

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



ĐỒ ÁN TỐT NGHIỆP

THIẾT KẾ HỆ THỐNG CHỨNG THỰC NGƯỜI BẰNG LÒNG BÀN TAY DỰA TRÊN HỌC SÂU

VŨ ĐỨC MẠNH

manh.vd174048@sis.hust.edu.vn

Ngành KT Điều khiển & Tự động hóa

Giảng viên hướng dẫn: TS. Trần Thị Thảo

Chữ ký của GVHD

Khoa: Tự động hóa

Trường: Điện – Điện tử

HÀ NỘI, 8/2022

**NHIỆM VỤ
ĐỒ ÁN TỐT NGHIỆP**

Họ và tên sinh viên: Vũ Đức Mạnh

Khóa: K62

Trường: Điện- Điện tử

Ngành: KT ĐK & TĐH

1. *Tên đề tài: Thiết kế hệ thống chứng thực người bằng lòng bàn tay dựa trên học sâu*
2. *Nội dung đề tài:*
 - *Xác định nội dung, yêu cầu bài toán và xây dựng giải pháp cho bài toán chứng thực người bằng lòng bàn tay.*
 - *Tìm hiểu và nghiên cứu các bài toán thị giác máy tính (Computer Vision) cần dùng: Trích xuất vùng quan tâm (ROI Extraction), Phân loại hình ảnh (Image Classification).*
 - *Chuẩn bị dữ liệu, thiết kế các thí nghiệm, đào tạo các mô hình để so sánh, đánh giá và lựa chọn các mô hình học sâu phù hợp với các phần của giải pháp để triển khai trong hệ thống ứng dụng.*
 - *Phân tích yêu cầu phần mềm, thiết kế hệ thống và lập trình giao diện người dùng trên máy tính cho hệ thống chứng thực.*
3. *Thời gian giao đề tài: Ngày 10/04/2022*
4. *Thời gian hoàn thành: Ngày 10/08/2022*

Ngày tháng năm 2022

CÁN BỘ HƯỚNG DẪN

Lời cảm ơn

Lời đầu tiên tôi xin được gửi lời cảm ơn chân thành tới gia đình, người thân và bạn bè đã luôn hỗ trợ tôi hết mình về cả vật chất và tinh thần. Để hoàn thành được đồ án trong thời gian quy định và đầy đủ yêu cầu thì ngoài việc bản thân đã cố gắng hết sức mình, tôi xin gửi lời cảm ơn đến TS. Trần Thị Thảo đã tận tình hướng dẫn, định hướng và giải đáp các thắc mắc của tôi trong quá trình thực hiện đồ án. Tôi cũng xin cảm ơn đến các thầy cô trong trường Điện-Điện tử nói riêng và các thầy cô trong trường Đại học Bách khoa Hà Nội nói chung đã truyền đạt cho tôi rất nhiều kiến thức và kinh nghiệm trong suốt 5 năm học tập trên ghế nhà trường.

Tôi xin chân thành cảm ơn!

Tóm tắt nội dung đồ án

Sinh trắc học là một là một công nghệ xác định một người bằng các đặc điểm sinh lý của người đó, chẳng hạn như mống mắt, vân tay, lòng bàn tay hoặc khuôn mặt hoặc sử dụng một số khía cạnh trong hành vi của người đó, chẳng hạn như giọng nói, chữ ký và cử chỉ... So với thông tin nhận dạng cá nhân dựa trên mã thông báo hoặc dựa trên kiến thức truyền thống có hạn chế là dễ bị đánh cắp hoặc bị lãng quên, sinh trắc học được coi là một phương tiện đáng tin cậy và hiệu quả để tự động nhận dạng danh tính của một người. Nhận dạng lòng bàn tay là một công nghệ sinh trắc học tương đối mới nhưng hiệu quả do không cần tới một hình ảnh thực sự sắc nét. Vì vậy một hệ thống chứng thực người được xây dựng dựa trên việc phân loại lòng bàn tay thay vì các đặc điểm sinh trắc học cũ có thể đem lại hiệu quả. Hệ thống được thiết kế chạy trên máy tính sử dụng các công nghệ của thị giác máy tính và học sâu giúp phân loại lòng bàn tay qua đó chứng thực xem một người có ở trong cơ sở dữ liệu hay không. Kết quả công việc của đồ án tốt nghiệp này đã đưa ra được giải pháp hoàn chỉnh cho bài toán chứng thực người đồng thời xây dựng được một hệ thống giao diện chứng thực bao gồm giao diện chứng thực trên máy tính giúp cho việc chứng cũng như kiểm tra thông tin người dùng dễ dàng hơn.

Sinh viên thực hiện

Ký và ghi rõ họ tên

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	2
1.1 Đặt vấn đề	2
1.2 Mục tiêu và phạm vi đề tài	2
1.3 Bố cục của đề tài	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	3
2.1 Cơ sở lý thuyết về nhận dạng lòng bàn tay	3
2.2 Cơ sở lý thuyết về máy biến đổi không gian.....	3
2.3 Cơ sở lý thuyết về mạng nơ-ron.....	5
2.3.1 Học sâu số liệu sâu.....	5
2.3.2 Mạng nơ-ron tích chập	6
2.3.3 Học chuyển đổi	9
2.3.4 Trình tối ưu hóa mô hình mạng nơ-ron.....	13
2.3.5 Các thước đo đánh giá mô hình	15
2.4 Cơ sở lý thuyết về cơ sở dữ liệu MongoDB	16
CHƯƠNG 3. PHƯƠNG PHÁP THỰC HIỆN.....	18
3.1 Phương pháp nhận diện lòng bàn tay	18
3.1.1 Giải pháp tổng thể	18
3.1.2 Quá trình phân loại lòng bàn tay	18
3.1.3 Trích xuất vùng quan tâm	19
3.1.4 Trích xuất vec-tơ đặc trưng và nhận diện lòng bàn tay	24
3.1.5 Nhận diện lòng bàn tay	24
3.2 Thiết kế giao diện người dùng	24
3.2.1 Các hoạt động chính trong hệ thống	25
3.2.2 Phân tích các yêu cầu chức năng	28
3.2.3 Các yêu cầu phi chức năng	32
3.3 Thiết kế cơ sở dữ liệu.....	33
CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ.....	35
4.1 Nhận diện lòng bàn tay	35
4.1.1 Trích xuất vùng quan tâm	35
4.1.2 Trích xuất vec-tơ đặc trưng.....	38
4.1.3 Nhận diện lòng bàn tay	38

4.2	Giao diện người dùng.....	40
4.2.1	Giao diện đăng nhập	40
4.2.2	Giao diện chính của hệ thống.....	40
4.2.3	Giao diện thêm người dùng.....	41
4.2.4	Giao diện xóa người dùng.....	41
4.2.5	Giao diện chứng thực người dùng.....	42
4.2.6	Giao diện kiểm tra thông tin người dùng	42
4.2.7	Giao diện kiểm tra thông tin chứng thực người dùng	43
4.3	Kết quả cài đặt hệ thống và hình ảnh minh họa các chức năng sử dụng của hệ thống	43
4.3.1	Chức năng đăng nhập:.....	44
4.3.2	Giao diện chính của hệ thống.....	45
4.3.3	Giao diện thêm người dùng.....	45
4.3.4	Giao diện xóa người dùng.....	46
4.3.5	Giao diện kiểm tra thông tin người dùng	48
4.3.6	Giao diện kiểm tra thông tin chứng thực người dùng	48
4.3.7	Giao diện chứng thực người dùng.....	48
	CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	50
5.1	Kết luận	50
5.2	Hướng phát triển	50

DANH MỤC HÌNH VẼ

Hình 2.1: Cấu trúc máy biến đổi không gian	3
Hình 2.2: Hai ví dụ về việc áp dụng lưới lấy mẫu được tham số hóa cho hình ảnh U tạo ra đầu ra V	4
Hình 2.3: Tích tích chập.....	7
Hình 2.4: Kỹ thuật zero-padding.....	8
Hình 2.5: Hai loại pooling thường được sử dụng	8
Hình 2.6: Cấu trúc mạng VGG-16	9
Hình 2.7: Tích chập chiều sâu.....	10
Hình 2.8: Tích chập điểm.....	11
Hình 2.9: Cấu trúc Residual	11
Hình 2.10: Mobilenet_V2 building block	12
Hình 2.11: Mô-đun SE-ResNet.....	12
Hình 2.12: Mobilenet_V3 block	13
Hình 2.13: Thuật toán cho hàm ProxyNCA Loss	15
Hình 2.14: Đường Precision-Recall	16
Hình 2.15: Giao diện MongoDB	17
Hình 3.1: Sơ đồ quá trình phân loại lòng bàn tay	18
Hình 3.2: Ảnh lòng bàn tay và biểu đồ của nó.....	19
Hình 3.3: Trọng tâm của lòng bàn tay	20
Hình 3.4: Đường ranh giới lòng bàn tay từ điểm ban đầu	21
Hình 3.5: Các điểm thung lũng trên lòng bàn tay	22
Hình 3.6: Các cột mốc trên lòng bàn tay.....	23
Hình 3.7: Cấu trúc của mạng ROILA-Net	23
Hình 3.8 : Biểu đồ hoạt động đăng nhập.....	25
Hình 3.9: Biểu đồ hoạt động thêm người dùng.....	26
Hình 3.10: Biểu đồ hoạt động xóa người dùng.....	26
Hình 3.11: Biểu đồ hoạt động kiểm tra thông tin người dùng	27
Hình 3.12: Biểu đồ hoạt động kiểm tra thông tin chứng thực	27
Hình 3.13: Biểu đồ hoạt động chứng thực người dùng.....	28
Hình 3.14: Biểu đồ Use case hệ thống	29
Hình 4.1: Ảnh đầu vào (kiểu ảnh xám) và ảnh nhị phân của nó.....	35
Hình 4.2: Tâm và ranh giới của lòng bàn tay.....	35
Hình 4.3: Các điểm thung lũng và vùng roi trên lòng bàn tay.....	35
Hình 4.4: Tâm và đường viền của lòng bàn tay khi trích xuất bị lỗi	36
Hình 4.5: Mất mát trong quá trình đào tạo mô hình ROILA-Net	37
Hình 4.6: Kết quả thực nghiệm trích xuất roi của mô hình ROI-LAnet với hình ảnh từ tập dữ liệu Tongji.....	37

Hình 4.7: Kết quả thực nghiệm trích xuất roi của mô hình ROI-LAnet với hình ảnh chụp từ điện thoại cá nhân.....	37
Hình 4.8: Nhận diện lòng bàn tay.	39
Hình 4.9: Nhận diện lòng bàn tay với độ sáng và độ tương phản thay đổi.....	39
Hình 4.10: Nhận diện lòng bàn tay khi thay đổi tư thế của bàn tay.....	39
Hình 4.11: Giao diện màn hình đăng nhập	40
Hình 4.12: Giao diện màn hình chính của hệ thống.....	41
Hình 4.13: Giao diện màn hình thêm người dùng	41
Hình 4.14: Giao diện xóa màn hình	42
Hình 4.15: Giao diện Chứng thực người dùng.....	42
Hình 4.16: Giao diện Kiểm tra thông tin người dùng	43
Hình 4.17: Giao diện Kiểm tra chứng thực người dùng	43
Hình 4.18: Cảnh báo sai thông tin đăng nhập	44
Hình 4.19: Cảnh báo thiếu thông tin đăng nhập	44
Hình 4.20: Giao diện chính của hệ thống.....	45
Hình 4.21: Giao diện thêm người dùng thành công	45
Hình 4.22: Giao diện thêm người với thông tin không chính xác	46
Hình 4.23: Giao diện thêm người dùng với thông tin bị thiếu.....	46
Hình 4.24: Giao diện xóa người dùng thành công	47
Hình 4.25: Giao diện xóa người dùng khi chưa điền tên người dùng.....	47
Hình 4.26: Giao diện kiểm tra thông tin người dùng	48
Hình 4.27: Giao diện kiểm tra thông tin chứng thực của người dùng	48
Hình 4.28: Giao diện chứng thực người dùng không thành công.....	49
Hình 4.29: Giao diện chứng thực người dùng thành công.....	49

DANH MỤC HÌNH VẼ

Bảng 3.1: Tác nhân sử dụng hệ thống.....	28
Bảng 3.2: Đặc tả use case Đăng nhập	29
Bảng 3.3: Đặc tả use case Đăng xuất	30
Bảng 3.4: Đặc tả use case thêm người dùng	30
Bảng 3.5: Đặc tả use case xóa người dùng	31
Bảng 3.6: Đặc tả use case Kiểm tra thông tin của người dùng	31
Bảng 3.7; Đặc tả use case Kiểm tra thông tin chứng thực của người dùng	32
Bảng 3.8: Bảng dữ liệu User	33
Bảng 3.9: Bảng dữ liệu Login	34
Bảng 3.10: Bảng dữ liệu Auth.....	34

DANH MỤC CÁC THUẬT NGỮ VÀ VIẾT TẮT

Thuật ngữ	Ý nghĩa
ROI	Vùng quan tâm
Deep learning	Học sâu
CNN	Mạng nơ-ron tích chập
Database	Cơ sở dữ liệu
Use case	Cả sử dụng
Computer Vision	Thị giác máy tính
Collection	Bản ghi
GD	Gradient Descent
SGD	Stochastic Gradient Descent
button	Nút bấm

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại ngày càng có nhiều đặc điểm sinh trắc học được đưa ra sử dụng làm đặc điểm phân loại con người thì lòng bàn tay có nhiều ưu điểm như là mang nhiều đặc điểm sinh trắc học, không yêu cầu chất lượng hình ảnh cao. Có nhiều phương pháp nhận dạng lòng bàn tay xuất sắc được đưa ra như Định hướng dòng (Robust line orientation code), Lập trình tuyến tính (Linear Programming) ... Các phương pháp này cho một kết quả rất tốt trên cơ sở dữ liệu cộng đồng PolyU. Tuy nhiên các phương pháp truyền thống này lại cho hiệu suất phân loại chưa được cao đối với những hình ảnh không có sự tách biệt giữa lòng bàn tay và nền. Gần đây với sự phát triển của các mô hình học sâu thì việc sử dụng các mô hình học sâu vào việc phân loại lòng bàn tay hứa hẹn sẽ cho một kết quả tốt.

1.2 Mục tiêu và phạm vi đề tài

Hệ thống chứng thực người dựa trên lòng bàn tay được đề xuất nhằm mục đích chứng thực người trong cơ sở dữ liệu hay không. Hệ thống bao gồm giao diện người dùng giúp cho việc chứng thực được diễn ra dễ dàng hơn, một cơ sở dữ liệu để lưu trữ dữ liệu người dùng.

1.3 Bố cục của đề tài

Phần còn lại của đề án tốt nghiệp được tổ chức như sau.

Chương 2 trình bày về cơ sở lý thuyết. Nội dung của chương này là những lý thuyết cơ bản sử dụng trong phạm vi đề án.

Chương 3 trình bày về phương pháp thực hiện. Nội dung của chương 3 trình bày về các phương pháp nhận dạng lòng bàn tay, phương pháp thiết kế giao diện người dùng và cơ sở dữ liệu.

Chương 4 trình bày về các kết quả đã đạt được. Nội dung chính của chương này trình bày về các kết quả đã thu được trong quá trình đào tạo các mô hình thông minh cũng như thiết kế và cài đặt hệ thống cùng với đánh giá kết quả.

Chương 5 trình bày về những gì đề án đã làm được, những hạn chế trong đề án và đưa ra định hướng cho đề án.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

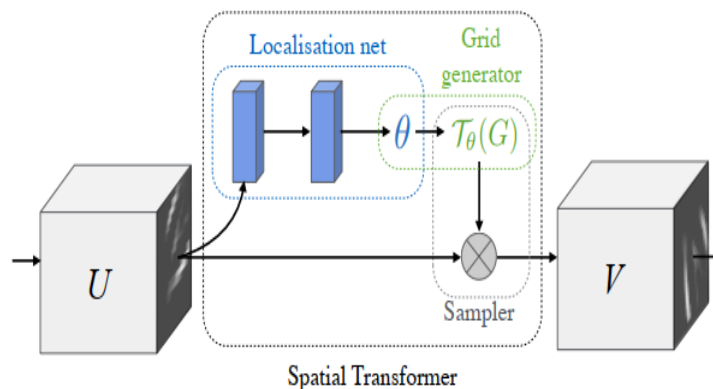
2.1 Cơ sở lý thuyết về nhận dạng lòng bàn tay

Sinh trắc học là một trong những chương trình dựa trên đặc điểm thể chất hoặc hành vi của con người và có thể được sử dụng để xác định hoặc xác minh con người. Có nhiều loại hệ thống nhận dạng sinh trắc học dựa trên dấu vân tay, mẫu móng mắt, khuôn mặt, chữ ký và lòng bàn tay. Trong số này, công nghệ sinh trắc học dựa trên lòng bàn tay đã và đang thu hút nhiều sự chú ý do có nhiều ưu điểm khác nhau như dễ xử lý với hình ảnh có độ phân giải thấp, phần cứng chi phí thấp và độ chính xác nhận dạng cao. Lòng bàn tay đề cập đến mô hình da trên bề mặt của lòng bàn tay và duy nhất cho mọi cá nhân. Việc nhận dạng lòng bàn tay thường sẽ thông qua các bước cơ bản sau:

- Trích xuất vùng quan tâm: đây là bước khởi đầu về nhận dạng lòng bàn tay giúp ta có thể thu được những hình ảnh được quan tâm (trên lòng bàn tay là các đường chỉ tay) và loại bỏ những khu vực không cần quan tâm. Việc trích xuất vùng quan tâm giúp cho việc nhận dạng lòng bàn tay sẽ chính xác và nhanh hơn.
- Trích xuất vec-tơ đặc trưng: giúp trích xuất ra những đặc trưng trên lòng bàn tay. Những đặc trưng này là đặc điểm nhận dạng của lòng bàn tay.
- Nhận diện lòng bàn tay: những vec-tơ đặc trưng được đem ra so sánh thông qua những thước đo độ tương tự như Cosine distance, Hamming distance, ...

2.2 Cơ sở lý thuyết về máy biến đổi không gian

Máy biến đổi không gian là một mô-đun có thể phân biệt được áp dụng một sự chuyển đổi không gian cho một bản đồ đối tượng địa lý trong một lần chuyển tiếp duy nhất, trong đó phép biến đổi được điều hoà trên một đầu vào cụ thể và tạo ra một bản đồ đặc điểm đầu ra. Cơ chế của máy biến áp không gian được chia làm ba phần được thể hiện trong hình dưới đây:



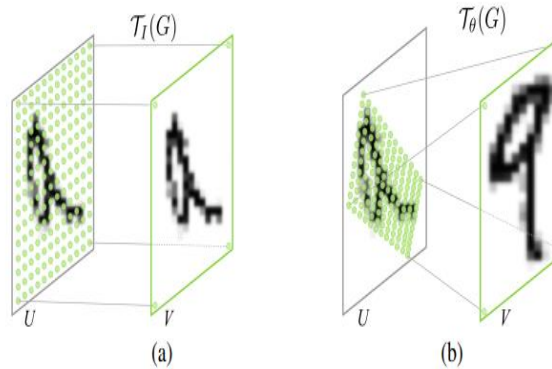
Hình 2.1: Cấu trúc máy biến đổi không gian

Để tính toán, đầu tiên một mạng cục bộ sẽ lấy bản đồ tính năng đầu vào và thông qua một số lớp ẩn sẽ xuất ra các tham số của phép biến đổi không gian sẽ được áp dụng cho bản đồ đối tượng - điều này đưa ra một phép biến đổi có điều

kiện đối với đầu vào. Sau đó, các tham số biến đổi dự đoán được sử dụng để tạo lưới lấy mẫu, là một tập hợp các điểm mà bản đồ đầu vào sẽ được lấy mẫu để tạo ra đầu ra được biến đổi. Điều này được thực hiện bởi máy tạo lưới. Cuối cùng, bản đồ đặc trưng và lưới lấy mẫu được lấy làm đầu vào cho bộ lấy mẫu, tạo ra bản đồ đầu ra được lấy mẫu từ đầu vào tại các điểm lưới.

Mạng cục bộ (Localisation Network) lấy bản đồ tính năng đầu vào $U \in \mathbb{R}^{H \times W \times C}$ với chiều rộng W , chiều cao H và C kênh và đầu ra θ , các tham số của phép biến đổi T_θ sẽ được áp dụng cho bản đồ đặc trưng: $\theta = f_{loc}(U)$. Kích thước của θ có thể thay đổi tùy thuộc vào kiểu biến đổi được tham số hóa. Hàm mạng cục bộ $f_{loc}()$ có thể ở bất kỳ dạng nào, chẳng hạn như mạng được kết nối đầy đủ (fully connected network) hoặc một mạng phức hợp (convolutional network), nhưng phải bao gồm một lớp hồi quy cuối cùng để tạo ra sự chuyển đổi tham số θ .

Lưới lấy mẫu tham số là thành phần để thực hiện sự cong vênh của bản đồ tính năng đầu vào, điểm ảnh đầu ra được tính toán bằng cách áp dụng một mẫu hạt nhân căn giữa tại một vị trí cụ thể trong bản đồ tính năng đầu vào. Theo điểm ảnh, thì việc lấy mẫu tham số đề cập đến một phần tử của bản đồ đối tượng địa lý chung, không nhất thiết phải là hình ảnh. Trong nói chung, các pixel đầu ra được xác định nằm trên một lưới thông thường $G = \{G_i\}$ của pixel $G_i = (x_i^s; y_i^s)$, tạo thành một bản đồ tính năng đầu ra $V \in \mathbb{R}^{H' \times W' \times C}$, trong đó H' và W' là chiều cao và chiều rộng của lưới, và C là số kênh, giống nhau ở đầu vào và đầu ra.



Hình 2.2: Hai ví dụ về việc áp dụng lưới lấy mẫu được tham số hóa cho hình ảnh U tạo ra đầu ra V

Trong đó (a) là lưới lấy mẫu là lưới thông thường $G = T_I(G)$, trong đó I là các tham số biến đổi nhận dạng. (b) là lưới lấy mẫu là kết quả của việc làm cong lưới thông thường với phép biến đổi affine $T_\theta(G)$.

Lấy mẫu hình ảnh có thể phân biệt (Differentiable Image Sampling): Để thực hiện phép biến đổi không gian của bản đồ đặc trưng đầu vào, người lấy mẫu phải lấy tập hợp các điểm lấy mẫu $T_\theta(G)$, cùng với bản đồ đặc trưng đầu vào U và tạo ra bản đồ đặc trưng đầu ra được lấy mẫu V . Mỗi tọa độ $(x_i^s; y_i^s)$ trong $T_\theta(G)$ xác định vị trí không gian trong đầu vào nơi hạt nhân lấy mẫu được áp dụng để nhận giá trị tại một pixel cụ thể trong đầu ra V . Điều này có thể triển khai như sau:

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - m; \Phi_y) \quad \forall i \in [1 \dots H'W'] \quad \forall c \in [1 \dots C]$$

Trong đó Φx và Φy là các tham số của nhân lấy mẫu chung $k()$ xác định hình ảnh nội suy (ví dụ: song tuyến), U_{nm}^c là giá trị tại vị trí $(n; m)$ trong kênh C của đầu vào và V_c^i là giá trị đầu ra cho pixel thứ i tại vị trí $(x_i^s; y_i^s)$ trong kênh C . Việc lấy mẫu được thực hiện giống hệt nhau cho mỗi kênh của đầu vào, vì vậy mọi kênh được chuyển đổi theo một cách giống hệt nhau (điều này duy trì sự nhất quán về không gian giữa các kênh). Về lý thuyết, bất kỳ hạt nhân lấy mẫu nào cũng có thể được sử dụng, miễn là có thể xác định các gradient liên quan đến $(x_i^s; y_i^s)$. Điều này cung cấp một cơ chế lấy mẫu có thể phân biệt, cho phép các gradient tổn thất không chảy ngược trở lại chỉ với bản đồ đối tượng đầu vào, mà còn tới tọa độ lưới lấy mẫu. Do không liên tục trong quá trình lấy mẫu, phải sử dụng các độ dốc phụ. Cơ chế lấy mẫu này có thể được triển khai rất hiệu quả trên GPU, bằng cách bỏ qua tổng tất cả các vị trí đầu vào và thay vào đó chỉ nhìn vào vùng hỗ trợ hạt nhân cho mỗi điểm ảnh đầu ra.

2.3 Cơ sở lý thuyết về mạng nơ-ron

2.3.1 Học sâu số liệu sâu

a) Học sâu:

Học sâu là một tập con của học máy, về cơ bản là một mạng nơ-ron có ba lớp trở lên. Những mạng nơ-ron cố gắng mô phỏng hành vi của não người cho phép nó “học” từ một lượng lớn dữ liệu. Mặc dù mạng nơ-ron với một lớp duy nhất vẫn có thể đưa ra các dự đoán gần đúng, nhưng các lớp ẩn bổ sung có thể giúp tối ưu hóa và tinh chỉnh để có độ chính xác. Học sâu thúc đẩy nhiều ứng dụng và dịch vụ trí tuệ nhân tạo (AI) nhằm cải thiện tự động hóa, thực hiện các tác vụ phân tích và vật lý mà không cần sự can thiệp của con người. Công nghệ học sâu nằm sau các sản phẩm và dịch vụ hàng ngày (chẳng hạn như trợ lý kỹ thuật số, điều khiển từ xa hỗ trợ giọng nói và phát hiện gian lận thẻ tín dụng) cũng như các công nghệ mới nổi (chẳng hạn như ô tô tự lái).

- Cách thức hoạt động của một mạng sâu: Mạng nơ-ron học sâu, hoặc mạng nơ-ron nhân tạo, cố gắng bắt chước não người thông qua sự kết hợp của dữ liệu đầu vào, trọng số, bias. Các yếu tố này làm việc cùng nhau để nhận biết, phân loại và mô tả chính xác các đối tượng trong dữ liệu. Mạng nơ-ron sâu bao gồm nhiều lớp nút được kết nối với nhau, mỗi lớp xây dựng dựa trên lớp trước đó để tinh chỉnh và tối ưu hóa dự đoán hoặc phân loại. Quá trình tính toán thông qua mạng này được gọi là quá trình truyền chuyển tiếp. Các lớp đầu vào và đầu ra của mạng nơ-ron sâu được gọi là các lớp khả kiến (visible layer). Lớp đầu vào là nơi mô hình học sâu nhập dữ liệu để xử lý và lớp đầu ra là nơi đưa ra dự đoán hoặc phân loại cuối cùng. Một quy trình khác được gọi là lan truyền ngược (backpropagation) sử dụng các thuật toán, như gradient descent, để tính toán sai số trong các dự đoán và sau đó điều chỉnh trọng số và độ chệch của hàm bằng cách di chuyển ngược lại qua các lớp trong nỗ lực đào tạo mô hình. Cùng với nhau, lan truyền chuyển tiếp và lan truyền ngược cho phép mạng nơ-ron đưa ra dự đoán và sửa bất kỳ lỗi nào cho phù hợp. Theo thời gian, thuật toán dần trở nên chính xác hơn. Ở trên mô tả loại mạng nơ-ron sâu đơn giản nhất bằng các thuật ngữ

đơn giản nhất. Tuy nhiên, các thuật toán học sâu vô cùng phức tạp và có nhiều loại mạng nơ-ron khác nhau để giải quyết các vấn đề hoặc tập dữ liệu cụ thể.

b) Học số liệu

Học số liệu là một nhánh của học máy. Mục tiêu của học số liệu là học một hàm biểu diễn ánh xạ các đối tượng vào một không gian nhúng. Khoảng cách trong không gian nhúng phải bảo toàn sự giống nhau của các đối tượng - các đối tượng tương tự đến gần và các đối tượng khác nhau ở xa. Học số liệu sẽ là một phương pháp rất tốt để có thể so sánh độ tương đồng giữa 2 hình ảnh. Tuy nhiên, khả năng của các phương pháp tiếp cận học số liệu truyền thống để xử lý dữ liệu trên dữ liệu thô đã hạn chế chúng. Do đó, cần phải có kỹ thuật tính năng trước khi phân loại hoặc phân nhóm các hoạt động, chẳng hạn như tiền xử lý và trích xuất tính năng. Vì vậy chúng ta cần đến học số liệu sâu để giải quyết vấn đề này.

c) Học số liệu sâu

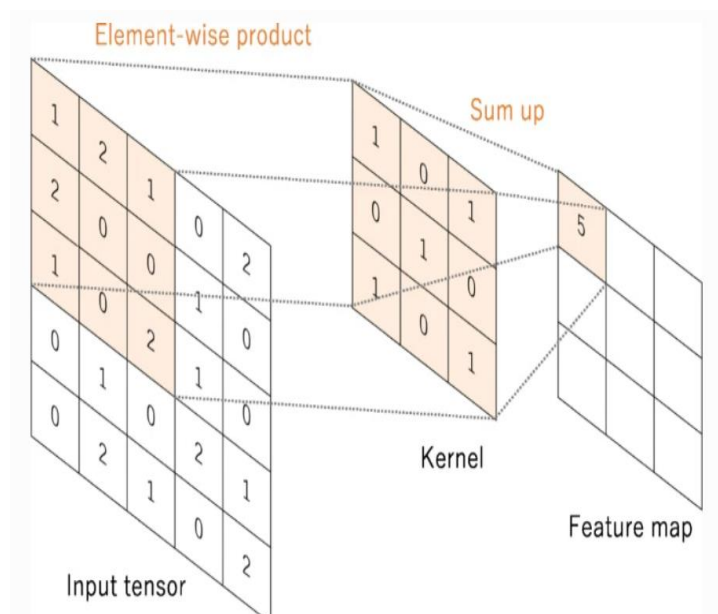
Phương pháp học theo số liệu sâu học cách đo lường sự tương đồng hoặc khoảng cách giữa các cặp điểm dữ liệu tùy ý, điều này có ý nghĩa rất quan trọng đối với một số ứng dụng máy tính và thị giác máy tính. Học số liệu sâu đã được áp dụng thành công cho tìm kiếm hình ảnh nhận dạng lại người và phương tiện, ... Các phương pháp học số liệu sâu học cách ánh xạ từ miền đầu vào, chẳng hạn như ảnh RGB đến không gian nhúng với chỉ số khoảng cách cố định đơn giản giúp nắm bắt rõ ràng sự giống nhau giữa các đầu vào. Để tìm hiểu một không gian nhúng như vậy, các lớp cơ sở thường được sử dụng làm đại diện cho sự giống nhau của hình ảnh đích. Các hàm mất mát của học số liệu sâu được thiết kế theo cách các hình ảnh tương tự được ánh xạ gần trong không gian nhúng đã học và các hình ảnh không giống nhau ở xa. Học số liệu sâu nhằm mục đích giải quyết một vấn đề chuyển giao, tức là đào tạo trên một tập hợp dữ liệu và đánh giá trên một tập dữ liệu thử nghiệm hoàn toàn khác. Học số liệu sâu là một phương pháp rất tốt cho việc nhận dạng hoặc phân loại.

2.3.2 Mạng nơ-ron tích chập

CNN là một loại mô hình học sâu để xử lý dữ liệu có dạng lưới, chẳng hạn như hình ảnh, được lấy cảm hứng từ tổ chức vỏ não thị giác của động vật và được thiết kế để học tự động và thích ứng các cấu trúc phân cấp không gian của các đối tượng địa lý, từ mức thấp - đến các mẫu cấp cao. CNN là một cấu trúc toán học thường bao gồm ba loại lớp (hoặc khối xây dựng): lớp chập, lớp gộp và lớp được kết nối đầy đủ. Hai lớp đầu tiên, lớp chập và lớp gộp, thực hiện trích xuất các đặc trưng cấp thấp đối tượng, trong khi lớp thứ ba, một lớp được kết nối đầy đủ, ánh xạ các đối tượng được trích xuất vào đầu ra cuối cùng, chẳng hạn như phân loại.

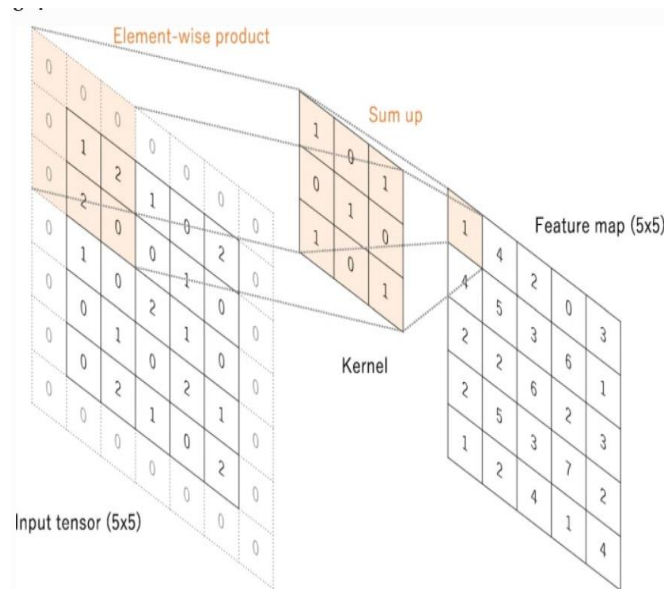
- Lớp tích chập (Convolution layer): Lớp tích chập là một thành phần cơ bản của kiến trúc CNN thực hiện việc trích xuất đối tượng địa lý, thường bao gồm sự kết hợp của các phép toán tuyến tính và phi tuyến tính, tức là hoạt động tích chập và hàm kích hoạt.
- Convolution: là một loại phép toán tuyến tính chuyên biệt được sử dụng để trích xuất đặc trưng, trong đó một mảng số nhỏ, được gọi là hạt nhân, được

áp dụng trên đầu vào, là một mảng số, được gọi là tensor. Tích số phần tử giữa mỗi phần tử của hạt nhân và tensor và được tổng hợp để có được giá trị đầu ra ở vị trí tương ứng của tensor đầu ra, được gọi là bản đồ đặc trưng. Quy trình này được lặp lại áp dụng nhiều hạt nhân để tạo thành một số lượng bản đồ đặc trưng tùy ý, biểu thị các đặc điểm khác nhau của các tensor đầu vào. Do đó, các nhân khác nhau có thể được coi là các bộ giải nén tính năng khác nhau. Hai siêu tham số chính xác định phép toán tích chập là kích thước và số lượng hạt nhân.



Hình 2.3: Tính tích chập

- **Padding:** Phép toán tích chập được mô tả ở trên không cho phép tâm của mỗi hạt nhân chồng lên phần tử ngoài cùng của tensor đầu vào và làm giảm chiều cao và chiều rộng của bản đồ đặc trưng đầu ra so với tensor đầu vào. Padding, thường là zero padding là một kỹ thuật để giải quyết vấn đề này, trong đó các hàng và cột của số không được thêm vào mỗi bên của tensor đầu vào, để vừa với tâm của hạt nhân trên phần tử ngoài cùng và giữ nguyên trong mặt phẳng thứ nguyên thông qua phép toán tích chập (Hình 4). Các kiến trúc CNN hiện đại thường sử dụng zero-padding để giữ lại kích thước trong mặt phẳng nhằm áp dụng nhiều lớp hơn. Nếu không có zero-padding, mỗi bản đồ đối tượng kế tiếp sẽ nhỏ hơn sau thao tác tích chập.

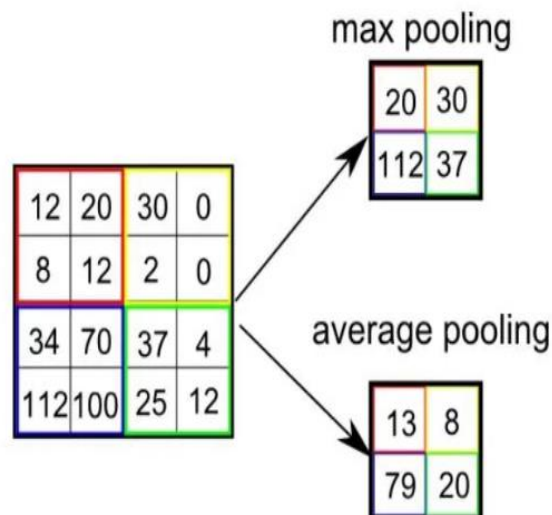


Hình 2.4: Kỹ thuật zero-padding

- Hàm kích hoạt phi tuyến: Các đầu ra của một phép toán tuyến tính như tích chập sau đó được chuyển qua một hàm kích hoạt phi tuyến. Có nhiều hàm phi tuyến được sử dụng như: Sigmoid, hyperbolic tangent, ReLU... trong đó thì hàm ReLU thường được sử dụng nhiều nhất. Công thức hàm kích hoạt ReLU:

$$f(x) = \max(0, x)$$

- Lớp Pooling: thường được sử dụng giữa các lớp tích chập để giảm kích thước dữ liệu đầu vào nhưng vẫn giữ được các đặc điểm quan trọng. Có 2 loại pooling thường được sử dụng là max pooling và average pooling.



Hình 2.5: Hai loại pooling thường được sử dụng

- Một số mạng nơ-ron tích chập tiêu biểu: cho đến nay đã có rất nhiều mạng nơ-ron tích chập ra đời và có kết quả thử nghiệm trên tập ImageNet (một tập dữ liệu với hơn 14 triệu ảnh và có hơn 1000 lớp). Trong đó các mạng điển hình là: AlexNet, VGG-Net, GoogleNet-Inception, ResNet, MobileNet, ...

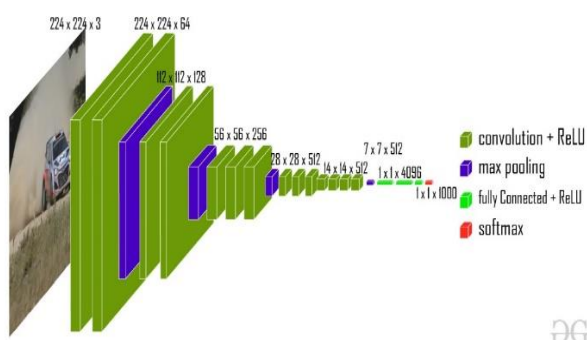
2.3.3 Học chuyển đổi

Học chuyển đổi là một phương pháp học máy trong đó chúng ta sử dụng lại một mô hình mạng nơ-ron được đào tạo trước làm điểm khởi đầu cho một mô hình trong một nhiệm vụ mới. So với phương pháp học truyền thống thì học chuyển đổi có một số ưu điểm:

- Các mô hình học sâu hay học máy truyền thống yêu cầu đào tạo từ đầu, rất tốn kém về mặt tính toán và yêu cầu một lượng lớn dữ liệu để đạt được hiệu suất cao. Mặt khác, học chuyển đổi hiệu quả về mặt tính toán và giúp đạt được kết quả tốt hơn bằng cách sử dụng một tập dữ liệu nhỏ.
- Các mô hình học chuyển đổi đạt được hiệu suất tối ưu nhanh hơn so với các mô hình học sâu hay học máy truyền thống. Đó là bởi vì các mô hình tận dụng kiến thức (tính năng, trọng số, ...) từ các mô hình được đào tạo trước đó đã hiểu các tính năng. Nó làm cho nó nhanh hơn so với đào tạo mạng nơ-ron từ đầu.

Trong bài toán nhận diện lòng bàn tay, với số lượng dữ liệu cộng đồng rất hạn chế thì việc sử dụng một mạng học chuyển đổi làm mạng cơ sở cho mạng trích xuất vec-tơ đặc trưng cũng như trích xuất vùng quan tâm sẽ giải quyết được vấn đề dữ liệu cũng như giúp việc đào tạo nhanh hơn. Dưới đây là chi tiết về một số mạng chuyển đổi được sử dụng trong đề án

- VGG-16:



Hình 2.6: Cấu trúc mạng VGG-16

Là một mô hình mạng nơ-ron tích chập được đề xuất bởi K. Simonyan và A. Zisserman từ Đại học Oxford. Mô hình này đạt độ chính xác trong bài kiểm tra top 5 là 92,7% trong ImageNet. VGG-16 là một trong những mô hình nổi tiếng được nộp cho ILSVRC-2014. Theo VGG-16, một mạng nơ-ron sâu hơn sẽ giúp ích cho cải thiện độ chính xác của mô hình tốt hơn. Về kiến trúc thì VGG-16 vẫn giữ các đặc điểm của AlexNet nhưng có những cải tiến:

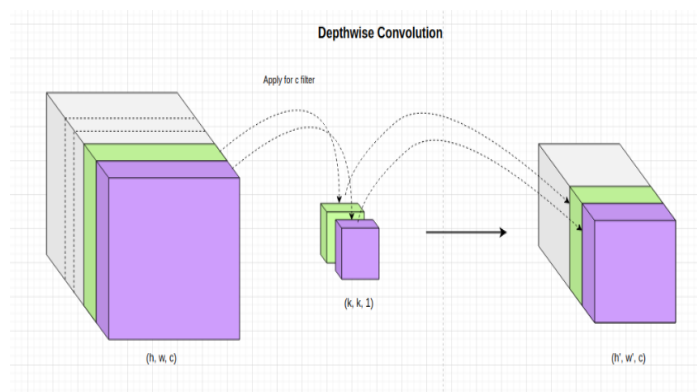
- Kiến trúc của VGG - 16 sâu hơn, bao gồm 13 lớp tích chập (thay vì 5 so với AlexNet) và 3 lớp kết nối đầy đủ.
- Lần đầu tiên trong VGG-16 xuất hiện khái niệm về khối tích chập (block). Đây là những kiến trúc gồm một tập hợp các lớp CNN được lặp lại giống nhau. Lần đầu tiên trong VGG-16 xuất hiện khái niệm về khối tích chập (block). Đây là những kiến trúc gồm một tập hợp các lớp CNN được lặp lại giống nhau. Kiến trúc khối đã khởi nguồn cho một dạng kiến trúc hình mẫu rất thường gặp ở các mạng CNN kể từ đó.

- VGG-16 cũng là kiến trúc đầu tiên thay đổi thứ tự của các block khi xếp nhiều lớp CNN cộng max pooling thay vì xen kẽ chỉ một lớp CNN cộng max pooling.
- VGG-16 chỉ sử dụng các bộ lọc kích thước nhỏ (thay vì nhiều kích thước bộ lọc như AlexNet). Kích thước bộ lọc nhỏ sẽ giúp giảm số lượng tham số cho mô hình và mang lại hiệu quả tính toán hơn.

- MobileNet:

Một mô hình được giới thiệu bởi Google cho các ứng dụng di động. Để có thể chạy được trên các thiết bị di động thì Mobilenet giới thiệu một khái niệm mới là: depthwise separable convolution (tích chập tách biệt theo chiều sâu). Tích chập tách biệt chiều sâu sẽ tìm cách loại bỏ sự phụ thuộc vào độ sâu khi tích chập mà vẫn tạo ra được một đầu ra có kích thước tương đương so với tích chập thông thường. Để có thể làm được điều này thì tích chập theo chiều sâu áp dụng tuần tự 2 bước là:

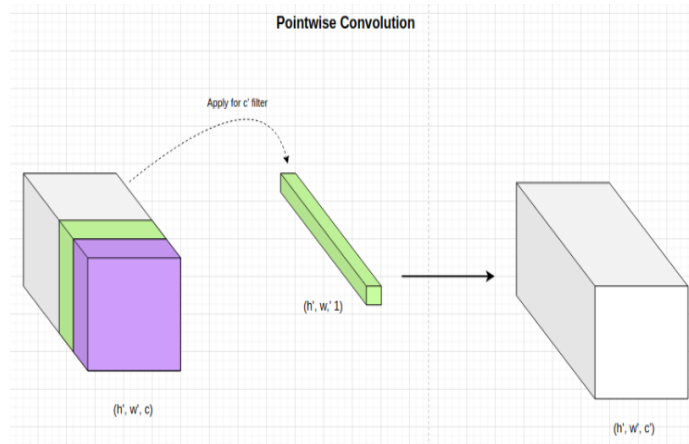
- Tích chập chiều sâu (Depthwise Convolution) [5]: Một đầu vào sẽ được chia thành các lát cắt ma trận theo độ sâu. Mỗi một kênh sẽ áp dụng một bộ lọc khác nhau và hoàn toàn không chia sẻ tham số. Điều này có nhiều tác dụng chính cho mô hình:
 - Nhận diện đặc trưng: Quá trình học và nhận diện đặc trưng sẽ được tách biệt theo từng bộ lọc. Nếu đặc trưng trên các channels là khác xa nhau thì sử dụng các bộ lọc riêng cho channel sẽ chuyên biệt hơn trong việc phát hiện các đặc trưng. Chẳng hạn như đầu vào là ba kênh RGB thì mỗi kênh áp dụng một bộ lọc khác nhau chuyên biệt.
 - Giảm thiểu khối lượng tính toán: Để tạo ra một điểm pixel trên đầu ra thì tích chập thông thường cần sử dụng $k \times k \times c$ phép tính trong khi tích chập chiều sâu tách biệt chỉ cần $k \times k$ phép tính.
 - Giảm thiểu số lượng tham số: Ở tích chập chiều sâu cần sử dụng $c \times k \times k$ tham số. Số lượng này ít hơn c' lần so với tích chập chiều sâu thông thường.



Hình 2.7: Tích chập chiều sâu

- Tích chập điểm (Pointwise Convolution) [5]: Có tác dụng thay đổi độ sâu của đầu ra bước trên từ c sang c' bằng cách áp dụng c' bộ lọc có kích thước $1 \times 1 \times c$. Điều giúp cho việc sử dụng tuần tự tích chập chiều

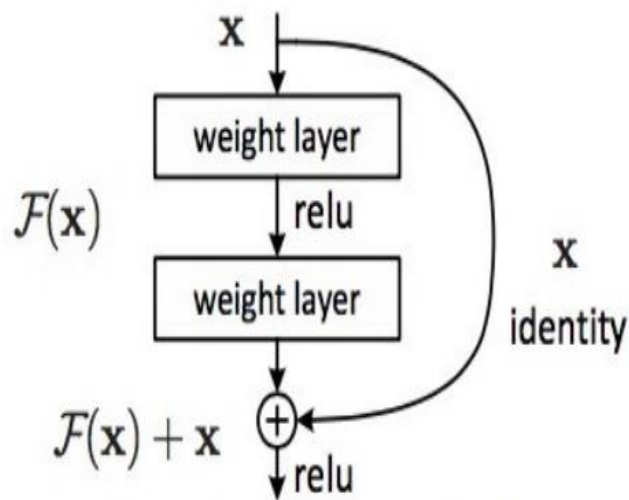
sâu và tích chập điểm thu được đầu ra tương tự so với việc sử dụng tích chập thông thường tuy nhiên phép tính cần thiết và tham số giảm đi đáng kể.



Hình 2.8: Tích chập điểm

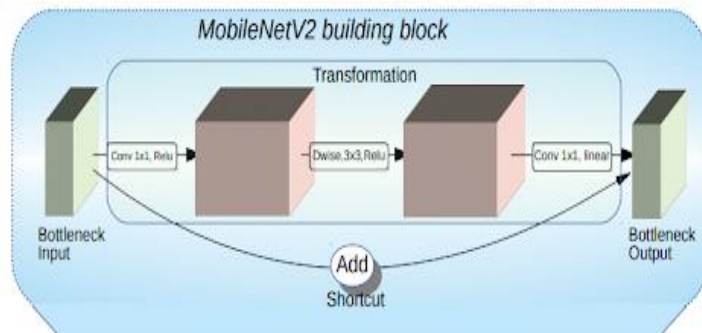
- MobileNet_V2:

Lần đầu giới thiệu cấu trúc Inverted Residual (kết nối tắt đảo ngược). MobileNet_V2 cũng sử dụng những kết nối tắt như ở mạng ResNet [7]. Các khối ở layer trước được cộng trực tiếp vào lớp liền sau. Nếu coi lớp liền trước là x , sau khi đi qua các xử lý tích chập hai chiều ta thu được kết quả $F(x)$ thì đầu ra cuối cùng là một residual block có giá trị $x + F(x)$.



Hình 2.9: Cấu trúc Residual

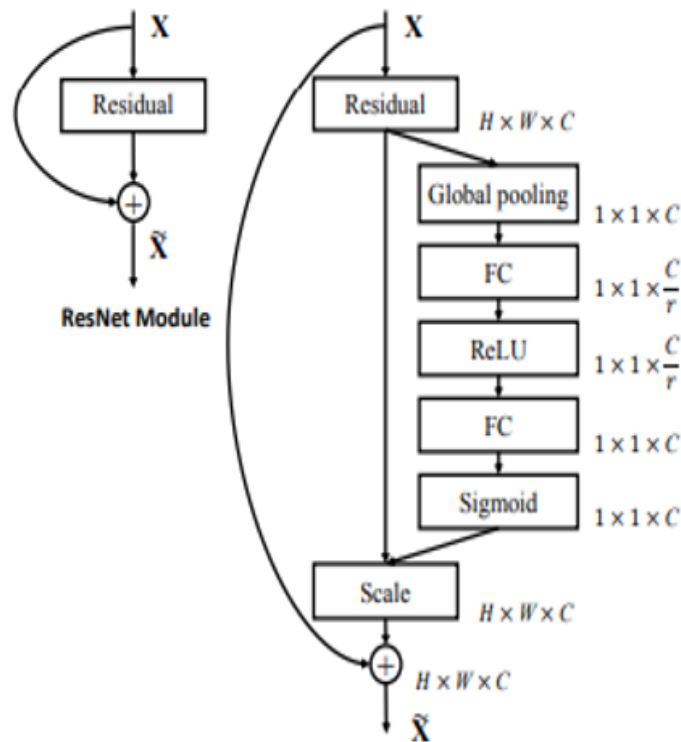
Tuy nhiên kết nối tắt ở MobileNet_V2 được điều chỉnh sao cho số kênh (hoặc chiều sâu) ở đầu vào và đầu ra của mỗi block residual [6] được thắt hẹp lại. Chính vì thế nó được gọi là các bottleneck layers (bottleneck [6] là một thuật ngữ thường được sử dụng trong học sâu để ám chỉ các kiến trúc thu hẹp kích thước theo một chiều nào đó).



Hình 2.10: Mobilenet_V2 building block

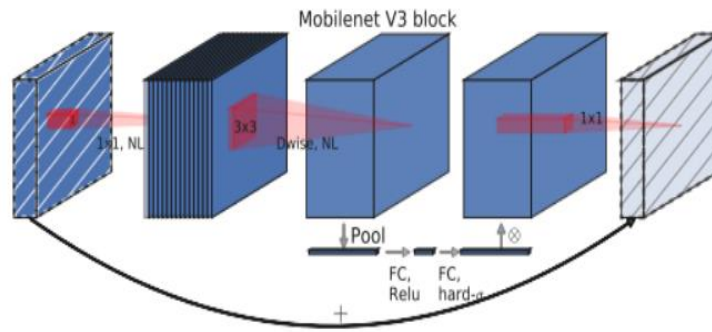
- MobileNet_V3:

Tiếp tục có sự nâng cấp từ Mobilenet_V2 bằng cách thêm vào khối Residual một mô-đun SE (Squeeze and Excitation) [9] để tạo thành một kiến trúc có độ chính xác cao hơn.



Hình 2.11: Mô-đun SE-ResNet

SE-ResNet áp dụng thêm một nhánh Global pooling có tác dụng ghi nhận bối cảnh của toàn bộ lớp trước đó. Kết quả sau cùng ở nhánh này ta thu được một véc-tơ global context được dùng để scale đầu vào X. Tương tự như vậy mô-đun SE được tích hợp vào kiến trúc của một inverted residual block trong mobilenet_V3 như sau:



Hình 2.12: Mobilenet_V3 block

Ở lớp thứ 3 có một nhánh SE có kích thước là 1x1 có tác dụng tổng hợp global context. Nhánh này lần lượt đi qua các biến đổi FC \rightarrow Relu \rightarrow FC \rightarrow hard sigmoid (FC là lớp kết nối đầy đủ). Cuối cùng được nhân trực tiếp vào nhánh đầu vào để scale đầu vào theo global context. Các kiến trúc còn lại hoàn toàn giữ nguyên như MobileNet_V2.

2.3.4 Trình tối ưu hóa mô hình mạng nơ-ron

Trình toán tối ưu là cơ sở để xây dựng mô hình mạng thần kinh với mục đích "học" được các đặc điểm (hay mẫu) của dữ liệu đầu vào, từ đó có thể tìm 1 cặp trọng số và thiên vị phù hợp để tối ưu hóa mô hình. Có rất nhiều trình tối ưu hóa được sử dụng trong quá trình đào tạo mô hình. Dưới đây là một số trình tối ưu hóa phổ biến:

- GD Optimizer (Gradient Descent): Trong sự tối ưu của bài toán, chúng ta thường tìm giá trị nhỏ nhất của 1 số hàm, mà giá trị nhỏ nhất khi đạo bằng 0. Nhưng không phải lúc nào đạo hàm đó cũng được, đối với nhiều biến chức năng thì đạo hàm rất phức tạp, thậm chí là bất khả thi. Nên thay vào đó người ta tìm điểm gần với điểm cực tiểu và xem đó là bài toán. Hướng tiếp cận ở đây là lựa chọn 1 ngẫu nhiên cứ sau mỗi vòng lặp (hay epoch) thì nó cứ tăng dần đến điểm cần tìm. Gradient Descent sử dụng công thức dưới đây để cập nhật trọng số sau mỗi vòng lặp:

$$x_i = x_{i-1} - \alpha \cdot \text{gradient}(x)$$

Trong đó gradient được tính theo công thức:

$$\nabla f(x, y) = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j$$

Gradient Descent là một thuật toán, dễ hiểu. Thuật toán đã được giải quyết vấn đề tối ưu hóa mạng nơ-ron mô hình bằng cách cập nhật các số quan trọng sau mỗi vòng lặp. Tuy nhiên vì đơn giản nên thuật toán Gradient Descent còn nhiều hạn chế như phụ thuộc vào hệ thống khởi tạo ban đầu và tốc độ học.

- SGD Optimizer (Stochastic Gradient Descent): là 1 biến thể của Gradient Descent. Thay vì sau mỗi kỷ nguyên, chúng ta sẽ cập nhật số quan trọng (Trọng lượng) 1 lần, trong mỗi kỷ nguyên có N điểm dữ liệu, chúng ta sẽ cập nhật số quan trọng N lần. Thuật toán giải quyết được đối với cơ sở dữ

liệu lớn mà GD không làm được. Thuật toán tối ưu này hiện nay vẫn hay được sử dụng. Thuật toán vẫn chưa giải quyết được 2 điểm lớn của GD (tỷ lệ học tập, dữ liệu khởi đầu). Vì vậy ta phải kết hợp SGD với 1 số khác thuật toán như: Momentum, AdaGrad

- Momentum: là một kỹ thuật rất phổ biến được sử dụng cùng với SGD. Momentum giới thiệu vec-tơ động lượng. Vec-tơ này được sử dụng để "lưu trữ" thay đổi trong các gradient trước đó. Nó giúp tăng tốc giảm độ dốc ngẫu nhiên trong hướng liên quan và làm tắt dần các dao động. Tại mỗi bước gradient được thêm vào vec-tơ động lượng. Sau đó, các thông số được cập nhật chỉ bằng cách trừ đi vec-tơ động lượng từ các giá trị tham số hiện tại. Vec-tơ động lượng được tính theo công thức:

$$m \leftarrow \beta m + \alpha \nabla SGD$$

- Adagrad Optimizer: Không giống như các thuật toán trước đó thì tốc độ học hầu như giống nhau trong quá trình training (learning rate là hằng số), Adagrad coi tốc độ học là 1 tham số. Công thức sử dụng cho Adagrad:

$$\theta_i = \theta_{i-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

Trong đó: η là hằng số; g_t là gradient tại thời điểm t ; ϵ là một số rất bé để tránh chia cho 0 và G là một ma trận chéo mà mỗi phần tử trên đường chéo là bình phương của đạo hàm vec-tơ tham số tại thời điểm t . Tổng bình phương biến thiên sẽ lớn dần theo thời gian cho đến khi nó làm tốc độ học cực kỳ nhỏ, làm việc training trở nên đóng băng.

- RMSprop Optimizer: RMSprop giải quyết vấn đề tỷ lệ học giảm dần của Adagrad bằng cách chia tỷ lệ học cho trung bình của bình phương gradient theo công thức:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{E[g^2]_{t-1} + \epsilon}}$$

- Adam Optimizer: là một thuật toán kết hợp kỹ thuật của RMSprop và momentum. Thuật toán sử dụng hai internal states momentum (m) và squared momentum (v) của độ dốc (gradient) cho các tham số. Sau mỗi lần huấn luyện, giá trị của m và v được cập nhật lại theo công thức:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Trong đó β được xem là một siêu tham số và theta được cập nhật theo công thức sau:

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t + \epsilon}}$$

Trong đó α là tốc độ học và ϵ là một số bé được thêm vào để tránh trường hợp chia cho 0.

2.3.5 Các thước đo đánh giá mô hình

a) Hàm mất mát

- MSE (Mean Squared Error) Loss: là một thước đo đánh giá mô hình thường được sử dụng với các mô hình hồi quy. Sai số bình phương trung bình của một mô hình đối với tập thử nghiệm là giá trị trung bình của các sai số dự đoán bình phương trên tất cả các trường hợp trong tập thử nghiệm được tính theo công thức:

$$\text{MSE}_{\text{loss}}(x, y) = (x - y)^2$$

- Cross Entropy Loss: là một số liệu được sử dụng để đo lường mức độ hoạt động của mô hình phân loại trong học máy. Tổn thất (hoặc sai số) được đo dưới dạng một số từ 0 đến 1, với 0 là một mô hình hoàn hảo. Mục tiêu nói chung là đưa mô hình về gần 0 nhất có thể. Mất entropy chéo thường được coi là có thể hoán đổi với nhau với tổn thất logistic (hoặc mất log, và đôi khi được gọi là mất entropy chéo nhị phân) nhưng điều này không phải lúc nào cũng chính xác. Mất mất entropy chéo đo lường sự khác biệt giữa phân phối xác suất được phát hiện của một mô hình phân loại học máy và phân phối được dự đoán. Tất cả các giá trị có thể có cho dự đoán được lưu trữ và mất mất entropy chéo nhị phân thì chỉ lưu trữ được 1 giá trị. Chính vì lý do này mà mất mất entropy chéo nhị phân (hoặc mất nhật ký) được sử dụng trong các tình huống chỉ có hai kết quả có thể xảy ra.
- ProxyNCA Loss: sử dụng proxy để tạo khoảng cách dựa trên công thức NCA. Việc tối ưu hóa được thực hiện bằng cách giảm dần độ dốc với thuật toán:

```

for  $i = 1 \dots T$  do
    Sample triplet  $(x, y, Z)$  from  $D$ 
    Formulate proxy triplet  $(x, p(y), p(Z))$ 
     $l = -\log \left( \frac{\exp(-d(x, p(y)))}{\sum_{p(z) \in p(Z)} \exp(-d(x, p(z)))} \right)$ 
     $\theta \leftarrow \theta - \lambda \partial_{\theta} l$ 
end for

```

Hình 2.13: Thuật toán cho hàm ProxyNCA Loss

Ở mỗi bước một mẫu gồm một bộ 3 của một điểm dữ liệu và hai proxy. Tuy nhiên, mỗi bộ ba được xác định trên các proxy giới hạn trên tất cả các bộ ba có điểm dữ liệu dương y' và z' âm có cùng proxy tương ứng với y và z . Điều này giúp tăng tốc độ hội tụ trong quá trình đào tạo.

b) Các thước đo đánh giá độ chính xác

- Precision và Recall: Trong bài toán phân loại nhị phân, ta gọi một lớp là positive và lớp còn lại là negative, TP là số điểm được dự đoán đúng là positive, TN là số điểm được dự đoán đúng là negative, FP là số điểm được dự đoán là positive nhưng thực tế lại là negative, FN là số điểm được dự

đoán là negative nhưng thực tế lại là positive. Ta có công thức tính các độ đo Precision và Recall:

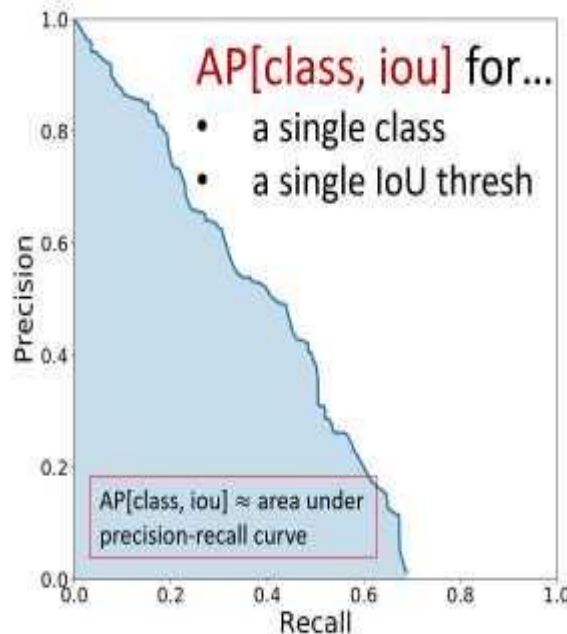
$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

- IoU (Intersection over Union): Gọi B_{pr} là nhãn được dự đoán bởi mô hình, B_{gt} là nhãn chính xác trong tập dữ liệu, IoU được tính bằng diện tích giao chia cho diện tích hợp của hai nhãn:

$$\text{IoU} = \frac{B_{pr} \cap B_{gt}}{B_{pr} \cup B_{gt}}$$

- AP: là diện tích phía dưới đường Precision-Recall với mỗi điểm tượng trưng cho một ngưỡng tin cậy:



Hình 2.14: Đường Precision-Recall

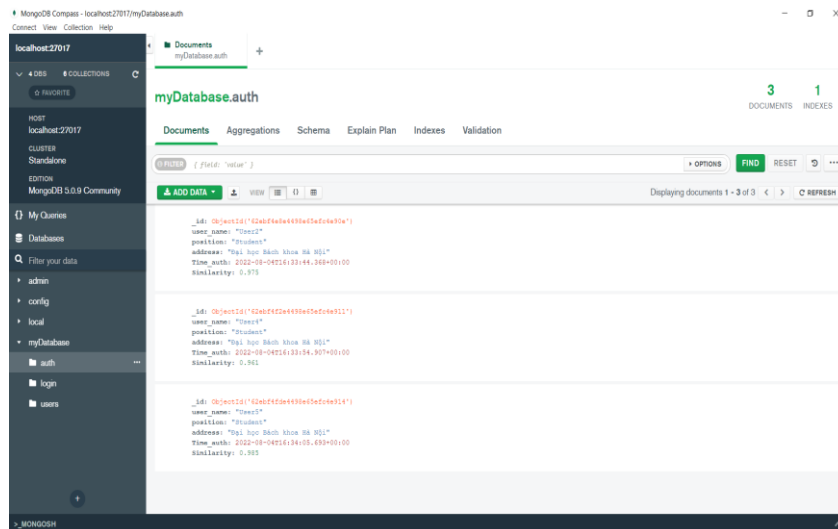
AP còn được tính đơn giản theo công thức sau (với n là số ngưỡng):

$$AP = \sum_n (R_n - R_{n-1}) P_n$$

mAP (Mean average precision): là trung bình cộng AP cho các lớp trong tập dữ liệu.

2.4 Cơ sở lý thuyết về cơ sở dữ liệu MongoDB

MongoDB là một cơ sở dữ liệu NoSQL hướng đối tượng, đơn giản, linh động và có thể mở rộng. Nó dựa trên mô hình lưu trữ NoSQL document. Các đối tượng dữ liệu được lưu trữ dưới dạng các tài liệu riêng biệt bên trong một collection – thay vì lưu trữ dữ liệu vào các cột và hàng của cơ sở dữ liệu quan hệ truyền thống. Ngôn ngữ MongoDB là triển khai một kho lưu trữ dữ liệu cung cấp hiệu suất cao, tính sẵn sàng cao và tự động mở rộng MongoDB sử dụng JSON hoặc BSON document để lưu trữ dữ liệu.



Hình 2.15: Giao diện MongoDB

- Ưu điểm:
 - Hiệu suất cao: Việc truy vấn cũng như thêm dữ liệu và cơ sở dữ liệu có tốc độ rất cao do MongoDB lưu dữ liệu dạng JSON, khi thêm nhiều đối tượng thì nó sẽ là thêm một mảng JSON gần như với trường hợp thêm 1 đối tượng. Một điều nữa là dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau như trong RDBMS, khi thêm, xóa hay chỉnh sửa nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các bảng liên quan như trong RDBMS. Dữ liệu trong MongoDB được đánh chỉ mục nên khi truy vấn nó sẽ tìm rất nhanh.
 - Dữ liệu linh hoạt: MongoDB là document database, dữ liệu lưu dưới dạng JSON, không bị bó buộc về số lượng trường, kiểu dữ liệu... do đó có thể thêm dữ liệu một cách tùy ý.
 - Khả năng mở rộng: Trong MongoDB có một khái niệm cluster là cụm các nút chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một nút vào cluster.
- Nhược điểm:
 - MongoDB không có các tính chất ràng buộc như trong RDBMS nên dễ làm sai sót dữ liệu.
 - Sử dụng nhiều bộ nhớ: do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại.
 - Bị giới hạn kích thước bản ghi: mỗi document không được có kích thước lớn hơn 16Mb và không mức độ các document con trong 1 document không được lớn hơn 100.

CHƯƠNG 3. PHƯƠNG PHÁP THỰC HIỆN

3.1 Phương pháp nhận diện lòng bàn tay

3.1.1 Giải pháp tổng thể

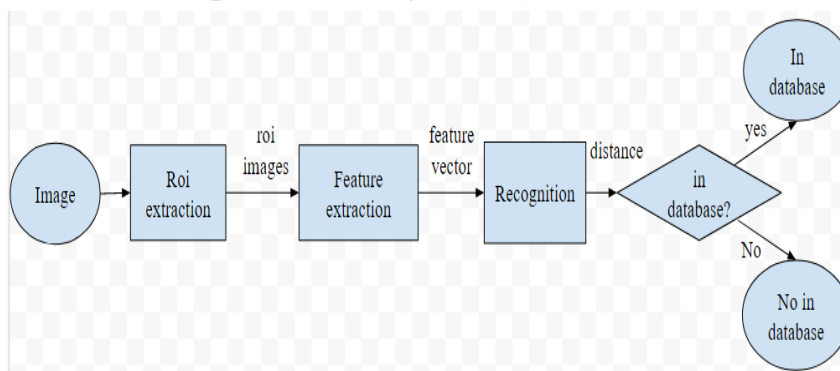
Phương pháp sử dụng các mô hình học sâu trong việc trích xuất và phân loại các đặc trưng của lòng bàn tay. Phương pháp này được phát biểu như một bài toán có đầu ra và đầu vào như sau:

- Đầu vào: Ảnh lòng bàn tay của một người.
- Đầu ra: Người đó có ở trong cơ sở dữ liệu hay không.

Để thực hiện công việc trích xuất và phân loại các lòng bàn tay, hệ thống sử dụng các công nghệ của thị giác máy tính sau đây:

- Trích xuất vùng quan tâm (ROI Extraction): nhằm xác định vùng quan tâm (ảnh roi) trên lòng bàn tay nơi lưu trữ các thông tin sinh trắc học.
- Trích xuất vec-tơ đặc trưng (Feature Extraction-Embedding Extraction): dựa trên các mô hình có trọng số được đào tạo sẵn để đào tạo lại trên tập dữ liệu lòng bàn tay đã thu thập được làm công cụ trích xuất vec-tơ đặc trưng.
- Nhận dạng (Recognition): nhằm mục đích đo lường sự giống nhau giữa các mẫu bằng cách tạo ra các mạng sâu và một thước đo khoảng cách như là Cosine hoặc Euclide.

3.1.2 Quá trình phân loại lòng bàn tay



Hình 3.1: Sơ đồ quá trình phân loại lòng bàn tay

- Image: ảnh đầu vào là ảnh của một lòng bàn tay.
- ROI Extraction: bộ trích xuất vùng quan tâm của ảnh lòng bàn tay, nhận đầu vào là một ảnh và đầu ra là một ảnh vùng quan tâm trên lòng bàn tay dưới thang độ xám.
- Feature Extraction: bộ trích xuất các vec-tơ đặc trưng của ảnh, nhận đầu vào là một ảnh roi và đầu ra là một vectơ đặc trưng đặc trưng của lòng bàn tay.
- Recognition: bộ phân loại, nhận đầu vào là một vec-tơ đặc trưng và đầu ra là khoảng cách của vec-tơ đó với các vec-tơ khác trong cơ sở dữ liệu.
- Khoảng cách giữa vec-tơ đầu vào và các vec-tơ khác trong cơ sở dữ liệu sẽ quyết định xem nó có ở trong cơ sở dữ liệu hay không.

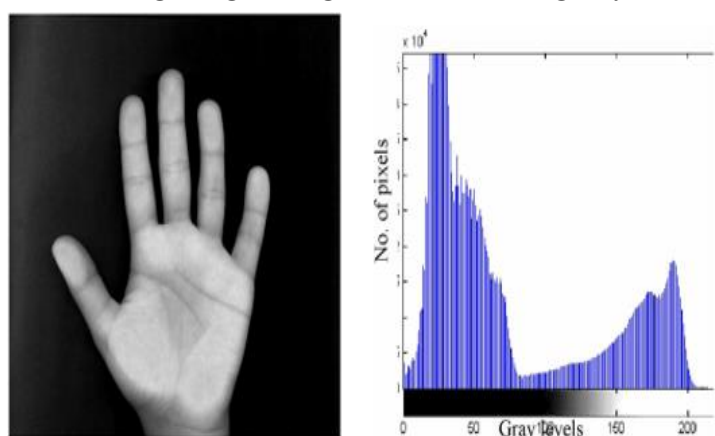
3.1.3 Trích xuất vùng quan tâm

Trích xuất vùng quan tâm (Extraction of ROI) thực hiện việc tìm ra vị trí được quan tâm trên bức ảnh. Việc trích xuất vùng quan tâm từ lòng bàn tay sẽ cải thiện đáng kể hệ thống nhận dạng lòng bàn tay. Trong bài toán nhận dạng lòng bàn tay thì việc trích xuất vùng quan tâm của lòng bàn tay là bước đầu tiên và quan trọng nhất. Hiện tại việc trích xuất lòng bàn tay được thực hiện theo hai phương pháp:

- Phương pháp truyền thống: phương pháp này dựa trên các kỹ thuật xử lý hình ảnh bao gồm ngưỡng (Thresholding) để mã hoá nhị phân, xác định tâm (centroid), khai thác ranh rới bằng cách sử dụng các phép toán.
- ROILA-Net: Sử dụng một mạng sâu được đào tạo và một mạng hồi quy để tìm ra các điểm cột mốc trên lòng bàn tay (Hand landmarks) qua đó trích xuất ra vùng quan tâm trên lòng bàn tay.

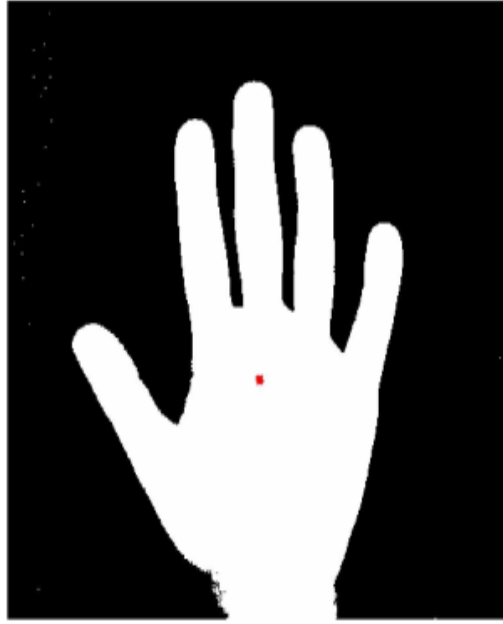
a) Phương pháp truyền thống

Lòng bàn tay được mã hoá nhị phân bằng cách sử dụng ngưỡng động là biểu đồ và tâm của nó được tính toán. Biểu đồ của một hình ảnh, $f(x, y)$ là xác suất tồn tại của các mức xám trong một bức ảnh. Biểu đồ mức xám của hình ảnh, $f(x, y)$ bao gồm các vật thể tối trong nền sáng hoặc ngược lại, được đưa ra theo cách mà cách điểm ảnh (pixel) của đối tượng và nền có mức xám được nhóm lại thành hai vùng ưu thế (dominant regions). Như vậy có thể trích xuất đối tượng từ nền bằng cách xác định ngưỡng 'T' ngăn cách hai vùng này.



Hình 3.2: Ảnh lòng bàn tay và biểu đồ của nó

Vùng bên trái của biểu đồ tương ứng với các điểm ảnh nền và vùng bên phải của biểu đồ tương ứng với các điểm ảnh của lòng bàn tay. Giá trị nhỏ nhất giữa các đỉnh của hai vùng được chọn là ngưỡng 'T'. Giá trị này thay đổi theo kích thước và sự chiếu sáng nên việc xác định ngưỡng này là sử dụng ngưỡng động. Ngưỡng động giúp ích trong việc giảm thiểu ảnh hưởng của nhiễu, bóng cũng như sự thay đổi của việc chiếu sáng.



Hình 3.3: Trọng tâm của lòng bàn tay

Xác định trọng tâm của lòng bàn tay: Trọng tâm của lòng bàn tay đã được mã hóa nhị phân được tính toán bằng cách sử dụng phương trình:

$$X_{\text{centroid}} = \frac{m_{01}}{m_{11}}, Y_{\text{centroid}} = \frac{m_{10}}{m_{11}}$$

Trong đó thì các mô-men được tính theo công thức:

$$m_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} i^p j^q$$

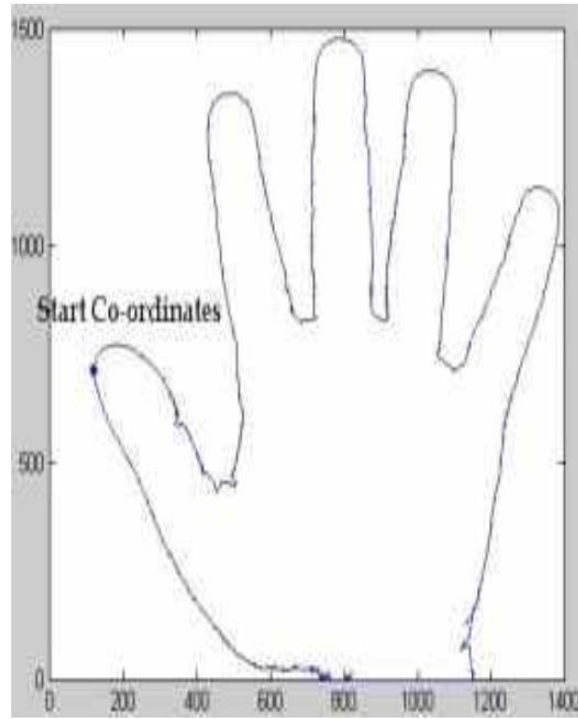
Trích xuất ranh giới: Ranh giới B (A) của lòng bàn tay A được cho dưới dạng phép trừ của lòng bàn tay nhị phân bị xói mòn (eroded) hình thái học (morphological) từ lòng bàn tay nhị phân ban đầu theo công thức:

$$B(A) = A \ominus (A - S)$$

Trong đó S là phần tử cấu trúc được sử dụng để xói mòn hình ảnh được xác định bởi công thức:

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Kỹ thuật xói mòn hình thái học giúp loại bỏ một số điểm ảnh lạc từ hình ảnh nhị phân. Điều này giúp việc xác định ranh giới tốt hơn. Tọa độ của các pixel ranh giới thu được bằng cách lần theo các ranh giới bên ngoài của các đối tượng bằng cách sử dụng kết nối tám điểm. Hình ảnh nhị phân chứa một số điểm ảnh lạc nhất định, một số điểm ảnh bị loại bỏ do xói mòn hình thái học. Các pixel lạc còn lại được loại bỏ bằng cách chọn đối tượng lớn nhất là ranh giới thực của lòng bàn tay. Các pixel biên của đối tượng lớn nhất được theo dõi từ pixel ngoài cùng bên trái. Hình dưới đây cho thấy ranh giới của lòng bàn tay được trích xuất với pixel ngoài cùng bên trái.



Hình 3.4: Đường ranh giới lòng bàn tay từ điểm ban đầu

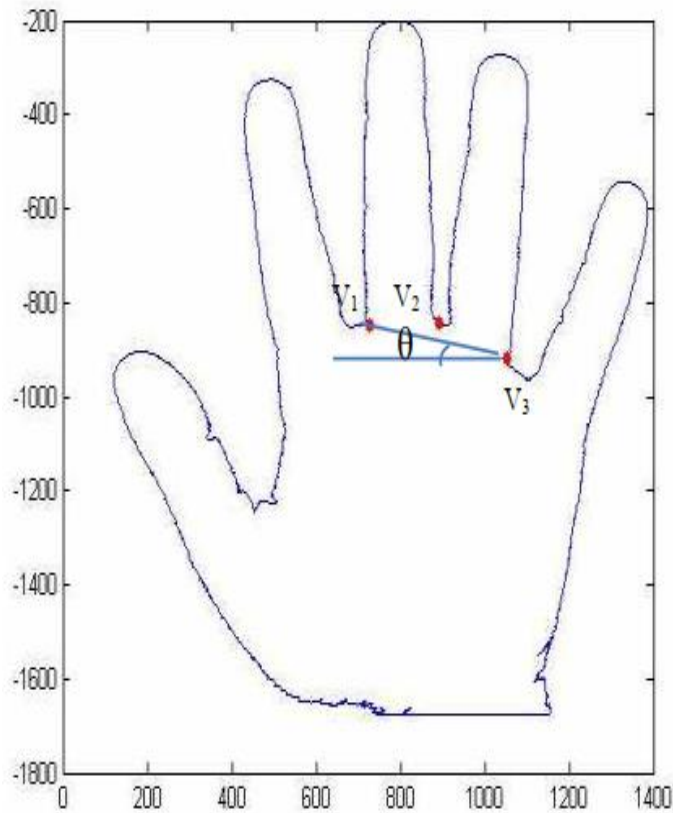
Khoảng cách (ED) giữa tâm và các pixel biên được tính bằng phương trình:

$$ED(x, y) = \sqrt{(x - x_{centroid})^2 + (y - y_{centroid})^2}$$

Các điểm thung lũng (khe giữa 2 ngón tay) sẽ gần tâm của bàn tay hơn vì vậy ED của chúng là giá trị cực tiểu. Để trích xuất những điểm này, trước tiên chúng ta phân biệt chức năng ED và lưu ý các điểm giao nhau không. Đồ thị vi phân của một hàm số sẽ đạt giá trị 0 tại cả điểm cực tiểu và cực đại của hàm số. Để phân biệt giữa hai điểm, chúng ta chọn các điểm giao nhau bằng 0, nơi đồ thị vi phân đi từ giá trị âm sang giá trị dương tại điểm uốn cong. Thực tế sẽ có thể có rất nhiều điểm giao nhau bằng 0 và đồ thị vi phân của chúng từ âm sang dương. Vấn đề này phát sinh do các thành phần tần số cao có trong chức năng ED. Các thành phần tần số cao này được tạo ra do đường viền của lòng bàn tay không đồng đều. Để loại bỏ các tần số cao này, ta có thể suy biến hàm ED thành các tần số cấu thành của nó bằng cách sử dụng biến đổi Fourier và loại bỏ các thành phần tần số cao và tái tạo lại như cũ chỉ sử dụng các tần số thấp. Biến đổi Fourier rời rạc của hàm Euclid điểm N được cho bởi:

$$F(k) = \sum_{x=0}^{N-1} f(x) * (e^{-2\pi j(\frac{kx}{N})})$$

Năng lượng của hàm ED tập trung chủ yếu ở dải tần số thấp hơn. Vì vậy, nếu chức năng ED được tái tạo chỉ xem xét các thành phần tần số thấp, nó sẽ giữ nguyên các đặc tính cơ bản của nó và do đó vị trí của các điểm thung lũng sẽ không thay đổi. Do đó, tất cả các thành phần tần số cao có năng lượng tích lũy nhỏ hơn một số năng lượng ngưỡng phần trăm (lấy 98% làm năng lượng ngưỡng) đều bị loại bỏ.



Hình 3.5: Các điểm thung lũng trên lòng bàn tay

Điểm thung lũng V_1 và V_2 được coi là điểm tham chiếu để trích xuất ROI. Đường nối hai điểm có thể ở góc θ so với phương ngang. Palm được xoay một góc θ để làm cho đường thẳng V_1V_3 nằm ngang. ROI có kích thước và hình dạng mong muốn sau đó có thể được trích xuất.

b) ROILA-Net

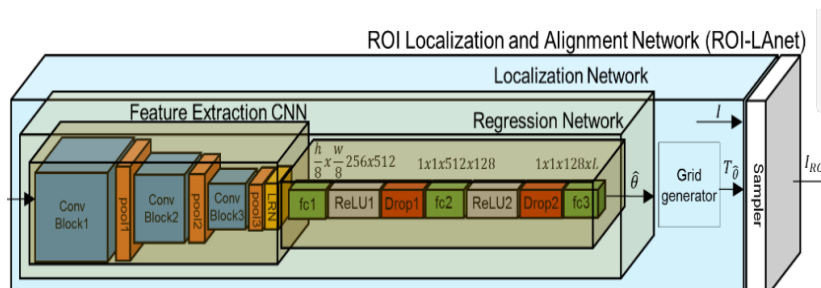
Cột mốc của lòng bàn tay (Hand landmarks) là các điểm dùng để điều chỉnh độ biến dạng đàn hồi do các tư thế tay khác nhau. Vì vậy mà thông thường sẽ có ít nhất ba điểm cột mốc để tham số hóa các phép biến đổi. Tuy nhiên ba điểm thì chỉ có thể dùng phép biến đổi affine hơn nữa các cột mốc sẽ kéo dài toàn bộ lòng bàn tay vì nhiều diện tích của lòng bàn tay hơn sẽ mang nhiều thông tin phân biệt hơn. Do đó, ban đầu các mốc được xác định ở bốn góc của lòng bàn tay là L1, L3, L7 và L9. Nếu chỉ lấy bốn điểm này thì đường nối L1 và L3 sẽ cắt ngang lòng bàn tay và một số thông tin về lòng bàn tay sẽ bị mất và mốc L2 sẽ bị bỏ qua. Khi điểm L2 được lấy thì những vấn đề trên sẽ không còn nữa và việc xác định cũng sẽ chính xác hơn. L1 và L3 được xác định rõ ràng nằm dưới cùng của các ngón tay, tuy nhiên L7 và L9 không phải lúc nào cũng rõ ràng L7 và L9 được đặt gần đường viền giữa lòng bàn tay và cổ tay. Ngoài những điểm mốc này, những điểm chính giữa của các dòng, $\{L1, L7\}$, $\{L7, L9\}$ và $\{L3, L9\}$ tự động được chiết xuất và được dán nhãn tương ứng là L4, L8 và L6. Các điểm giữa giữa L8 và L2 cũng được trích xuất và dán nhãn là L5. Chín điểm kể trên được trích xuất để đào tạo cho một mạng có tên là mạng bản địa hóa (Localization Network).



Hình 3.6: Các cột mốc trên lòng bàn tay

Dựa trên cấu trúc của một máy biến đổi không gian thì mạng sử dụng cho việc trích xuất vùng quan tâm được gọi là ROI-LAnet. Phần đầu tiên của mạng ROI-LAnet là phần chỉnh sửa của mạng VGG-16. VGG-16 là một mạng thần kinh tích chập tương đối sâu nên với một lượng dữ liệu nhỏ của lòng bàn tay thì dễ xảy ra hiện tượng overfitting. Tuy nhiên với việc sử dụng mạng được đào tạo trước và giảm độ sâu của mạng thì nhu cầu sử dụng dữ liệu sẽ giảm bớt và cải thiện được vấn đề overfitting.

Cụ thể mạng VGG-16 được đào tạo trên tập ImageNet sẽ cắt bỏ các lớp sau lớp pool3 trên cùng là lớp chuẩn hóa phản hồi cục bộ. Mạng phản hồi cục bộ là một mạng hồi quy bao gồm 2 lớp ẩn fc1 và fc2 hay lớp này là 2 lớp fully connected và cả 2 lớp này đều có hàm kích hoạt là leakyReLU và dropout để tránh sự kết hợp của các nơ-ron để có thể giảm sự tăng ROI trong quá trình đào tạo. Đầu ra của mạng là tọa độ của các mốc trên lòng bàn tay đã được chuẩn hóa và phạm vi chuẩn hóa là từ -1 đến 1. Tọa độ đã được chuẩn hóa sau đó được đưa đến bộ tạo lưới (grid generator) cùng với phép biến đổi không gian trình bày ở trên để tạo ra đầu ra kích thước $h_{ROI} \times w_{ROI}$ là vùng ROI để cung cấp cho mạng trích xuất vector đặc trưng. Dưới đây là cấu trúc của mạng ROILA-Net.



Hình 3.7: Cấu trúc của mạng ROILA-Net

Thông số chi tiết các lớp và số parameters của mạng:

Layer (type)	Output Shape	Param #
Conv2d-1	[48, 64, 224, 224]	1,792
ReLU-2	[48, 64, 224, 224]	0
Conv2d-3	[48, 64, 224, 224]	36,928
ReLU-4	[48, 64, 224, 224]	0
MaxPool2d-5	[48, 64, 112, 112]	0
Conv2d-6	[48, 128, 112, 112]	73,856
ReLU-7	[48, 128, 112, 112]	0
Conv2d-8	[48, 128, 112, 112]	147,584
ReLU-9	[48, 128, 112, 112]	0
MaxPool2d-10	[48, 128, 56, 56]	0
Conv2d-11	[48, 256, 56, 56]	295,168
ReLU-12	[48, 256, 56, 56]	0
Conv2d-13	[48, 256, 56, 56]	590,080
ReLU-14	[48, 256, 56, 56]	0
Conv2d-15	[48, 256, 56, 56]	590,080
ReLU-16	[48, 256, 56, 56]	0
MaxPool2d-17	[48, 256, 28, 28]	0
LocalResponseNorm-18	[48, 256, 28, 28]	0
Linear-19	[48, 512]	6,423,040
LeakyReLU-20	[48, 512]	0
Dropout-21	[48, 512]	0
Linear-22	[48, 128]	65,664
LeakyReLU-23	[48, 128]	0
Dropout-24	[48, 128]	0
Linear-25	[48, 18]	2,322
Total params: 8,226,514		
Trainable params: 8,226,514		
Non-trainable params: 0		

3.1.4 Trích xuất vec-tơ đặc trưng

Cho tới nay có nhiều phương pháp được đưa ra để trích xuất vec-tơ đặc trưng được đưa ra như dùng bộ lọc Gabor, ... Với sự phát triển các mạng nơ-ron thì mạng học chuyển đổi được sử dụng như một công cụ để trích xuất vec-tơ đặc trưng có độ chính xác cao. Trong khuôn khổ đề án, một mạng học chuyển đổi được sử dụng để trích xuất vec-tơ đặc trưng. Cụ thể, một mạng nơ-ron được bỏ lớp kết nối đầy đủ để trích xuất các đặc trưng cấp thấp. Vec-tơ này sau đó được làm phẳng và được đưa vào một lớp kết nối đầy đủ để thay đổi kích thước đầu ra theo ý muốn. Cuối cùng vec-tơ này được chuẩn hóa trong một khoảng nhất định. Dưới đây là thông số về số parameters của các mạng được sử dụng trong khuôn khổ đề án:

- MobileNet_V2: mạng gồm 158 lớp với tổng số params là 2,387,840
- MobileNet_V3_small: mạng gồm 142 lớp với tổng số params là 1,000,864
- MobileNet_V3_large: mạng gồm 179 lớp với tổng số params là 3,094,960

3.1.5 Nhận diện lòng bàn tay

Độ tương tự cosine đo độ giống nhau giữa hai vector khác không của một không gian tích vô hướng. Nó được đo bằng cô-sin của góc giữa hai vector và xác định xem hai vector có hướng gần giống nhau hay không. Với vec-tơ đặc trưng được trích xuất với các mô hình trình bày ở trên thì việc sử dụng độ tương tự cô-sin để so sánh là tương đối dễ dàng. Độ tương tự cô-sin được tính toán theo công thức dưới đây:

$$\text{Similarity} = \frac{x \cdot y}{||x|| ||y||}$$

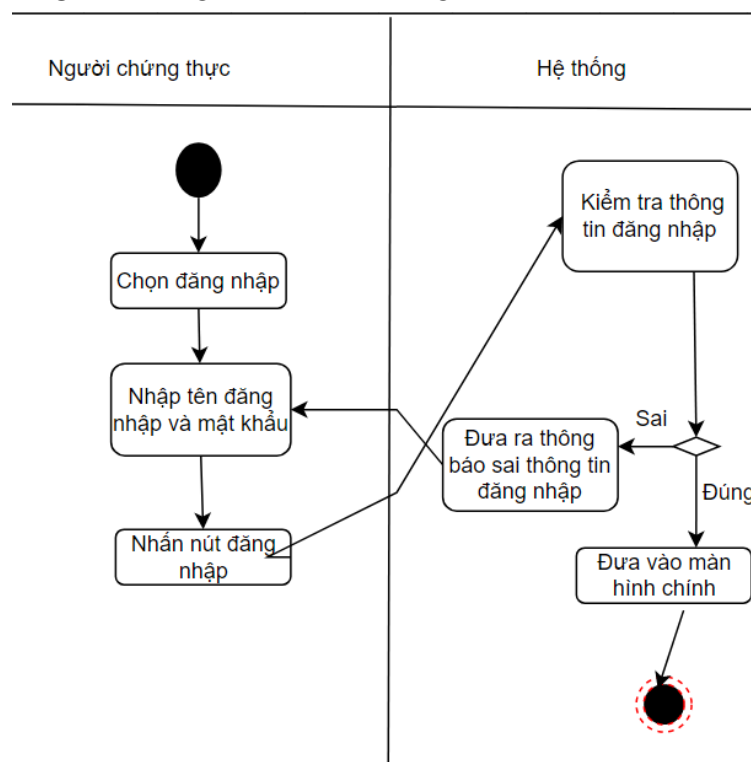
Hai lòng bàn tay càng giống nhau thì độ tương tự cô-sin giữa 2 vec-tơ đặc trưng càng gần với 1.

3.2 Thiết kế giao diện người dùng

3.2.1 Các hoạt động chính trong hệ thống

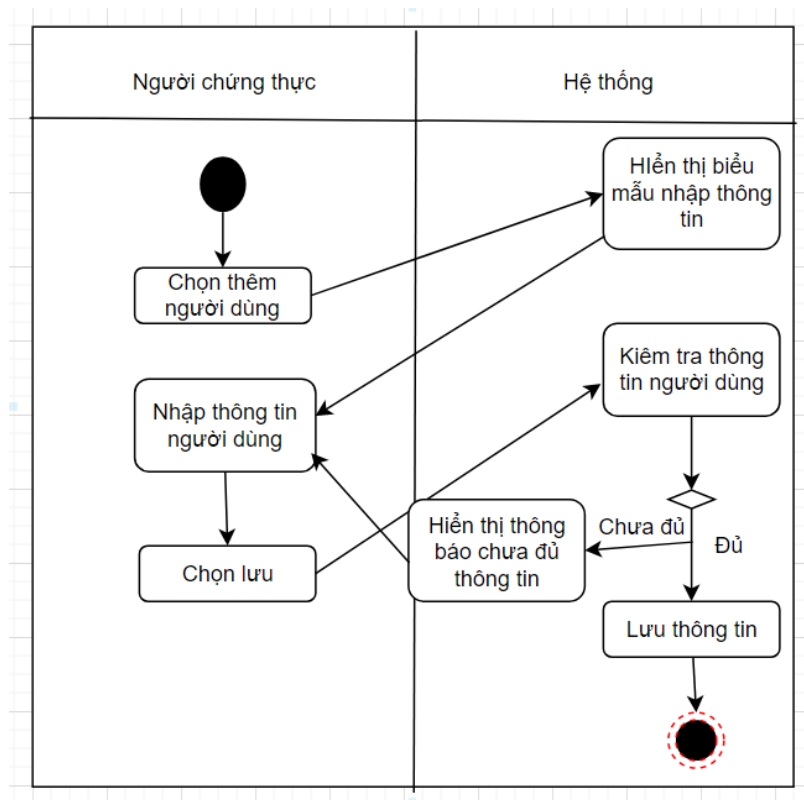
Dưới đây là các sơ đồ hoạt động trình bày về các quy trình quan trọng trong hệ thống chứng thực:

- Hoạt động đăng nhập: Người chứng thực sử dụng tài khoản được cấp đăng nhập vào hệ thống. Dưới đây là biểu đồ thể hiện hoạt động đăng nhập tương tác giữa người chứng thực và hệ thống:



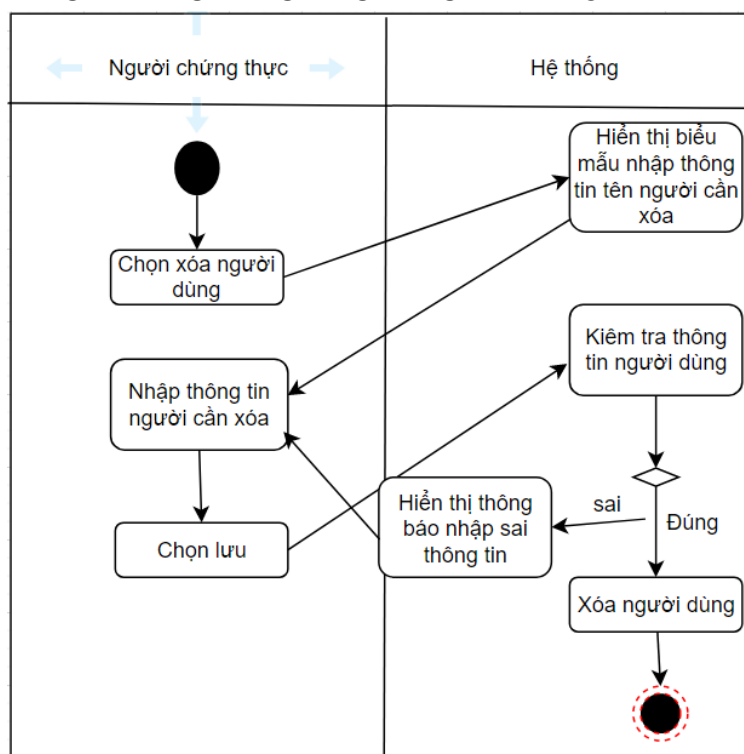
Hình 3.8 : Biểu đồ hoạt động đăng nhập

- Hoạt động thêm người dùng: Người chứng thực thêm người dùng vào cơ sở dữ liệu thông qua giao diện của hệ thống. Dưới đây là biểu đồ thể hiện hoạt động thêm người dùng tương tác giữa người chứng thực và hệ thống:



Hình 3.9: Biểu đồ hoạt động thêm người dùng

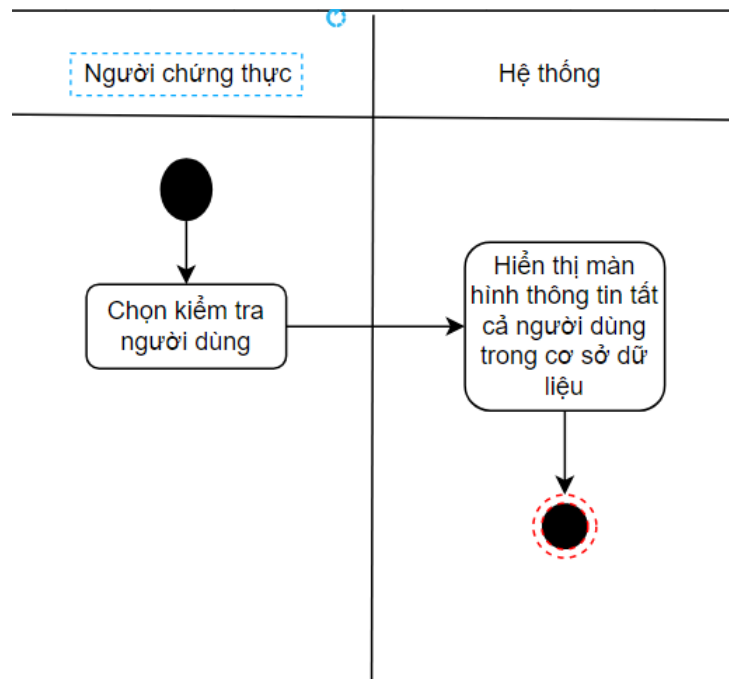
- Hoạt động xóa người dùng: Người chứng thực xóa người dùng khỏi cơ sở dữ liệu thông qua giao diện của hệ thống. Dưới đây là biểu đồ thể hiện hoạt động xóa người dùng tương tác giữa người chứng thực và hệ thống:



Hình 3.10: Biểu đồ hoạt động xóa người dùng

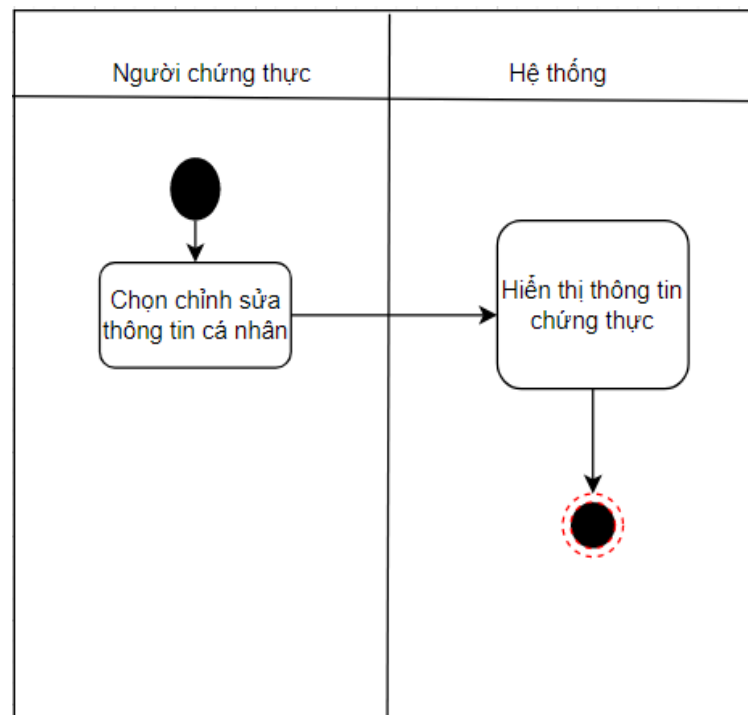
- Hoạt động kiểm tra thông tin người dùng: Người chứng thực kiểm tra thông tin toàn bộ người dùng trong cơ sở dữ liệu thông qua giao diện của hệ thống.

Dưới đây là biểu đồ thể hiện hoạt động kiểm tra thông tin người dùng tương tác giữa người chứng thực và hệ thống:



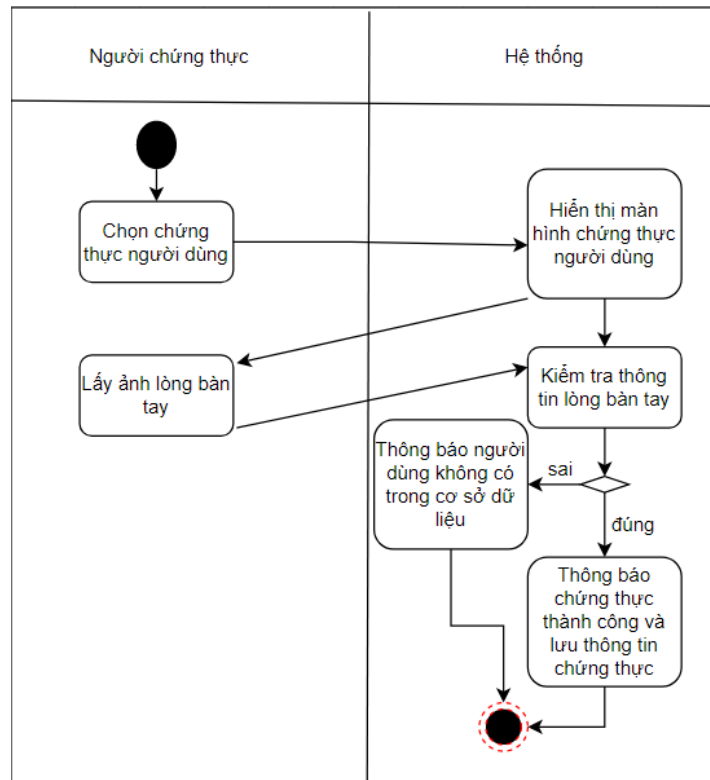
Hình 3.11: Biểu đồ hoạt động kiểm tra thông tin người dùng

- Hoạt động kiểm tra thông tin chứng thực người dùng: Người chứng thực kiểm tra thông tin chứng thực toàn bộ người dùng trong cơ sở dữ liệu thông qua giao diện của hệ thống. Dưới đây là biểu đồ thể hiện hoạt động kiểm tra thông tin chứng thực của người dùng tương tác giữa người chứng thực và hệ thống:



Hình 3.12: Biểu đồ hoạt động kiểm tra thông tin chứng thực

- Hoạt động chứng thực người dùng: Người chứng thực chứng thực người dùng thông qua giao diện của hệ thống. Dưới đây là biểu đồ thể hiện hoạt động chứng thực người dùng tương tác giữa người chứng thực và hệ thống:



Hình 3.13: Biểu đồ hoạt động chứng thực người dùng

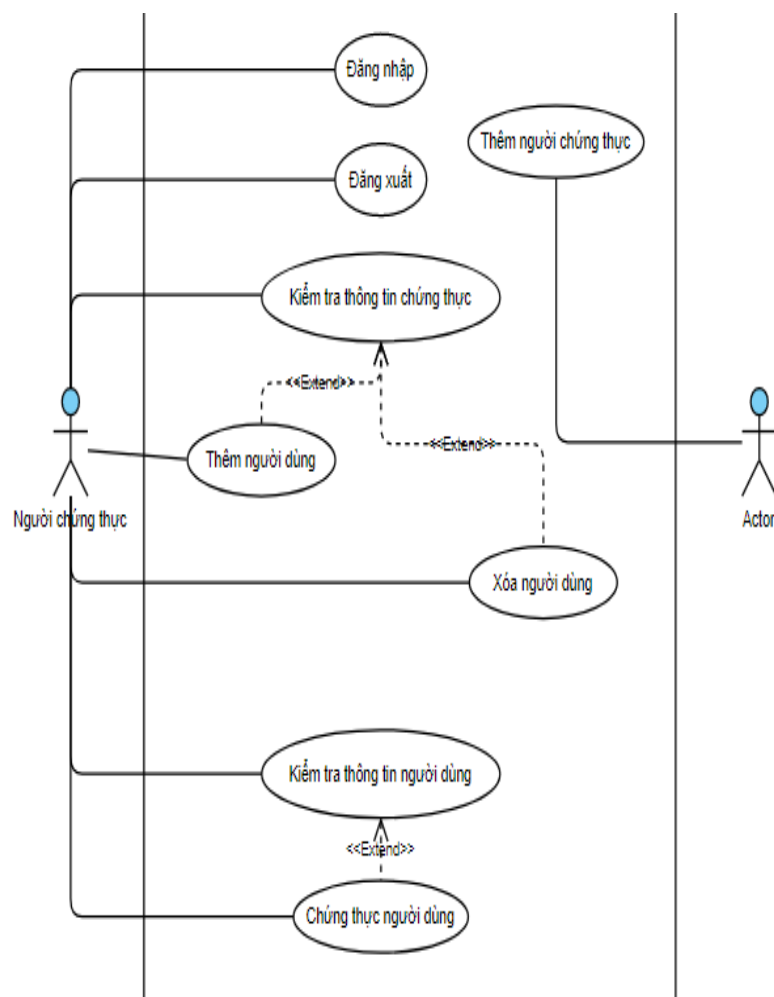
3.2.2 Phân tích các yêu cầu chức năng

a) Tác nhân sử dụng hệ thống

Bảng 3.1: Tác nhân sử dụng hệ thống

Tác nhân	Mô tả
Người chứng thực	Người đã có thông tin trong cơ sở dữ liệu tương tác với hệ thống thông qua giao diện người dùng trên máy tính. Có thể thêm, xóa, kiểm tra thông tin người dùng và chứng thực người dùng.
Quản trị viên	Người có thể thêm người chứng thực vào cơ sở dữ liệu.

b) Biểu đồ Use case hệ thống



Hình 3.14: Biểu đồ Use case hệ thống

c) Đặc tả chi tiết các Use case

- Use case đăng nhập:

Bảng 3.2: Đặc tả use case Đăng nhập

Mã chức năng	UC01
Tên chức năng	Đăng nhập
Mô tả	Đăng nhập vào hệ thống
Tác nhân	Người chứng thực
Sự kiện kích hoạt	Ấn nút đăng nhập trên màn hình đăng nhập
Điều kiện tiên quyết	Không có
Trạng thái hệ thống sau khi thực hiện chức năng	Đăng nhập vào hệ thống và chuyển sang giao diện người dùng
Luồng xử lý chính	1. Người chứng thực nhập tên đăng nhập và mật khẩu vào form đăng nhập 2. Người chứng thực ấn nút đăng nhập

	3. Hệ thống kiểm tra điền form đầy đủ và hợp lệ 4. Hệ thống xác nhận thông tin tài khoản thành công 5. Hệ thống đăng nhập thành công và chuyển sang giao diện người dùng
Luồng thay thế	3b. Hệ thống phát hiện thông tin đăng nhập không đủ. Hệ thống thông báo sai sót cho người dùng 4b. Hệ thống phát hiện thông tin đăng nhập không chính xác. Hệ thống thông báo sai thông tin đăng nhập cho người dùng

- Use case đăng xuất:

Bảng 3.3: Đặc tả use case Đăng xuất

Mã chức năng	UC02
Tên chức năng	Đăng xuất
Mô tả	Đăng xuất khỏi hệ thống
Tác nhân	Người chứng thực
Sự kiện kích hoạt	Ấn nút đăng xuất trên màn hình đăng nhập
Điều kiện tiên quyết	Không có
Trạng thái hệ thống sau khi thực hiện chức năng	Đăng xuất và thoát khỏi hệ thống chứng thực
Luồng xử lý chính	1. Người dùng ấn Đăng xuất 2. Hệ thống đăng xuất tài khoản
Luồng thay thế	Không có

- Use case thêm người dùng

Bảng 3.4: Đặc tả use case thêm người dùng

Mã chức năng	UC03
Tên chức năng	Thêm người dùng
Mô tả	Người chứng thực thêm người dùng vào cơ sở dữ liệu
Tác nhân	Người người chứng thực
Sự kiện kích hoạt	Người chứng thực kích hoạt nút Thêm người dùng
Điều kiện tiên quyết	Không có

Trạng thái hệ thống sau khi thực hiện chức năng	Đưa ra thông báo thêm thành công hoặc cảnh báo thiếu thông tin người dùng
Luồng xử lý chính	<ol style="list-style-type: none"> 1. Người chứng thực ấn nút thêm người dùng và điền thông tin người dùng 2. Hệ thống kiểm tra thông tin người dùng có đủ không 3. Hệ thống đưa ra thông báo thêm người dùng thành công
Luồng thay thế	3b. Hệ thống đưa ra thông báo thông tin người dùng chưa đủ hoặc chưa chính xác

- Use case xóa người dùng

Bảng 3.5: Đặc tả use case xóa người dùng

Mã chức năng	UC04
Tên chức năng	Xóa người dùng
Mô tả	Người chứng thực xóa người dùng khỏi cơ sở dữ liệu
Tác nhân	Người người chứng thực
Sự kiện kích hoạt	Người chứng thực kích hoạt nút Xóa người dùng
Điều kiện tiên quyết	Không có
Trạng thái hệ thống sau khi thực hiện chức năng	Đưa ra thông báo xóa thành công hoặc cảnh báo thông tin người dùng không đúng
Luồng xử lý chính	<ol style="list-style-type: none"> 1. Người chứng thực điền thông tin người dùng cần xóa 2. Hệ thống kiểm tra thông tin người dùng 3. Hệ thống đưa ra thông tin xóa người dùng thành công
Luồng thay thế	3b. Hệ thống đưa ra thông báo thông tin người dùng chưa chính xác

- Use case kiểm tra thông tin người dùng

Bảng 3.6: Đặc tả use case Kiểm tra thông tin của người dùng

Mã chức năng	UC05
Tên chức năng	Xem thông tin người dùng
Mô tả	Người chứng thực kiểm tra thông tin của người dùng

Tác nhân	Người người chứng thực
Sự kiện kích hoạt	Người chứng thực kích hoạt nút xem thông tin người dùng
Điều kiện tiên quyết	Không có
Trạng thái hệ thống sau khi thực hiện chức năng	Xem được màn hình thông tin người dùng
Luồng xử lý chính	1. Người chứng thực ấn nút kiểm tra thông tin người dùng 2. Hệ thống hiển thị thông tin người dùng
Luồng thay thế	Không có

- Use case kiểm tra thông tin chứng thực

Bảng 3.7; Đặc tả use case Kiểm tra thông tin chứng thực của người dùng

Mã chức năng	UC06
Tên chức năng	Xem thông tin chứng thực của người dùng
Mô tả	Người chứng thực kiểm tra thông tin chứng thực của người dùng
Tác nhân	Người người chứng thực
Sự kiện kích hoạt	Người chứng thực kích hoạt nút xem thông tin chứng thực người dùng
Điều kiện tiên quyết	Không có
Trạng thái hệ thống sau khi thực hiện chức năng	Xem được màn hình thông tin chứng thực của người dùng
Luồng xử lý chính	1. Người chứng thực ấn nút kiểm tra thông tin chứng thực của người dùng 2. Hệ thống hiển thị thông tin chứng thực của người dùng
Luồng thay thế	Không có

3.2.3 Các yêu cầu phi chức năng

- Yêu cầu về hiệu suất: Ngoài độ chính xác thì thời gian thực thi của các mô hình lựa chọn cũng rất quan trọng và phải thực hiện được trên cả CPU và GPU.
- Yêu cầu bảo mật: Người chứng thực chỉ có thể truy cập được các chức năng và dữ liệu với tài khoản đăng nhập được cung cấp bởi quản trị viên.

- Yêu cầu phần mềm: Giao diện người dùng có thể chạy trên hầu hết các hệ điều hành phổ biến như là: Windows, Mac OS, ...

3.3 Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu được thiết kế trên mongoDB gồm các collection chính như sau:

- User (Người dùng): Là thông tin của người dùng.
- Login (Đăng nhập): Các tài khoản của người chứng thực dùng để đăng nhập vào hệ thống.
- Auth (Chứng thực): Là thông tin về việc chứng thực của người dùng

Dưới đây là chi tiết về thông tin các collection bao gồm: các trường, kiểu dữ liệu, mô tả về các trường đó:

- Bảng dữ liệu User:

Bảng 3.8: Bảng dữ liệu User

Tên trường	Kiểu dữ liệu	Mô tả
id	ObjectID	Định danh của người dùng
name	String	Tên của người dùng
address	String	Địa chỉ của người dùng
gender	String	Giới tính của người dùng
email	String	Email của người dùng
phoneNumber	Int	Số điện thoại của người dùng
right_hand	Binary	Lưu trữ thông tin vec-tơ đặc trưng của tay phải người dùng
left_hand	Binary	Lưu trữ thông tin vec-tơ đặc trưng của tay trái người dùng

- Bảng dữ liệu Login:

Bảng 3.9: Bảng dữ liệu Login

Tên trường	Kiểu dữ liệu	Mô tả
id	ObjectID	Định danh của người chứng thực
username	String	Tên của người chứng thực
email	String	Email của người chứng thực
password	String	Mật khẩu đăng nhập của người chứng thực

- Bảng dữ liệu Auth:

Bảng 3.10: Bảng dữ liệu Auth

Tên trường	Kiểu dữ liệu	Mô tả
id	ObjectID	Định danh của hành động chứng thực
user_name	String	Tên của người đã chứng thực chứng thực
email	String	Chức vụ của người đã chứng thực
address	String	Địa chỉ của người đã chứng thực
phoneNumber	Int	Số điện thoại của người dùng
Time_auth	Date	Thời gian người dùng đã chứng thực
Similarity	Double	Độ tương đồng của bàn tay dùng để chứng thực và bàn tay trong cơ sở dữ liệu

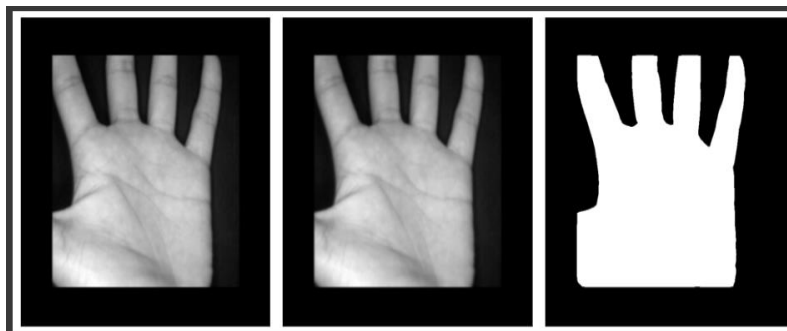
CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ

4.1 Nhận diện lòng bàn tay

4.1.1 Trích xuất vùng quan tâm

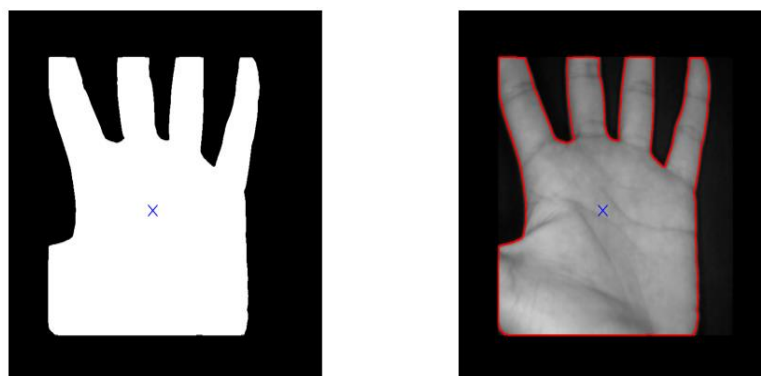
a) Phương pháp truyền thống

- Môi trường thực hiện: Việc trích xuất vùng quan tâm heo phương pháp truyền thống được làm tất cả Google Colab với các tham số:
 - Python: 3.7
 - OpenCV: 4.1.2
- Nhị phân ảnh đầu vào



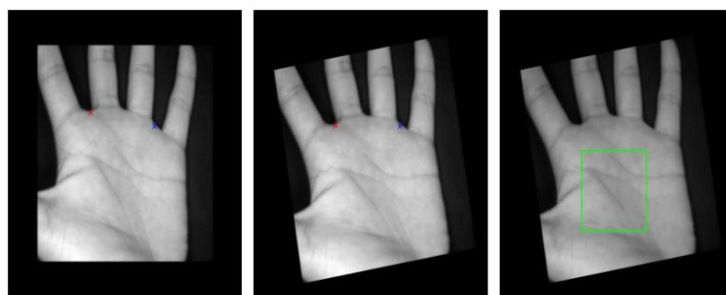
Hình 4.1: Ảnh đầu vào (kiểu ảnh xám) và ảnh nhị phân của nó.

- Trích xuất ranh giới của lòng bàn tay sau khi đã áp dụng kỹ thuật xói mòn hình thái học và xác định trọng tâm của lòng bàn tay:



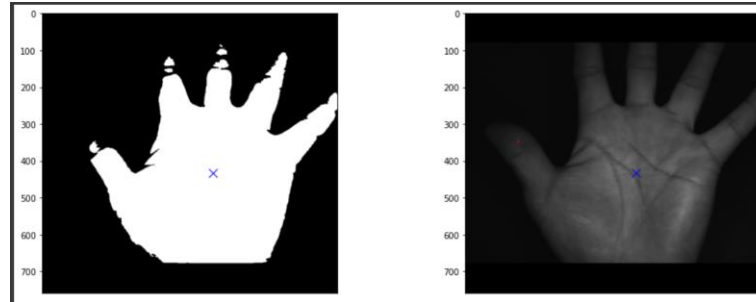
Hình 4.2: Tâm và ranh giới của lòng bàn tay

- Trích xuất roi:



Hình 4.3: Các điểm trung lũng và vùng roi trên lòng bàn tay

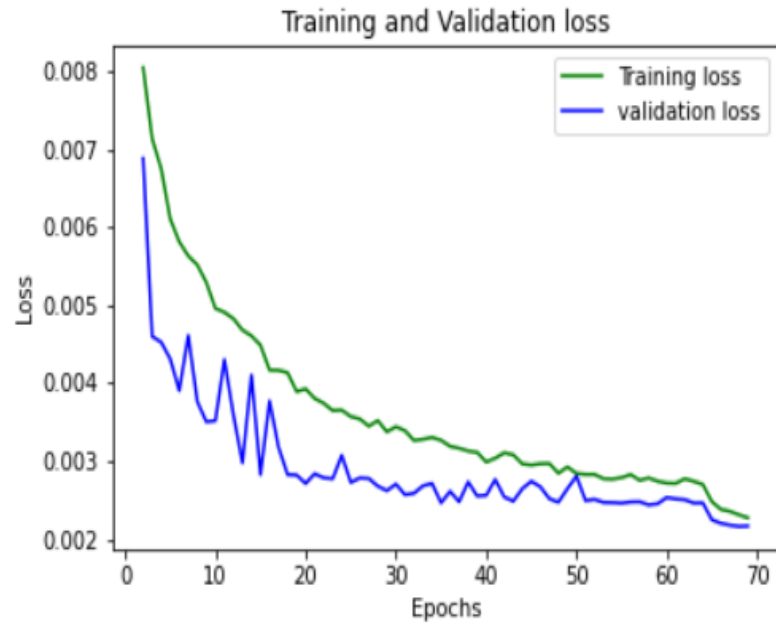
- Đánh giá kết quả: Phương pháp truyền thống cho kết quả trích xuất vùng quan tâm rất tốt. Tuy nhiên một nhược điểm rất lớn là yêu cầu sự phân biệt rõ ràng của lòng bàn tay và nền xung quanh (như trong hình là một nền đen tuyệt đối). Với hầu hết các dữ liệu ảnh lòng bàn tay trong tập Tongji thì phương pháp trích xuất roi truyền thống này không áp dụng được. Dưới đây là minh họa về kết quả không thể trích xuất đường viền bằng phương pháp này:



Hình 4.4: Tâm và đường viền của lòng bàn tay khi trích xuất bị lỗi

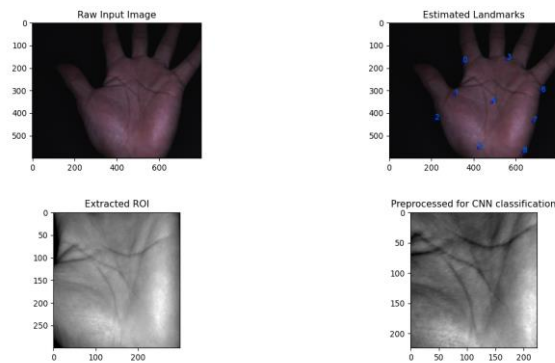
b) ROILA-Net:

- Tập dữ liệu: Tập dữ liệu lòng bàn tay được lấy từ cơ sở dữ liệu NTU Palmprints từ Internet (NTU-Plv1) bao gồm bao gồm 7781 hình ảnh bàn tay được thu thập từ 2035 lòng bàn tay khác nhau của 1093 đối tượng với các dân tộc, giới tính và độ tuổi, được thiết lập cùng với 1000 lòng bàn tay ngẫu nhiên từ tập dữ liệu Tongji. Trong đó, 7903 hình ảnh được sử dụng cho tập huấn luyện và 878 hình ảnh được sử dụng cho tập đánh giá.
- Môi trường thí nghiệm: Tất cả công việc đào tạo và huấn luyện được thực hiện trên Google Colab.
 - Ngôn ngữ: Python 3
 - Framework: Pytorch 1.11
 - GPU: Tesla T4
 - Hỗ trợ: Pillow 9.2, OpenCV 4.1.2, ...
- Các siêu tham số:
 - Số kỷ nguyên (epochs): 70
 - Batch_size: 32
 - Optimizer: Adam
 - Learning rate: 0.001
- Kết quả thí nghiệm:

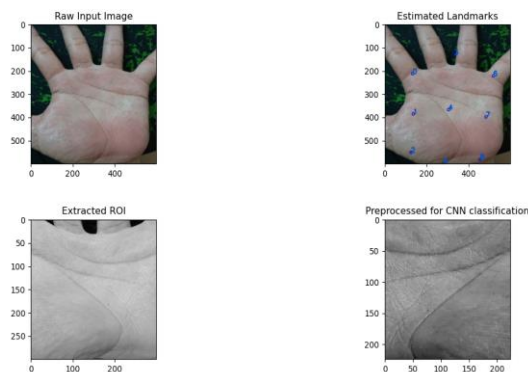


Hình 4.5: Mất mát trong quá trình đào tạo mô hình ROILA-Net

Quá trình đào tạo cho một kết quả tương đối tốt với mất mát thấp và việc trích xuất các điểm cột mốc bàn tay đạt kết quả rất tốt trên những hình ảnh lòng bàn tay từ tập Tongji. Tuy nhiên việc trích xuất các điểm cột mốc trên lòng bàn tay với một hình ảnh được chụp từ điện thoại di động cũng chưa thực sự tốt và vẫn còn một số sai số nhất định nhưng chấp nhận được.



Hình 4.6: Kết quả thực nghiệm trích xuất roi của mô hình ROI-LAnet với hình ảnh từ tập dữ liệu Tongji



Hình 4.7: Kết quả thực nghiệm trích xuất roi của mô hình ROI-LAnet với hình ảnh chụp từ điện thoại cá nhân

4.1.2 Trích xuất vec-tơ đặc trưng

- Tập dữ liệu: sử dụng hình ảnh roi được trích xuất từ ROI-LAnet với toàn bộ hình ảnh cơ sở dữ liệu Tongji. Trong bộ dữ liệu Tongji, hình ảnh được thu thập từ 300 tình nguyện viên, bao gồm 192 nam và 108 nữ. Trong số đó, 235 đối tượng từ 20 ~ 30 tuổi và những người khác từ 30 ~ 50 tuổi. Bộ dữ liệu đã thu thập các mẫu trong hai phiên riêng biệt. Trong mỗi phiên, đối tượng được yêu cầu cung cấp 10 hình ảnh cho mỗi lòng bàn tay. Do đó, 40 hình ảnh từ 2 lòng bàn tay đã được thu thập từ mỗi đối tượng. Tổng cộng, cơ sở dữ liệu chứa 12.000 hình ảnh được chụp từ 600 lòng bàn tay khác nhau. Trong đó 9584 hình ảnh dữ liệu được sử dụng cho tập huấn luyện và 2396 hình ảnh dữ liệu sử dụng cho tập đánh giá.
- Môi trường thí nghiệm: Tất cả công việc đào tạo và huấn luyện được thực hiện trên Google Colab.
 - Ngôn ngữ: Python 3
 - Framework: Pytorch 1.11
 - GPU: Tesla T4
 - Hỗ trợ: Pillow 9.2, OpenCV 4.1.2, ...
- Các siêu tham số:
 - Số kỷ nguyên (epochs): 120
 - Batch_size: 48
 - Optimizer: RAdam
 - Learning rate: 0.0001
- Kết quả thí nghiệm: Độ chính xác của các mô hình so sánh trên tập val:

Mô hình	mAP	Accuracy
MobileNet_V2	0.997	0.999
MobileNet_V3_small	0.995	0.998
Mobilenet_V3_large	0.997	0.999

- Đánh giá kết quả: Mobilenet_V3_small là mạng nhẹ nhất cho kết quả xấp xỉ với Mobilenet_V2 và Mobilenet_V3_large là 0.997. Kết quả cho thấy độ chính xác của các mô hình được chọn có kết quả tương đối tốt. Tuy nhiên thực nghiệm khi so sánh độ tương đồng để tìm kiếm thì Mobilenet_V2 cho kết quả trích xuất vec-tơ đặc trưng tốt hơn nên có kết quả so sánh khả quan hơn. Vì vậy Mobilenet_V2 sẽ được lựa chọn làm mạng cơ sở cho việc trích xuất vec-tơ đặc trưng.

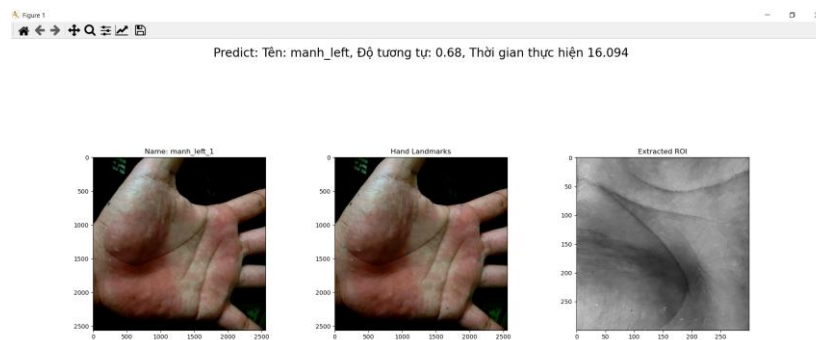
4.1.3 Nhận diện lòng bàn tay

Việc nhận diện được thực hiện trên 14 bàn tay thu thập từ 7 người khác nhau bao gồm 5 nam và 2 nữ. Quá trình nhận diện lòng bàn tay bao gồm trích xuất vùng quan tâm, trích xuất và nhận diện trên toàn bộ 14 bàn tay thực hiện trong khoảng thời gian 14s. Dưới đây là hình ảnh nhận diện của lòng bàn tay.



Hình 4.8: Nhận diện lòng bàn tay.

Khi thay đổi một số yếu tố như màu sắc độ sáng của ảnh lòng bàn tay thì độ tương tự của lòng bàn tay bị giảm tuy nhiên nhận diện vẫn chính xác.



Hình 4.9: Nhận diện lòng bàn tay với độ sáng và độ tương phản thay đổi

Tư thế của lòng bàn tay cũng là một yếu tố ảnh hưởng đến việc trích xuất vùng quan tâm dẫn đến ảnh hưởng tới việc nhận diện.



Hình 4.10: Nhận diện lòng bàn tay khi thay đổi tư thế của bàn tay

Đánh giá kết quả: Mô hình trích xuất vec-tơ đặc trưng của lòng bàn tay tương đối tốt nhưng mô hình trích xuất các điểm cột mốc trên lòng bàn tay chưa thực sự tốt. Nó chỉ hoạt động thực sự tốt trên các ảnh của tập dữ liệu Tongji. Với

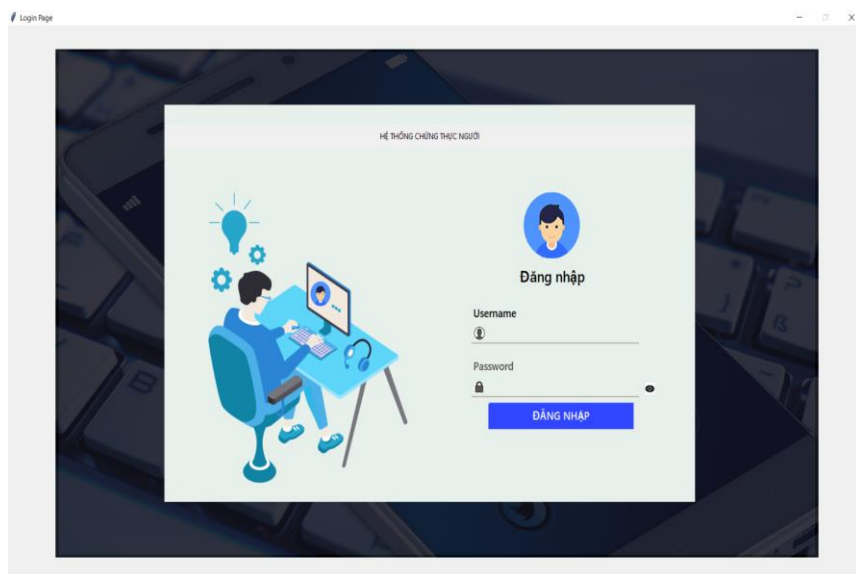
khoảng ngẫu nhiên khoảng 20 hình ảnh trên tập Tongji thì kết quả dự đoán đều cho kết quả chính xác với những hình ảnh là một cặp có độ tương đồng lớn hơn 0.9 còn những hình ảnh khác nhau có độ tương đồng thấp (nhỏ hơn 0.3). Còn với những hình ảnh chụp từ điện thoại cá nhân thì mô hình nhận diện vẫn chính xác. Tuy nhiên độ tương đồng giữa 2 lòng bàn tay thuộc một cặp sẽ lớn hơn 0.6 còn giữa những bàn tay khác nhau sẽ nhỏ hơn 0.3. Khi những hình ảnh lòng bàn tay thuộc cùng một bàn tay nhưng có tư thế của bàn tay khác nhau sẽ cho ra độ tương đồng tương đối thấp như ở ví dụ ở phía trên (độ tương đồng khác nhau 0.15).

4.2 Giao diện người dùng

Dưới đây là những hình ảnh về giao diện cơ bản của hệ thống chứng thực người dùng:

4.2.1 Giao diện đăng nhập

Giao diện đăng nhập để cho những người chứng thực đăng nhập vào trong hệ thống thông qua email và mật khẩu được cung cấp bởi các quản trị viên. Khi điền đúng và đủ email và mật khẩu thì hệ thống đưa người chứng thực vào màn hình chính của hệ thống, nếu sai thì hệ thống sẽ đưa ra thông báo lỗi cho người dùng. Hình ảnh các trường hợp sử dụng được trình bày ở mục 4.3 bên dưới.



Hình 4.11: Giao diện màn hình đăng nhập

4.2.2 Giao diện chính của hệ thống

Giao diện chính của hệ thống nơi chứa những nội dung chính của hệ thống. Tùy vào nhiệm vụ muốn thực hiện thì người chứng thực sẽ lựa chọn các button trên màn hình giao diện hệ thống. Khi muốn thoát khỏi màn hình chính và tắt giao diện của hệ thống người dùng có thể bấm vào button Đăng xuất.



Hình 4.12: Giao diện màn hình chính của hệ thống

4.2.3 Giao diện thêm người dùng

Giao diện thêm người dùng cho phép người chứng thực thêm một người dùng vào hệ thống dữ liệu. Trên màn hình người chứng thực cần lấy đầy đủ thông tin của người dùng như tên, địa chỉ, giới tính, thông tin 2 lòng bàn tay, ... Khi ấn button thêm người dùng thì người chứng thực cần điền đầy đủ và chính xác các thông tin nếu không hệ thống sẽ thông báo lỗi. Hình ảnh các trường hợp sử dụng được trình bày ở mục 4.3 bên dưới.

Hình 4.13: Giao diện màn hình thêm người dùng

4.2.4 Giao diện xóa người dùng

Giao diện xóa người dùng cho phép người chứng thực xóa một người dùng khỏi hệ thống dữ liệu. Trên màn hình người chứng thực cần nhập chính xác tên của người cần xóa để có thể thực hiện chức năng. Nếu như thông tin điền chưa chính xác hệ thống sẽ thông báo lỗi cho người chứng thực. Hình ảnh các trường hợp sử dụng được trình bày ở mục 4.3 bên dưới.



Hình 4.14: Giao diện xóa màn hình

4.2.5 Giao diện chứng thực người dùng

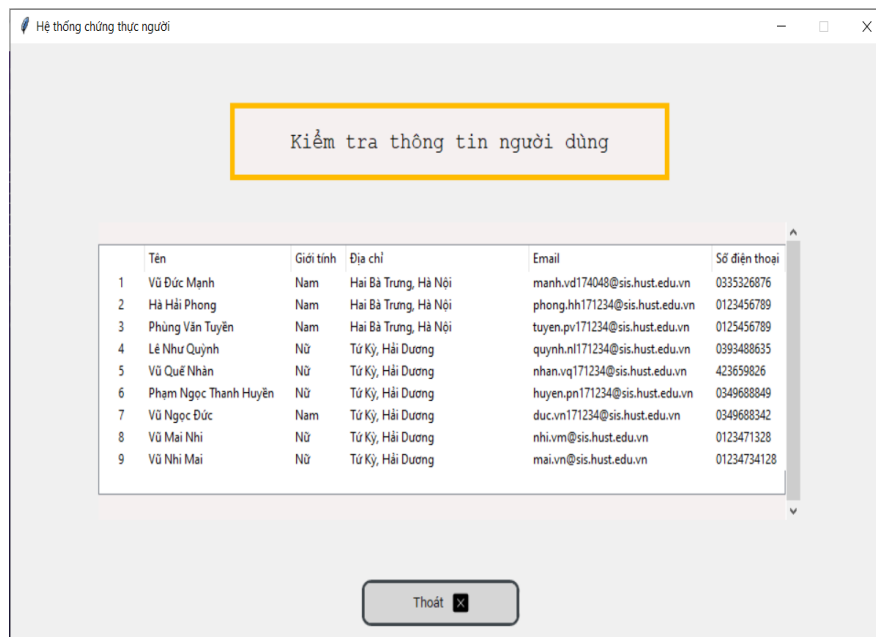
Giao diện chứng thực người dùng để giúp người chứng thực chứng thực một người nào đó. Để sử dụng chức năng người chứng thực cần cung cấp một hình ảnh lòng bàn tay của người cần chứng thực thông qua một button. Nếu chứng thực thành công thì hệ thống sẽ đưa ra thông báo thành công và lưu việc chứng thực vào cơ sở dữ liệu. Nếu không thành công hệ thống đưa ra cảnh báo cho người chứng thực. Hình ảnh các trường hợp sử dụng được trình bày ở mục 4.3 bên dưới.



Hình 4.15: Giao diện Chứng thực người dùng

4.2.6 Giao diện kiểm tra thông tin người dùng

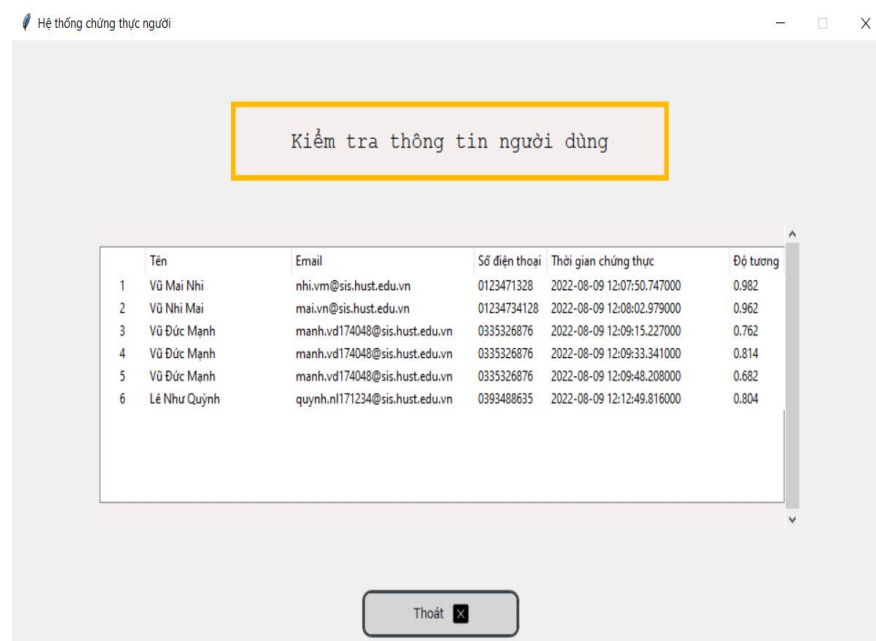
Giao diện kiểm tra thông tin người dùng giúp người chứng thực theo dõi tất cả người dùng trong cơ sở dữ liệu như tên, giới tính, địa chỉ, email, số điện thoại.



Hình 4.16: Giao diện Kiểm tra thông tin người dùng

4.2.7 Giao diện kiểm tra thông tin chứng thực người dùng

Giúp cho người chứng thực kiểm tra thông tin của những người đã chứng thực như tên, email, số điện thoại, thời gian đã chứng thực cũng như độ tương đồng của bàn tay đã chứng thực và bàn tay trong cơ sở dữ liệu thông qua màn hình giao diện của hệ thống.



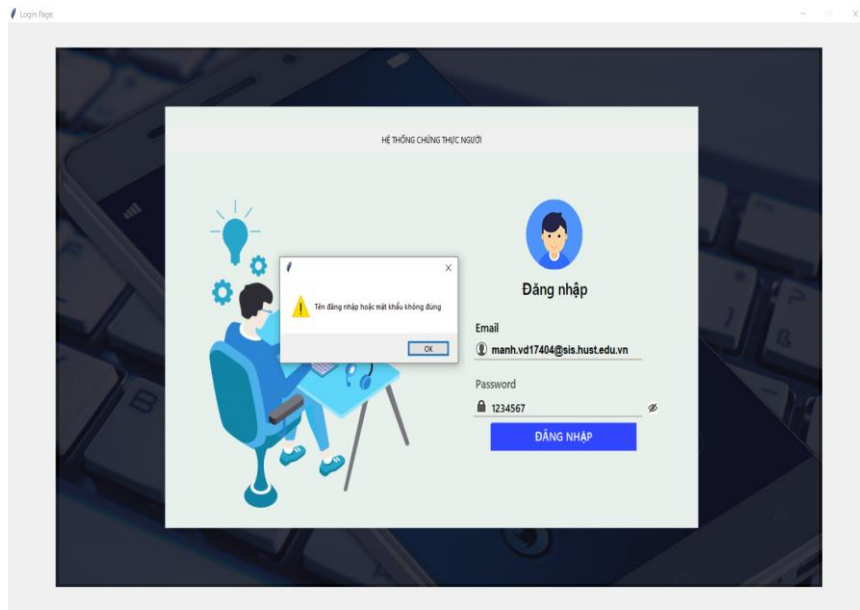
Hình 4.17: Giao diện Kiểm tra chứng thực người dùng

4.3 Kết quả cài đặt hệ thống và hình ảnh minh họa các chức năng sử dụng của hệ thống

Hệ thống được cài đặt thành công trên windows sử dụng CPU để thực hiện các chức năng của hệ thống. Dưới đây là hình ảnh minh họa cho các trường hợp sử dụng hệ thống với cơ sở dữ liệu gồm 14 bàn tay của 7 người, trong đó có 5 nam và 2 nữ:

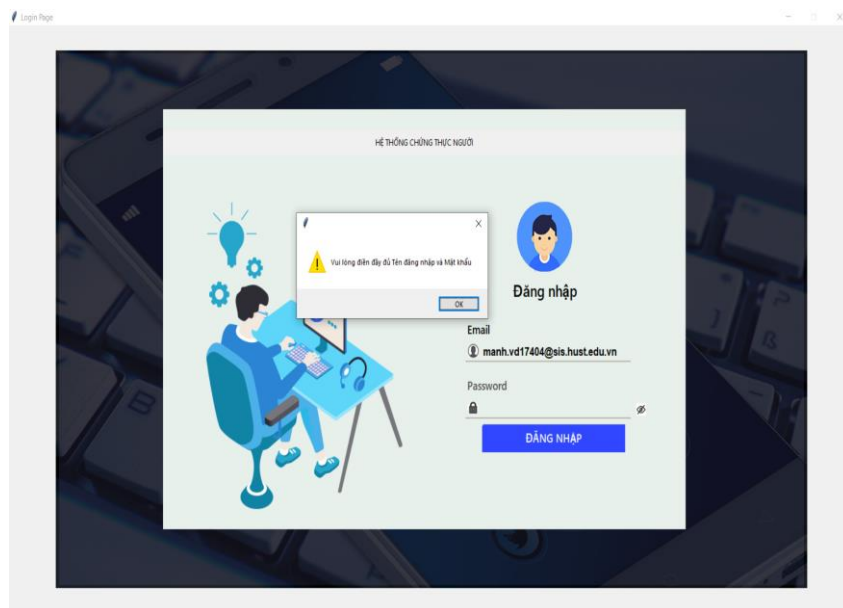
4.3.1 Chức năng đăng nhập:

- Sai thông tin đăng nhập hoặc mật khẩu



Hình 4.18: Cảnh báo sai thông tin đăng nhập

- Thiếu thông tin đăng nhập



Hình 4.19: Cảnh báo thiếu thông tin đăng nhập

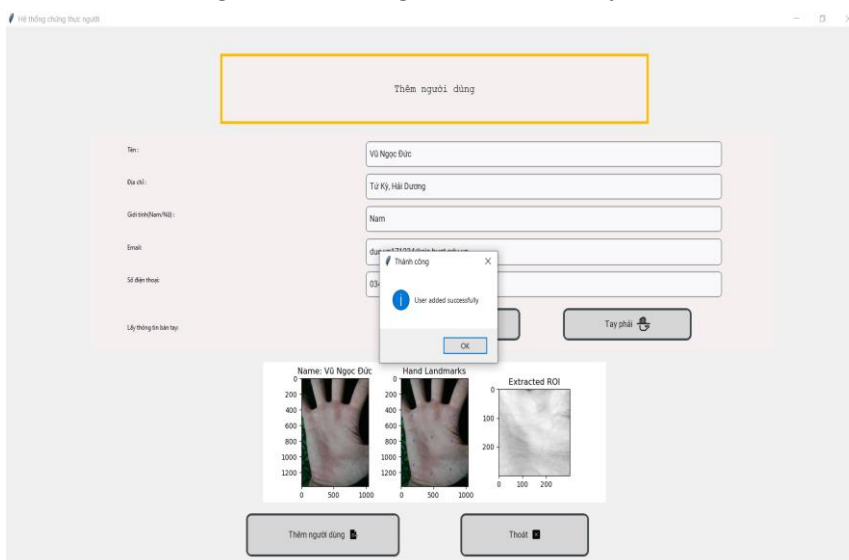
4.3.2 Giao diện chính của hệ thống



Hình 4.20: Giao diện chính của hệ thống

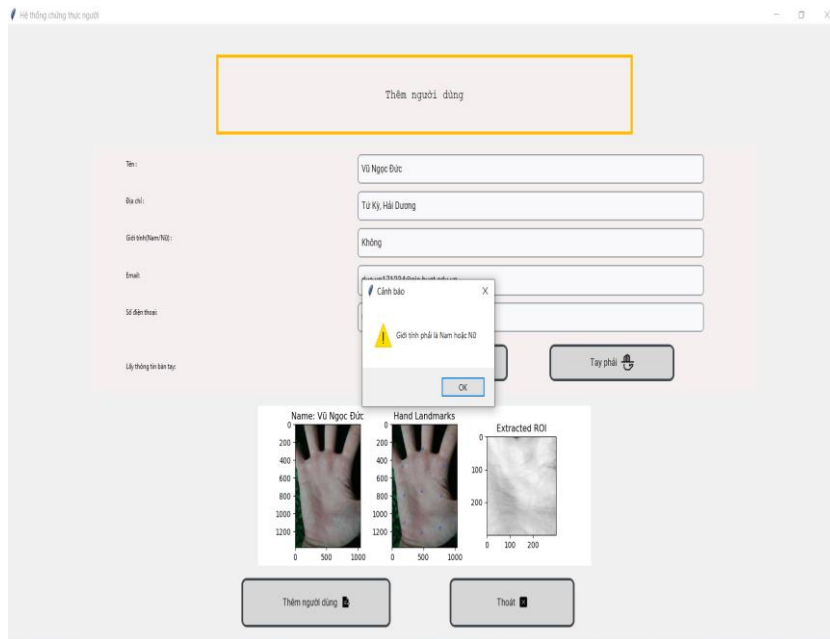
4.3.3 Giao diện thêm người dùng

- Giao diện thêm người dùng thành công: Thời gian lấy thông tin lòng bàn tay và lưu trữ trung bình khoảng 2s một bàn tay.



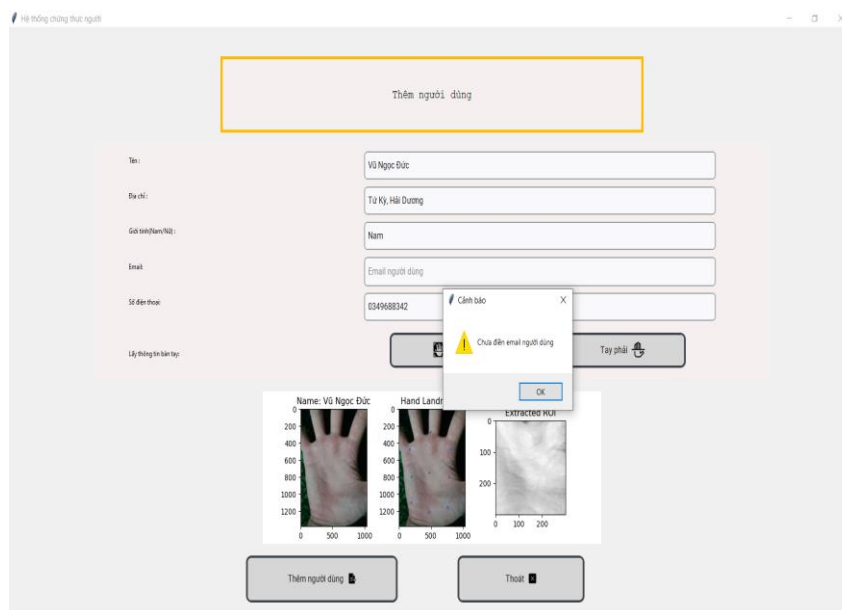
Hình 4.21: Giao diện thêm người dùng thành công

- Giao diện khi sai thông tin người dùng:



Hình 4.22: Giao diện thêm người với thông tin không chính xác

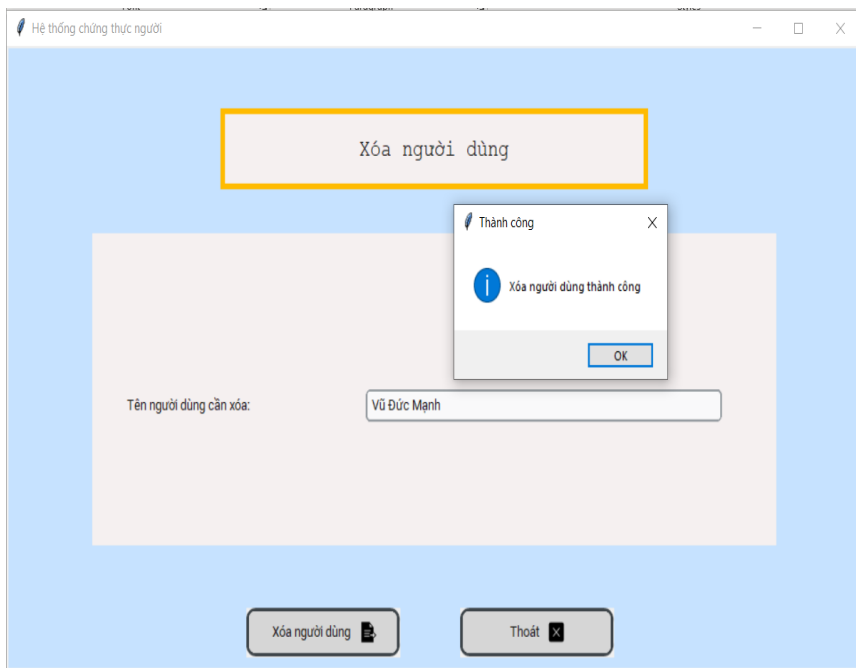
- Giao diện thêm người dùng khi thiếu thông tin người dùng



Hình 4.23: Giao diện thêm người dùng với thông tin bị thiếu

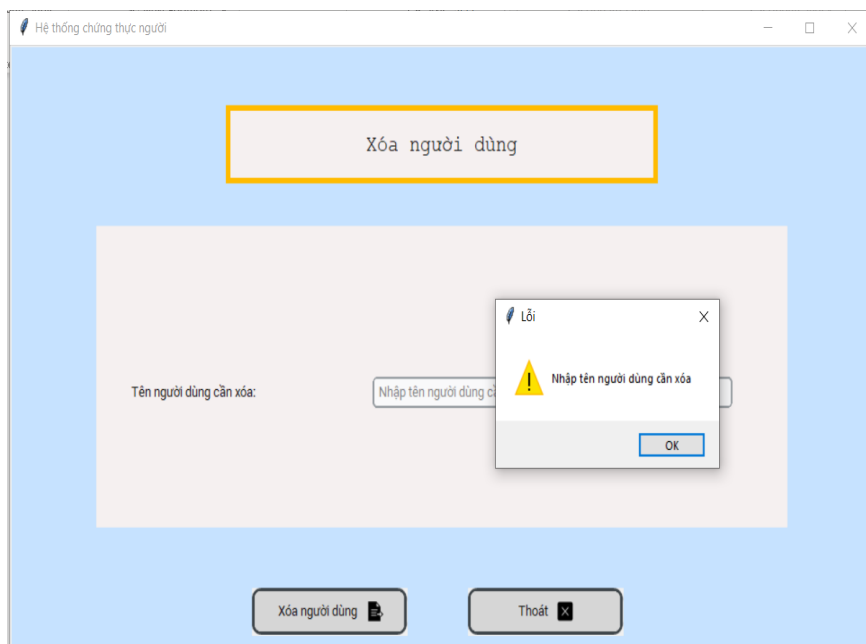
4.3.4 Giao diện xóa người dùng

- Giao diện xóa người dùng thành công



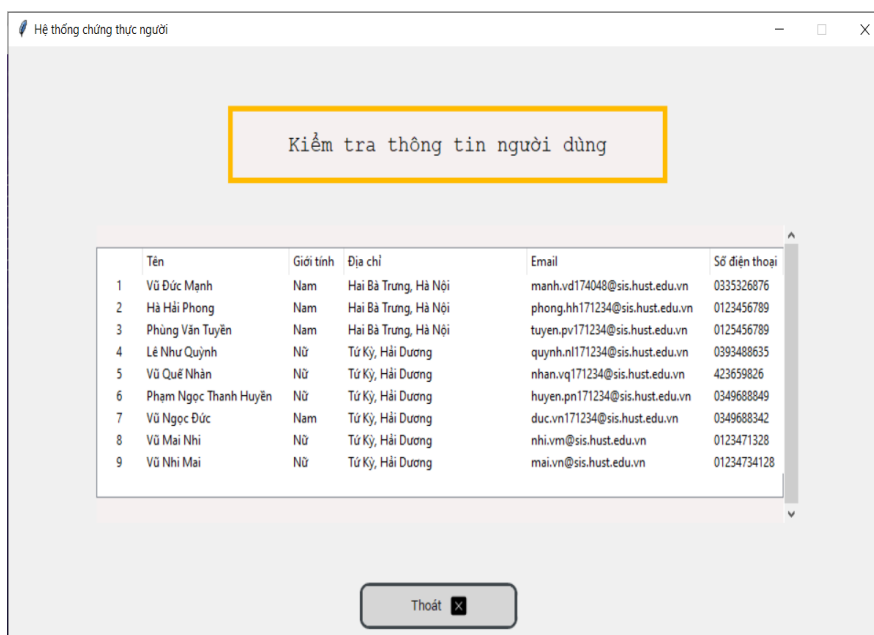
Hình 4.24: Giao diện xóa người dùng thành công

- Giao diện xóa người khi chưa điền tên người dùng



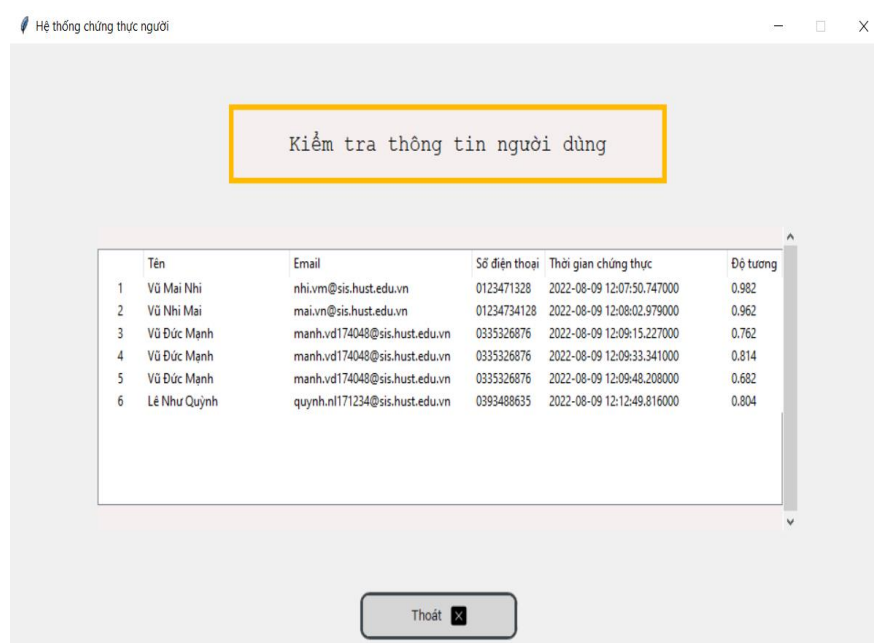
Hình 4.25: Giao diện xóa người dùng khi chưa điền tên người dùng

4.3.5 Giao diện kiểm tra thông tin người dùng



Hình 4.26: Giao diện kiểm tra thông tin người dùng

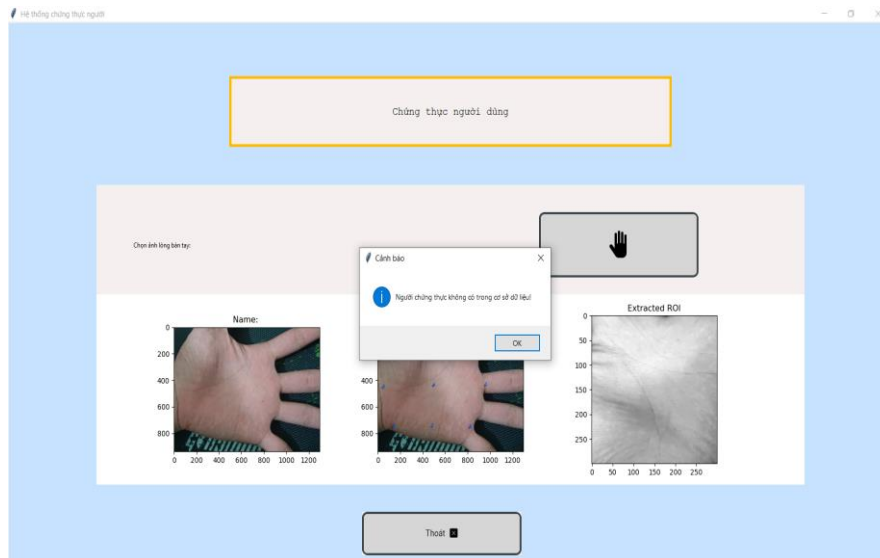
4.3.6 Giao diện kiểm tra thông tin chứng thực người dùng



Hình 4.27: Giao diện kiểm tra thông tin chứng thực của người dùng

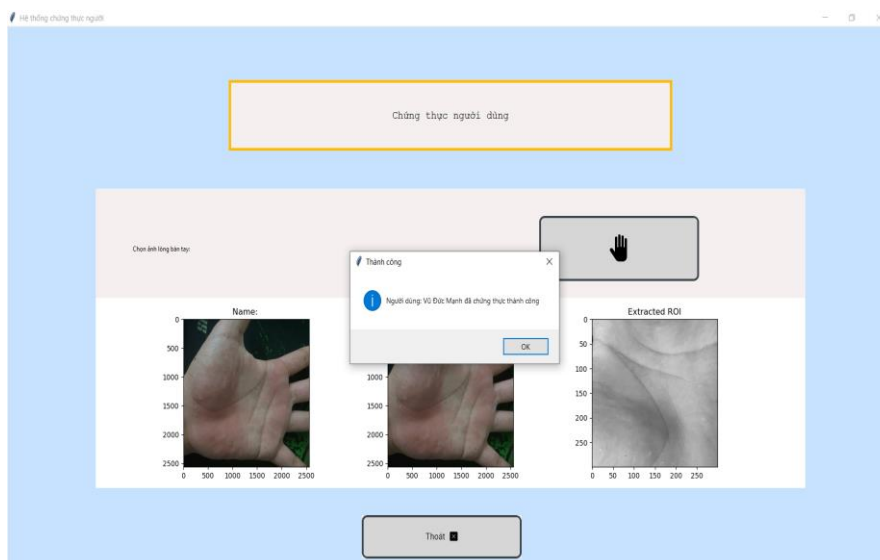
4.3.7 Giao diện chứng thực người dùng

- Giao diện chứng thực không thành công:



Hình 4.28: Giao diện chứng thực người dùng không thành công

- Giao diện chứng thực người dùng thành công: Thời gian xử lý trích xuất đặc trưng cũng như truy vấn và đối sánh vec-tơ đặc trưng hết khoảng 3s trên 14 bàn tay trong cơ sở dữ liệu:



Hình 4.29: Giao diện chứng thực người dùng thành công

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Đồ án tốt nghiệp đã phát triển một hệ thống chứng thực người thông qua lòng bàn tay đã đạt được một số thành công nhất định. Thực hiện được việc phân loại lòng bàn tay cũng như thiết kế được một giao diện người dùng để thuận tiện hơn cho việc chứng thực trên máy tính và một cơ sở dữ liệu cho việc lưu trữ thông tin người dùng cũng như việc lưu trữ thông tin chứng thực. Hệ thống áp dụng nhiều công nghệ về thị giác máy tính làm giải pháp cho việc đào tạo các mô hình trích xuất vùng quan tâm và trích xuất vec-tơ đặc trưng trên lòng bàn tay con người. Các mô hình này có thể triển khai tương đối đơn giản trên các hệ điều hành thông dụng như Windows, Mac OS.

Ngoài ra trong quá trình thực hiện đồ án tốt nghiệp bản thân em cũng đã học được nhiều kiến thức và kỹ năng như cách suy nghĩ và đưa ra giải pháp, hướng tiếp cận cho vấn đề và bài toán đã đặt ra; kiến thức lý thuyết và kỹ năng tiến hành thí nghiệm, triển khai các mô hình học máy, học sâu trong lĩnh vực Computer vision; khả năng tự học và tìm hiểu các công nghệ lập trình mới.

5.2 Hướng phát triển

Mặc dù hệ thống có nhiều ưu điểm tuy nhiên trong khuôn khổ của đồ án hệ thống không tránh khỏi được những sai sót. Dưới đây là một số sai sót, khuyết điểm của hệ thống

- Vấn đề về dữ liệu: đây là vấn đề lớn nhất với hệ thống. Hiện nay các dữ liệu về lòng bàn tay công khai trên mạng còn tương đối ít đặc biệt là dữ liệu cho việc đào tạo mạng ROILA-Net cho việc trích xuất các cột mốc của lòng bàn tay. Vì vậy việc trích xuất các cột mốc trên lòng bàn tay từ các thiết bị di động còn một số sai sót.
- Giao diện chứng thực người dùng chưa thực sự thân thiện và thiết kế còn chưa thực sự đẹp như còn thiếu một số chức năng cơ bản: chỉnh sửa thông tin cá nhân người dùng ... Do lần đầu tiên tìm hiểu cũng như triển khai cho nên thiết kế chưa thực sự tốt.

Hướng phát triển cho hệ thống trong tương lai:

- Thực hiện việc tìm kiếm và chỉnh sửa dữ liệu và đào lại các mô hình trích xuất vùng quan tâm và trích xuất vec-tơ đặc trưng để đem lại độ chính xác cao hơn. Đặc biệt là dữ liệu cho mạng ROILA-Net để tăng tính chính xác cho việc trích xuất vùng quan tâm kể cả với các hình ảnh được chụp từ các thiết bị điện thoại.
- Thực hiện việc chỉnh sửa lại giao diện người dùng để thân thiện với người dùng hơn. Thiết kế giao diện đẹp hơn và thêm một số chức năng phụ cho việc chứng thực người dùng.
- Triển khai hệ thống trên một trang web để có thể thuận tiện cho việc kiểm tra cũng như kiểm soát. Chuyển đổi source code về file thực thi hệ thống để người sử dụng thuận tiện hơn.

- Tăng tính bảo mật của hệ thống bằng cách sử dụng một số kỹ thuật như: hash password cũng tăng tính bảo mật của hệ thống cơ sở dữ liệu.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] M. Z. B. C. D. K. W. W. o. W. M. A. H. A. Andrew G. Howard, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision," *arXiv:1704.04861*, 2017.
- [2] X. Z. S. R. J. S. Kaiming He, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385v1*, 2015.
- [3] A. H. H. M. Z. A. Z. L.-C. C. Mark Sandler, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv:1801.04381*, 2019.
- [4] L. S. S. A. S. E. W. Jie Hu, "Squeeze-and-Excitation Networks," *arXiv:1709.01507*, 2017.
- [5] D. S. R. V. Dr.H.B.Kekre, "An Effectual Method for Extraction of ROI of Palmprints," *International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India*, 2012.
- [6] K. S. A. Z. K. K. Max Jaderberg, "Spatial Transformer Networks," *arXiv:1506.02025*, 2016.
- [7] T. C. A. W. K. K. Wojciech Michal Matkowski, "Palmprint Recognition in Uncontrolled and Uncooperative Environment," *arXiv:1911.12514*, 2019.
- [8] A. Z. Karen Simonyan, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION," *ICLR*, 2015.
- [9] M. S. G. C. L.-C. C. B. C. M. T. W. W. Y. Z. R. P. V. V. Q. V. L. H. A. Andrew Howard, "Searching for MobileNetV3," *arXiv:1905.02244*, 2019.
- [10] A. T. T. K. L. S. I. S. S. Yair Movshovitz-Attias, "No Fuss Distance Metric Learning using Proxies," *arXiv:1703.07464*, 2017.
- [11] T. T. Trức, "Optimizer- Hiểu sâu về các thuật toán tối ưu," *Viblo*, 2020.
- [12] D.-S. H. D. Z. Wei Jia, "Palmprint verification based on robust line orientation code," *Pattern Recognition*, 2008.
- [13] L. W. T. T. Zhenan Sun, "Ordinal Feature Selection for Iris and Palmprint Recognition," *IEEE Transactions on Image Processing*, 2014.