hình 3. 5 Độ chính xác phân lớp của CNN qua các lần lặp 40](#_Toc167034054)

[Hình 3. 6 Giá trị hàm mất mát của CNN qua các lần lặp 41](#_Toc167034055)

[Hình 3. 7 Giá trị learning rate qua các lần lặp 41](#_Toc167034056)

[Hình 4. 1 Biểu đồ use case tổng quát 45](#_Toc167034057)

[Hình 4. 2 Phân rã use case Nhận dạng ký hiệu 45](#_Toc167034058)

[Hình 4. 3 Giao diện màn hình chính 49](#_Toc167034059)

[Hình 4. 4 Chuyển văn bản/giọng nói sang ngôn ngữ ký hiệu 49](#_Toc167034060)

[Hình 4. 5 Chuyển ngôn ngữ ký hiệu sang văn bản 50](#_Toc167034061)

[Hình 4. 6 Mô hình hoạt động chức năng nhận diện ảnh 51](#_Toc167034062)

# LỜI CẢM ƠN

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép em được bày tỏ lòng biết ơn chân thành đến cô Chu Thị Quyên, cô đã tận tâm hướng dẫn chúng em qua từng buổi nói chuyện, thảo luận về các lĩnh vực trong đề tài. Cảm ơn cô đã dành thời gian, kiến thức và sự đồng lòng để giúp em hoàn thành kỳ đồ án này.

Em xin gửi lời cảm ơn chân thành đến các giảng viên, thầy cô trong trường đại học đã truyền đạt kiến thức và kinh nghiệm quý báu, và tạo điều kiện thuận lợi để chúng em có thể thực hiện đề tài này. Những kiến thức và sự hướng dẫn của quý thầy cô đã giúp chúng em xác định hướng đi chính xác và phát triển kỹ năng nghiên cứu của mình.

Trong quá trình làm đồ án, cũng như là trong quá trình làm bài báo cáo, khó tránh khỏi sai sót. Em rất mong nhận được ý kiến đóng góp từ thầy cô để học thêm được nhiều kinh nghiệm quý giá.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

*Nguyễn Minh Hiếu*

# LỜI NÓI ĐẦU

Tại Việt Nam, việc ứng dụng các công nghệ trí tuệ nhân tạo nói chung và các kỹ thuật học máy nói riêng đang nở rộ. Các kết quả nghiên cứu/ ứng dụng đã cho ra đời một loạt các sản phẩm mang hơi hướng của trí tuệ nhân tạo. Vai trò của trí tuệ nhân tạo ngày càng rõ rệt và quan trọng trong các lĩnh vực.

Sự phát triển nhanh chóng của trí tuệ nhân tạo (AI) đã mở ra những cơ hội vô cùng hứa hẹn trong việc giải quyết các thách thức xã hội, bao gồm việc hỗ trợ và cải thiện cuộc sống của những người khiếm thính. Theo ước tính của Tổ chức Y tế Thế giới (WHO), có khoảng 430 triệu người trên toàn thế giới cần phục hồi chức năng để giải quyết vấn đề mất thính lực và trong số đó có khoảng 34 triệu trẻ em. Ở Việt Nam ta, theo số liệu của Tổng cục Thống kê về Dân số và Nhà ở, hiện có khoảng 2,5 triệu người điếc/nghe kém. Người khiếm thính sử dụng ngôn ngữ ký hiệu là ngôn ngữ cử chỉ tay với dấu hiệu truyền trực quan bằng tay để truyền đạt ý nghĩa từ thay vì sử dụng âm thanh. Ngôn ngữ này được sử dụng trong cộng đồng người khiếm thính, tuy nhiên không được phổ biến trong cộng đồng giao tiếp.

Vấn đề phát triển các phương pháp nhận dạng ngôn ngữ ký hiệu chuyển đổi các ký hiệu thực hiện bằng cử chỉ tay thành văn bản nhằm tạo ra sự giao tiếp thuận tiện giữa người khuyết tật và người bình thường. Việc nghiên cứu cải tiến các phương pháp nhận dạng cử chỉ tay có ý nghĩa quan trọng, giúp người khiếm thính hòa nhập tốt với cộng đồng.

Quyển báo cáo này giới thiệu về đề tài "Nghiên cứu và ứng dụng thuật toán AI hỗ trợ người khiếm thính". Đề tài này nhằm mục đích nghiên cứu và ứng dụng các thuật toán trí tuệ nhân tạo để tạo ra các giải pháp hữu ích và hiệu quả trong việc hỗ trợ người khiếm thính.

Trong thực tế, người khiếm thính gặp rất nhiều khó khăn trong việc giao tiếp. Sử dụng trí tuệ nhân tạo (AI) trong việc nhận diện và chuyển đổi ngôn ngữ ký hiệu có thể tạo ra những công cụ hỗ trợ thông minh, giúp người khiếm thính giao tiếp một cách tự tin và hiệu quả hơn.

Với mục tiêu áp dụng sự phát triển mạnh mẽ của trí tuệ nhân tạo để giải quyết bài toán nhận dạng ngôn ngữ ký hiệu, nghiên cứu sẽ tập trung vào ứng dụng kỹ thuật học sâu Deep Learning và phương pháp theo dõi bàn tay Mediapipe Hands được phát triển bởi Google.

Để thực hiện nghiên cứu này, trước tiên tôi nghiên cứu thực hiện khảo sát và phân tích bài toán nhận dạng ngôn ngữ ký hiệu. Sau đó, tôi tiến hành thu thập và tiền xử lý bộ dữ liệu huấn luyện. Tiếp theo, tôi sử dụng MediaPipe Hands Tracking để hỗ trợ trong việc phát hiện và theo dõi bàn tay trong các hình ảnh. Sau đó, tôi tập trung vào việc nghiên cứu, phát biểu bài toán nhận dạng ngôn ngữ ký hiệu qua hình ảnh bằng kỹ thuật học sâu Deep Learning sử dụng mạng CNNs. Đồng thời, chúng tôi cũng đã nắm vững các công cụ và ngôn ngữ lập trình để triển khai thuật toán và tạo ra một hệ thống đáng tin cậy và dễ sử dụng cho nhận diện và chuyển đổi ngôn ngữ ký hiệu.

*Nội dung quyển báo cáo sẽ bao gồm các chương như sau:*

**Chương 1: Bài toán nhận dạng ngôn ngữ ký hiệu**

Trong chương 1, tôi trình bày về tổng quan về thị giác máy tính và bài toán nhận dạng ngôn ngữ ký hiệu.

**Chương 2: Các kỹ thuật giải quyết bài toán**

Sau khi đã phát biểu và xác định rõ ràng được yêu cầu bài toán, tôi trình bày các kỹ thuật giải quyết bài toán hiện có cùng các ưu và nhược điểm của chúng

**Chương 3: Thực nghiệm**

Tại chương 3, tôi tập trung trình bày về quá trình thực nghiệm cũng như các kết quả đạt được với kỹ thuật giải quyết bài toán được đề xuất ở chương 2

**Chương 4: Xây dựng sản phẩm demo**

Để tận dụng kết quả thực nghiệm đã thu được, tôi tiến hành xây dựng sản phẩm demo là một ứng dụng có chức năng chính là nhận diện và chuyển đổi ngôn ngữ ký hiệu trong thời gian thực. Nội dung chương bao gồm: trình bày các framework được sử dụng và lý do lựa chọn, trình bày các kết quả của hệ thống.

**Phần kết luận:**

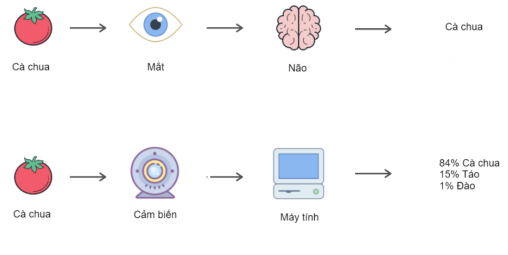
Cuối cùng trong phần kết luật, tôi tổng hợp các kết quả đạt được, các hướng phát triển và mở rộng đề tài nghiên cứu trong tương lai.

# TỔNG QUAN BÀI TOÁN NHẬN DẠNG NGÔN NGỮ KÝ HIỆU

## Giới thiệu về thị giác máy tính

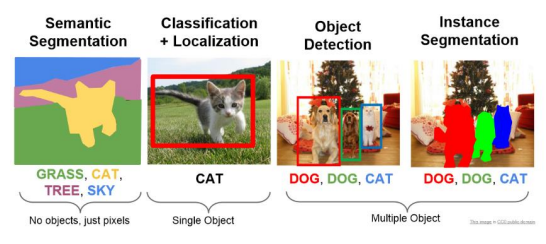
**Thị giác máy tính** là một lĩnh vực trí tuệ nhân tạo cho phép máy tính và hệ thống lấy được thông tin có ý nghĩa từ dữ liệu dạng hình ảnh kỹ thuật số - đồng thời thực hiện hành động hoặc đưa ra đề xuất dựa trên thông tin đó. Cụ thể hơn, nhiệm vụ thị giác máy tính bao gồm các phương pháp thu thập, xử lý, phân tích và hiểu hình ảnh kỹ thuật số cũng như trích xuất dữ liệu nhiều chiều từ thế giới thực để tạo ra thông tin số hoặc ký hiệu. Các thông tin số là đại lượng có thể được đo lường hoặc đếm, ví dụ: Tọa độ của một vật trong ảnh, số lượng các vật được phát hiện, v.v. Thông tin dạng ký hiệu là đại lượng được biểu diễn bởi kí hiệu hoặc danh mục, các thông tin đó có thể là nhãn gán cho đối tượng, mô tả của ảnh v.v. Tóm lại, nếu trí tuệ nhân tạo cho phép máy tính suy nghĩ thì thị giác máy tính cho phép máy tính nhìn, quan sát và hiểu.

**Nguyên lý** của thị giác máy tính tương đối giống như thị giác của con người, chính xác hơn là bắt chước thị giác của con người. Máy tính tiếp nhận thông tin dạng hình ảnh về thế giới thông qua các thiết bị cảm biến và các thông tin sẽ được xử lý tại bộ phận xử lý trung tâm. Cách thức trên khá giống với thị giác của con người, đều có bộ phận tiếp nhận thông tin và bộ phận xử lý. Tuy nhiên, thị giác của con người có lợi thế vượt trội hơn là trải qua quá trình tiến hóa sinh học và quá trình học hỏi không ngừng mỗi ngày. Điều này làm cho thị giác của con người trở nên nhạy bén hơn do có kinh nghiệm tích lũy từ cuộc sống. Ngoài ra, thị giác của con người có khả năng phân tích và hiểu ngữ cảnh, nhận diện vật thể từ nhiều góc độ, ánh sáng khác nhau – điều mà thị giác máy tính chưa thể hiện tốt. Song, thị giác máy tính lại có khả năng xử lý một cách nhanh chóng, vượt trội ở các tác vụ đếm các đối tượng trong ảnh. Như vậy, thị giác máy trở nên khả thi là có những hiểu biết về thị giác của con người. Mặc dù chưa thể hiện tốt như thị giác của con người nhưng có những tác vụ mà thị giác máy tính thể hiện vượt trội hoàn toàn so với con người.



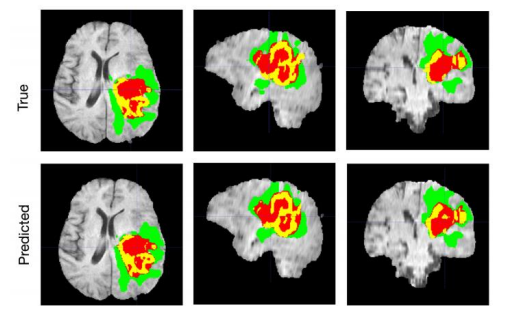
Hình 1. 1 So sánh giữa thị giác máy tính và con người

Trong thị giác máy tính, có rất nhiều các tác vụ được đề ra nhằm giải quyết một yêu cầu xác định. Một số các tác vụ phổ biến có thể kể đến có trong thị giác máy tính là: Phân loại ảnh (Image Classification), chọn vùng đối tượng (Object Localization), phát hiện đối tượng (Object Detection), phân đoạn ảnh (Semantic Segmentation), theo dõi (Tracking), v.v. Các tác vụ này có thể được giải quyết bằng nhiều các kỹ thuật khác nhau, trong đó Học sâu (Deep learning) và Mạng nơ-ron tích chập (CNNs) là hai kỹ thuật thiết yếu.



Hình 1. 2 Một số tác vụ của thị giác máy tính

Tính đến thời điểm hiện tại, thị giác máy tính đã phát triển vượt bậc và được ứng dụng trong nhiều lĩnh vực khác nhau. Trong lĩnh vực sản xuất, thị giác máy tính được sử dụng cho việc giám sát chất lượng, quản lý từ xa và tự động hóa hệ thống. Trong lĩnh vực y tế, các quá trình phân tích hình ảnh X-ray, MRI nhằm chẩn đoán được thực hiện một cách tự động nhờ có thị giác máy tính và đem lại kết quả tốt. Trong việc bảo mật, các hệ thống nhận dạng, phát hiện các đối tượng xâm nhập trở nên khả thi là nhờ có thị giác máy tính. Ngoài ra, vô số các ứng dụng khác của thị giác máy tính có thể được đưa vào trong đời sống hằng ngày, cải thiện, nâng cao chất lượng cuộc sống.



Hình 1. 3 Ví dụ về thị giác máy tính trong chẩn đoán

Mặc dù đã có những sự phát triển vượt bậc nhất định và được ứng dụng nhiều trong đời sống nhưng thị giác máy tính vẫn còn những thử thách, khó khăn nhất định. Ở thời điểm hiện tại giới hạn về chất bán dẫn đang dần được thể hiện rõ nét, hiệu năng của các vi xử lý thế hệ tăng không nhiều so với với thế hệ cũ. Một khó khăn khác là thiếu nguồn dữ liệu huấn luyện cả về chất lượng lẫn số lượng do thị giác máy tính cần rất nhiều dữ liệu để học. Ngoài ra còn nhiều những khó khăn khác về tối ưu hóa thuật toán thu thập, lưu trữ và xử lý dữ liệu phục vụ cho mục đích huấn luyện. Tóm lại, thị giác máy tính vẫn còn nhiều những thử thách và cần có giải pháp mang tính toàn diện.

Nhìn chung, thị giác máy tính là một lĩnh vực phát triển nhanh chóng với nhiều ứng dụng vào cuộc sống và nhiều những cơ hội cải tiến, thúc đẩy công nghệ. Với việc tiếp tục nghiên cứu và phát triển, hoàn toàn có thể mong đợi được thấy những ứng dụng thú vị hơn nữa về thị giác máy tính trong tương lai.

## Giới thiệu bài toán

### Giới thiệu chung

Ngôn ngữ ký hiệu tay đóng vai trò quan trọng trong việc giao tiếp và truyền đạt ý nghĩa cho những người gặp khó khăn khi sử dụng ngôn ngữ nói, như người câm và điếc. Mặc dù ngôn ngữ ký hiệu mang lại khả năng giao tiếp cho cộng đồng này nhưng nó đối mặt với nhiều thách thức, đặc biệt là sự hiểu biết và sử dụng rộng rãi từ phía cộng đồng người bình thường.

Tầm quan trọng của vấn đề này không chỉ nằm ở việc cung cấp một phương tiện giao tiếp cho nhóm người có khuyết tật, mà còn là việc tạo ra một môi trường hòa nhập và đồng thuận trong cộng đồng lớn hơn. Việc nắm bắt và hiểu biết ngôn ngữ ký hiệu tay không chỉ mở rộng khả năng giao tiếp của những người sử dụng mà còn đóng góp vào sự đa dạng và sự hiểu biết trong xã hội.

Mặc dù có nhiều cố gắng để phát triển và giảng dạy ngôn ngữ ký hiệu tay, thách thức lớn nhất vẫn là thiếu sự nhất quán và hiểu biết về ngôn ngữ này trong cộng đồng người bình thường. Điều này tạo ra một khoảng cách giao tiếp giữa những người sử dụng ngôn ngữ ký hiệu tay và những người không quen thuộc với nó.

Bài toán nhận dạng ngôn ngữ ký hiệu là một trong những ứng dụng quan trọng của trí tuệ nhân tạo đối với người khiếm thính. Bài toán này tập trung vào việc nhận dạng và phân loại các ký hiệu tay từ đó chuyển đổi thành ngôn ngữ văn bản hoặc giọng nói để giao tiếp.

Mục tiêu của đề tài "Nhận diện Ngôn ngữ ký hiệu tay" là phát triển một hệ thống hiệu quả và đáng tin cậy, có khả năng nhận diện và dịch ngôn ngữ ký hiệu tay sang ngôn ngữ bình thường. Cụ thể đề tài nhằm hướng đến các mục tiêu sau:

* Xây dựng hệ thống nhận diện và chuyển đổi ngôn ngữ ký hiệu: Ứng dụng mô hình học sâu (deep learning) và Mediapipe Hands Tracking để nhận diện các cử chỉ trong ngôn ngữ ký hiệu tay. Mô hình này cần có khả năng chính xác cao và đồng thời phải linh hoạt để nhận diện đa dạng các cử chỉ tay.
* Xây dựng bộ dữ liệu mẫu ngôn ngữ ký hiệu tay áp dụng triển khai hệ thống nhận dạng theo thời gian thực.

### Các khó khăn thách thức của bài toán

Đối với bài toán nhận dạng ngôn ngữ ký hiệu chúng ta phải đối mặt với một số thách thức sau:

* Đa dạng ngôn ngữ ký hiệu: Ngôn ngữ ký hiệu khá phong phú và đa dạng, với nhiều biểu hiện và biến thể khác nhau. Điều này tạo ra thách thức trong việc xây dựng một hệ thống nhận dạng chính xác và đa dạng cho các cử chỉ tay và ngôn ngữ ký hiệu.
* Độ phức tạp của cử chỉ: Một số cử chỉ có biểu hiện khá giống nhau gây nhầm lẫn trong nhận dạng, phân loại.
* Hiệu suất và tốc độ: Trong một số tình huống, hiệu suất và tốc độ của hệ thống là yếu tố quan trọng. Điều này đặt ra thách thức trong việc phát triển các thuật toán và hệ thống có thể hoạt động nhanh chóng và hiệu quả trong môi trường thời gian thực.
* Độ chính xác và độ tin cậy của hệ thống nhận dạng: Việc xây dựng một hệ thống nhận dạng cử chỉ tay phải đạt được độ chính xác cao và độ tin cậy cao để đảm bảo rằng thông tin được chuyển đổi từ ngôn ngữ ký hiệu sang văn bản là chính xác và đầy đủ.

### Mô tả

Đối với các bài toán phân lớp, dữ liệu đầu vào phục vụ cho mục đích phân lớp là đặc trưng của đối tượng và kết quả của bài toán phân lớp phải thuộc vào một trong các lớp đã được định nghĩa trước. Các đặc trưng được sử dụng để phân biệt giữa các đối tượng với nhau phải là các đặc trưng mang nhiều thông tin quan trọng trong quá trình phân lớp. Các lớp cần được phân lớp phải rõ ràng và không chồng chéo, điều này đảm bảo rằng một mẫu dữ liệu cụ thể chỉ thuộc vào một lớp duy nhất.

Dữ liệu đầu vào của mô hình là ảnh màu có kích thước 200x200 có chứa đặc trưng bàn tay của một người bất kỳ. Nếu như ảnh chưa đúng với kích thước trên thì tiến hành thay đổi kích thước ảnh. Tuy nhiên, điều này có thể làm mất thông tin về đối tượng trong ảnh, ảnh hưởng trực tiếp đến kết quả phân lớp.

Kết quả đầu ra của mô hình là một vector với giá trị thể hiện xác suất một dự đoán có thể xảy ra. Trong bài toán trên, một giá trị sẽ thể hiện xác suất một chữ cái.

### Ứng dụng

Bài toán nhận diện ngôn ngữ ký hiệu mang lại lợi ích to lớn đối với cộng đồng người khiếm thính, khó nghe. Họ có thể sử dụng ứng dụng để có thể giao tiếp một cách linh hoạt, thuận tiện hơn. Một số ứng dụng cụ thể:

* Dịch ngôn ngữ ký hiệu tay: Dịch ngôn ngữ ký hiệu tay trong thời gian và chuyển đổi chúng thành văn bản hoặc giọng nói. Điều này giúp người khiếm thính có thể giao tiếp dễ dàng hơn với những người không biết ngôn ngữ ký hiệu.
* Bộ từ điển ngôn ngữ ký hiệu tay: Cung cấp một bộ từ điển đầy đủ các biểu hiện ngôn ngữ ký hiệu tay, cũng như cách diễn đạt chúng. Người khiếm thính có thể sử dụng ứng dụng này để tìm kiếm và học các biểu hiện mới hoặc để tham khảo khi cần.
* Học ngôn ngữ ký hiệu tay: Cung cấp các bài học, bài kiểm tra, và trò chơi để giúp người khiếm thính học và thực hành ngôn ngữ ký hiệu tay một cách hiệu quả và thú vị.
* Hỗ trợ giáo dục: Sử dụng trong các lớp học để hỗ trợ giao tiếp giữa giáo viên và học sinh khiếm thính. Nó có thể cung cấp các tài liệu giáo trình, bài giảng, và tương tác trực tiếp qua ngôn ngữ ký hiệu tay.
* Ứng dụng giao tiếp xã hội cho người khiếm thính: Tương tự như các ứng dụng mạng xã hội khác, ứng dụng này cho phép người khiếm thính kết nối, chia sẻ và tương tác với nhau thông qua ngôn ngữ ký hiệu tay. Nó cung cấp một nền tảng cho cộng đồng khiếm thính để giao lưu, học hỏi và hỗ trợ lẫn nhau.

# CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN

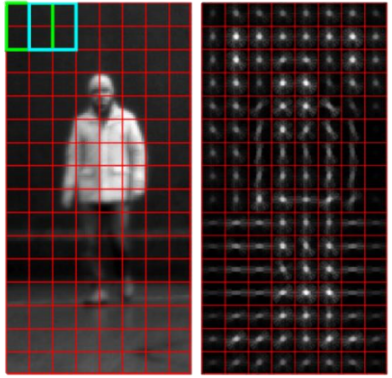
## Một số kỹ thuật giải quyết bài toán

### Các kỹ thuật trích chọn đặc trưng

Trong học máy, có rất nhiều các kỹ thuật trích chọn đặc trưng từ ảnh được sử dụng và mỗi kỹ thuật sẽ trích chọn ra một số đặc trưng nhất định. Do đó, việc sử dụng các kỹ thuật trích chọn đặc trưng phù hợp sẽ mang lại kết kết quả tốt. Dưới đây là một số kỹ thuật trích chọn đặc trưng.

* **Histogram of Oriented Gradients (HOG)**

HOG là một kỹ thuật trích xuất đặc trưng từ ảnh, được sử dụng trong các bài toán nhận dạng đối tượng trong thị giác máy tính và xử lý hình ảnh. HOG có khả năng đếm số lần xuất hiện của các hướng cạnh (edge orientations) trong các vùng nhỏ của ảnh, đặc biệt là để nhận dạng các cạnh và hình dạng trong hình ảnh. Bản chất của phương pháp HOG là sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directins) để mô tả các đối tượng cục bộ trong ảnh.



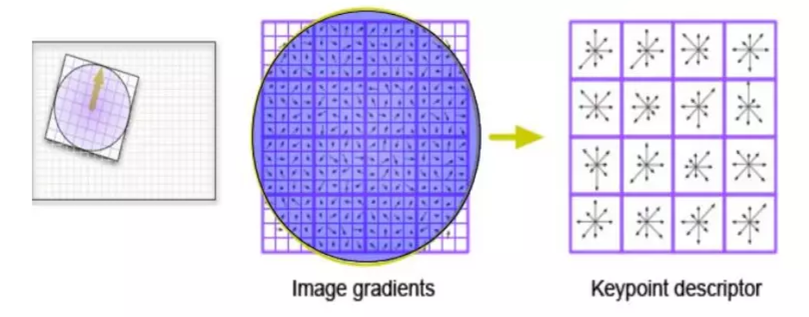
Hình 2. 1 Minh họa thuật toán HOG

* ƯU điểm:
* HOG là một phương pháp mạnh mẽ để mô tả các đặc trưng hình dạng và cạnh trong hình ảnh.
* Không yêu cầu dữ liệu huấn luyện và có khả năng tổng quát hóa tốt trong việc nhận dạng đối tượng.
* Nhược điểm:
* HOG có thể không thực hiện tốt trong môi trường với sự biến đổi ánh sáng mạnh hoặc nhiễu cao.
* Không thể xử lý được các biến đổi phi tuyến tính lớn của hình ảnh.
* **Scale-Invariant Feature Transform (SIFT)**

Scale-invariant feature transform (SIFT) là một feature descriptor được sử dụng trong computer vision và xử lý hình ảnh được dùng để nhận dạng đối tượng, matching image, hay áp dụng cho các bài toán phân loại...

Với đầu vào là một hình ảnh >> SIFT >> các keypoint. Mỗi đối tượng trong hình ảnh sẽ cho ra rất nhiều các keypoint khác nhau, để ta phân biệt được các keypoint này với nhau sẽ thông qua một vector 128 chiều hay còn gọi là descriptor. Các descriptor này sẽ được dùng để nhận dạng đối tượng trong ảnh, hay dùng cho các bài toán classification.

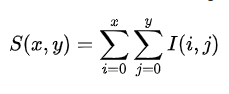
Hình ảnh sau khi áp dụng biến đổi SIFT, ứng với mỗi keypoint ta sẽ thu được: toạ độ keypoint, scale và orientation của keypoint, descriptor. Các mũi tên trong hình dưới vẽ nhờ vào scale và orientation.



Hình 2. 2 Minh họa thuật toán SIFT

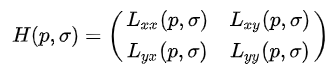
* Ưu điểm:
* Các keypoint sẽ ít bị phụ thuộc bởi cường độ sáng, nhiễu, góc xoay của ảnh do các descriptor được tạo ra từ gradients do đó nó đã bất biến với các thay đổi về độ sáng.
* Nhanh và hiệu quả, tốc độ xử lý gần như với thời gian thực (realtime).
* Có thể xử lý khi xoay ảnh
* Nhược điểm:
* Tính toán phức tạp, cần nhiều bộ nhớ
* Khả năng phát hiện kém trong điều kiện ánh sáng yếu hoặc nhiễu cao
* **Speeded Up Robust Features (SURF)**

Speeded Up Robust Features (SURF) là một kỹ thuật hiệu quả cho trích xuất điểm đặc trưng. SURF sử dụng các bộ lọc hình vuông như một phép gần đúng của làm mịn Gaussian. (Phương pháp SIFT sử dụng các bộ lọc xếp tầng để phát hiện các điểm đặc trưng bất biến theo tỷ lệ, trong đó sự khác biệt của Gaussian (DoG) được tính toán dần dần trên các hình ảnh được thay đổi tỷ lệ.) Lọc hình ảnh bằng hình vuông nhanh hơn nhiều nếu sử dụng hình ảnh tích phân:



Tổng của hình ảnh ban đầu trong một hình chữ nhật có thể được đánh giá nhanh chóng bằng cách sử dụng hình ảnh tích phân, yêu cầu đánh giá ở bốn góc của hình chữ nhật.

SURF sử dụng máy dò đốm màu dựa trên ma trận Hessian để tìm các điểm quan tâm. Định thức của ma trận Hessian được sử dụng làm thước đo sự thay đổi cục bộ xung quanh điểm và các điểm được chọn khi yếu tố quyết định này là cực đại. Trái ngược với máy dò Hessian-Laplacian của Mikolajczyk và Schmid, SURF cũng sử dụng yếu tố quyết định Hessian để chọn thang đo, cũng được thực hiện bởi Lindeberg. Cho điểm p = (x, y) trong ảnh I, ma trận Hessian H (p, σ) tại điểm p và tỷ lệ σ, là:

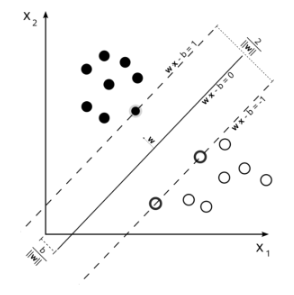


Trong đó Lxx, v.v ... là tích chập của đạo hàm bậc hai của gaussian với ảnh I(x,y) tại điểm p. Bộ lọc hộp có kích thước 9 × 9 là xấp xỉ của một Gaussian với σ = 1,2 và đại diện cho mức thấp nhất (độ phân giải không gian cao nhất) cho các bản đồ phản hồi đốm màu.

### Các kỹ thuật phân lớp

* **SVM (Support Vector Machine)**

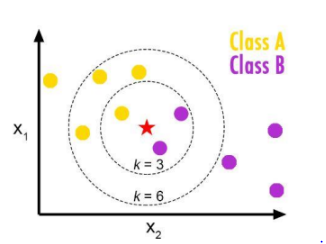
SVM là một kỹ thuật học máy phổ biến được phát triển bởi Vapnik và các cộng sự vào những năm 1990. SVM là một phương pháp phân loại tuyến tính dựa trên việc tìm ra siêu mặt phẳng phân chia hai lớp dữ liệu sao cho khoảng cách từ siêu mặt phẳng đến các điểm dữ liệu gần nhất (các vector hỗ trợ) là lớn nhất. SVM có thể xử lý các bài toán phân loại phi tuyến bằng cách sử dụng các hàm nhân (kernel) để ánh xạ dữ liệu vào không gian nhiều chiều hơn. SVM có nhiều ưu điểm như khả năng hoạt động tốt với dữ liệu nhiễu, ít bị ảnh hưởng bởi số chiều của dữ liệu, có thể tùy biến được các hàm nhân theo nhu cầu. Tuy nhiên, SVM cũng có một số nhược điểm như khó chọn được tham số tối ưu cho hàm nhân, khó mở rộng cho các bài toán phân loại nhiều lớp, yêu cầu nhiều tài nguyên tính toán và bộ nhớ.



Hình 2. 3 Minh họa SVM

* **KNN**

KNN (K-Nearest Neighbors) một thuật toán học máy phân loại dựa trên khoảng cách. Thuật toán này được phát triển bởi Thomas Cover và Peter Hart vào năm 1968. Đặc điểm của KNN là không cần huấn luyện mô hình trước, mà chỉ cần lưu trữ tập dữ liệu huấn luyện và so sánh dữ liệu mới với các điểm dữ liệu gần nhất để đưa ra nhãn. Các ưu điểm của KNN là đơn giản, dễ hiểu, linh hoạt, có thể áp dụng cho nhiều loại dữ liệu và khả năng mở rộng số lớp. Nhược điểm của KNN là tốn nhiều bộ nhớ, thời gian và tài nguyên để tính toán khoảng cách, khó xác định giá trị K phù hợp và nhạy cảm với nhiễu, dữ liệu không cân bằng.



Hình 2. 4 Minh họa KNN

### Mạng nơ ron tích chập (Convolutional Neural Networks)

#### Nguồn gốc

Convolutional Neural Networks (CNN) là một trong những mô hình deep learning phổ biến nhất và có ảnh hưởng nhiều nhất trong Computer Vision. CNN được dùng trong trong nhiều bài toán như nhân dạng ảnh, phân tích video, ảnh MRI, hoặc cho bài các bài của lĩnh vự xử lý ngôn ngữ tự nhiên, và hầu hết đều giải quyết tốt các bài toán này.

CNN cũng có lịch sử khá lâu đời. Kiến trúc gốc của mô hình CNN được giới thiệu bởi một nhà khoa học máy tính người Nhật vào năm 1980. Sau đó, năm 1998, Yan LeCun lần đầu huấn luyện mô hình CNN với thuật toán backpropagation cho bài toán nhận dạng chữ viết tay. Tuy nhiên, mãi đến năm 2012, khi một nhà khoa học máy tính người Ukraine Alex Krizhevsky xây dựng mô hình CNN (AlexNet) và sử dụng GPU để tăng tốc quá trình huấn luyện deep nets đã đạt được top 1 trong cuộc thi Computer Vision thường niên ImageNet với độ lỗi phân lớp top 5 giảm hơn 10% so với những mô hình truyền thống trước đó, đã tạo nên làn sóng mạnh mẽ sử dụng deep CNN với sự hỗ trợ của GPU để giải quyết càng nhiều các vấn đề trong Computer Vision.

#### Cấu trúc

Mạng CNN là một trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định và nhận dạng hình ảnh là một trong số những lĩnh vực mà CNN được sử dụng rộng rãi. CNN được thiết kế để giúp mô hình học máy tự động trích xuất và hiểu các đặc trưng từ dữ liệu hình ảnh một cách hiệu quả.



Hình 2. 5 Minh họa mô hình CNN

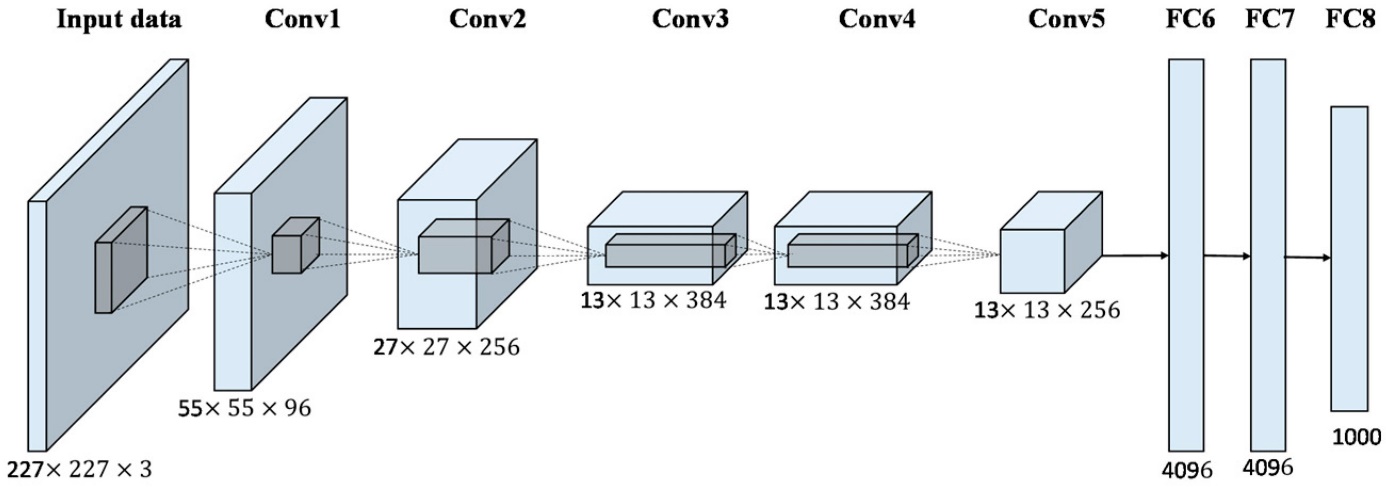
Mạng CNN gồm có ba loại lớp chính: lớp tích chập, lớp gộp và lớp kết nối đầy đủ. Các lớp này được sắp xếp theo một thứ tự nhất định để tạo thành một kiến trúc mạng CNN :

* Lớp tích chập (convolutional layer): Đây là lớp quan trọng nhất của mạng CNN, nơi xảy ra quá trình trích xuất đặc trưng từ ảnh. Lớp tích chập sử dụng một bộ lọc (filter) hay kernel để quét qua toàn bộ ảnh hoặc một vùng nhỏ của ảnh và tính toán tích chập (convolution) giữa bộ lọc và ảnh. Kết quả của quá trình này là một ma trận mới gọi là feature map, biểu diễn cho các đặc trưng được phát hiện bởi bộ lọc.
* Lớp gộp (pooling layer): Đây là lớp dùng để giảm kích thước của các feature map, do đó giảm số lượng tham số và ngăn ngừa quá khớp. Lớp gộp sử dụng một hàm gộp (pooling function) để tổng hợp thông tin từ một vùng nhỏ của feature map thành một giá trị duy nhất. Có nhiều loại hàm gộp khác nhau, một trong những hàm gộp phổ biến nhất là max pooling, lấy giá trị lớn nhất trong một vùng nhỏ của feature map. Các tham số quan trọng của lớp gộp là kích thước của vùng gộp, bước nhảy và đệm.
* Lớp kết nối đầy đủ (fully connected layer): Đây là lớp cuối cùng của mạng CNN, nơi xảy ra quá trình phân lớp hoặc hồi quy dựa trên các đặc trưng đã được trích xuất. Lớp kết nối đầy đủ sử dụng một mạng nơ-ron truyền thống, nơi mỗi nơ-ron ở lớp này được kết nối với tất cả các nơ-ron ở lớp trước đó. Lớp kết nối đầy đủ có thể có nhiều lớp ẩn (hidden layer) và một lớp đầu ra (output layer). Lớp đầu ra có số lượng nơ-ron bằng với số lượng lớp hoặc giá trị cần dự đoán. Các hàm kích hoạt (activation function) được sử dụng để tính toán giá trị đầu ra của các nơ-ron. Các hàm kích hoạt phổ biến là sigmoid, tanh, ReLU, softmax.

#### Một số kiến trúc mạng CNN phổ biến

1. Mạng nơ ron tích chập sâu AlexNet (Deep Convolutional Neural Networks)

Mạng AlexNet được giới thiệu vào năm 2012, được đặt theo tên của Alex Krizhevsky, tác giả thứ nhất của bài báo đột phá trong phân loại ImageNet [Krizhevsky et al., 2012]. Mạng AlexNet bao gồm 8 tầng mạng nơ-ron tích chập, đã chiến thắng cuộc thi ImageNet Large Scale Visual Recognition Challenge năm 2012 với cách biệt không tưởng. AlexNet lần đầu tiên đã chứng minh được rằng các đặc trưng thu được bởi việc học có thể vượt qua các đặc trưng được thiết kế thủ công, phá vỡ định kiến trước đây trong nghiên cứu thị giác máy tính. AlexNet có tám tầng gồm: năm tầng tích chập, hai tầng ẩn kết nối đầy đủ, và một tầng đầu ra kết nối đầy đủ. AlexNet sử dụng ReLU thay vì sigmoid làm hàm kích hoạt.

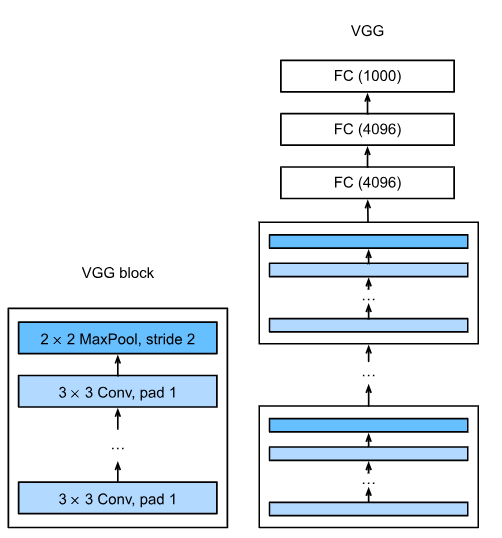


Hình 2. 6 Minh họa mô hình Alexnet

Ngày nay AlexNet đã bị vượt qua bởi các kiến trúc hiệu quả hơn nhiều nhưng nó là một bước quan trọng để phát triển các mạng sâu được sử dụng ngày nay.

1. Mạng sử dụng khối VGG (Networks Using Blocks)

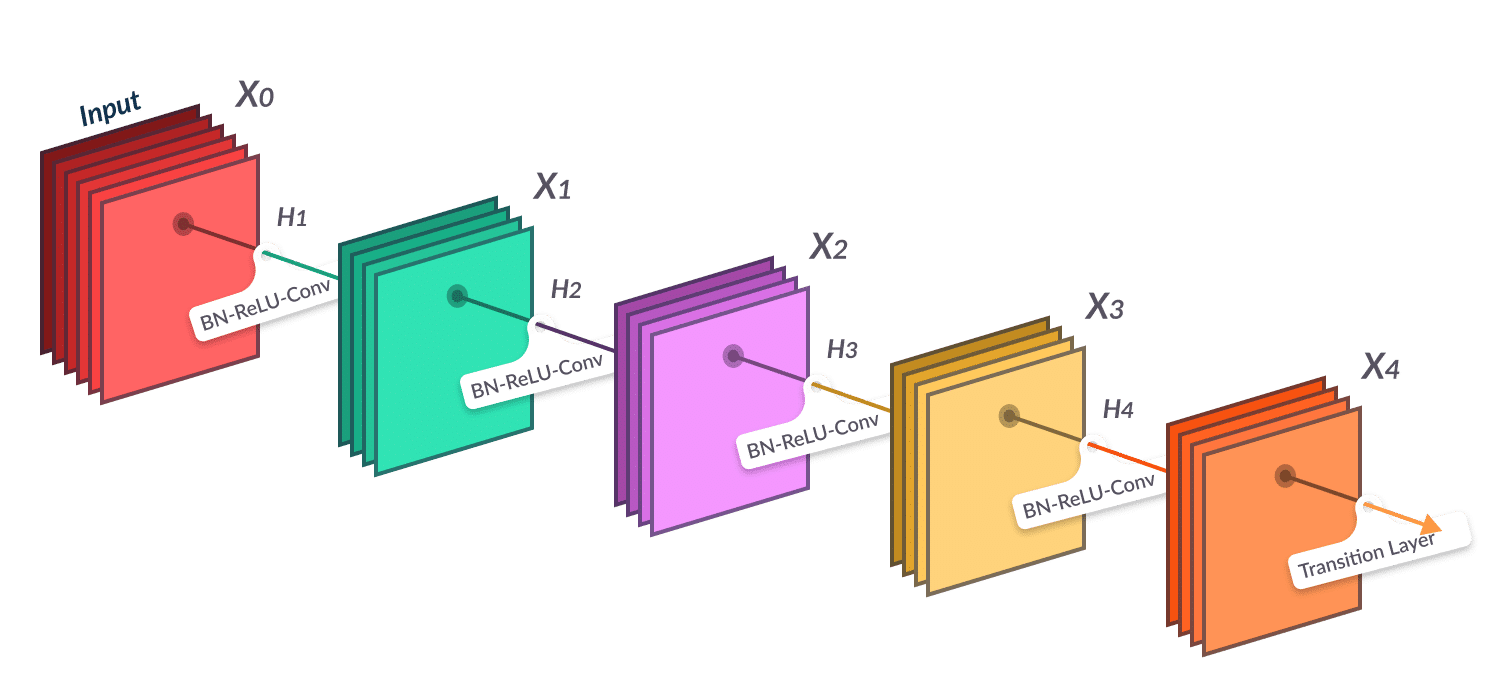
Ý tưởng sử dụng các khối lần đầu xuất hiện trong mạng VGG, được đặt theo tên của nhóm VGG thuộc Đại học Oxford. Sử dụng bất kỳ các framework học sâu hiện đại nào với vòng lặp và chương trình con để xây dựng các cấu trúc lặp lại này là tương đối dễ dàng. Khối cơ bản của mạng tích chập cổ điển là một chuỗi các tầng sau đây: một tầng tích chập (với phần đệm để duy trì độ phân giải), một tầng phi tuyến như ReLU, một tầng gộp như tầng gộp cực đại. Một khối VGG gồm một chuỗi các tầng tích chập, tiếp nối bởi một tầng gộp cực đại để giảm chiều không gian.



Hình 2. 7 Minh họa mô hình VGG

Việc sử dụng các khối giúp ta định nghĩa mạng bằng các đoạn mã nguồn ngắn gọn và thiết kế các mạng phức tạp một cách hiệu quả hơn.

1. Mạng phần dư ResNet (Residual Networks)

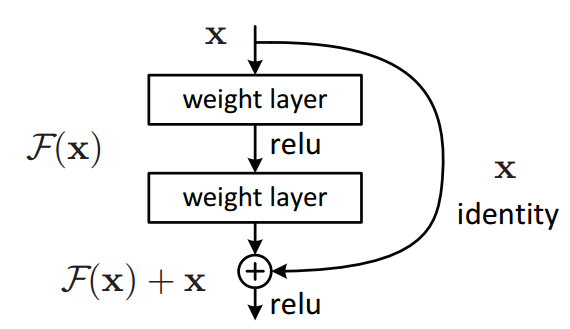


Hình 2. 8 Minh họa kiến trúc mạng Resnet

ResNet (viết tắt của residual network), là mạng học sâu nhận được quan tâm từ những năm 2012 sau cuộc thi LSVRC2012 và trở nên phổ biến trong lĩnh vực thị giác máy. ResNet khiến cho việc huấn luyện hàng trăm thậm chí hàng nghìn lớp của mạng nơ ron trở nên khả thi và hiệu quả.

Nhờ khả năng biểu diễn mạnh mẽ của ResNet, hiệu suất của nhiều ứng dụng thị giác máy, không chỉ các ứng dụng phân loại hình ảnh được tăng cường. Một số ví dụ có thể kể đến là các ứng dụng phát hiện đồ vật.

Ý tưởng chính của ResNet là sử dụng kết nối tắt đồng nhất để xuyên qua một hay nhiều lớp. Một khối như vậy được gọi là một residual block như trong hình dưới đây:



Hình 2. 9 Kiến trúc Residual

Việc xếp chồng các lớp sẽ không làm giảm hiệu suất mạng. Chúng ta có thể đơn giản xếp chồng các ánh xạ đồng nhất lên mạng hiện tại và hiệu quả của kiến trúc không thay đổi. Điều này giúp cho kiến trúc sâu ít nhất là không kém hơn các kiến trúc nông. Hơn nữa, với kiến trúc này, các lớp ở phía trên có được thông tin trực tiếp hơn từ các lớp dưới nên sẽ điều chỉnh trọng số hiệu quả hơn.

Sau ResNet, hàng loạt những biến thể của kiến trúc này được giới thiệu. Thực nghiệm cho thấy những kiến trúc sau này có thể được huấn luyện mạng nơ ron với độ sâu hàng nghìn lớp. ResNet nhanh chóng trở thành kiến trúc phổ biến nhất trong thị giác máy tính.

#### Ưu nhược điểm của mạng CNN trong bài toán nhận diện

Ưu điểm:

* Hiệu suất cao trong ứng dụng xử lý hình ảnh và thị giác máy tính: Một trong những ưu điểm chính của CNN là khả năng xử lý hình ảnh hiệu quả. Điều này là do CNN sử dụng một kỹ thuật gọi là tích chập, bao gồm việc áp dụng bộ lọc vào hình ảnh để trích xuất các đặc điểm có liên quan đến hình ảnh hiện tại. Bằng cách này, CNN có thể giảm lượng thông tin cần xử lý, giúp chúng nhanh hơn và hiệu quả hơn các loại thuật toán khác.
* Tỷ lệ chính xác cao: Một ưu điểm khác của CNN là khả năng đạt được tỷ lệ chính xác cao. Điều này là do CNN có thể học cách nhận biết các mẫu phức tạp trong hình ảnh bằng cách phân tích các tập dữ liệu lớn. Điều này có nghĩa là CNN có thể được huấn luyện để nhận dạng các vật thể hoặc đặc điểm cụ thể với độ chính xác cao, khiến chúng trở nên lý tưởng cho các nhiệm vụ như nhận dạng khuôn mặt hoặc phát hiện vật thể.
* Chống nhiễu: CNN cũng có khả năng chống nhiễu tốt, điều đó có nghĩa là CNN có thể nhận dạng các mẫu trong hình ảnh ngay cả khi chúng bị biến dạng hoặc hỏng. Điều này là do CNN sử dụng nhiều lớp bộ lọc để trích xuất các đặc điểm từ hình ảnh, khiến chúng có khả năng chống nhiễu tốt hơn các loại thuật toán khác.
* Học chuyển tiếp: CNN cũng hỗ trợ việc học chuyển tiếp, có nghĩa là chúng có thể được đào tạo về một nhiệm vụ và sau đó được sử dụng để thực hiện một nhiệm vụ khác mà không cần đào tạo thêm. Điều này là do các tính năng được CNN trích xuất thường đủ chung để sử dụng cho nhiều nhiệm vụ, khiến chúng trở thành một công cụ linh hoạt cho nhiều ứng dụng khác nhau.
* Khả năng tự động trích xuất các đặc trưng cấp cao từ dữ liệu: Có nghĩa là CNN có thể học cách nhận dạng các mẫu trong hình ảnh mà không cần các kỹ thuật hay tính năng thủ công. Điều này khiến chúng trở nên lý tưởng cho các nhiệm vụ mà các tính năng liên quan đến nhiệm vụ không được biết trước vì CNN có thể học cách xác định các tính năng liên quan thông qua đào tạo.

Nhược điểm:

* Yêu cầu tính toán cao: Một trong những nhược điểm chính của CNN là yêu cầu tính toán cao. Điều này là do CNN thường có số lượng lớp và tham số lớn, đòi hỏi cần xử lý và bộ nhớ nhiều để huấn luyện và chạy. Điều này có thể làm cho chúng không thực tế để sử dụng trong một số ứng dụng có nguồn lực hạn chế.
* Khó khăn trên các tập dữ liệu nhỏ: CNN cũng yêu cầu các tập dữ liệu lớn để đạt được tỷ lệ chính xác cao. Điều này là do CNN học cách nhận biết các mẫu hình trong hình ảnh bằng cách phân tích nhiều ví dụ về các mẫu đó. Nếu tập dữ liệu quá nhỏ, CNN có thể quá phù hợp, nghĩa là nó trở nên quá chuyên biệt đối với tập dữ liệu huấn luyện và hoạt động kém trên dữ liệu mới.
* Khả năng khái quát hóa hạn chế: Cuối cùng, CNN có khả năng khái quát hóa hạn chế cho các tình huống mới. Điều này có nghĩa là chúng có thể hoạt động kém trên những hình ảnh rất khác so với những hình ảnh trong tập dữ liệu huấn luyện. Đây có thể là một vấn đề trong các ứng dụng mà CNN cần làm việc với nhiều loại hình ảnh khác nhau.

### Mô hình học kết hợp (Hybrid Learning Model)

Trong thời gian gần đây, một trong những lý do chính khiến các mô hình học sâu (Deep Learning) trở nên phổ biến là bởi chúng có khả năng tính toán trên dữ liệu phi cấu trúc mà không phụ thuộc quá nhiều vào các hàm tiền xử lý dữ liệu, điều cực kỳ khó và hầu như tất cả các thuật toán học máy (Machine Learning) cổ điển chưa đạt được. Trong các thuật toán học máy cổ điển, việc tìm hiểu và phân tích về tập dữ liệu huấn luyện đóng vai trò vô cùng quan trọng để dẫn tới hiệu suất và độ chính xác cuối cùng của thuật toán. Mặt khác, lớp phân lớp (classification layer) của mô hình học sâu hay lớp kết nối đầy đủ (Fully Connected Layers) cuối cùng của mô hình học sâu thông thường sử dụng hàm Softmax cho mục đích phân lớp lại dễ dẫn tới tình trạng học quá khớp (overfitting). Do đó, tận dụng điểm mạnh của cả 2 kỹ thuật Deep Learning và Machine Learning và kết hợp chúng lại với nhau, chúng ta áp dụng phương pháp học sâu lai (Deep Hybrid Learning) để tạo nên các mô hình học kết hợp.

Mô hình học kết hợp là sự tích hợp của mô hình học sâu có vai trò trích xuất các đặc trưng từ dữ liệu và các thuật toán học máy cổ điển đóng vai trò làm mô hình phân loại tính toán kết quả cuối cùng dựa trên các đặc trưng đã được trích xuất. Bằng cách áp dụng thuật toán học sâu lai, mô hình học kết hợp vừa tận dụng được điểm mạnh từ mô hình học sâu và thuật toán học máy, đồng thời giảm bớt nhược điểm của cả hai kỹ thuật trên và cung cấp các giải pháp có độ chính xác cao cũng như bớt tốn kém về mặt tính toán cho vấn đề cần giải quyết.

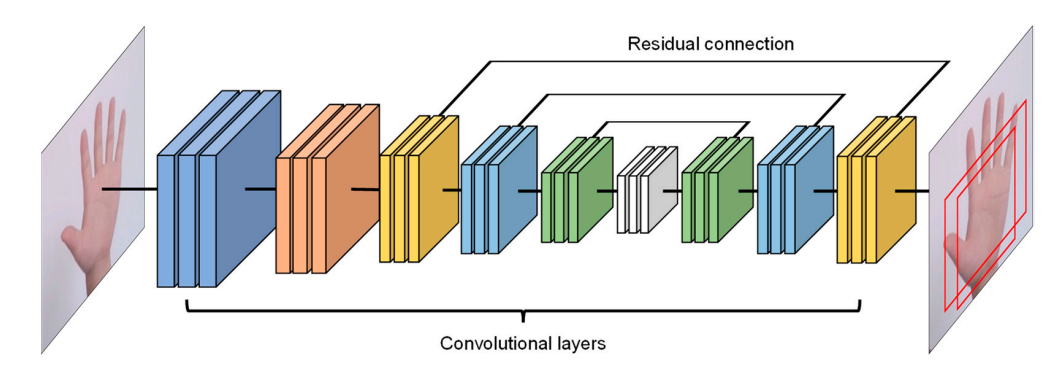
### Phương pháp Hand Tracking

"Hand Tracking" là một phương pháp được sử dụng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo để nhận diện, theo dõi và định vị vị trí của bàn tay và các ngón tay trong không gian 2D hoặc 3D. Đây là một kỹ thuật được áp dụng rộng rãi trong các ứng dụng liên quan đến tương tác người-máy, thực tế ảo, tăng cường thực tế và nhiều lĩnh vực khác. Hiện nay, với sự phát triển không ngừng của AI, trí tuệ nhân tạo, Google đã đưa ra MediaPipe Hands, một công cụ trong bộ công cụ MediaPipe, được sử dụng để thực hiện hand tracking (theo dõi bàn tay). MediaPipe Hands sử dụng các thuật toán và mô hình học máy để nhận diện, theo dõi và định vị vị trí của bàn tay và các ngón tay trong không gian 3D từ hình ảnh hoặc video.

MediaPipe Hands thực hiện hand tracking thông qua một ML pipeline bao gồm:

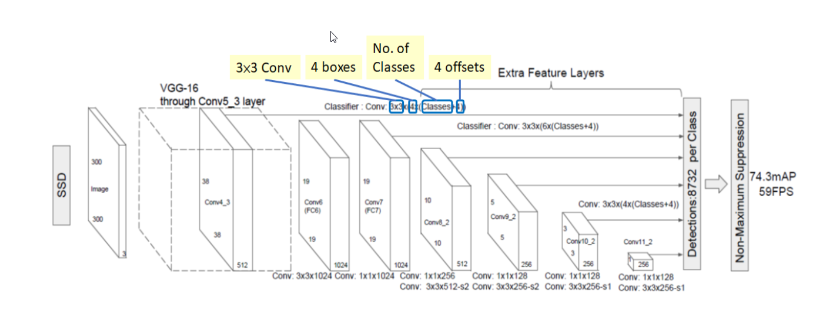
* **Palm Detector**

Đầu tiên, một mô hình được sử dụng để phát hiện vị trí và hình dạng của lòng bàn tay trong hình ảnh hoặc video. So với bàn tay có hình dạng thay đổi, hình dạng của lòng bàn tay đẹp hơn cố định, thường là hình chữ nhật và do đó dễ bị phát hiện hơn. Diện tích bàn tay có thể đạt được bằng cách mở rộng vùng lòng bàn tay. Ngoài ra, mô hình mạng có đồng hồ cát, cấu trúc có thể trích xuất các kích thước khác nhau của các khung giới hạn, do đó các tỷ lệ khác nhau của các vùng bàn tay có thể đươc tìm thấy. Mô hình này tạo ra một hộp giới hạn (bounding box) xác định vị trí của lòng bàn tay.



Hình 2. 10 Mô hình Palm Detector

Palm Detector là mô hình được thiết kế từ mô hình SSD (Single Shot Detector). Không giống như các kiến trúc cũ hơn có thể kể đến như R-CNN, fast R-CNN, đặc điểm của chúng là tốc độ xử lý thấp, không đáp ứng được trong việc object dection realtime. SSD có tốc độ xử lý nhanh mà vẫn đảm bảo về độ chính xác nhờ những thay đổi trong kiến trúc mạng nhằm gói gọn quá trình phát hiện và phân loại vật thể trong 1 lần và cắt bớt được các xử lý không cần thiết. Minh họa kiến trúc SSD



Hình 2. 11 Mô hình SSD

SSD dựa trên một tiến trình lan truyền thuận của một kiến trúc chuẩn (chẳng hạn VGG16) để tạo ra một khối feature map output gồm 3 chiều ở giai đoạn sớm. Chúng ta gọi kiến trúc mạng này là base network (tính từ input Image đến Conv7 trong hình 2.11). Sau đó chúng ta sẽ thêm những kiến trúc phía sau base network để tiến hành nhận diện vật thể như phần Extra Feature Layers trong sơ đồ.

Ưu điểm:

* Tốc độ xử lý rất nhanh, phù hợp cho việc phát hiện vật thể trong thời gian thực. Điều này đạt được nhờ việc loại bỏ bước tạo region proposal network và tích hợp toàn bộ quá trình phát hiện và phân loại vật thể vào một mạng neural duy nhất.
* SSD duy trì được độ chính xác cao nhờ sử dụng nhiều feature map với độ phân giải khác nhau, giúp nhận diện được các vật thể với kích thước và hình dạng đa dạng. Việc này đảm bảo rằng cả các vật thể lớn và nhỏ đều được phát hiện chính xác.
* SSD có kiến trúc đơn giản hơn so với các mô hình trước đây như R-CNN hay Fast R-CNN. Sự đơn giản này không chỉ giúp giảm thiểu tài nguyên tính toán mà còn dễ dàng hơn trong việc triển khai và huấn luyện mô hình.

Nhược điểm:

* SSD thường gặp khó khăn khi phát hiện các vật thể nhỏ trong hình ảnh. Mặc dù sử dụng nhiều feature map với độ phân giải khác nhau, độ phân giải của các feature map cuối cùng vẫn có thể không đủ chi tiết để phát hiện các vật thể rất nhỏ.
* Để đạt được hiệu suất tối ưu, SSD yêu cầu việc tối ưu hóa và tinh chỉnh cẩn thận, đòi hỏi nhiều tài nguyên tính toán và thời gian huấn luyện.
* **Hand Landmark Model**

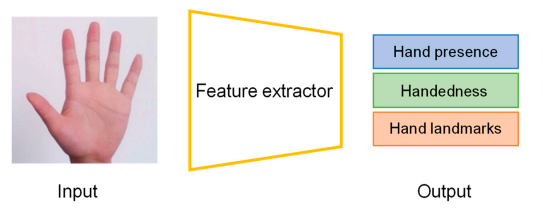
Sau khi chạy tính năng palm detector trên toàn bộ hình ảnh, mô hình mốc tay tiếp theo thực hiện định vị mốc chính xác của 21 tọa độ 2,5D bên trong vùng được phát hiện vùng tay thông qua hồi quy. Mô hình học cách biểu diễn tư thế bàn tay ngay cả khi bàn tay lộ rõ ​​một phần. Mô hình có ba đầu ra (xem hình 2.12):

1. Tham số biểu thị xác suất xuất hiện tay trong hình ảnh đầu vào.

2. Phân loại tay thuận, ví dụ: trái hay phải tay.

3. 21 mốc tay, điểm đặc trưng trên bàn tay trong không gian ảnh.

Model sử dụng cấu trúc liên kết tương tự cho 21 điểm mốc. Tọa độ 2D được học từ cả hình ảnh trong thế giới thực cũng như các bộ dữ liệu tổng hợp. Khi phát hiện bàn tay, để tránh sai sót sẽ có một ngưỡng chính xác để theo dõi bàn tay. Nếu điểm số là thấp hơn ngưỡng thì bộ dò sẽ được kích hoạt để đặt lại theo dõi. Thuận tay là một thuộc tính quan trọng để tương tác hiệu quả bằng tay.

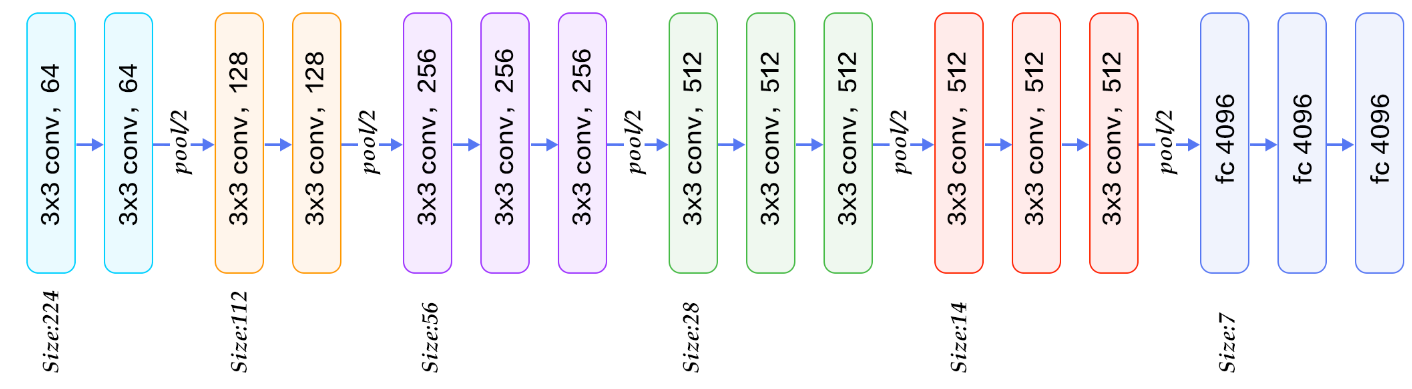


Hình 2. 12 Mô hình Hand Landmark Model

## Mô hình đề xuất cho bài toán

### Mô hình VGG16

VGG16 là mạng convolutional neural network được đề xuất bởi K. Simonyan and A. Zisserman, University of Oxford. Model sau khi train bởi mạng VGG16 đạt độ chính xác 92.7% top-5 test trong dữ liệu [ImageNet](http://www.image-net.org/)gồm 14 triệu hình ảnh thuộc 1000 lớp khác nhau.



Hình 2. 13. Minh hoa kiến trúc mạng VGG16

Cấu trúc:

* Convolutional layer: kích thước 3\*3, padding=1, stride=1. Mặc định sẽ là stride=1 và padding để cho output cùng width và height với input.
* Pool/2 : max pooling layer với size 2\*2
* 3\*3 conv, 64: thì 64 là số kernel áp dụng trong layer đấy, hay depth của output của layer đấy.
* Càng các convolutional layer sau thì kích thước width, height càng giảm nhưng depth càng tăng.
* Sau khá nhiều convolutional layer và pooling layer thì dữ liệu được flatten và cho vào fully connected layer.

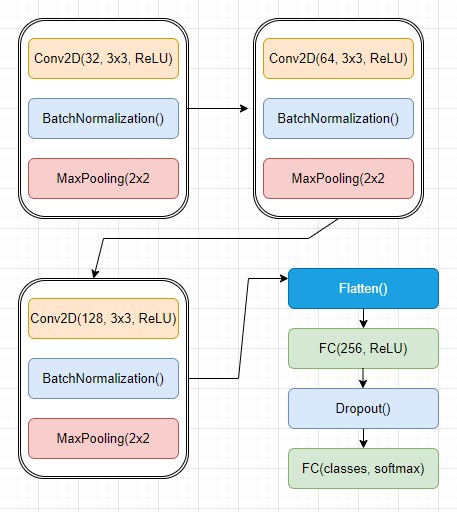
### Mô hình mạng nơ ron tích chập (CNN)

Hiệu quả của mô hình CNN phụ thuộc chủ yếu vào kiến trúc của mạng. Trong phần này, tôi sẽ giới thiệu một kiến trúc mạng nơ ron tích chập được áp dụng để giải quyết bài toán. Mô hình này đồng thời cũng được sử dụng như một bộ trích xuất thuộc tính sử dụng trong mô hình học kết hợp được trình bày ở các phần sau. Mục tiêu là sử dụng một mạng không quá phức tạp nhưng vừa đủ cho các mục đích trích chọn đặc trưng và cắt giảm số chiều của ảnh. Mô hình được đề xuất (Hình 2.14) bao gồm 3 khối chính với 3 lớp tích chập (Convolutional layer). Lớp tích chập có số lượng các bộ lọc (Filter) được đề xuất lần lượt là 32, 64, 128.

Trong mỗi lớp tích chập, hàm kích hoạt phi tuyến ReLU có dạng 𝑓(𝑥) = max (0, 𝑥) sẽ được thực hiện để chuyển các giá trị âm thành 0. Tiếp theo, lớp Batch Normalization (BN) có vai trò chuẩn hóa các đặc trưng về trạng thái zero-mean với độ lệch chuẩn bằng 1. Lớp BN giúp tránh được hiện tượng các giá trị rơi vào khoảng bão hòa sau khi đi qua các hàm kích hoạt phi tuyến, tức là đảm bảo rằng không có giá trị nào quá cao hoặc quá thấp. Điều này có thể giúp giảm thiểu hiện tượng quá khớp (overfitting).

Lớp gộp (Max pooling layer) với kích thước 2x2 được đặt cuối mỗi khối nhằm chắt lọc thông tin hữu ích, loại bỏ các thông tin gây nhiễu cũng như giảm chiều của dữ liệu ảnh đầu vào giúp giảm thời gian huấn luyện của mô hình. Dữ liệu sau khi qua 3 khối chính nêu trên sẽ được làm phẳng để đưa vào phân lớp. Hai lớp kết nối đầy đủ (Fully Connected layer - FC) được triển khai. Lớp FC đầu tiên có kích thước 256 node. Dữ liệu đầu ra của lớp này cũng được chuyển tới các lớp ReLU và GN trước khi đưa tới lớp FC cuối cùng để phân lớp. Lớp FC này có số node bằng với số lớp trong bộ dữ liệu huấn luyện (classes) và sử dụng hàm phân loại Softmax.

Hàm tổn thất (loss function) Sparse categorical cross entropy và thuật toán tối ưu (optimizer) Adam được sử dụng trong quá trình huấn luyện. Mô hình sẽ được huấn luyện trong 50 epoch và batch size là 32.



Hình 2. 14 Mô tả kiến trúc mô hình CNN

### Mô hình học kết hợp CNN-SVM

Tiếp đó, tôi sẽ trình bày về mô hình học kết hợp CNN-SVM là sự kết hợp của mạng nơ ron nhân tạo CNN và SVM, tuy nhiên quá trình huấn luyện của CNN và SVM được tách rời. Trong đó, mô hình CNN đóng vai trò như một bộ trích xuất đặc trưng có nhiệm vụ lọc nhiễu, trích rút đặc trưng và giảm số chiều trên dữ liệu ảnh đầu vào. Véc tơ đặc trưng thu được từ mô hình này được xem như đầu vào của mô hình SVM đa lớp. Sự kết hợp này được kỳ vọng sẽ tận dụng được ưu điểm của cả hai mô hình khi mà CNN thực hiện việc trích chọn đặc trưng rất hiệu quả trên ảnh, trong khi SVM lại có độ chính xác phân lớp tốt nếu dữ liệu đầu vào được tiền xử lý hiệu quả.

Mô hình của hệ thống phân loại sẽ gồm 2 thành phần chính:

• Mô hình trích xuất đặc trưng (Feature Extractor): đặc trưng (feature) là những thuộc tính quan trọng và độc nhất của một bức ảnh. Việc trích xuất các đặc trưng cần được thực hiện bằng một thuật toán phù hợp nhằm đạt được kết quả tốt nhất tại mô hình phân lớp. Mạng nơ ron tích chập CNN là một lựa chọn tiêu biểu cho vấn đề trên với khả năng lọc nhiễu, giảm số chiều trên ảnh và có thể tự động học cách trích xuất các đặc trưng phức tạp bằng cách sử dụng nhiều bộ lọc khác nhau trong quá trình huấn luyện.

• Mô hình phân lớp (Classifier): Sau khi trích xuất các đặc trưng, các kỹ thuật học máy dựa trên cơ sở toán học chặt chẽ được áp dụng để xây dựng một bộ phân lớp các ký hiệu. Quá trình nghiên cứu sẽ tập trung vào thuật toán SVM trong việc xây dựng mô hình phân lớp.

Mạng nơ ron tích chập được giới thiệu trong Hình 2.13 có thể tóm lược lại thành các bước như sau:

(1) CONV2D: 3x3, 32 filters, activation: ReLU max(0, x)

(2) BatchNormalization

(3) MaxPooling: 2x2

(4) CONV2D: 3x3, 64 filters, activation: ReLU max(0, x)

(5) BatchNormalization

(6) MaxPooling: 2x2

(7) CONV2D: 3x3, 128 filters, activation: ReLU max(0, x)

(8) BatchNormalization

(9) MaxPooling: 2x2

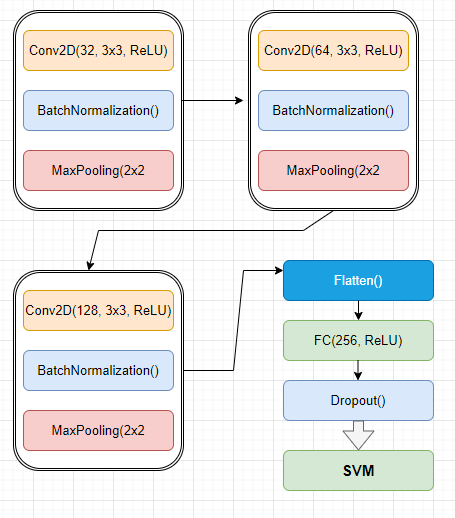
(10) Flatten

(11) FC: 256 hidden neurons, ReLu max(0, x)

(12) GroupNormalization

(13) FC: classes, Softmax

Để sử dụng mạng nơ ron này cho mô hình học chuyển tiếp CNN-SVM, trước tiên mạng CNN được huấn luyện với đầy đủ các bước. Quá trình tiếp theo được thực hiện như sau: Mỗi ảnh đầu vào được đưa qua mạng CNN đã được huấn luyện ở trên nhưng bỏ đi các bước (12) và (13). Đầu ra của bước (11) được lưu lại dưới dạng véc tơ đặc trưng 256 chiều. Trong pha tiếp theo, véc tơ đặc trưng này được chuyển tiếp tới một mô hình SVM sử dụng chiến lược phân đa lớp one vs one và đóng vai trò như đầu vào của mô hình này. Hình 2.15 dưới đây minh họa một cách trực quan hơn mô hình CNN-SVM.



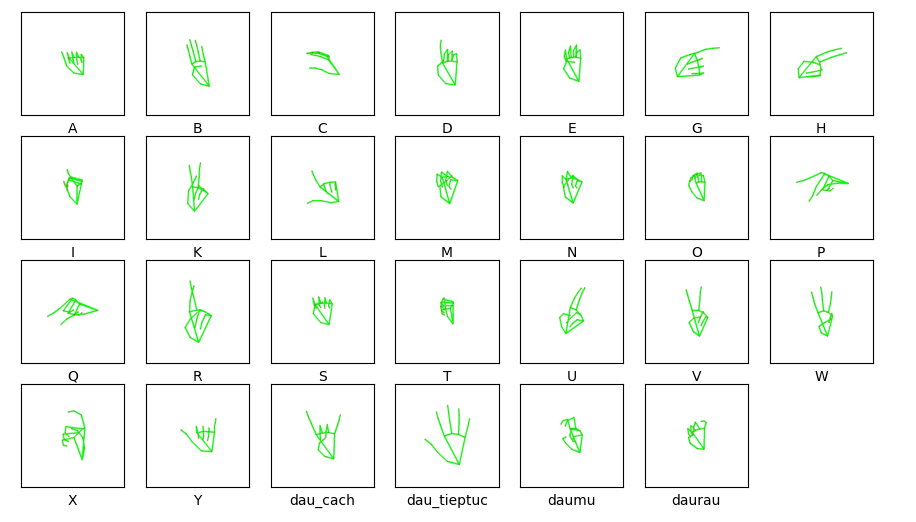
Hình 2. 15 Minh họa mô hình học kết hợp CNN-SVM

# THỰC NGHIỆM

## Bộ dữ liệu

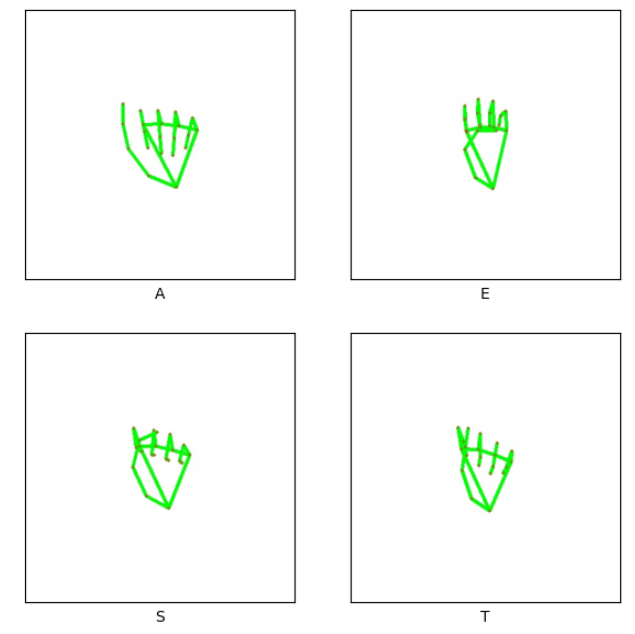
Quá trình thực nghiệm được tôi tiến hành trên bộ dữ liệu ký hiệu tay là VN\_Handsign. Bộ dữ liệu sử dụng ảnh màu RGB.

Bộ dữ liệu VN\_Handsign do tôi nghiên cứu tự xây dựng và xử lý. Bộ dữ liệu được tạo nên sau khi sử dụng kỹ thuật Hand Tracking để phát hiện và vẽ các điểm mốc bàn tay lên ảnh. Bộ dữ liệu gồm 27 ký tự phổ biến của ngôn ngữ ký hiệu.

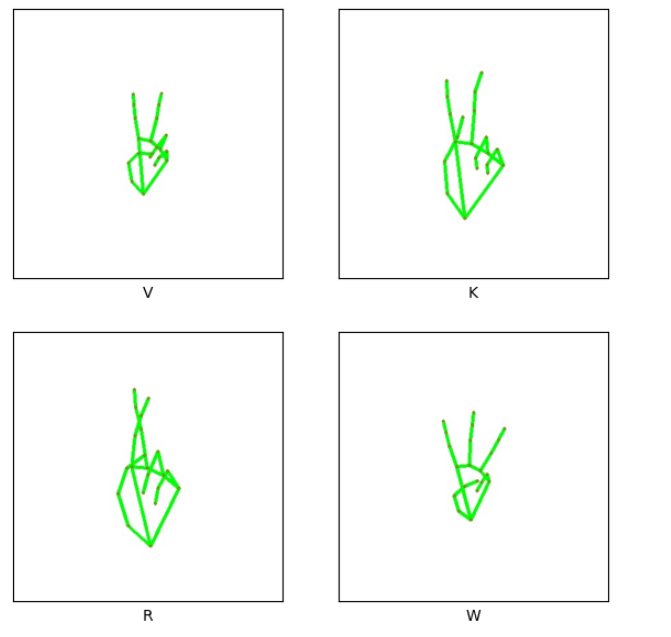


Hình 3. 1 Minh họa dataset

Các mẫu ảnh trong bộ dữ liệu có số lượng giống ở tất cả ký hiệu. Có một số ký hiệu khá giống nhau nếu ta không để ý kỹ. Ví dụ ký hiệu A, E, S, T.



Hình 3. 2 Một số ký hiệu hay nhầm



Hình 3. 3 Một số ký hiệu hay nhầm khác

Thông số chi tiết của các bộ dữ liệu, xin xem trong bảng dưới đây

Bảng 3. 1 Dữ liệu thực nghiệm

|  |  |  |  |
| --- | --- | --- | --- |
| Dữ liệu | Số mẫu | Số lớp | Kích thước(pixel) |
| VN\_handsign | 12960 | 27 | 400x400x3 |

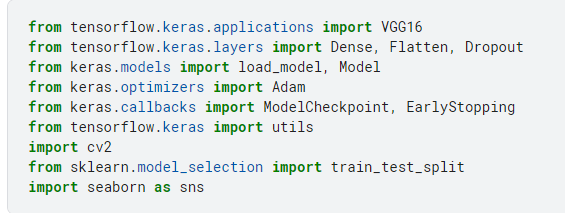
## Huấn luyện mô hình

Các mô hình được tôi nghiên cứu tiến hành thực nghiệm trên môi trường Kaggle với ngôn ngữ lập trình python. Kaggle hỗ trợ GPU Tesla T4 trong 30 giờ sử dụng liên tục giúp quá trình huấn luyện các mô hình diễn ra nhanh hơn.

Import các thư viện cần thiết

### Huấn luyện mô hình VGG16

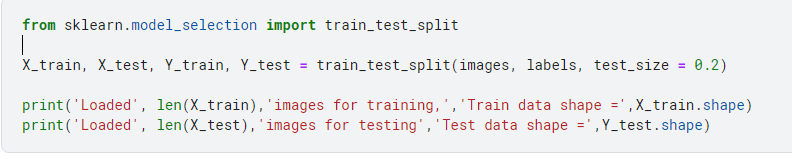
Khai báo các thư viện cần thiết



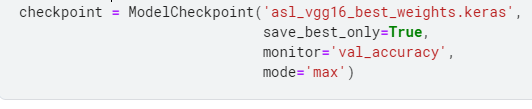
Ánh xạ các từ khóa vào các giá trị số để sử dụng trong quá trình huấn luyện mô hình. Khởi tạo hai danh sách rỗng images và labels để lưu trữ dữ liệu hình ảnh và nhãn tương ứng, thay đổi kích thước về 200x200 pixels và chuyển đổi ảnh thành một mảng numpy. Chuẩn hóa giá trị pixel của hình ảnh về khoảng từ 0 đến 1 bằng cách chia cho 255.0. Chuyển đổi nhãn thành dạng one-hot encoding sử dụng utils.to\_categorical.



Sử dụng hàm train\_test\_split từ module sklearn.model\_selection để chia dữ liệu thành hai phần: dữ liệu huấn luyện (X\_train, Y\_train) và dữ liệu kiểm tra (X\_test, Y\_test). Dữ liệu được chia 20% dữ liệu sẽ được sử dụng cho việc kiểm tra, còn lại 80% sẽ được sử dụng cho việc huấn luyện.



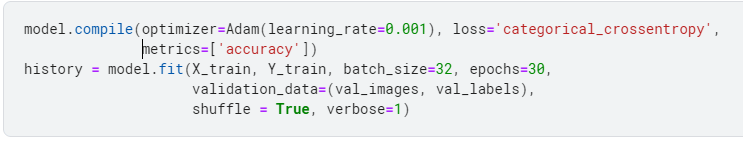
Thiết lập checkpoints và mô hình (Setting Checkpoints and Model): Sử dụng ModelCheckpoint để lưu lại model tốt nhất.



Khởi tạo mô hình và thêm các lớp (Initialize Model and Add Layers):



Sử dụng compile để cấu hình mô hình và sau đó sử dụng fit để huấn luyện mô hình trên dữ liệu huấn luyện. Sử dụng save để lưu mô hình đã được huấn luyện ra file.



### Huấn luyện mô hình CNN

Mô hình được xây dựng bằng Sequential API của Keras. Nó bao gồm các lớp Convolutional (Conv2D), hàm kích hoạt ReLU (Activation('relu')), lớp MaxPooling (MaxPooling2D), lớp BatchNormalization (BatchNormalization), lớp Flatten (Flatten), các lớp Dense (Dense) và lớp Dropout (Dropout):

* Các lớp Convolutional với các bộ lọc có kích thước khác nhau.
* Hàm kích hoạt ReLU được sử dụng sau mỗi lớp Convolutional.
* Lớp MaxPooling được sử dụng để giảm kích thước của feature map.
* Lớp BatchNormalization được sử dụng để chuẩn hóa các giá trị đầu ra của các lớp trước khi đưa vào hàm kích hoạt.
* Lớp Flatten được sử dụng để chuyển đổi feature map thành một vector 1D.
* Lớp Dense với các đơn vị kích hoạt ReLU và Softmax.
* Lớp Dropout để giảm overfitting bằng cách tắt ngẫu nhiên một số đơn vị đầu ra trong quá trình huấn luyện.

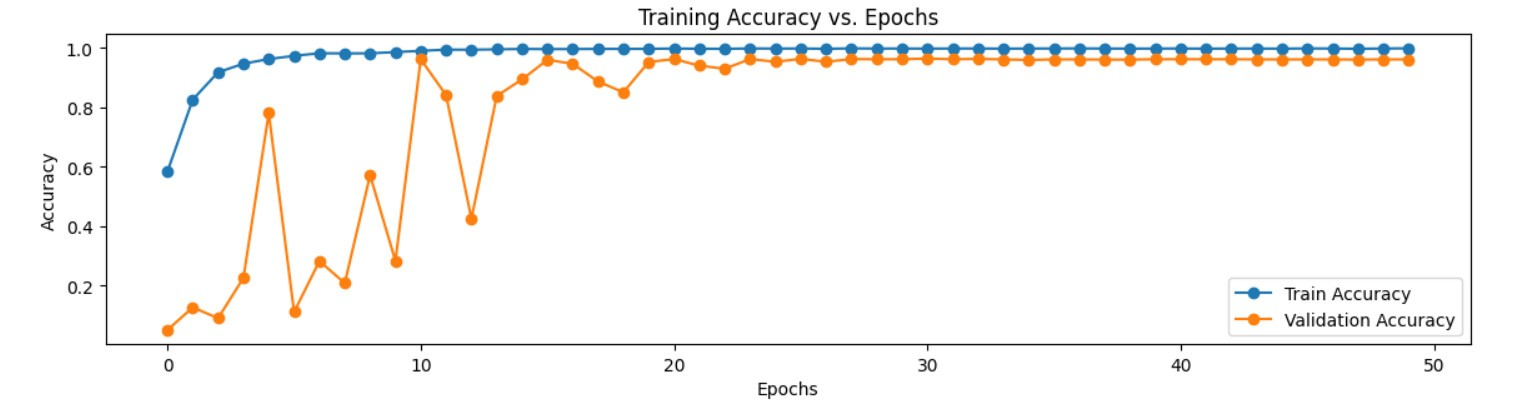


Hình 3. 4 Minh họa mô hình CNN

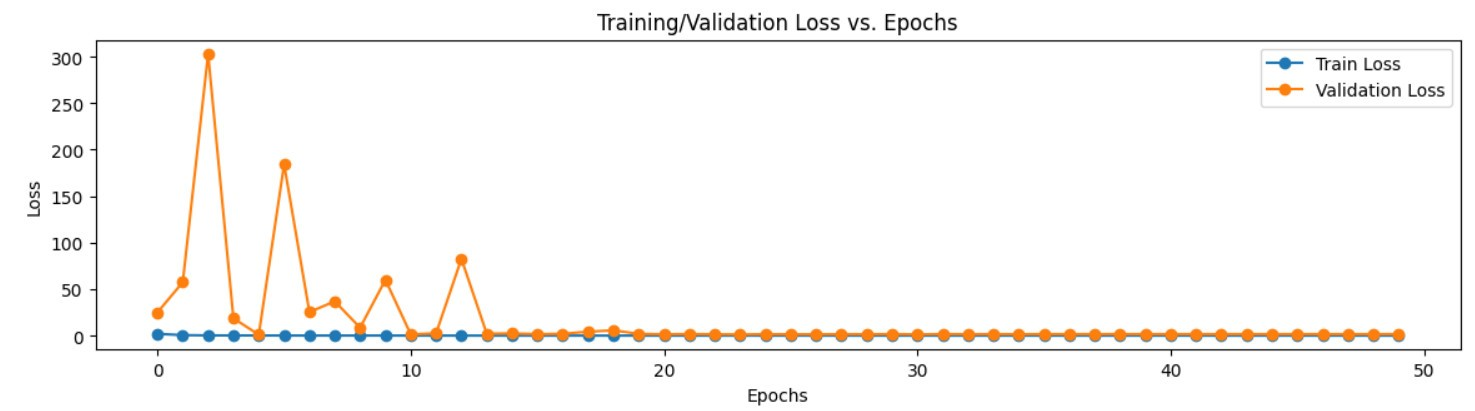
### Huấn luyện mô hình kết hợp CNN-SVM

Các tham số cho mô hình CNN-SVM được xác định như sau: Tham số điều chỉnh C được chọn lần lượt trong tập {10-1, 1, 10}. Hàm nhân được sử dụng là Gaussian Kernel hay Radial Basic Function (RBF) với tham số 𝜎 được chọn lần lượt trong tập {10-6, 10-5, 10-4, 10-3, 10-2, 10-1}.

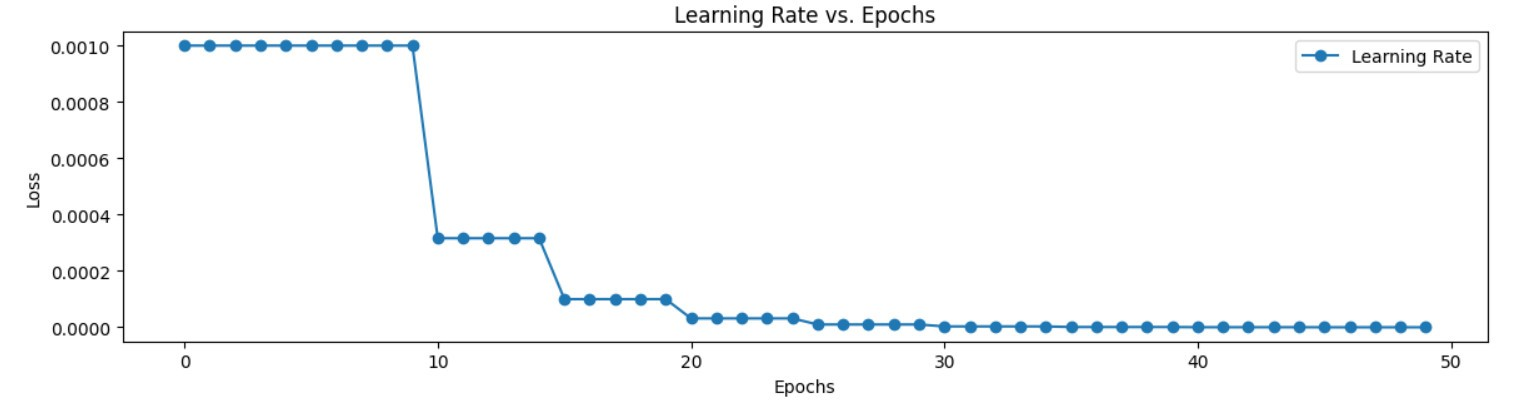
## Các kết quả thực nghiệm

Dưới đây là kết quả sau khi tôi tiến hành thử nghiệm mô hình CNN với bộ dữ liệu VN\_handsign. Hình bên dưới lần lượt biểu thị độ chính xác phân lớp và giá trị hàm tổn thất qua các lần lặp (Epochs) trong quá trình huấn luyện với bộ dữ liệu VN\_handsign. 

Hình 3. 5 Độ chính xác phân lớp của CNN qua các lần lặp



Hình 3. 6 Giá trị hàm mất mát của CNN qua các lần lặp



Hình 3. 7 Giá trị learning rate qua các lần lặp

Các kết quả cho thấy quá trình huấn luyện diễn ra không ổn định lúc ban đầu và sau đó dần dần ổn định lại, không còn hiện tượng giảm đột ngột (drop) của Accuracy sau 50 epoch. Tiếp theo là kết quả của các mô hình VGG16, mô hình kết hợp CNN-SVM.

Bên cạnh đó, tôi cũng tiến hành thử nghiệm với mô hình rất nổi tiếng trong bài toán nhận dạng qua hình ảnh là MobileNetV2 (kèm với trọng số pre-trained trên tập dữ liệu ImageNet được cung cấp sẵn nhằm áp dụng Transfer Learning) trên cùng các bộ dữ liệu và cùng phương pháp huấn luyện với mô hình CNN để so sánh. Chi tiết các số liệu được thể hiện trong bảng dưới đây.

Bảng 3. 2 Độ chính xác phân lớp của các mô hình trên độ đo Accuracy

|  |  |  |
| --- | --- | --- |
|  | Train dataset | Test dataset |
| VGG16 | 90.32% | 90.12% |
| CNN | 96.48% | 95.61% |
| CNN-SVM | **97.2%** | **96.17%** |
| MobileNetV2 | 89.8% | 89.4% |

Qua các kết quả có thể thấy rằng mô hình CNN được đề xuất đảm bảo được độ chính xác phân lớp tốt đạt kết quả tốt hơn với 2 mô hình VGG16 và MobileNetV2 với bộ dữ liệu VN\_Handsign trên cả tập dữ liệu huấn luyện (training set) và tập dữ liệu kiểm tra (test set) dù cần ít thời gian huấn luyện hơn.

Cuối cùng là khả năng phân lớp tốt của mô hình CNN-SVM khi đạt kết quả ấn tượng 96.17% trên bộ dữ liệu kiểm tra.

Không chỉ đánh giá trên mỗi Accuracy, tôi cũng tiến hành thực nghiệm chi tiết hơn trên các mô hình với nhiều độ đo khác nhau. Các kết quả thu được trong Bảng 3.3 được thực hiện chỉ trên tập dữ liệu kiểm tra (test set) của 2 bộ dữ liệu nêu trên.

Bảng 3. 3 Kết quả thử nghiệm các mô hình trên một số độ đo khác

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision (%) | Recall (%) | F1-score (%) |
| VGG16 | 90.42 | 90.36 | 90.67 |
| CNN | 94.32 | 96.35 | 95.03 |
| CNN-SVM | 95.56 | 96.28 | 96.15 |
| MobileNetV2 | 89.76 | 89.32 | 89.12 |

Các kết quả thử nghiệm trên một số độ đo khác như Precision, Recall, F1- score trên bảng 4 cũng cho kết quả gần như tương tự với độ đo Accuracy: mô hình CNN được để xuất thể hiện tốt trên bộ dữ liệu, mô hình CNN-SVM cung cấp kết quả phân lớp tốt nhất và ổn định nhất trong tất cả các mô hình thực nghiệm.

# XÂY DỰNG SẢN PHẨM DEMO

## Giới thiệu về công cụ Tkinter

Tkinter là một thư viện được tích hợp sẵn trong Python, chuyên dành cho việc phát triển giao diện người dùng đồ họa (GUI). Được xây dựng trên nền tảng toolkit đồ họa Tk, Tkinter cung cấp một cách đơn giản và hiệu quả để tạo ra các ứng dụng có giao diện đồ họa trực quan và dễ sử dụng.

Dù đã được phát hành vào nhiều năm trước, nhưng tkinter đã sớm trở nên nổi tiếng nhờ những điểm mạnh của mình:

* Dễ học và sử dụng: Tkinter được biết đến với cú pháp đơn giản và dễ học, làm cho nó là một lựa chọn lý tưởng cho người mới bắt đầu với lập trình GUI trong Python.
* Tích hợp sẵn trong python: Tkinter là một phần của thư viện tiêu chuẩn của Python, không yêu cầu cài đặt bổ sung. Điều này giúp giảm độ phức tạp và giữ cho mã nguồn có thể chạy trên nhiều nền tảng mà không cần thay đổi.
* Đa dạng trong xây dựng giao diện: Tkinter hỗ trợ nhiều loại widget như nút, ô văn bản, hộp thoại, menu, và nhiều widget khác, cho phép bạn xây dựng giao diện người dùng phong phú và đa dạng.
* Tương thích đa nền tảng: Ứng dụng sử dụng Tkinter có thể chạy trên nhiều hệ điều hành khác nhau mà không cần sửa đổi mã nguồn, tăng tính tương thích và di động.

Lợi ích của việc sử dụng Tkinter trong python:

* Phát triển nhanh chóng: Tkinter giúp người phát triển nhanh chóng xây dựng giao diện người dùng mà không cần nhiều đoạn mã phức tạp.
* Phù hợp cho ứng dụng nhỏ và trung bình: Đối với các ứng dụng nhỏ và trung bình, Tkinter cung cấp một giải pháp hiệu quả và đủ mạnh mẽ để tạo ra giao diện người dùng thân thiện.
* Cộng đồng lớn và tài nguyên phong phú: Tkinter có sự hỗ trợ từ cộng đồng lớn người phát triển Python, điều này có nghĩa là có nhiều tài liệu, ví dụ và hỗ trợ trực tuyến.
* Sự kết hợp với python: Tkinter tích hợp tốt với Python, cho phép bạn kết hợp lập trình hàm và lập trình hướng đối tượng một cách linh hoạt.
* Ứng dụng phổ quát: Tkinter thích hợp cho nhiều loại ứng dụng, từ các ứng dụng desktop đơn giản đến các ứng dụng quản lý dự án và các công cụ biểu đồ.

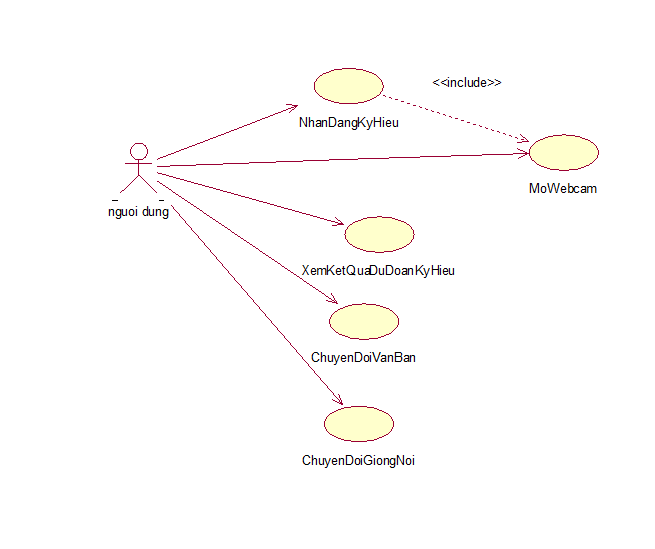
Các bước xây dựng phần mềm nhận diện ngôn ngữ ký hiệu tay

1. Đầu tiên, cài đặt thư viện Tkinter bằng lệnh pip install tkinter và import tkinter.
2. Tạo cửa sổ chính của ứng dụng GUI bao gồm các nút chức năng và giao diện tương tích.
3. Tạo các màn hình con tương ứng với từng chức năng để thực hiện một chức năng cụ thể.
4. Gọi các màn hình con để liên kết với các chức năng của màn hình chính để tạo thành một ứng dụng hoàn chỉnh.

## Phân tích thiết kế hệ thống

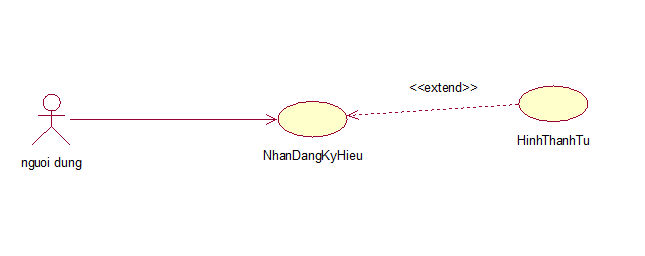
### Biểu đồ use case

* + - 1. Biểu đồ use case tổng quát



Hình 4. 1 Biểu đồ use case tổng quát

* + - 1. Phân rã use case Nhận dạng ký hiệu



Hình 4. 2 Phân rã use case Nhận dạng ký hiệu

### Mô tả chi tiết các use case

4.2.2.1. Mô tả use case Nhận dạng ký hiệu

|  |
| --- |
| **Tên use case**: Nhận dạng ký hiệu |
| **Mô tả tóm tắt**: Use case cho phép người dùng nhận dạng ngôn ngữ ký hiệu cử chỉ tay qua webcam |
| **Luồng sự kiện**  - Luồng cơ bản:  1. Người dùng có thể nhấp chọn nút “Chuyển ngôn ngữ ký hiệu sang văn bản/giọng nói” trên giao diện để chuyển sang màn hình thực hiện nhận dạng:  1.1. Người dùng nhấp chọn nút “Chuyển ngôn ngữ ký hiệu sang văn bản/giọng nói” trên giao diện trang chủ, hệ thống sẽ truy cập tới webcam trên thiết bị của người dùng.  1.1.1 Trường hợp người dùng không có webcam, hệ thống báo lỗi, quay trở lại giao diện ban đầu.  1.1.2 Trường hợp người dùng có webcam, hệ thống chuyển sang giao diện nhận dạng.  2. Sau khi tiến hành nhận dạng ký hiệu từ người dùng chọn, hệ thống sẽ trả về kết quả cho người dùng theo thời gian thực ở bên ô dự đoán kết kết quả ký hiệu.  - Luồng rẽ nhánh: Không có |
| **Các yêu cầu đặc biệt**: Không có |
| **Tiền điều kiện**: Người dùng đã bật webcam |
| **Hậu điều kiện**: Không có |
| **Các điểm mở rộng**: Không có |

4.2.2.2. Mô tả use case Chuyển đổi văn bản

|  |
| --- |
| **Tên use case**: Chuyển đổi văn bản |
| **Mô tả tóm tắt**: Use case cho phép người dùng chuyển đổi văn bản sang hình ảnh ngôn ngữ ký hiệu |
| **Luồng sự kiện**  - Luồng cơ bản:  1. Người dùng nhập văn bản cần chuyển đổi sang ngôn ngữ ký hiệu vào ô “Văn bản” và nhấn vào nút “Chuyển đổi” trên giao diện hệ thống.  1.1. Người dùng nhấp chọn nút “Chuyển đổi” trên giao diện trang chủ, hệ thống sẽ truy cập tới bộ lưu trữ ảnh ngôn ngữ ký hiệu trên thiết bị của người dùng.  1.1.1 Trường hợp người dùng không nhập văn bản vào ô “Văn bản” nhưng chọn nút “Chuyển đổi” của hệ thống, ứng dụng tự động quay về giao diện trang chuyển đổi.  1.1.2 Trường hợp người dùng nhập văn bản vào ô “Văn bản” và chọn nút “Chuyển đổi” của hệ thống. Hình ảnh ngôn ngữ ký hiệu sẽ được hiển thị bên cạnh ô văn bản.  2. Sau khi tiến hành chuyển đổi văn bản sang ngôn ngữ ký hiệu từ người dùng, hệ thống sẽ trả kết quả ngay bên cạnh lần lượt ký hiệu để tạo nên văn bản.  - Luồng rẽ nhánh: Không có |
| **Các yêu cầu đặc biệt**: Không có |
| **Tiền điều kiện**: Người dùng nhập văn bản cần chuyển đổi |
| **Hậu điều kiện**: Không có |
| **Các điểm mở rộng**: Không có |

4.2.2.3 Mô tả use case Chuyển đổi giọng nói

|  |
| --- |
| **Tên use case**: Chuyển đổi giọng nói |
| **Mô tả tóm tắt**: Use case cho phép người dùng chuyển đổi giọng nói sang hình ảnh ngôn ngữ ký hiệu |
| **Luồng sự kiện**  - Luồng cơ bản:  1. Người dùng chọn nút “Giọng nói” và bắt đầu nói thông tin cần chuyển đổi sang ngôn ngữ ký hiệu và nhấn vào nút “Chuyển đổi” trên giao diện hệ thống.  1.1. Người dùng nhấp chọn nút “Chuyển đổi” trên giao diện trang chủ, hệ thống sẽ truy cập tới bộ lưu trữ ảnh ngôn ngữ ký hiệu trên thiết bị của người dùng.  1.1.1 Trường hợp người dùng không chọn nút giọng nói nhưng chọn nút “Chuyển đổi” của hệ thống, ứng dụng tự động quay về giao diện trang chuyển đổi.  1.1.2 Trường hợp người dùng Chọn nút “Giọng nói” nhưng không nói gì và chọn nút “Chuyển đổi” của hệ thống, ứng dụng tự động quay về giao diện trang chuyển đổi.  1.1.3 Trường hợp người dùng Chọn nút “Giọng nói”, nói thông tin cần chuyển đổi sang ngôn ngữ ký hiệu. Thông tin cần chuyển đổi sang ngôn ngữ ký hiệu sẽ hiện ở ô Văn bản và hình ảnh ngôn ngữ ký hiệu sẽ được hiển thị bên cạnh ô văn bản. Người dùng chọn nút “Chuyển đổi” của hệ thống, hình ảnh ngôn ngữ ký hiệu sẽ được hiển thị bên cạnh ô văn bản.  2. Sau khi tiến hành chuyển đổi giọng nói sang ngôn ngữ ký hiệu từ người dùng, hệ thống sẽ trả kết quả ngay bên cạnh lần lượt ký hiệu sau khi chuyển đổi của thông tin giọng nói.  - Luồng rẽ nhánh: Không có |
| **Các yêu cầu đặc biệt**: Không có |
| **Tiền điều kiện**: Người dùng nói thông tin chuyển đổi |
| **Hậu điều kiện**: Không có |
| **Các điểm mở rộng**: Không có |

## Giao diện ứng dụng

Giao diện được thiết kế với tông màu sáng, thao tác đơn giản và dễ sử dụng. Hiện tại hệ thống chỉ hỗ trợ ngôn ngữ Tiếng Việt với mục đích chính hướng tới người sử dụng trong nước nhưng trong tương lai hệ thống sẽ được nâng cấp để hỗ thêm một số ngôn ngữ thông dụng như tiếng Anh và tiếng Trung Quốc. Chi tiết giao diện được trình bày qua các hình bên dưới.



Hình 4. 3 Giao diện màn hình chính

Các chức năng chính của hệ thống như chuyển đổi văn bản/ giọng nói sang ngôn ngữ ký hiệu hay chuyển đổi ngôn ngữ ký hiệu sang văn bản/giọng nói đều đặt ở trang chủ để thuận tiện cho người sử dụng.



Hình 4. 4 Chuyển văn bản/giọng nói sang ngôn ngữ ký hiệu

Từ văn bản được người dùng nhập vào từ bàn phím hoặc từ giọng nói mà người dùng khi bấm vào nút “Giọng nói” thì người dùng có thể dễ dàng chuyển đổi sang ngôn ngữ ký hiệu được hiện ở phần kết quả.



Hình 4. 5 Chuyển ngôn ngữ ký hiệu sang văn bản

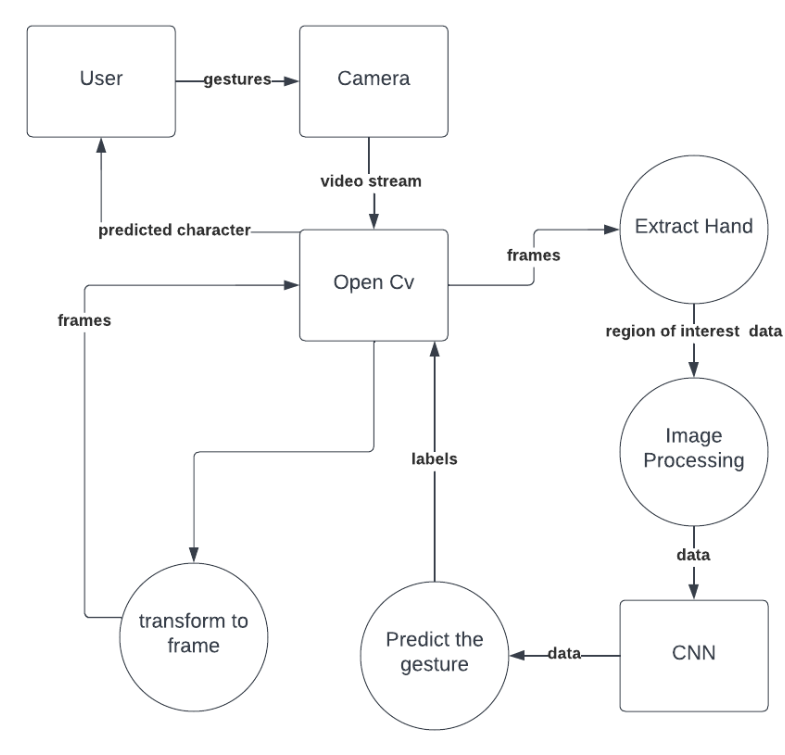
Từ hình ảnh theo thời gian thực được thu bởi webcam, người dùng có thể dễ dàng chuyển đổi từ ngôn ngữ ký hiệu sang văn bản hoặc giọng nói. Ngoài ra, hệ thống có nút “Trở lại” giúp người dùng có thể thuận tiện quay lại màn hình chính để dùng chức năng khác của hệ thống.

## Chức năng chính của hệ thống

Hệ thống có chức năng chính là nhận dạng ngôn ngữ ký hiệu Tiếng Việt theo thời gian thực qua webcam, camera. Hệ thống có thể giúp chuyển đổi 27 ký hiệu cơ bản của ngôn ngữ ký hiệu tiếng việt. Đặc biệt, hệ thống sử dụng lại các ký hiệu đã được nhận diện để có thể làm các thanh dấu khi cần thiết, điều này giúp hạn chế huấn luyện nhiều cử chỉ gần giống nhau gây sai lệch cho mô hình. Các ký hiệu sau khi nhận diện có thể ghép thành câu và sử dụng chức năng “Nói” để phát âm thanh câu văn đó.

Khi người dùng thực hiện nhận diện qua webcam máy tính, ta sử dụng opencv để đọc các frame từ webcam sau đó chuyển đổi frame vừa đọc dưới dạng numpy. Sau đó, ta sử dụng Hand Tracking để phát hiện bàn tay. Sau khi thực hiện xử lý dữ liệu sau khi phát hiện bàn tay ta thực hiện cho vào mô hình CNN để dự đoán cử chỉ tay. Tiếp đó, kết quả dự đoán cử chỉ tay được hiện thị trên giao diện cho người dùng.

Mô hình hoạt động của chức năng nhận diện được khái quát và thể hiện qua Hình 4.6 dưới đây.



Hình 4. 6 Mô hình hoạt động chức năng nhận diện ảnh

# KẾT LUẬN

Thời gian làm đồ án tốt nghiệp vừa qua là trải nghiệm vô cùng thú vị và đáng giá với bản thân em. Em đã tìm hiểu các kỹ thuật khác nhau của trí tuệ nhân tạo nhằm giải quyết bài toán nhận dạng ngôn ngữ ký hiệu tay-hỗ trợ giao tiếp người khiếm thính, qua đó không chỉ tích lũy được các kinh nghiệm về chuyên môn mà còn học được cái kĩ năng làm việc độc lập, quản lý thời gian. Đây chắc chắn sẽ là hành trang quý giá trong sự nghiệp tương lai của em.

Em đã tìm hiểu, nghiên cứu, ứng dụng các mô hình học máy cũng như tận dụng các nghiên cứu từng được công bố về nhận dạng cử chỉ tay để hoàn thành đề tài Nghiên cứu các kỹ thuật của Trí tuệ nhân tạo cùng các mô hình học máy Deep Learning, Support Vector Machine và ứng dụng cho bài toán nhận dạng ngôn ngữ ký hiệu. Dưới sự hướng dẫn của cô Chu Thị Quyên, em đã xây dựng thành công hệ thống nhận dạng ngôn ngữ ký hiệu tay, hỗ trợ giao tiếp cho người khiếm thính. Hy vọng rằng nghiên cứu của em sẽ đóng góp vào việc tạo ra một môi trường hòa nhập và thuận lợi hơn cho người khiếm thính trong xã hội.

Mặc dù đã đạt được một số thành công nhất định nhưng hệ thống vẫn còn một số hạn chế như nhận diện nhầm một số ký hiệu gần giống nhau. Em nhận thức rằng hành trình xây dựng và phát triển một hệ thống hoàn thiện mang lợi ích lớn cho cộng đồng này vẫn còn rất xa. Em rất mong được được các thầy cô đóng góp ý kiến để có thể cái thiện, nâng cao hiệu suất và tính ứng dụng của sản phẩm.

Em xin được gửi lời cảm ơn chân thành nhất tới cô giáo, Thạc sĩ Chu Thị Quyên đã tận tình hướng dẫn em thực hiện đề tài này. Em xin chúc cô luôn luôn mạnh khỏe và thành công trong những nghiên cứu sắp tới.

Em xin trân trọng cảm ơn!

Nguyễn Minh Hiếu

# TÀI LIỆU THAM KHẢO

[1]. MachineLearningCoBan, (2017), Tiep Vu Huu, <https://machinelearningcoban.com/2017/04/09/smv/>

[2]. Hội Thính học Việt Nam: Vì con người, cho con người,

<https://vietnamhoinhap.vn/vi/hoi-thinh-hoc-viet-nam-vi-con-nguoi-cho-con-nguoi-46616.htm>.

[3]. MediaPipe Hands: On-device Real-time Hand Tracking.

[4]. iHand: Hand Recognition-Based Text Input Method for Wearable Devices.

[5]. Deafness and hearing loss,

<https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>.

[6]. Nguyen Trong-Nguyen, Duc-Hoang Vo, Huu-Hung Huynh, and Jean Meunier. "Geometry-based static hand gesture recognition using support vector machine." In Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on, pp. 769-774. IEEE, 2014.

[7]. Static Hand Gesture Recognition Using Novel Convolutional Neural Network and Support Vector Machine, July 2023.