

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**

ĐỒ ÁN TỐT NGHIỆP

**NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG HỖ TRỢ DỰ ĐOÁN BỆNH DA LIỄU CHO NGƯỜI VIỆT NAM

**Người hướng dẫn: TS. LÊ THỊ MỸ HẠNH
Sinh viên thực hiện: ĐỖ MINH NHẬT
Số thẻ sinh viên: 102150055
Lớp: 15T1**

Đà Nẵng, 12/2019

[illegible]

TS. Lê Thị Mỹ Hạnh

[illegible]

Đà Nẵng, ngày ... tháng ... năm 2019

PHIẾU DUYỆT ĐỒ ÁN TỐT NGHIỆP

I. Phần dành cho Sinh viên

1. *Họ và tên:* ĐỖ MINH NHẬT 2. *Mã Sinh viên:* 102150055 3. *Lớp:* 15T1
4. *Tên đề tài:* Xây dựng ứng dụng hỗ trợ dự đoán bệnh da liễu cho người Việt Nam
5. *Số điện thoại:* 0766800577 6. *E-mail:* dominhnhat2311@gmail.com
7. *Họ và tên GVHD:* TS. Lê Thị Mỹ Hạnh

II. Phần dành cho Hội đồng

STT	Nội dung đánh giá	Kết luận
1.	<i>Trình bày báo cáo theo đúng mẫu qui định của Khoa</i>	
2.	<i>Không có sự sao chép nội dung báo cáo và chương trình đã có</i>	
3.	Biên dịch mã nguồn và chạy được chương trình	
4.	Có kịch bản thực hiện với dữ liệu thử nghiệm	
5.	Kết quả thực hiện chương trình đúng theo báo cáo	
6.	Có sự đóng góp, phát triển của tác giả trong đồ án	

Ý kiến khác:

Kết luận: ☐ Đạt yêu cầu ☐ Phải sửa chữa lại ☐ Không đạt yêu cầu

Đà Nẵng, ngày tháng 12 năm 2019

Chủ tịch Hội đồng
(Ký và ghi họ tên)

Cán bộ duyệt kiểm tra
(Ký và ghi họ tên)

TÓM TẮT

Tên đề tài: Xây dựng ứng dụng hỗ trợ dự đoán bệnh da liễu cho người Việt Nam

Sinh viên thực hiện: Đỗ Minh Nhật

Số thẻ SV: 102150055 Lớp: 15T1

Ứng dụng hỗ trợ dự đoán bệnh da liễu cho người Việt Nam là một ứng dụng di động giúp hỗ trợ dự đoán một số bệnh da liễu thường gặp ở người Việt Nam và cung cấp nhiều kiến thức về những bệnh da liễu. Ngoài ra, đây cũng là một cách thức để tác động vào sự chủ động của người dùng trong việc khám chữa bệnh và phòng tránh bệnh. Ứng dụng sẽ hỗ trợ người dùng trong việc dự đoán vấn đề da liễu mà người đó đang gặp phải cũng như đưa ra các cách phòng ngừa.

Ngoài các mục đích hỗ trợ cho việc dự đoán bệnh thì ứng dụng cũng sẽ xây dựng một tập các dữ liệu thông qua dữ liệu được xây dựng trước cũng như dữ liệu trong quá trình sử dụng của người dùng.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: ĐỖ MINH NHẬT

Số thẻ sinh viên: 102150055

Lớp: 15T1

Khoa: Công nghệ thông tin

Ngành: Công nghệ phần mềm

1. Tên đề tài đồ án:

Xây dựng ứng dụng hỗ trợ dự đoán bệnh da liễu cho người Việt Nam

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

Không có.

4. Nội dung các phần thuyết minh và tính toán:

Nội dung của thuyết minh gồm:

Mở đầu: Phần mở đầu của luận văn, giới thiệu về nhu cầu thực tế và lý do thực hiện đề tài, đồng thời giới thiệu sơ lược về đề tài và mục tiêu phải đạt được.

Chương 1. Tổng quan về đề tài: trình bày về đề tài, bối cảnh đề tài cũng như hoàn cảnh lựa chọn đề tài.

Chương 2. Cơ sở lý thuyết: trình bày những lý thuyết học được và đã áp dụng vào hệ thống.

Chương 3. Phân tích và thiết kế: trình bày các hồ sơ phân tích và hồ sơ thiết kế trong xây dựng hệ thống và luồng hoạt động của hệ thống.

Chương 4. Triển khai và đánh giá kết quả: mô tả cách cài đặt, vận hành hệ thống và đánh giá kết quả đạt được.

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

Không có.

6. Họ tên người hướng dẫn: TS. LÊ THỊ MỸ HẠNH

7. Ngày giao nhiệm vụ đồ án:/...../ 2019

8. Ngày hoàn thành đồ án:/...../ 2019

Đà Nẵng, ngày tháng 12 năm 2019

Trưởng Bộ môn

Người hướng dẫn

Lê Thị Mỹ Hạnh

LỜI MỞ ĐẦU

Lời đầu tiên, tôi muốn gửi những lời cảm ơn và biết ơn chân thành nhất của mình tới tất cả những người đã hỗ trợ, giúp đỡ chúng tôi về kiến thức và tinh thần trong quá trình thực hiện đồ án.

Trước hết tôi xin chân thành cảm ơn TS. Lê Thị Mỹ Hạnh, Giảng viên Khoa Công Nghệ Thông Tin, Trường Đại Học Bách Khoa – Đại Học Đà Nẵng, người đã trực tiếp hướng dẫn, nhận xét, giúp đỡ tôi trong suốt quá trình thực hiện đồ án.

Xin chân thành cảm ơn ban giám hiệu nhà trường, các thầy cô trong Khoa Công Nghệ Thông Tin và các phòng ban nhà trường đã tạo điều kiện tốt nhất cho tôi trong suốt thời gian làm đồ án.

Với đề tài thiết thực và hữu ích cho xã hội, tôi đã làm việc một cách đầy nỗ lực với mục tiêu sản phẩm được đến với người dùng một cách dễ dàng và đầy sự hấp dẫn.

Tuy nhiên với vốn kiến thức hạn hẹp, và thời gian làm việc có hạn nên tôi không tránh khỏi những thiếu sót. Tôi rất mong nhận được những ý kiến, đóng góp, phê bình của Quý thầy cô để sản phẩm được hoàn thiện hơn. Đó sẽ là hành trang quý giá để tôi tiếp tục hoàn thiện kiến thức của mình sau này.

Tôi xin chân thành cảm ơn.

CAM ĐOAN

Tôi xin cam đoan:

Tôi hiểu tính chất của đạo, copy và tôi hiểu chính sách của trường với việc đó như thế nào. Tôi xin cam kết rằng:

1. Những nội dung của phần Đồ án tốt nghiệp này được thực hiện bởi chính tôi và được sự hướng dẫn của giảng viên hướng dẫn TS. Lê Thị Mỹ Hạnh;
2. Mọi tài liệu tham khảo, được dùng trong Đồ Án Tốt Nghiệp này, đều được trích dẫn với đầy đủ tên của tác giả, tên của bào báo/nghiên cứu và được phát hành một cách công khai và trung thực;
3. Mọi sự sao chép không hợp lệ, vi phạm quy chế và gian lận sẽ do chính tôi chịu trách nhiệm hoàn toàn

Nếu có những sao chép không hợp lệ, vi phạm quy chế đào tạo, tôi xin chịu hoàn toàn trách nhiệm.

Đà Nẵng, ngày tháng 12 năm 2019
Sinh viên thực hiện

Đỗ Minh Nhật

MỤC LỤC

LỜI MỞ ĐẦU	1
CAM ĐOAN	2
MỤC LỤC	3
DANH SÁCH CÁC BẢNG	6
DANH SÁCH HÌNH ẢNH	7
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT	8
MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	3
1.1. Bối cảnh đề tài	3
1.2. Các báo cáo liên quan đến đề tài	3
1.3. Mô tả kiến trúc ứng dụng	4
1.4. Kết chương	5
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	6
2.1. Học máy	6
2.1.1. Giới thiệu	6
2.1.2. Phân loại thuật toán học máy	6
2.1.3. Các bước học máy	7
2.2. Mạng nơ-ron nhân tạo	8
2.2.1. Giới thiệu	8
2.2.2. Kiến trúc chung	9
2.3. Học sâu	10
2.4. Mạng nơ-ron tích chập	10
2.4.1. Giới thiệu	10
2.4.2. Kiến trúc của mạng nơ-ron tích chập	11
2.5. Học chuyển đổi	14
2.6. Python	15
2.6.1. Giới thiệu chung	15
2.6.2. Đặc điểm ngôn ngữ Python	15
2.6.3. Flask python	16
2.7. Android	16
2.7.1. Giới thiệu	16
2.7.2. Cơ chế hoạt động	18
2.7.3. Thành phần	19
2.7.4. Môi trường phát triển	20

2.8 Kết chương	21
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	22
3.1. Mô tả hệ thống.....	22
3.2. Phân tích và thiết kế hệ thống	22
3.2.1. Phân tích đối tượng.....	22
3.2.2. Hệ thống sơ đồ hoạt động chức năng người dùng	23
3.3. Kế hoạch làm việc	24
3.4. Đánh giá dữ liệu huấn luyện.....	26
3.4.1. Thông tin tập dữ liệu huấn luyện.....	26
3.4.2. Quá trình xây dựng tập dữ liệu	27
3.5. Giải pháp đề xuất.....	28
3.5.1. Kỹ thuật Data Augmentation	28
3.5.2. Kỹ thuật Transfer Learning – Fine tuning	31
3.5.4. Mô hình ResNet.....	32
3.5.2. Mô hình ResNet152.....	34
3.5.3. Thông số mô hình lựa chọn	35
3.5.4. Hàm mất mát Negative Log-Likelihood.....	36
3.5.5. Hàm tối ưu Adam	37
3.6. Kết chương	38
CHƯƠNG 4: TRIỂN KHAI THỰC TẾ VÀ ĐÁNH GIÁ KẾT QUẢ	39
4.1. Triển khai huấn luyện mô hình.....	39
4.1.1. Môi trường huấn luyện	39
4.1.2. Framework sử dụng huấn luyện	39
4.1.3. Tổng quan quá trình huấn luyện mô hình và dự đoán kết quả	40
4.1.4. Thông số huấn luyện.....	40
4.1.3. Quá trình huấn luyện mô hình	41
4.1.4. Kiểm nghiệm mô hình	43
4.2. Triển khai hệ thống Back-End.....	43
4.2.1. Thư viện cài đặt	43
4.2.2. Danh sách API	45
4.3. Triển khai hệ thống Front-End	46
4.4. Kết quả sản phẩm	47
4.5. Kết luận và hướng phát triển	51
4.5.1. Đạt được.....	51
4.5.2. Chưa đạt.....	51
4.5.3. Hướng phát triển.....	52
4.6. Kết chương	52
TÀI LIỆU THAM KHẢO.....	53
PHỤ LỤC 1	54

DANH SÁCH CÁC BẢNG

Bảng 3.1 Bảng phân tích các tác nhân đối với hệ thống.....	22
Bảng 3.2 Bảng kế hoạch làm việc.....	24
Bảng 3.3 Bảng thông số các lớp tùy biến sau cùng của mô hình	36

DANH SÁCH HÌNH ẢNH

Hình 1.1 Sơ đồ kiến trúc ứng dụng.....	4
Hình 2.1 Mối liên hệ giữa trí tuệ nhân tạo với nhánh học máy học	6
Hình 2.2 Kiến trúc chung của một mạng nơ ron.....	9
Hình 2.3 Kiến trúc nơ ron có nhiều tầng ẩn.....	9
Hình 2.4 So sánh giữa học máy và học sâu	10
Hình 2.5 Kiến trúc của mạng nơ-ron tích chập.....	11
Hình 2.6 Ví dụ về lớp tích chập	11
Hình 2.7 Kết quả đầu ra của lớp tích chập.....	12
Hình 2.8 Ví dụ cho lớp lấy mẫu.....	14
Hình 2.9 Framework Flask python	16
Hình 2.10 Hệ điều hành Android.....	17
Hình 2.11 Cấu trúc của ứng dụng Android.....	18
Hình 2.12 Thành phần trong ứng dụng Android	19
Hình 3.1 Biểu đồ hoạt động của người dùng	23
Hình 3.2 Số lượng ảnh và nhãn trong tập dữ liệu	26
Hình 3.3 Cấu trúc của kỹ thuật Fine tuning.....	32
Hình 3.4 Cấu trúc của một Residual block cơ bản	32
Hình 3.5 Tổng quan kiến trúc của các mô hình ResNet	33
Hình 3.6 Khối ResNet cơ bản và khối ResNet cải tiến.....	34
Hình 3.7 Cấu trúc mạng ResNet 152	34
Hình 3.8 Đồ thị hàm mất mát negative log-likelihood	37
Hình 3.9 Quá trình tính toán giá trị mất mát.....	37
Hình 3.10 Đồ thị hàm số trong hàm tối ưu Adam	38
Hình 3.11 Công thức cập nhật giá trị của hàm tối ưu Adam	38
Hình 4.1 Tổng quan quá trình huấn luyện mô hình và dự đoán kết quả.....	40
Hình 4.2 Biểu đồ giá trị mất mát trong quá trình huấn luyện và xác nhận	41
Hình 4.3 Biểu đồ tỉ lệ chính xác trên tập xác nhận trong quá trình huấn luyện	42
Hình 4.4 Hình ảnh bộ dữ liệu được kiểm nghiệm	43
Hình 4.5 Tỉ lệ dự đoán trên tập kiểm nghiệm	43
Hình 4.6 Thư viện NumPy.....	44
Hình 4.7 Màn hình hướng dẫn sử dụng ứng dụng	47
Hình 4.8 Màn hình cấp phép quyền truy cập	48
Hình 4.9 Màn hình chính cho người sử dụng	48
Hình 4.10 Màn hình kết quả dự đoán	49
Hình 4.11 Màn hình tùy chọn tải dữ liệu và thông tin bệnh	50
Hình phụ lục 1.1 Dấu hiệu của bệnh hắc bào	54
Hình phụ lục 1.2 Dấu hiệu của bệnh lang ben	55
Hình phụ lục 1.3 Dấu hiệu của bệnh tổ đỉa.....	56
Hình phụ lục 1.4 Dấu hiệu của bệnh vảy nến	57
Hình phụ lục 1.5 Dấu hiệu của bệnh zona	57
Hình phụ lục 1.6 Dấu hiệu của bệnh mụn cóc	58

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	CNN	Convolutional Neural Network – Mạng nơ ron tích chập
2	ReLU	Rectified Linear Unit – Tính chỉnh đơn vị tuyến tính
3	GPU	Graphics Processing Unit – Bộ vi xử lý đồ họa
4	ANN	Artificial Neural network – Mạng nơ ron nhân tạo
5	AI	Artificial Intelligence – Trí tuệ nhân tạo
6	API	Application Programming Interface – giao diện lập trình ứng dụng

MỞ ĐẦU

1. Giới thiệu đề tài

Việc chẩn đoán các bệnh da liễu hiện nay ở Việt Nam vẫn đang được thực hiện theo cách truyền thống, chính vì thế có thể có rất nhiều yếu tố ảnh hưởng đến kết quả chẩn đoán. Việc xây dựng ứng dụng hỗ trợ dự đoán các bệnh da liễu ở người Việt Nam sẽ là hướng đi ứng trí tuệ nhân tạo vào quá trình dự đoán bệnh da liễu.

1.1. Mục đích

Một trong những mục đích của đề án là xây dựng được một ứng dụng đơn giản trên điện thoại hỗ trợ dự đoán bệnh da liễu cho người Việt Nam. Thêm vào đó, việc hỗ trợ dự đoán này có thể được ứng dụng trong các trung tâm y khoa với mục đích hỗ trợ cho các chẩn đoán của bác sỹ.

Ứng dụng cung cấp nhiều kiến thức về những bệnh da liễu thường gặp ở các nước Châu Á và đặc biệt là Việt Nam. Ngoài ra, đây cũng là một cách thức để tác động vào sự chủ động của người dùng trong việc khám chữa bệnh và phòng tránh bệnh.

1.2. Ý nghĩa

Ứng dụng là bước đầu trong việc ứng dụng trí tuệ nhân tạo cho việc nhận dạng và hỗ trợ dự đoán các bệnh trong y sinh nói chung và các bệnh da liễu nói riêng. Đưa bài toán nhận dạng bệnh da liễu vào quá trình nghiên cứu, phát triển trong tương lai

Việc xây dựng bộ dữ liệu về các bệnh da liễu thường gặp ở Việt Nam là một đóng góp mang tính tích cực trong việc phát triển và cải thiện hiệu quả trong việc khám chữa bệnh. Ngoài ra còn là một sự bổ sung dữ liệu cần thiết cho việc nghiên cứu cũng như học tập.

2. Phạm vi của đề tài

Để xác định một bệnh da liễu cần rất nhiều yếu tố về hình dáng, màu sắc, tiền sử bệnh lý, quá trình hình thành bệnh,... Và triệu chứng cũng như đặc điểm các giai đoạn phát triển bệnh của mỗi nơi trên thể giới lại khác nhau tùy vào điều kiện thời tiết, thổ nhưỡng. Chính vì thế trong phạm vi thời gian cho phép của đề án thì phạm vi các bệnh được hỗ trợ dự đoán sẽ là các bệnh thường gặp ở người Việt Nam.

Ứng dụng là bước đầu trong việc áp dụng trí tuệ nhân tạo vào việc nhận dạng các bệnh da liễu, tuy nhiên có rất nhiều yếu tố ảnh hưởng đến việc dự đoán bệnh. Do

đó trong phạm vi của ứng dụng chỉ chọn ra hai yếu tố là hình dạng và màu sắc để làm hai yếu tố chính để nhận dạng bệnh da liễu thông qua ứng dụng.

Dữ liệu đầu vào dành cho ứng dụng bao gồm ảnh người dùng chụp, và video quay lại bệnh của người sử dụng. Việc cho phép dự đoán trên nhiều ảnh hay video sẽ giúp tăng độ chính xác của việc nhận diện bệnh.

3. Phương pháp thực hiện

- Phương pháp học sâu;
- Phương pháp phân tích tổng hợp từ nguồn tài liệu và dữ liệu trên nhiều nguồn khác nhau(sách, tạp chí, các trang blog, diễn đàn,...);
- Phương pháp phân tích thiết kế hệ thống theo hướng đối tượng;
- Phương pháp thử nghiệm, đánh giá kết quả.

4. Bố cục của đề án

Đề án bao gồm các nội dung sau:

Mở đầu

Chương 1: TỔNG QUAN VỀ ĐỀ TÀI

Chương 2: CƠ SỞ LÝ THUYẾT

Chương 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Chương 3: TRIỂN KHAI THỰC TẾ VÀ ĐÁNH GIÁ KẾT QUẢ

Kết luận và hướng phát triển.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Bối cảnh đề tài

Trong những năm gần đây, chủ đề sức khỏe con người luôn được quan tâm và cải thiện không ngừng. Việc khám và chữa bệnh trong nước hiện nay đang được nâng cao chất lượng hằng ngày để đảm bảo nhu cầu về bảo vệ sức khỏe của con người. Tuy nhiên, vẫn còn rất nhiều hạn chế như thủ tục khám bệnh quá phức tạp và tốn nhiều thời gian, ý thức của con người trong việc phòng ngừa bệnh cũng như tự giác khám chữa bệnh vẫn còn chưa cao...

Đặc biệt một trong những loại bệnh thường gặp nhất ở người Việt Nam chính là bệnh da liễu. Với việc ô nhiễm môi trường sống hiện nay, tiếp xúc với nhiều chất độc hại trong công việc cũng như sinh hoạt hằng ngày, nguy cơ mắc phải bệnh da liễu là rất cao. Khi hậu nhiệt đới ở khu vực Việt Nam là một môi trường lý tưởng để việc các mầm bệnh phát triển một cách nhanh chóng. Các loại bệnh da liễu rất dễ nhầm lẫn với nhau vì có triệu chứng khá giống nhau về mặt hình dạng và màu sắc. Vì vậy để phân loại bệnh da liễu thì cần rất nhiều yếu tố và quá trình theo dõi sát sao.

Trên thế giới ngày nay việc áp dụng trí tuệ nhân tạo vào việc hỗ trợ chẩn đoán đang được thực hiện một cách rộng rãi. Như đã biết, lỗi chẩn đoán trong ngành y là một mối đe dọa nghiêm trọng đối với chất lượng và an toàn trong chăm sóc sức khỏe. Tại Mỹ, ước tính tỷ lệ lỗi chẩn đoán ngoại trú là 5,08% tương đương 12 triệu người mỗi năm. Khoảng một nửa trong số các lỗi này là có khả năng gây hại. Công nghệ AI đã được sử dụng để cải thiện chất lượng chẩn đoán. AI dựa trên nguồn dữ liệu 129.450 hình ảnh lâm sàng để chẩn đoán bệnh ngoài da, kết quả đã chứng minh rằng hệ thống này có thể phân loại ung thư da ở mức tương đương với các bác sĩ da liễu. Với việc có thể phân loại ung thư da thì một bài toán tương đương là phân loại các bệnh da liễu thường gặp ở người có thể được giải quyết bởi trí tuệ nhân tạo là điều hoàn toàn có thể xảy ra.

1.2. Các báo cáo liên quan đến đề tài

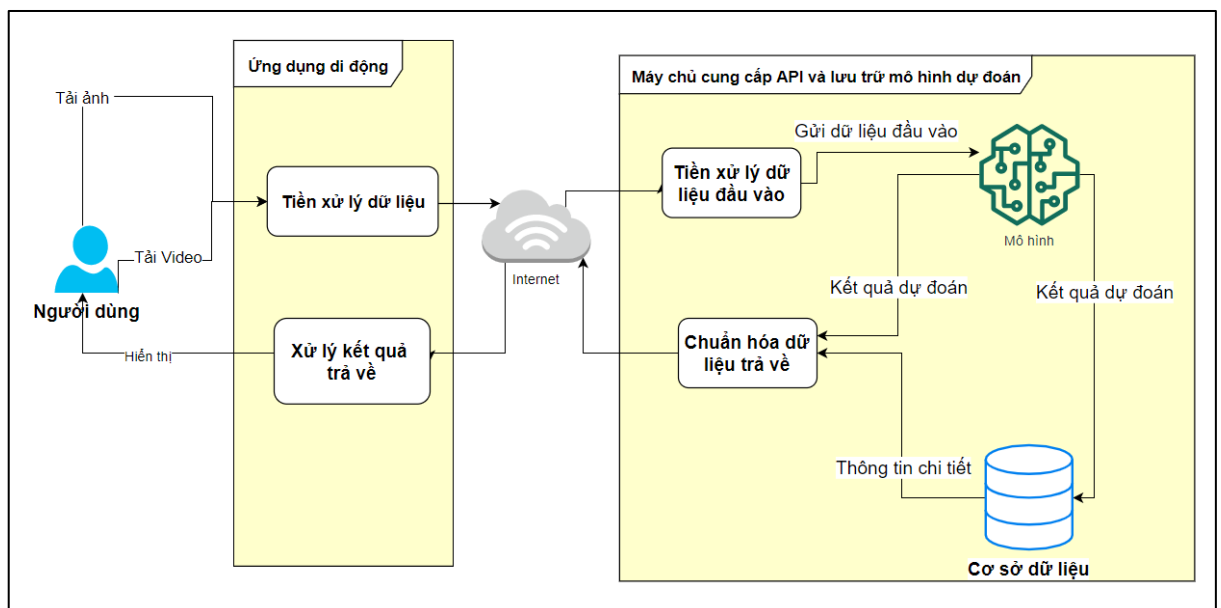
Theo nghiên cứu của Giáo sư Alastair Denniston thuộc Đại học Hospitals Birmingham NHS Foundation Trust nằm trong báo cáo [13], AI có thể chẩn đoán hiệu quả bệnh ung thư và các bệnh khác như một chuyên gia y tế. Nghiên cứu đã kiểm tra dữ liệu từ 14 thử nghiệm, cho thấy AI đã phát hiện chính xác tới 87% các trường hợp, trong khi con số này của các bác sĩ chỉ đạt 86%. AI cũng loại trừ những

bệnh nhân không có bệnh với tỷ lệ chính xác lên đến 93%, nhỉnh hơn một chút so với chuyên gia y tế (91%).

Ngoài ra, bài báo khoa học [9] của Li-sheng Wei, Quan Gan, and Tao Ji được xuất bản trên Computational and Mathematical Methods in Medicine đã cho thấy kết quả nghiên cứu về việc sử dụng trí tuệ nhân tạo vào việc trích xuất đặc trưng và nhận dạng ba loại bệnh ngoài da đó là mụn cóc sinh dục (herpes), viêm da và bệnh vẩy nến da. Phương pháp đề ra như sau: Ban đầu, hình ảnh da được xử lý trước để loại bỏ nhiễu và nền không liên quan bằng cách lọc và biến đổi. Sau đó, phương pháp đồng cấp ma trận xuất hiện mức xám (GLCM) đã được sử dụng để phân đoạn hình ảnh của bệnh da. Thông qua đó, kết cấu hình dạng và đặc điểm màu sắc của các hình ảnh bệnh da khác nhau có thể được lấy chính xác. Cuối cùng, bằng cách sử dụng phương pháp phân loại máy vector hỗ trợ (SVM), ba loại bệnh ngoài da đã được xác định. Kết quả thí nghiệm điện tử chứng minh tính hiệu quả và tính khả thi của phương pháp được đề xuất.

Có rất nhiều đề xuất giải pháp trong việc giải quyết bài toán nhận dạng bệnh da liễu mắc phải trên thế giới. Chính vì thế việc phát triển ứng dụng hỗ trợ dự đoán bệnh da liễu thường gặp ở người Việt Nam cũng là một giải pháp được đề xuất cho việc giải quyết bài toán này.

1.3. Mô tả kiến trúc ứng dụng



Hình 1.1 Sơ đồ kiến trúc ứng dụng

Hình 1.1 mô tả các hoạt động chính của người dùng trên ứng dụng di động và hoạt động của máy chủ. Hệ thống của ứng dụng sẽ thực hiện các công việc sau:

- Ứng dụng di động thu thập dữ liệu người dùng đưa vào
- Xử lý và truyền yêu cầu đến máy chủ chứa API
- Máy chủ nhận yêu cầu từ ứng dụng di động
- Chuẩn hóa dữ liệu đầu vào và đầu ra
- Mô hình đã huấn luyện dự đoán bệnh qua dữ liệu ảnh hoặc video người dùng tải lên
- Lấy thông tin từ cơ sở dữ liệu và trả về người dùng
- Ứng dụng di động nhận thông tin kết quả trả về và hiển thị

Mô hình dự đoán sẽ được lưu trữ trên máy chủ. Việc xử lý dữ liệu được gửi lên từ người dùng, đưa ra dự đoán và xử lý trả về dữ liệu kết quả cho người dùng sẽ được thực hiện trên máy chủ.

Ứng dụng di động có vai trò hiển thị cũng như xử lý các thao tác người dùng để gửi lên máy chủ dữ liệu được tải lên bởi người sử dụng.

Kiến trúc ứng dụng gồm 3 phần:

- Ứng dụng di động
- Máy chủ cung cấp API
- Mô hình dự đoán

1.4. Kết chương

Chương 1 cung cấp cho người đọc thực trạng vấn đề khám chữa bệnh và phòng bệnh da liễu của người Việt Nam hiện nay và việc áp dụng khoa học công nghệ, trí tuệ nhân tạo vào việc nhận dạng các bệnh da liễu để hỗ trợ cho y học. Do đó việc xây dựng ứng dụng hỗ trợ dự đoán bệnh da liễu cho người Việt Nam là bước đầu trong việc ứng dụng trí tuệ nhân tạo vào trong nghiên cứu và hỗ trợ chẩn đoán trong y học. Ngoài ra chương 1 cho người đọc thấy được phạm vi của ứng dụng này và mục tiêu cần đạt được khi xây dựng ứng dụng hỗ trợ dự đoán bệnh da liễu cho người Việt Nam.

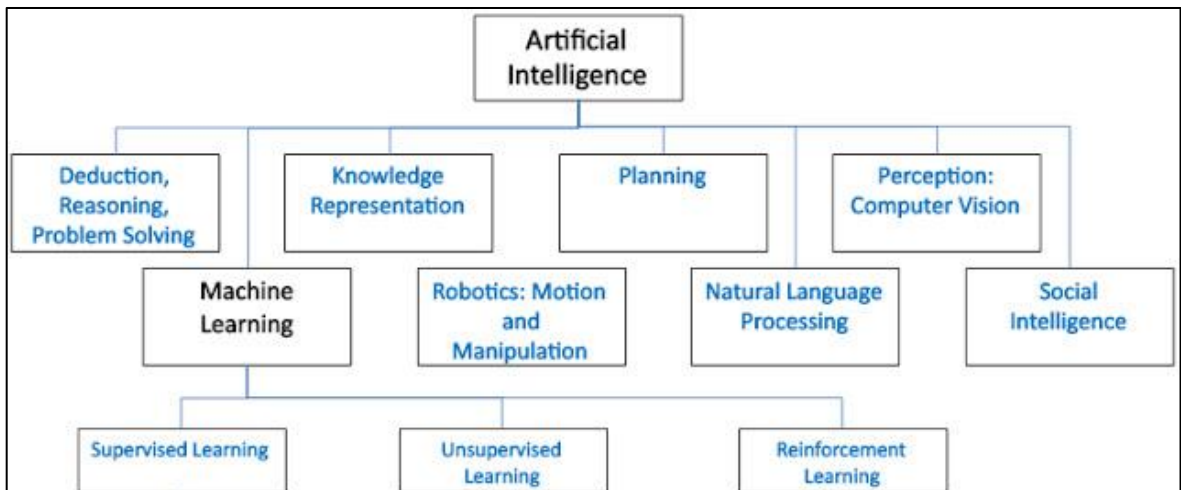
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Học máy

2.1.1. Giới thiệu

Học máy, có tài liệu gọi là Máy học, (tiếng Anh: machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc phát triển các kỹ thuật cho phép các máy tính có thể "học". Cụ thể hơn, học máy là một phương pháp để tạo ra các chương trình máy tính bằng việc phân tích các tập dữ liệu. Học máy có liên quan lớn đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán.

Học máy có tính ứng dụng rất cao bao gồm máy truy tìm dữ liệu, chẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, phân loại các chuỗi DNA, nhận dạng tiếng nói và chữ viết, dịch tự động, chơi trò chơi và cử động rô-bốt (robot locomotion).



Hình 2.1 Mối liên hệ giữa trí tuệ nhân tạo với nhánh học máy học

2.1.2. Phân loại thuật toán học máy

Các thuật toán học máy được phân loại theo kết quả mong muốn của thuật toán. Các loại thuật toán thường dùng bao gồm:

- Học có giám sát: Trong đó, thuật toán tạo ra một hàm ánh xạ dữ liệu vào tới kết quả mong muốn. Một phát biểu chuẩn về một việc học có giám sát là bài toán phân loại: chương trình cần học (cách xấp xỉ biểu hiện của) một hàm ánh xạ một vector $[X_1, X_2, \dots, X_n]$ tới một vài lớp bằng cách xem xét một số mẫu dữ liệu - kết quả của hàm đó.

- Học không giám sát: Mô hình hóa một tập dữ liệu, không có sẵn các ví dụ đã được gắn nhãn.
- Học nửa giám sát: Kết hợp các ví dụ có gắn nhãn và không gắn nhãn để sinh một hàm hoặc một bộ phân loại thích hợp.
- Học tăng cường: trong đó, thuật toán học một chính sách hành động tùy theo các quan sát về thế giới. Mỗi hành động đều có tác động tới môi trường, và môi trường cung cấp thông tin phản hồi để hướng dẫn cho thuật toán của quá trình học.
- Học chuyển đổi: tương tự học có giám sát nhưng không xây dựng hàm một cách rõ ràng. Thay vì thế, cố gắng đoán kết quả mới dựa vào các dữ liệu huấn luyện, kết quả huấn luyện, và dữ liệu thử nghiệm có sẵn trong quá trình huấn luyện.
- Học cách học: trong đó thuật toán học thiên kiến quy nạp của chính mình, dựa theo các kinh nghiệm đã gặp.

2.1.3. Các bước học máy

Một bài toán học máy cần trải qua 3 bước chính:

- Chọn mô hình: Chọn một mô hình thống kê cho tập dữ liệu.
- Tìm tham số: Các mô hình thống kê có các tham số tương ứng, nhiệm vụ lúc này là tìm các tham số này sao cho phù hợp với tập dữ liệu nhất có thể.
- Suy luận: Sau khi có được mô hình và tham số, có thể dựa vào chúng để đưa ra kết quả cho một đầu vào mới nào đó.

Bất cứ một bài toán học máy nào cũng đều cần có dữ liệu để huấn luyện, ta có thể coi nó là điều kiện tiên quyết. Dữ liệu sau khi có được cần phải:

- Chuẩn hoá: Tất cả các dữ liệu đầu vào đều cần được chuẩn hoá để máy tính có thể xử lý được. Quá trình chuẩn hoá bao gồm số hoá dữ liệu, co giãn thông số cho phù hợp với bài toán. Việc chuẩn hoá này ảnh hưởng trực tiếp tới tốc độ huấn luyện cũng như cả hiệu quả huấn luyện. Cụ thể ra sao thì ta sẽ cùng thảo luận trong một bài viết khác.
- Phân chia: Việc mô hình được chọn rất khớp với tập dữ liệu đang có không có nghĩa là giả thuyết của ta là đúng mà có thể xảy ra tình huống dữ liệu thật lại không khớp. Vấn đề này trong học máy được gọi là khớp quá (*Overfitting*). Vì vậy khi

huấn luyện người ta phải phân chia dữ liệu ra thành 3 loại để có thể kiểm chứng được phần nào mức độ tổng quát của mô hình. Cụ thể 3 loại đó là:

- Tập huấn luyện (Training set): Dùng để học khi huấn luyện.
- Tập kiểm chứng (Cross validation set): Dùng để kiểm chứng mô hình khi huấn luyện.
- Tập kiểm tra (Test set): Dùng để kiểm tra xem mô hình đã phù hợp chưa sau khi huấn luyện.

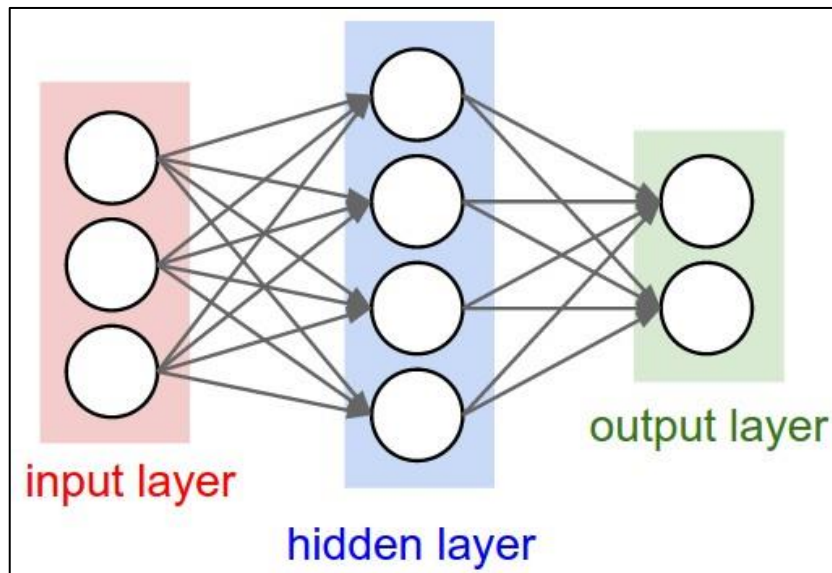
2.2. Mạng nơ-ron nhân tạo

2.2.1. Giới thiệu

Mạng nơ-ron nhân tạo hay thường gọi ngắn gọn là mạng nơ-ron (tiếng Anh là Artificial Neural network - ANN hay Neural Network) là một mô hình toán học hay mô hình tính toán được xây dựng dựa trên các mạng neural sinh học. Nó gồm có một nhóm các neural nhân tạo (nút) nối với nhau, và xử lý thông tin bằng cách truyền theo các kết nối và tính giá trị mới tại các nút (cách tiếp cận connectionism đối với tính toán). Trong nhiều trường hợp, mạng neural nhân tạo là một hệ thống thích ứng (adaptive system) tự thay đổi cấu trúc của mình dựa trên các thông tin bên ngoài hay bên trong chảy qua mạng trong quá trình học.

Trong thực tế sử dụng, nhiều mạng neural là các công cụ mô hình hóa dữ liệu thống kê phi tuyến. Chúng có thể được dùng để mô hình hóa các mối quan hệ phức tạp giữa dữ liệu vào và kết quả hoặc để tìm kiếm các dạng/mẫu trong dữ liệu.

2.2.2. Kiến trúc chung

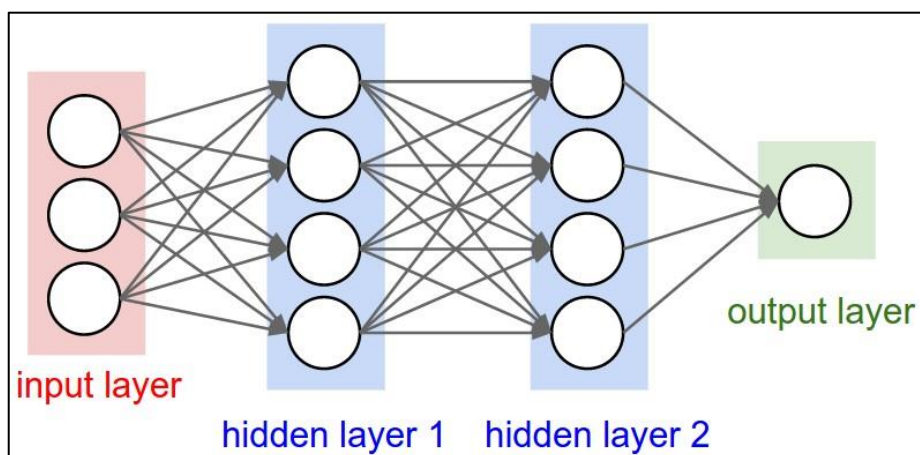


Hình 2.2 Kiến trúc chung của một mạng nơ ron

Một mạng nơ-ron sẽ có 3 kiểu tầng:

- Tầng vào (*input layer*): Là tầng bên trái cùng của mạng thể hiện cho các đầu vào của mạng.
- Tầng ra (*output layer*): Là tầng bên phải cùng của mạng thể hiện cho các đầu ra của mạng.
- Tầng ẩn (*hidden layer*): Là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng.

Một nơ-ron chỉ có 1 tầng vào và 1 tầng ra nhưng có thể có nhiều tầng ẩn.

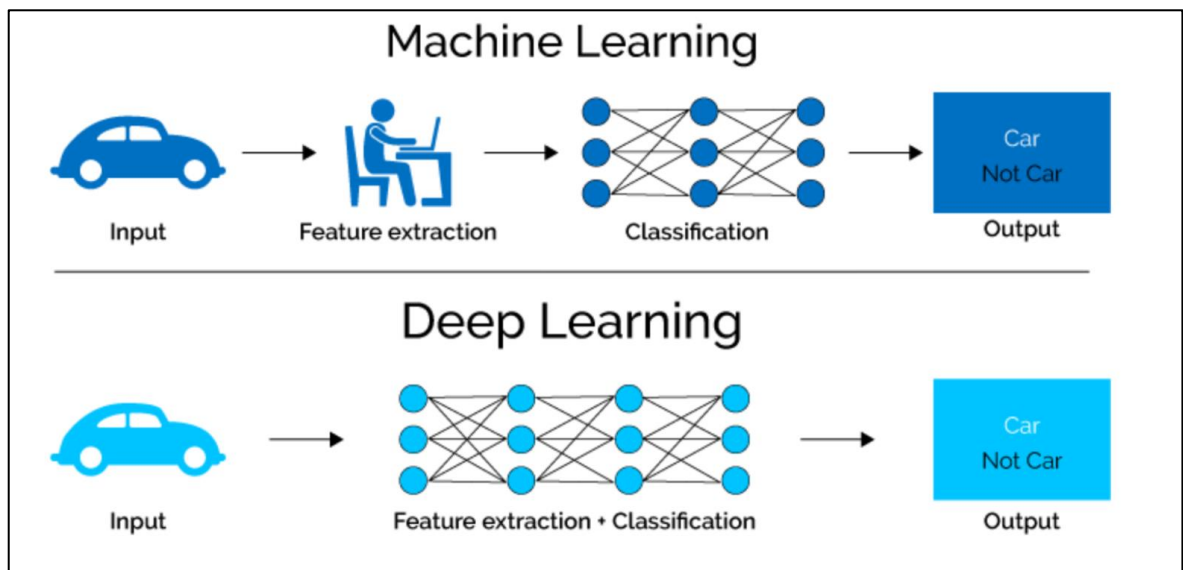


Hình 2.3 Kiến trúc nơ ron có nhiều tầng ẩn

2.3. Học sâu

Học sâu (tiếng Anh: *deep learning*) là một chi của ngành máy học dựa trên một tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, hoặc bằng cách khác bao gồm nhiều biến đổi phi tuyến.

Các thuật toán học sâu dựa trên các đại diện phân phối. Giả định tiềm ẩn đằng sau các đại diện phân phối là các dữ liệu được quan sát là được tạo ra bởi sự tương tác của các yếu tố được tổ chức theo lớp. Học sâu thêm giả định rằng các lớp của các yếu tố này tương ứng với các mức độ trừu tượng hay theo thành phần. Các con số khác nhau của các lớp và kích thước của lớp có thể được sử dụng để quy định các lượng trừu tượng khác.



Hình 2.4 So sánh giữa học máy và học sâu

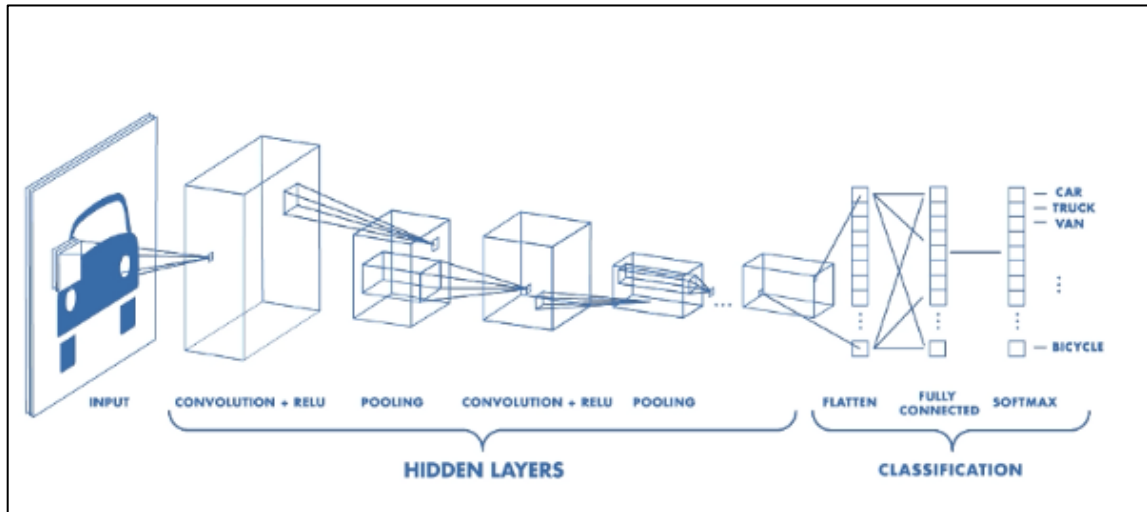
2.4. Mạng nơ-ron tích chập

2.4.1. Giới thiệu

Mạng nơ-ron tích chập (CNNs – Convolutional Neural Network) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Như hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động.

CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. Để tìm hiểu tại sao thuật toán này được sử dụng rộng rãi cho việc nhận dạng (detection), chúng ta hãy cùng tìm hiểu về thuật toán này.

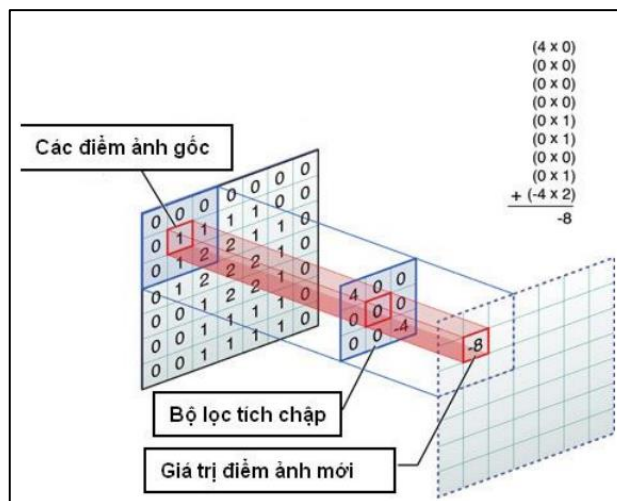
2.4.2. Kiến trúc của mạng nơ-ron tích chập



Hình 2.5 Kiến trúc của mạng nơ-ron tích chập

Các lớp cơ bản trong một mạng CNN bao gồm: Lớp tích chập (Convolutional), Lớp kích hoạt phi tuyến ReLU (Rectified Linear Unit), Lớp lấy mẫu (Pooling) và Lớp kết nối đầy đủ (Fully-connected), được thay đổi về số lượng và cách sắp xếp để tạo ra các mô hình huấn luyện phù hợp cho từng bài toán khác nhau.

- Lớp tích chập: Đây là thành phần quan trọng nhất trong mạng CNN, cũng là nơi thể hiện tư tưởng xây dựng sự liên kết cục bộ thay vì kết nối toàn bộ các điểm ảnh. Các liên kết cục bộ này được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc – filters – có kích thước nhỏ.



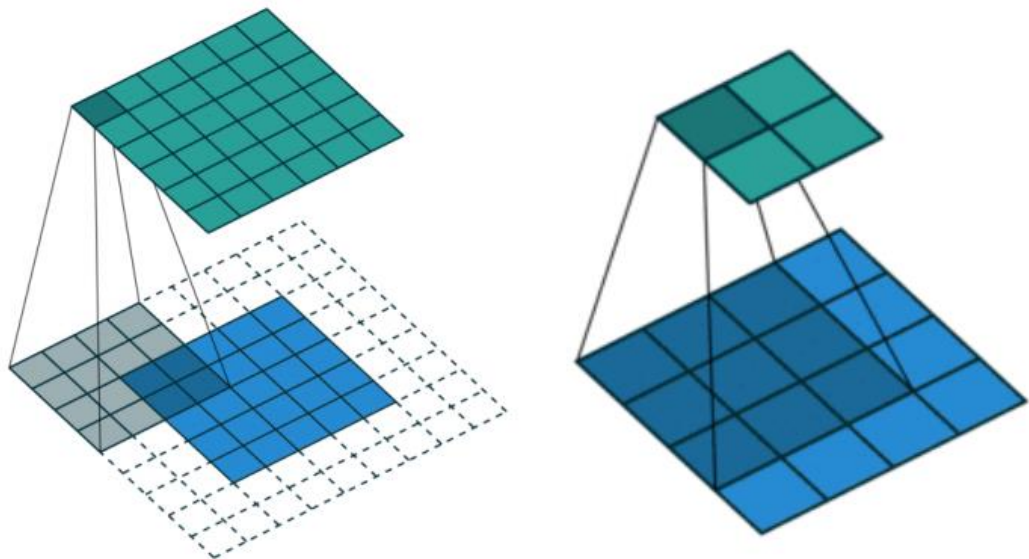
Hình 2.6 Ví dụ về lớp tích chập

Trong ví dụ ở trên, ta thấy bộ lọc được sử dụng là một ma trận có kích thước 3x3. Bộ lọc này được dịch chuyển lần lượt qua từng vùng ảnh đến khi hoàn thành quét toàn bộ bức ảnh, tạo ra một bức ảnh mới có kích thước nhỏ hơn hoặc bằng với kích thước ảnh đầu vào. Kích thước này được quyết định tùy theo kích thước các khoảng trắng được thêm ở viền bức ảnh gốc và được tính theo công thức:

$$o = i + 2 * p - k * s + 1$$

Trong đó:

- o: kích thước ảnh đầu ra
- i: kích thước ảnh đầu vào
- p: kích thước khoảng trắng phía ngoài viền của ảnh gốc
- k: kích thước bộ lọc
- s: bước trượt của bộ lọc



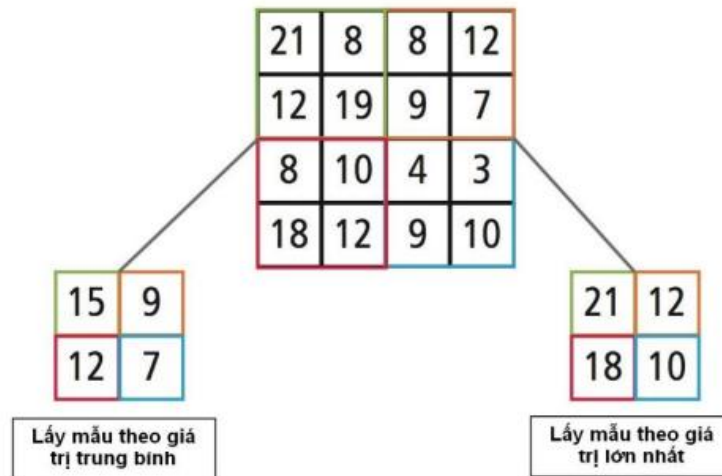
Hình 2.7 Kết quả đầu ra của lớp tích chập

Như vậy, sau khi đưa một bức ảnh đầu vào cho lớp Tích chập ta nhận được kết quả đầu ra là một loạt ảnh tương ứng với các bộ lọc đã được sử dụng để thực hiện phép tích chập. Các trọng số của các bộ lọc này được khởi tạo ngẫu nhiên trong lần đầu tiên và sẽ được cải thiện dần xuyên suốt quá trình huấn luyện.

- kích hoạt phi tuyến ReLU:
 - Lớp này được xây dựng với ý nghĩa đảm bảo tính phi tuyến của mô hình huấn luyện sau khi đã thực hiện một loạt các phép tính toán tuyến tính qua các lớp Tích chập. Lớp Kích hoạt phi tuyến nói chung sử dụng các hàm kích hoạt phi tuyến như ReLU hoặc sigmoid, tanh... để giới hạn phạm vi biên độ cho phép của giá trị đầu ra. Trong số các hàm kích hoạt này, hàm ReLU được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả. Cụ thể, phép tính toán của hàm ReLU chỉ đơn giản là chuyển tất cả các giá trị âm thành giá trị 0.

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

- Thông thường, lớp ReLU được áp dụng ngay phía sau lớp Tích chập, với đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh cũng hoàn toàn tương tự trừ các giá trị âm đã bị loại bỏ.
- Lớp lấy mẫu:
 - Một thành phần tính toán chính khác trong mạng CNN là lấy mẫu (Pooling), thường được đặt sau lớp Tích chập và lớp ReLU để làm giảm kích thước kích thước ảnh đầu ra trong khi vẫn giữ được các thông tin quan trọng của ảnh đầu vào. Việc giảm kích thước dữ liệu có tác dụng làm giảm được số lượng tham số cũng như tăng hiệu quả tính toán. Lớp lấy mẫu cũng sử dụng một cửa sổ trượt để quét toàn bộ các vùng trong ảnh tương tự như lớp Tích chập, và thực hiện phép lấy mẫu thay vì phép tích chập – tức là ta sẽ chọn lưu lại một giá trị duy nhất đại diện cho toàn bộ thông tin của vùng ảnh đó.



Hình 2.8 Ví dụ cho lớp lấy mẫu

Như vậy, với mỗi ảnh đầu vào được đưa qua lấy mẫu ta thu được một ảnh đầu ra tương ứng, có kích thước giảm xuống đáng kể nhưng vẫn giữ được các đặc trưng cần thiết cho quá trình tính toán sau này.

- Lớp kết nối đầy đủ: Lớp kết nối đầy đủ này được thiết kế hoàn toàn tương tự như trong mạng nơ-ron truyền thống, tức là tất cả các điểm ảnh được kết nối đầy đủ với node trong lớp tiếp theo. So với mạng nơ-ron truyền thống, các ảnh đầu vào của lớp này đã có kích thước được giảm bớt rất nhiều, đồng thời vẫn đảm bảo các thông tin quan trọng cho việc nhận dạng. Do vậy, việc tính toán nhận dạng sử dụng mô hình truyền thẳng đã không còn phức tạp và tốn nhiều thời gian như trong mạng nơ-ron truyền thống.

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

Mạng CNN sử dụng 3 ý tưởng cơ bản:

- Các trường tiếp nhận cục bộ (local receptive field)
- Trọng số chia sẻ (shared weights)
- Tổng hợp (pooling).

2.5. Học chuyển đổi

Trong thời gian đầu khi các phương pháp Học sâu mới đạt được nhiều thành tựu và được áp dụng phổ biến, trong cộng đồng Học sâu trên thế giới đã tồn tại một quan niệm không chính xác nhưng hết sức phổ biến. Nói chính xác hơn, đây đã từng là một quan niệm đúng và hợp lý, bởi mỗi mô hình huấn luyện này đều sử dụng rất nhiều các lớp ẩn, với hàng nghìn nơ-ron và hàng triệu tham số. Đồng thời quá trình

huấn luyện mô hình cũng được gắn liền với các kiến thức riêng và bài toán phân tích, nhận dạng... cụ thể, và nếu cố gắng áp dụng mô hình đó với một CSDL khác, chắc chắn độ chính xác sẽ bị suy giảm đáng kể. Tuy nhiên, trong thời gian sau đó, một phương pháp học mới được đưa ra và đã giải quyết được điểm hạn chế này của Học sâu, đó chính là Học chuyển đổi – Transfer Learning.

Học chuyển đổi là quá trình khai thác, tái sử dụng các tri thức đã được học tập bởi một mô hình huấn luyện trước đó vào giải quyết một bài toán mới mà không phải xây dựng một mô hình huấn luyện khác từ đầu. Đây được coi là một trong những kỹ thuật được xếp mức độ quan trọng hàng đầu trong cộng đồng khoa học dữ liệu, nhằm hướng tới mục đích chung là phát minh ra một thuật toán học tự động mạnh mẽ.

Học chuyển đổi là một phương pháp học máy mà trong đó, một mô hình đã được phát triển cho một tác vụ (pretrained network) được tái sử dụng ở một tác vụ khác. Phương pháp này mang đến một cách tiếp cận phổ biến và hiệu quả cao trong deep learning khi bạn có một tập dữ liệu vừa và nhỏ.

Pretrained network là một mạng đã lưu trước đó, được đào tạo trên một tập dữ liệu lớn (thường là trên một task phân loại ảnh quy mô lớn). Một số mạng pretrained thường được sử dụng như: VGG, ResNet, Inception, Inception-ResNet, Xception....

2.6. Python

2.6.1. Giới thiệu chung

Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level) ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.

2.6.2. Đặc điểm ngôn ngữ Python

- Ngữ pháp đơn giản, dễ đọc.
- Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented)
- Hỗ trợ module và hỗ trợ gói (package)
- Xử lý lỗi bằng ngoại lệ (Exception)

- Kiểu dữ liệu động ở mức cao.
- Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
- Có khả năng tương tác với các module khác viết trên C/C++.
- Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

2.6.3. Flask python

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các Api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.



Hình 2.9 Framework Flask python

Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

2.7. Android

2.7.1. Giới thiệu

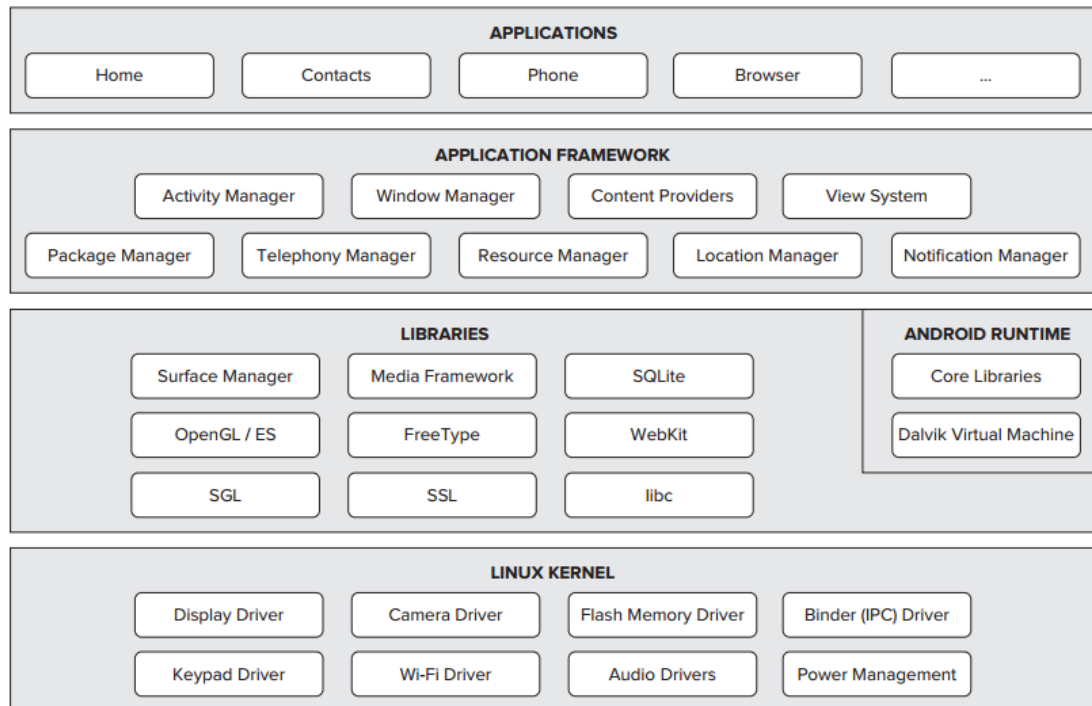
- Android là một hệ điều hành có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng.



Hình 2.10 Hệ điều hành Android

- Android gồm 5 phần chính sau được chứa trong 4 lớp:
 - Nhân Linux: Đây là nhân nền tảng mà hệ điều hành Android dựa vào nó để phát triển. Đây là lớp chứa tất cả các thiết bị giao tiếp ở mức thấp dùng để điều khiển các phần cứng khác trên thiết bị Android.
 - Thư viện: Chứa tất cả các mã cái mà cung cấp các tính năng chính của hệ điều hành Android, đôi với ví dụ này thì SQLite là thư viện cung cấp việc hỗ trợ làm việc với database dùng để chứa dữ liệu. Hoặc Webkit là thư viện cung cấp những tính năng cho trình duyệt Web.
 - Android runtime: Là tầng cùng với lớp thư viện Android runtime cung cấp một tập các thư viện cốt lõi để cho phép các lập trình viên phát triển viết ứng dụng bằng việc sử dụng ngôn ngữ lập trình Java. Android Runtime bao gồm máy ảo Dalvik (ở các version < 4.4, hiện tại là phiên bản máy ảo ART được cho là mạnh mẽ hơn trong việc xử lý biên dịch). Là cái để điều khiển mọi hoạt động của ứng dụng Android chạy trên nó, tương tự như các ứng dụng được biên dịch trên máy ảo Java vậy). Ngoài ra máy ảo còn giúp tối ưu năng lượng pin cũng như CPU của thiết bị Android
 - Android framework: Là phần thể hiện các khả năng khác nhau của Android (kết nối, thông báo, truy xuất dữ liệu) cho nhà phát triển ứng dụng, chúng có thể được tạo ra để sử dụng trong các ứng dụng của họ.

- **Application:** Tầng ứng dụng là tầng bạn có thể tìm thấy chuyên các thiết bị Android như Contact, trình duyệt... Và mọi ứng dụng bạn viết đều nằm trên tầng này.



Hình 2.11 Cấu trúc của ứng dụng Android

2.7.2. Cơ chế hoạt động

Khi bạn biên dịch một ứng dụng, bộ SDK Android sẽ đóng gói ứng dụng của bạn vào một file có hậu tố .apk. Một tệp APK chứa tất cả nội dung của một ứng dụng Android và các thiết bị sử dụng nền tảng Android sẽ sử dụng chúng để cài đặt ứng dụng vào hệ thống.

Sau khi được cài đặt vào một thiết bị, ứng dụng sẽ được đặt vào bên trong một hộp cát bảo mật của chính nó:

- Vì hệ điều hành Android là một hệ thống Linux đa người dùng, trong đó mỗi ứng dụng là một người dùng.
- Theo mặc định, hệ thống gán cho từng ứng dụng một ID người dùng Linux duy nhất (ID chỉ được dùng bởi hệ thống và không xác định với người dùng). Hệ thống sẽ gán quyền tất cả các tệp trong một ứng dụng sao cho ID người dùng được gán cho ứng dụng đó thì mới có thể truy cập.
- Mỗi một tiến trình có máy ảo riêng của mình, vì thế mã của một ứng dụng sẽ chạy độc lập với các ứng dụng khác.

2.7.3. Thành phần



Hình 2.12 Thành phần trong ứng dụng Android

Có 4 kiểu thành phần trong ứng dụng của Android bao gồm:

- **Hoạt động (Activity):** Biểu diễn một màn hình đơn với giao diện người dùng. Hoạt động triển khai như một lớp con của Activity. Activity là một thành phần ứng dụng cung cấp một màn hình mà với nó người dùng có thể tương tác để thực hiện một điều gì đó. Ví dụ như quay số điện thoại, chụp ảnh, gửi e-mail... Mỗi hoạt động được đưa vào một cửa sổ là nơi để vẽ giao diện người dùng, cửa sổ này thường lấp đầy màn hình nhưng cũng có thể nhỏ hơn màn hình và nổi lên trên màn hình khác. Ứng dụng bao gồm nhiều hoạt động liên kết lỏng lẻo với nhau. Thường thì một hoạt động trong một ứng dụng, sẽ được quy định là hoạt động chính, nó được trình bày trước khi ứng dụng khởi chạy lần đầu tiên. Sau đó, mỗi hoạt động có thể bắt đầu hoạt động khác. Mỗi khi một hoạt động mới được khởi tạo, hành động trước đó sẽ dừng lại, nhưng hệ thống vẫn giữ nguyên hoạt động đó trong một ngăn xếp (back-stack). Khi một hoạt động mới được khởi tạo, nó sẽ được đưa lên đỉnh ngăn xếp, chiếm lấy tiêu điểm của người dùng và theo cơ chế vào sau ra trước của ngăn xếp. Khi một hoạt động bị dừng vì hoạt động khác được khởi tạo, nó được thông báo về sự thay đổi này thông qua vòng đời của hoạt động.
- **Dịch vụ (Service):** Là một thành phần chạy ngầm để thực hiện các thao tác chạy lâu hoặc để thực hiện công việc việc cho các ứng dụng từ xa. Dịch vụ không cung cấp một giao diện người dùng. Ví dụ, một ứng dụng có thể phát

nhạc dưới nền trong khi người dùng đang ở một ứng dụng khác hoặc nó có thể tải ứng dụng qua mạng mà không chặn người dùng tương tác với hoạt động. Dịch vụ triển khai như một lớp con của Service.

- **BroadcastsReceive:** Là một thành phần có chức năng hồi đáp lại các thông báo trên hệ thống, có nhiều broadcast receiver trên hệ thống như: pin yếu hoặc pin đã tắt hoặc một bức ảnh đã được chụp.
- **Intent:** Các loại thành phần trên sẽ được kích hoạt bằng một thông báo không đồng bộ gọi là Intent. Intent sẽ gắn kết các thành phần với nhau lúc runtime. Nó được khởi tạo từ một đối tượng Intent, nó định nghĩa một thông báo để kích hoạt một thành phần. Đối với các Activity và Service Intent có chức năng định nghĩa một hoạt động sẽ thực hiện và có thể chỉ định URI của dữ liệu để hành động dựa trên đó.
- **AndroidManifest (bản kê khai):** Kê khai các thành phần trong ứng dụng. Xác định quyền của người dùng mà ứng dụng yêu cầu, chẳng hạn truy cập Internet.
 - Mô tả mức API tối thiểu mà ứng dụng yêu cầu dựa trên những API mà ứng dụng sử dụng.
 - Khai báo các tính năng phần cứng phần mềm được sử dụng hoặc yêu cầu bởi hệ thống.
 - Các thư viện API mà ứng dụng cần liên kết.

2.7.4. Môi trường phát triển

- Hệ điều hành:
Ứng dụng Android có thể được lập trình trên cả 3 nền tảng hệ điều hành phổ biến nhất hiện nay: Microsoft Windows, Linux và Mac.
- Java development kit (JDK):
JDK là một hệ tiêu chuẩn trong việc triển khai nền tảng Java, bao gồm các trình thông dịch dịch và thư viện lớp.
- Android Software Development Kit (SDK):
SDK là bộ Công Cụ Phát Triển Cho Android. SDK này sẽ cung cấp cho chúng ta một bộ các thư viện và công cụ cần thiết để chúng ta có thể build, kiểm tra và debug cho các ứng dụng Android
- Android Studio:
Chức năng của Android Studio là cung cấp giao diện để tạo các ứng dụng và xử lý phần lớn các công cụ quản lý file phức tạp đằng sau hậu trường. Ngôn

ngữ lập trình được sử dụng ở đây là Java và được cài đặt riêng trên thiết bị của bạn. Android Studio rất đơn giản, bạn chỉ cần viết, chỉnh sửa và lưu các dự án của mình và các file trong dự án đó. Đồng thời, Android Studio sẽ cấp quyền truy cập vào Android SDK.

2.8 Kết chương

Chương 2 cung cấp cho người đọc các kiến thức cơ sở liên quan đến việc xây dựng ứng dụng hỗ trợ dự đoán bệnh da liễu thường gặp ở người Việt Nam. Phần cơ sở lý thuyết này là nền tảng trong việc thiết kế và triển khai xây dựng mô hình huấn luyện, hệ thống API và giao diện người sử dụng.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Mô tả hệ thống

Xây dựng một hệ thống có khả năng dự đoán một số bệnh phổ biến đặc trưng về da liễu ở người Việt Nam thông qua thuật toán Deep learning. Hệ thống được xây dựng trên nền tảng ứng dụng di động. Hệ thống có thể bổ sung dữ liệu của người sử dụng để phát triển bộ dữ liệu để tăng độ chính xác của mô hình. Thêm vào đó ứng dụng sẽ cung cấp các lý thuyết về nguyên nhân, ảnh hưởng và cách sơ cấp cứu của mỗi loại bệnh được chẩn đoán.

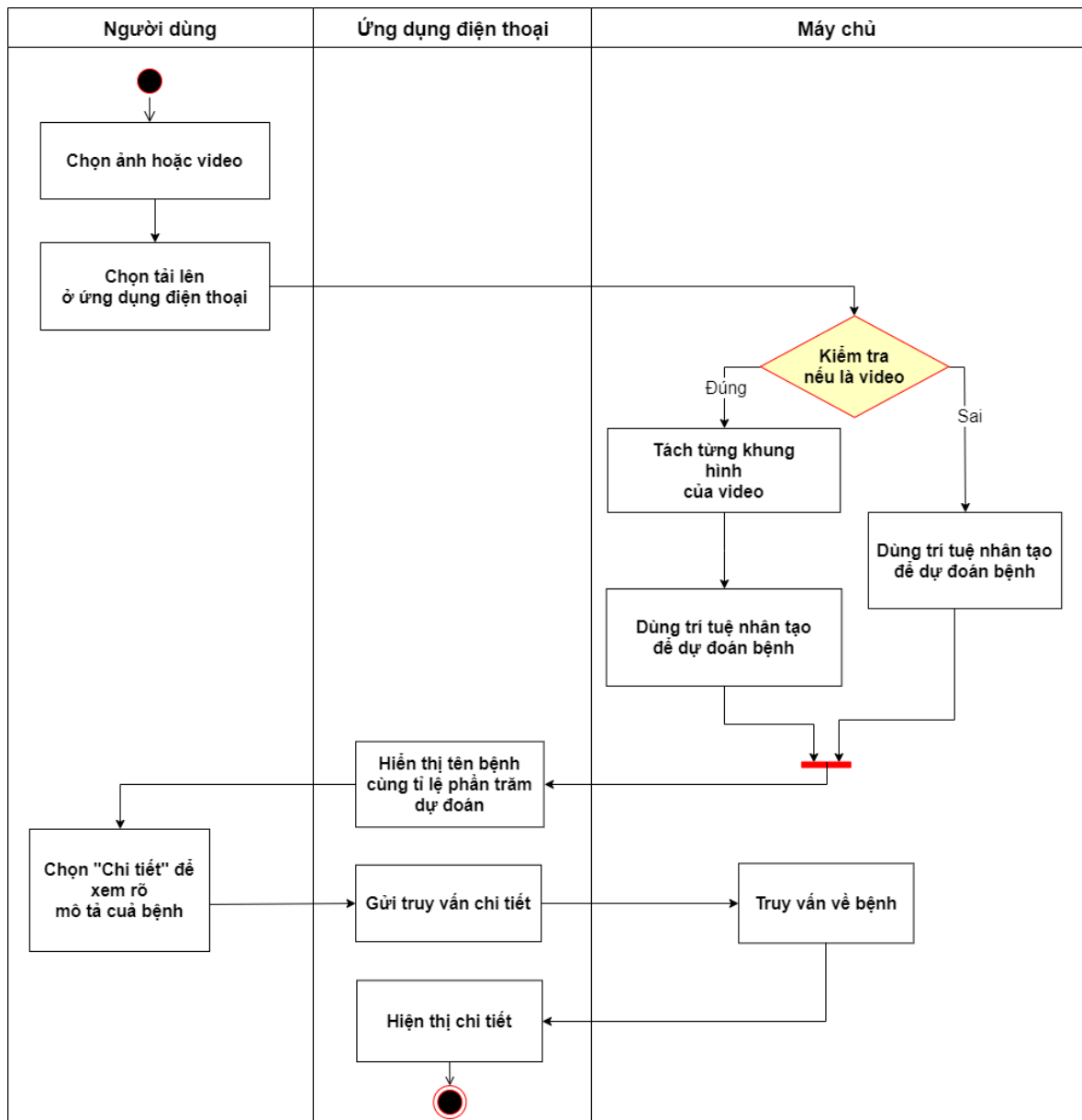
3.2. Phân tích và thiết kế hệ thống

3.2.1. Phân tích đối tượng

Bảng 3.1 Bảng phân tích các tác nhân đối với hệ thống

Tác nhân	Description
Người dùng	<ul style="list-style-type: none">- Tải lên ảnh triệu chứng, dấu hiệu của bệnh cần dự đoán- Xem thông tin về bệnh da liễu được dự đoán- Tìm hiểu và đọc thêm kiến thức về các bệnh da liễu thường gặp- Nhận kết quả dự đoán bệnh da liễu từ hệ thống- Tải lên video cần dự đoán để tăng thêm độ chính xác về nhiều góc độ

3.2.2. Hệ thống sơ đồ hoạt động chức năng người dùng



Hình 3.1 Biểu đồ hoạt động của người dùng

Biểu đồ hoạt động mô tả đầy đủ quá trình các hoạt động xảy ra khi người dùng đưa dữ liệu và nhận về kết quả được dự đoán. Người dùng có thể chọn ảnh hoặc video để tải lên máy chủ, máy chủ sau khi nhận được dữ liệu sẽ kiểm tra loại dữ liệu. Nếu là video thì máy chủ sẽ cắt ảnh từ video để phục vụ cho việc đưa dữ liệu vào mô hình dự đoán, còn nếu định dạng dữ liệu đưa lên là ảnh thì sẽ tiến xử lý dữ liệu và đưa vào mô hình dự đoán. Sau khi nhận được kết quả dự đoán từ mô hình thì máy chủ sẽ đóng gói kết quả và trả về ứng dụng điện thoại để hiển thị cho người dùng. Nếu người dùng muốn biết thêm thông tin về chi tiết của bệnh được dự đoán thì có thể gửi truy vấn thông qua giao diện người dùng của ứng dụng điện thoại. Yêu cầu truy vấn sau khi

được đưa lên máy chủ sẽ được xử lý và trả về kết quả cho ứng dụng di động thể hiện thị cho người dùng xem.

3.3. Kế hoạch làm việc

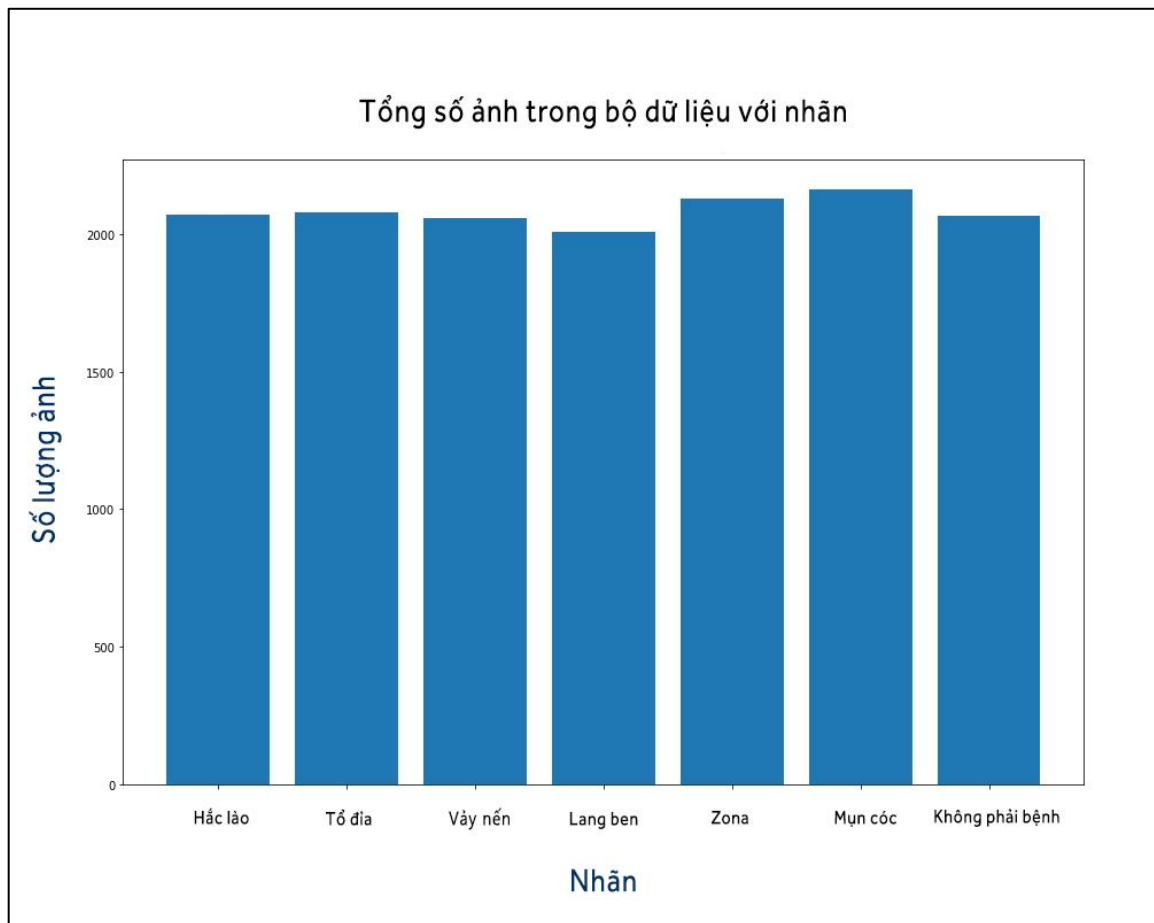
Bảng 3.2 Bảng kế hoạch làm việc

Thời gian	Kế hoạch công việc	Phiên bản
06-09-2019 09-11-2019	<ul style="list-style-type: none"> - Thu thập dữ liệu về bệnh da liễu thường gặp ở Việt Nam - Gán nhãn cho dữ liệu - Xử lý dữ liệu trước khi đưa vào huấn luyện 	1.0
12-09-2019 17-09-2019	<ul style="list-style-type: none"> - Tìm hiểu kiến thức chung về vấn đề nhận dạng phân loại của với kỹ thuật học sâu - Tìm hiểu về cách thực thi Pytorch - Tìm hiểu về mô hình ResNet 	1.0
18-09-2019 30-09-2019	<ul style="list-style-type: none"> - Tiền xử lý tập dữ liệu - Xây dựng mô hình phiên bản 1 với Keras mô hình ResNet 50 - Đánh giá mô hình vừa triển khai 	1.0
03-10-2019 15-10-2019	<ul style="list-style-type: none"> - Vẽ mockup cho phần Front-End. - Chỉnh sửa Usecase - Lên mức độ ưu tiên cho các chức năng đề ra - Xây dựng hệ thống Back-End API 	1.0
16-10-2019 23-10-2019	<ul style="list-style-type: none"> - Viết Api phục vụ cho ứng dụng - Huấn luyện mô hình phiên bản 2 với Pytorch ResNet 50 - Thu thập dữ liệu bổ sung cho tập dữ liệu 	2.0

24-10-2019 08-11-2019	<ul style="list-style-type: none">- Viết UI cho 3 màn hình chính của ứng dụng- Đánh giá mô hình huấn luyện phiên bản 2 với bộ dữ liệu thử nghiệm- Tăng cường dữ liệu bằng Augmentation	2.0
01-11-2019 25-11-2019	<ul style="list-style-type: none">- Hoàn thiện giao diện người dùng- Kết nối mô hình và hệ thống API- Thay đổi theo nhu cầu của khách hàng	2.0
26-11-2019 05-11-2019	<ul style="list-style-type: none">- Tìm hiểu và sửa lỗi sản phẩm theo nhu cầu người dùng- Lọc các dữ liệu không tốt, không phù hợp- Xử lý dữ liệu để chuẩn bị cho huấn luyện- Hiệu chỉnh hệ thống API phù hợp cho giao diện người dùng	2.0
06-12-2019 10-12-2019	<ul style="list-style-type: none">- Viết báo cáo cho ứng dụng- Xây dựng mô hình phiên bản 3 với Pytorch ResNet 152	3.0
11-12-2019 16-12-2019	<ul style="list-style-type: none">- Hoàn thiện báo cáo- Hoàn thiện mô hình dự đoán- Kết nối chặt chẽ giữa giao diện người dùng và máy chủ chứa API và mô hình dự đoán	3.0

3.4. Đánh giá dữ liệu huấn luyện

3.4.1. Thông tin tập dữ liệu huấn luyện



Hình 3.2 Số lượng ảnh và nhãn trong tập dữ liệu

Tập dữ liệu đầu vào gồm tổng cộng 14406 ảnh và 7 nhãn. Trong đó bao gồm 6 nhãn bệnh da liễu và 1 nhãn không phải là bệnh. Dữ liệu ảnh có kích thước mặc định 256x256. Các nhãn được gán trong tập dữ liệu bao gồm:

- Bệnh hắc bào (2071 ảnh)
- Bệnh lang ben (2008 ảnh)
- Bệnh tổ đũa (2078 ảnh)
- Bệnh vảy nến (2059 ảnh)
- Bệnh zona (2131 ảnh)
- Bệnh mụn cóc (2162 ảnh)
- Không phải bệnh (1897 ảnh)

Tập dữ liệu xác nhận gồm 3621 ảnh trong đó có 6 nhãn bệnh và 1 nhãn không phải bệnh.

Nguồn gốc để thu thập dữ liệu bao gồm:

Atlas bệnh da liễu: Dữ liệu thu thập được từ nguồn dữ liệu này chiếm 15% trong tổng số dữ liệu tổng.

- Atlas các bệnh da liễu thường gặp: được biên soạn bởi BS. Việt Hà, BS. Phan Hoa, BS. Bích Thủy, BS. Hải Yến. Đường đính kèm tài liệu ¹
- Atlas Fitzpatrick's color atlas and synopsis of clinical dermatology: Đường dẫn đính kèm tài liệu ²
- Giáo trình bài giảng bệnh da liễu: Chủ biên. GS. TS Nguyễn Văn Út.

Bác sỹ chuyên khoa da liễu: Bác sỹ Chuyên khoa 1 Da Liễu Trần Nguyễn Như Hương – Thạc sỹ Da Liễu Đại học Y Dược Thành phố Hồ Chí Minh. Dữ liệu thu thập được chiếm 25% trong tổng số dữ liệu tổng.

Nhóm cộng đồng da liễu trên mạng xã hội Facebook: Dữ liệu thu thập được từ nguồn dữ liệu này chiếm 40% trong tổng số dữ liệu tổng.

DermNet New Zealand Trust: Dữ liệu thu thập được từ nguồn dữ liệu này chiếm 20% trong tổng số dữ liệu tổng. DermNet NZ là một nguồn cung cấp dữ liệu nổi tiếng thế giới về da. Trang web này được sở hữu bởi DermNet New Zealand Trust. Nó đã được HON (Health-on-the-Net) chứng nhận từ năm 1996 và đang được phát triển liên tục cung cấp thông tin về da trên bất kỳ máy tính để bàn hoặc trình duyệt web di động. Đường dẫn đến trang website là dermnetnz.org.

3.4.2. Quá trình xây dựng tập dữ liệu

Để nhận dạng một loại bệnh da liễu cần rất nhiều yếu tố, chính vì thế nếu khoanh vùng phạm vi sẽ nhận dạng được sẽ giúp nâng cao độ chính xác cũng như tính ổn định của ứng dụng. Sau khi tham khảo cũng như nghe các lời khuyên từ phía các bác sỹ chuyên khoa da liễu và người hướng dẫn đề án, quá trình xây dựng tập dữ liệu ban đầu chỉ lấy các bệnh có đặc điểm nổi bật về hình dáng và màu sắc, những

¹ <https://drive.google.com/file/d/0Bw8HNtsJWDpKdVFzcmdUOC1iZ1U/view>

² <https://drive.google.com/file/d/0B2hwoMA4XfWCVnBndy1sSmgtRXM/view>

bệnh này thường xuất hiện ở người sinh sống tại Việt Nam. Các yếu tố để đưa các bệnh vào danh sách tập huấn luyện là bao gồm:

- Bệnh về da thường xuyên gặp phải ở Việt Nam
- Bệnh có đặc trưng về hình dáng của triệu chứng rõ ràng
- Bệnh có đặc trưng về màu sắc của triệu chứng

3.5. Giải pháp đề xuất

Với bộ dữ liệu thu thập còn nhỏ và chưa thực sự được chuẩn hóa một cách hoàn chỉnh thì việc overfit trong quá trình huấn luyện rất dễ xảy ra. Chính vì vậy nên giải pháp đưa ra là sử dụng kết hợp hai kỹ thuật là học chuyển đổi (transfer learning) và Data Augmentation. Ngoài ra để giảm vấn đề về việc độ chính xác bị giảm đi khi mô hình được xây dựng với số lớp quá lớn hoặc kết quá bị quá khớp với bộ dữ liệu huấn luyện, việc sử dụng mô hình ResNet 152 sẽ giúp chúng ta huấn luyện mô hình với mạng nơ ron sâu mà không mắc phải các vấn đề nêu trên.

Sử dụng Pre-trained model là ResNet 152 của ImageNet. Mô hình ResNet152 của ImageNet dataset, phân loại ảnh thuộc 1000 thể loại khác nhau. Nên có thể hiểu là nó đủ tổng quát để tách ra các đặc điểm của bức ảnh, cụ thể ở đây là bệnh về da liễu.

3.5.1. Kỹ thuật Data Augmentation

Augmentor là gói phần mềm dùng để tăng cường hình ảnh, tập trung vào việc cung cấp các hoạt động thường được sử dụng trong việc tạo dữ liệu hình ảnh cho các vấn đề về máy học. Là kỹ thuật đơn giản nhất bằng việc xử lý đơn giản dữ liệu sẵn có bằng các phép tuyến tính hay phi tuyến (như tạo dữ liệu qua mạng GAN). Việc sử dụng data augmentation sẽ giúp bộ dữ liệu huấn luyện được tăng cường một cách nhanh chóng. Trong y sinh việc thực hiện sinh ảnh bằng các ảnh gốc thường không làm ảnh hưởng quá nhiều trong việc huấn luyện mô hình dự đoán, phân loại ảnh. Việc thay đổi vẫn giữ lại những đặc trưng nổi bật của dữ liệu gốc.

Về nguyên tắc, Augmentor bao gồm một số lớp cho các hàm thao tác hình ảnh tiêu chuẩn, chẳng hạn như lớp Rotate hoặc lớp Crop. Người dùng tương tác và sử dụng các lớp này bằng cách sử dụng một số lượng lớn các hàm tiện lợi, bao gồm hầu hết các chức năng bạn có thể yêu cầu khi tăng bộ dữ liệu hình ảnh cho các vấn đề về máy học.

Bởi vì việc tăng cường hình ảnh thường là một thủ tục nhiều giai đoạn, Augmentor sử dụng phương pháp dựa trên đường ống , trong đó các hoạt động được thêm liên tục để tạo ra một đường ống dẫn. Hình ảnh sau đó được chuyển qua đường ống này, trong đó mỗi thao tác được áp dụng cho hình ảnh khi nó đi qua.

Ngoài ra, Augmentor áp dụng các thao tác cho hình ảnh một cách ngẫu nhiên khi chúng đi qua đường ống, theo giá trị xác suất do người dùng xác định cho mỗi thao tác. Do đó, mọi hoạt động đều có tối thiểu một tham số xác suất, điều khiển khả năng hoạt động sẽ được áp dụng cho từng hình ảnh được xem khi hình ảnh đi qua đường ống.

Các hàm được sử dụng trong quá trình tăng cường dữ liệu ảnh bao gồm:

- **rotate(probability=1, max_left_rotation=14, max_right_rotation=14):** Đây là thao tác xoay hình ảnh. Trong đó probability là xác suất thao tác được áp dụng, max_left_rotation và max_right_rotation là tham số cho việc kiểm soát mức độ mà ảnh có thể xoay.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.
- grid_width (Integer) - Số lượng hình chữ nhật trong trục ngang của lưới.
- max_left_rotation (Integer) – Độ xoay lớn nhất về phía trái.
- max_right_rotation (Integer) - Độ xoay lớn nhất về phía phải.
- **zoom_random(probability=0.5, percentage_area=0.8):** Phóng to hình ảnh tại một vị trí ngẫu nhiên trong hình ảnh.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.
- percentage_area (Integer) - Diện tích, tính theo phần trăm diện tích của hình ảnh hiện tại, để cắt.
- **flip_left_right(probability=0.3):** Là thao tác lật ảnh từ trái sang phải.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1
- **flip_top_bottom(probability=0.4):** Là thao tác lật ảnh từ trên xuống dưới.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.

- **rotate_random_90(probability=0.3):** Xoay hình ảnh theo 90, 180 hoặc 270 độ, được chọn ngẫu nhiên. Chức năng này sẽ xoay theo 90, 180 hoặc 270 độ. Điều này rất hữu ích để tránh các tình huống trong đó hình ảnh có thể được quay trở lại vị trí ban đầu của chúng.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.
- **skew(probability=0.5, magnitude=0.7):** Thao tác làm nghiêng ảnh theo 4 hướng trái, phải, trên và dưới.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.
- magnitude (Integer) - Độ lớn của các biến dạng.
- **skew_tilt(probability=0.6, magnitude =0.7):** Thao tác làm nghiêng ảnh theo một hướng ngẫu nhiên, về phía trước, ngược, trái hoặc phải, một lượng ngẫu nhiên.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.
- magnitude (Integer) - Độ lớn của các biến dạng.
- **random_distortion(probability=0.7, grid_width=5, grid_height=5, magnitude=8):** Thực hiện một biến dạng ngẫu nhiên, đàn hồi trên một hình ảnh. Hàm này thực hiện một biến dạng ngẫu nhiên, đàn hồi được điều khiển bởi các tham số đã chỉ định. Chiều rộng và chiều cao của lưới điều khiển mức độ biến dạng tốt. Kích thước nhỏ hơn sẽ dẫn đến biến dạng lớn hơn, rõ hơn và ít hạt hơn. Số lượng lớn hơn sẽ dẫn đến biến dạng mịn hơn, chi tiết hơn. Độ lớn của các biến dạng có thể được kiểm soát bằng cách sử dụng cường độ. Điều này có thể là ngẫu nhiên hoặc cố định.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.
- grid_width (Integer) - Số lượng hình chữ nhật trong trục ngang của lưới.
- grid_height (Integer) - Số lượng hình chữ nhật trong trục dọc của lưới.
- magnitude (Integer) - Độ lớn của các biến dạng.

- **flip_left_right(probability=0.3):** Là thao tác lật ảnh từ trái sang phải.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.

- **flip_top_bottom(probability=0.4):** Là thao tác lật ảnh từ trên xuống dưới.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.

- **rotate_random_90(probability=0.3):** Xoay hình ảnh theo 90, 180 hoặc 270 độ, được chọn ngẫu nhiên. Chức năng này sẽ xoay theo 90, 180 hoặc 270 độ. Điều này rất hữu ích để tránh các tình huống trong đó hình ảnh có thể được quay trở lại vị trí ban đầu của chúng.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.

- **skew(probability=0.5, magnitude=0.7):** Thao tác làm nghiêng ảnh theo 4 hướng trái, phải, trên và dưới.

Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.
- magnitude (Integer) - Độ lớn của các biến dạng.

- **skew_tilt(probability=0.6, magnitude =0.7):** Thao tác làm nghiêng ảnh theo một hướng ngẫu nhiên, về phía trước, ngược, trái hoặc phải, một lượng ngẫu nhiên.

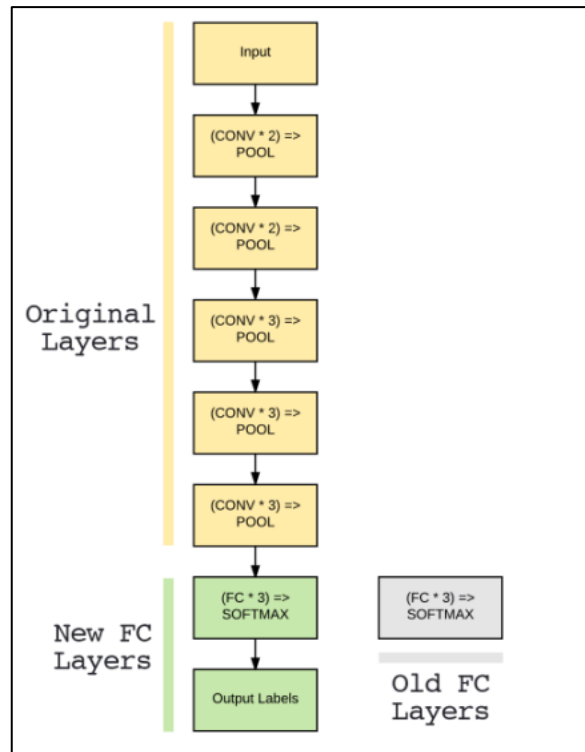
Thông số:

- probability (Float) - Giá trị từ 0 đến 1 đại diện cho xác suất mà hoạt động nên được thực hiện.
- magnitude (Integer) - Độ lớn của các biến dạng.

3.5.2. Kỹ thuật *Transfer Learning* – *Fine tuning*

Sau khi lấy ra các đặc điểm của ảnh bằng việc sử dụng ConvNet của pre-trained model, đây được xem như là một input của một mạng nơ ron tích chập mới bằng cách thêm các ConvNet và Fully Connected layer vào sau để phù hợp với dữ liệu đầu ra mong muốn.

Mô tả cấu trúc của Fine tuning:

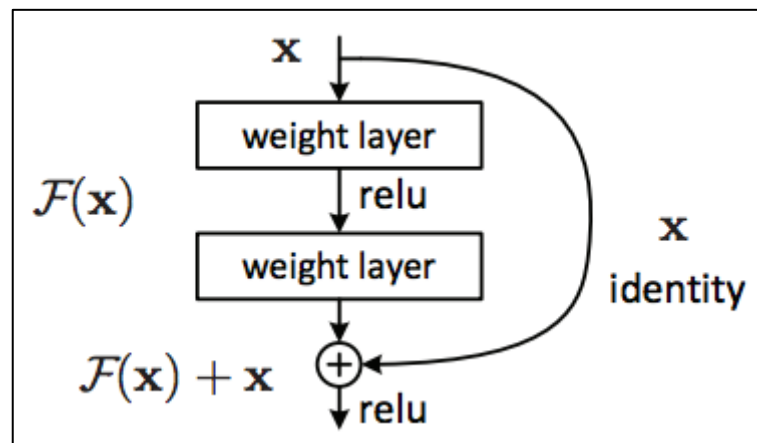


Hình 3.3 Cấu trúc của kỹ thuật Fine tuning

Ta chỉ giữ lại phần ConvNet trong CNN và bỏ đi Fully Connected layers. Sau đó thêm các Fully Connected layer mới vào output của ConvNet.

3.5.4. Mô hình ResNet

Khi train các mô hình Deep CNN (số lượng layers lớn, số lượng param lớn,...) ta thường gặp phải vấn đề về vanishing gradient hoặc exploding gradient. Thực tế cho thấy khi số lượng layer trong CNN model tăng, độ chính xác của mô hình cũng tăng theo, tuy nhiên khi tăng số layers quá lớn (>50 layers) thì độ chính xác lại bị giảm đi.



Hình 3.4 Cấu trúc của một Residual block cơ bản

Residual block ra đời nhằm giải quyết vấn đề trên, với Residual block, ta hoàn toàn có thể train các mô hình CNN có kích thước và độ phức tạp "khủng" hơn mà không lo bị exploding/vanishing gradient. Mấu chốt của Residual block là cứ sau 2 layer, ta cộng input với output: $F(x) + x$. Resnet là một mạng CNN bao gồm nhiều Residual block nhỏ tạo thành.

Mạng lưới nơ ron tích chập sâu phát hiện các đối tượng bằng cách học các tính năng. Về mặt lý thuyết, việc thêm nhiều lớp vào CNN cho phép nó tìm hiểu thêm các tính năng và đạt được độ chính xác cao hơn; tuy nhiên, nó không phải là một trường hợp lý tưởng trong thực tế. Thực tế đã được thừa nhận rằng độ chính xác đào tạo có xu hướng đạt đến bão hòa và sau đó là một sự xuống cấp nhanh chóng khi thêm nhiều lớp.

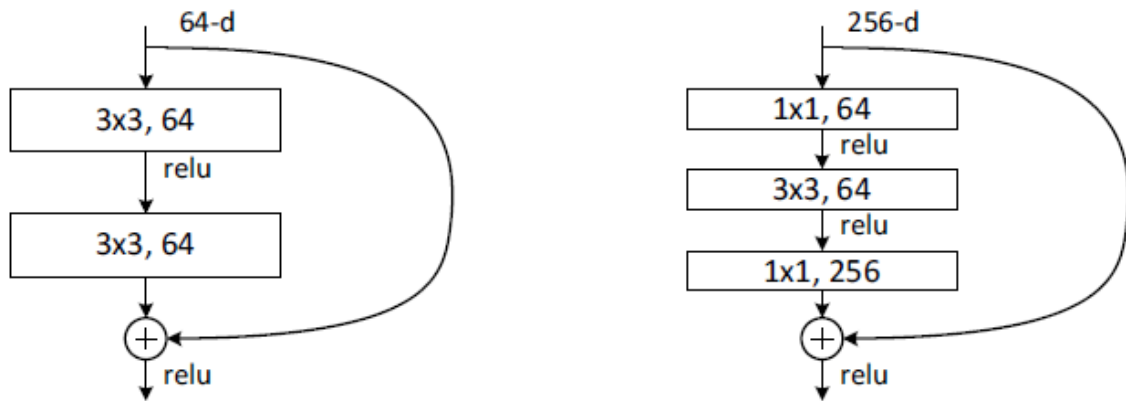
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Hình 3.5 Tổng quan kiến trúc của các mô hình ResNet

ResNet cho phép đào tạo các mạng nơ ron rất sâu mà không mất đi tính chính xác như một sự đánh đổi. Mạng dư được đặc trưng bởi một kết nối phím tắt, thay vì đi từ lớp này sang lớp khác trong các trường hợp thông thường, hai lớp (hai lớp được hiển thị trong biểu đồ, nhưng có thể là một hoặc nhiều lớp) được bỏ qua để thực hiện ánh xạ định danh, có nghĩa là đầu ra từ lớp n-2 được thêm trực tiếp vào đầu ra của lớp n hiện tại. Kết nối bỏ qua như vậy giúp cho sự giảm độ chính xác học tập có thể tránh được vì ngay cả khi các kết nối bị bỏ qua không đạt được bất cứ điều gì trong học tập, độ chính xác đầu ra từ lớp hiện tại được duy trì bằng ánh xạ định danh từ lớp n-2 trước đó. ResNet đã được báo cáo để đạt được độ chính xác được cải thiện trên các bộ dữ liệu như ImageNet và COCO³, do đó, đây là một mạng thích hợp để thực hiện trên bộ dữ liệu bệnh da đã được xây dựng và cần mạng lưới thần kinh rất sâu.

3.5.2. Mô hình ResNet152

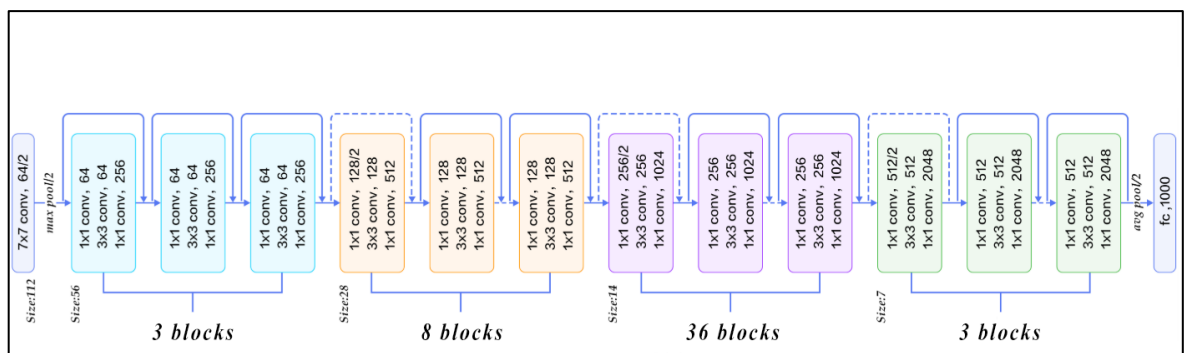
Khác với các mạng ResNet 18 và 34, ResNet 152 được xây dựng dựa trên việc kết hợp rất nhiều lớp tích chập. Cũng vì vậy nên độ sâu của mạng ResNet 152 rất lớn và độ phức tạp cao. Do đó giải pháp được đưa ra để cải tiến khối ResNet cơ bản để giảm độ phức tạp cho mạng chính là thiết kế nút cổ chai



Hình 3.6 Khối ResNet cơ bản và khối ResNet cải tiến

Các lớp đối lưu 1×1 được thêm vào đầu và cuối của mạng như trong hình (bên phải). Đây là một kỹ thuật được đề xuất trong Network In Network và GoogLeNet (Inception-v1). Nó chỉ ra rằng 1×1 có thể giảm số lượng kết nối (tham số) trong khi không làm giảm hiệu suất của mạng rất nhiều.

Với thiết kế nút cổ chai, ResNet 34 lớp trở thành ResNet 50 lớp. Và có mạng lưới sâu hơn với thiết kế nút cổ chai: ResNet-101 và ResNet-152.



Hình 3.7 Cấu trúc mạng ResNet 152

Mạng ResNet 152 gồm 1 lớp tích chập 1_x với kích thước dữ liệu đầu ra là 112×112 , 3 lớp tích chập 2_x , 8 lớp tích chập 3_x , 36 lớp tích chập 4_x và 3 lớp tích chập 5_x .

Cấu trúc các lớp như sau:

- Lớp Conv 2_x : Kích thước dữ liệu đầu vào là 56×56

1x1 conv, 64
3x3 conv, 64
1x1 conv, 253

- Lớp Conv3_x: Kích thước dữ liệu đầu vào là 28x28

1x1 conv, 64
3x3 conv, 64
1x1 conv, 253

- Lớp Conv4_x: Kích thước dữ liệu đầu vào là 14x14

1x1 conv, 64
3x3 conv, 64
1x1 conv, 253

- Lớp Conv5_x: Kích thước dữ liệu đầu vào là 7x7

1x1 conv, 64
3x3 conv, 64
1x1 conv, 253

3.5.3. Thông số mô hình lựa chọn

- Tổng tham số: 60,899,911
- Tham số có thể huấn luyện: 60,899,911
- Tham số không thể huấn luyện: 0
- Kích thước dữ liệu đầu vào (MB): 0.68
- Kích thước lan truyền tiến/lan truyền ngược (MB): 763.77
- Kích thước tham số (MB): 232.31
- Kích thước ước tính (MB): 996.77

Thông số các lớp tùy biến sau cùng:

Bảng 3.3 Bảng thông số các lớp tùy biến sau cùng của mô hình

Tên lớp	Đầu ra	Kích thước tham số
AdaptiveAvgPool2d-513	[1, 2048, 1, 1]	0
Linear-514	[1, 1024]	2,098,176
ReLU-515	[1, 1024]	0
Dropout-516	[1, 1024]	0
Linear-517	[1, 512]	524,800
ReLU-518	[1, 512]	0
Dropout-519	[1, 512]	0
Linear-520	[1, 256]	131,328
ReLU-521	[1, 256]	0
Dropout-522	[1, 256]	0
Linear-523	[1, 7]	1,799
LogSoftmax-524	[1, 7]	0

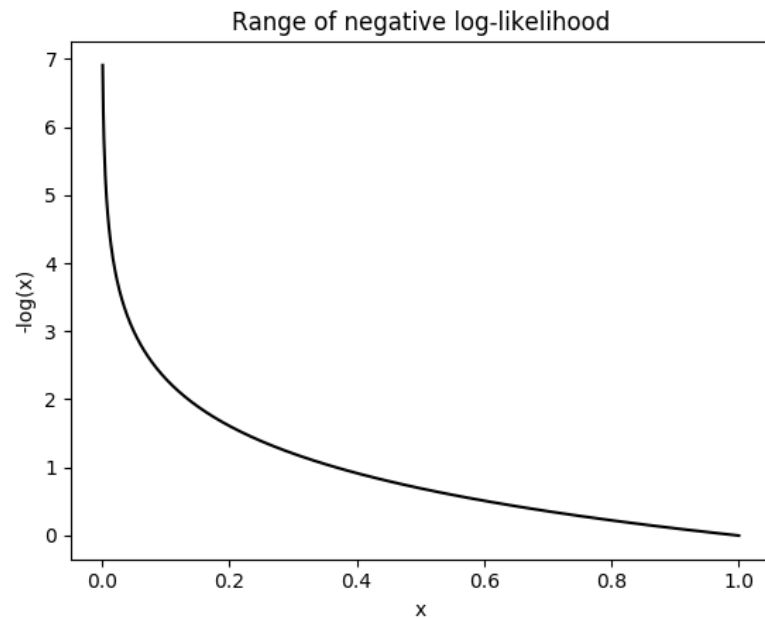
3.5.4. Hàm mất mát Negative Log-Likelihood

Trong thực tế, hàm softmax được sử dụng song song với khả năng log âm. Công thức toán học của hàm mất mát:

$$L(y) = -\log(y)$$

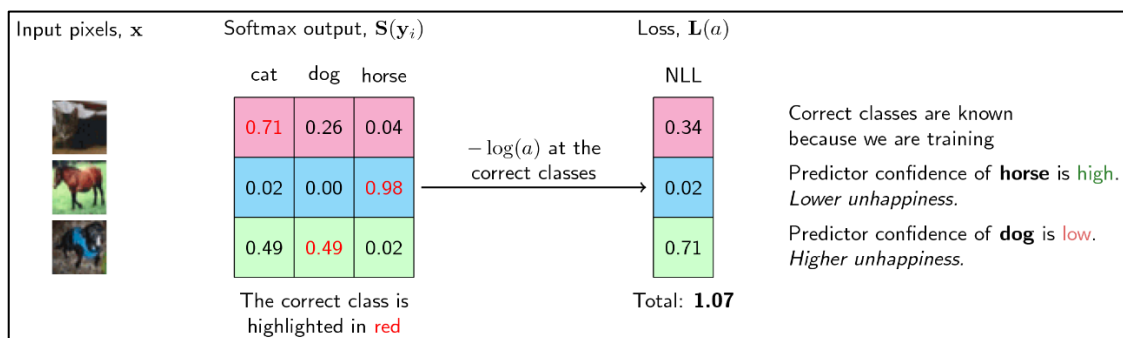
Khi đào tạo một mô hình, chúng ta mong muốn tìm ra cực tiểu của hàm mất mát cho một tập các tham số (trong mạng nơ ron, đây là các trọng số và độ lệch). Có thể hiểu rằng sự mất mát là sự không mong muốn của mạng lưới đối với các tham số của mạng. Mất mát càng cao, sự thất vọng càng cao.

Kết quả cho việc sử dụng hàm mất mát negative log-likelihood được đánh giá bởi đồ thị bên dưới:



Hình 3.8 Đồ thị hàm mất mát negative log-likelihood

Hàm mất mát negative log-likelihood sẽ cho ra kết quả không tốt ở các giá trị nhỏ hơn, nơi nó có thể đạt giá trị mất mát vô hạn (điều đó rất không mong đợi) và trả về giá trị mất mát nhỏ hơn ở các giá trị lớn hơn.



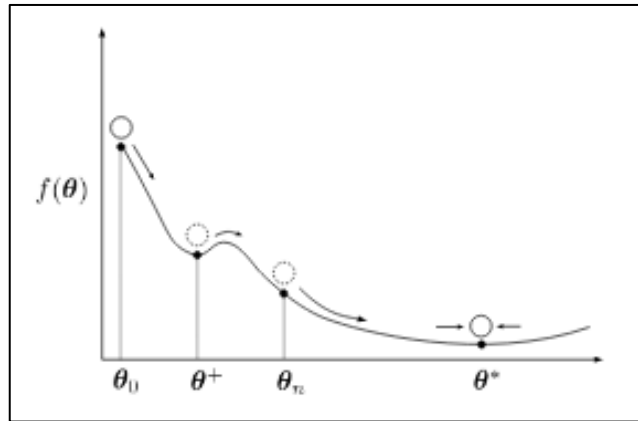
Hình 3.9 Quá trình tính toán giá trị mất mát

Hình trên cho thấy quá trình tính toán tổn thất, sau đó có thể thấy rằng độ tin cậy cao hơn ở lớp chính xác dẫn đến tổn thất thấp hơn và ngược lại.

3.5.5. Hàm tối ưu Adam

- Giống với hàm tối ưu Adadelta và RMSprop, Adam duy trì trung bình bình phương độ dốc (slope) quá khứ vt và cũng đồng thời duy trì trung bình độ dốc quá khứ mt, giống momentum.
- Trong khi momentum giống như một quả cầu lao xuống dốc, thì Adam lại giống như một quả cầu rất nặng và có ma sát (friction), nhờ vậy nó dễ dàng vượt qua local minimum và đạt tới điểm tối ưu nhất (flat minimum)

- Nó đạt được hiệu ứng Heavy Ball with Friction nhờ vào hệ số (mt/\sqrt{vt})



Hình 3.10 Đồ thị hàm số trong hàm tối ưu Adam

- Công thức cập nhật giá trị của nó là:

$$\begin{aligned} \mathbf{g}_n &\leftarrow \nabla f(\boldsymbol{\theta}_{n-1}) \\ \mathbf{m}_n &\leftarrow (\beta_1/(1-\beta_1^n)) \mathbf{m}_{n-1} + ((1-\beta_1)/(1-\beta_1^n)) \mathbf{g}_n \\ \mathbf{v}_n &\leftarrow (\beta_2/(1-\beta_2^n)) \mathbf{v}_{n-1} + ((1-\beta_2)/(1-\beta_2^n)) \mathbf{g}_n \odot \mathbf{g}_n \\ \boldsymbol{\theta}_n &\leftarrow \boldsymbol{\theta}_{n-1} - a \mathbf{m}_n / (\sqrt{\mathbf{v}_n} + \epsilon), \end{aligned}$$

Hình 3.11 Công thức cập nhật giá trị của hàm tối ưu Adam

Các lợi ích của thuật toán Adam

- Không khó khăn để thực thi và triển khai với các mô hình huấn luyện
- Độ phức tạp hiệu quả
- Bộ nhớ yêu cầu ít.
- Thích hợp với các bài toán có độ biến thiên không ổn định và dữ liệu training phân mảnh.
- Các siêu tham số được biến thiên một cách hiệu quả và yêu cầu ít điều chỉnh

3.6. Kết chương

Chương 3 cung cấp cho người đọc giải pháp được đề xuất để giải quyết bài toán nhận dạng bệnh da liễu thường gặp ở người Việt Nam thông qua ảnh hoặc video từ người dùng tải lên. Các kỹ thuật được áp dụng để giải quyết bao gồm kỹ thuật học chuyển đổi và tăng cường ảnh dữ liệu. Ngoài ra còn mô tả thêm về hàm mất mát cũng như hàm tối ưu trong quá trình huấn luyện.

CHƯƠNG 4: TRIỂN KHAI THỰC TẾ VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Triển khai huấn luyện mô hình

4.1.1. Môi trường huấn luyện

Google colab là một dịch vụ đám mây miễn phí, hiện nay có hỗ trợ GPU (Tesla K80) và TPU (TPUv2). Do được phát triển dựa trên Jupiter Notebook nên việc sử dụng Google Colab cũng tương tự như việc sử dụng Jupyter Notebook. Google Colab là một công cụ lý tưởng để chúng ta rèn luyện kỹ năng lập trình với ngôn ngữ Python thông qua các thư viện của deep learning. Google Colab cài đặt sẵn cho chúng ta những thư viện rất phổ biến trong nghiên cứu Deep Learning như PyTorch, TensorFlow, Keras và OpenCV.

4.1.2. Framework sử dụng huấn luyện

PyTorch là một framework được xây dựng dựa trên python cung cấp nền tảng tính toán khoa học phục vụ lĩnh vực Deep learning. Pytorch tập trung vào 2 khả năng chính:

- Một sự thay thế cho bộ thư viện numpy để tận dụng sức mạnh tính toán của GPU.
- Một platform Deep learning phục vụ trong nghiên cứu, mang lại sự linh hoạt và tốc độ.

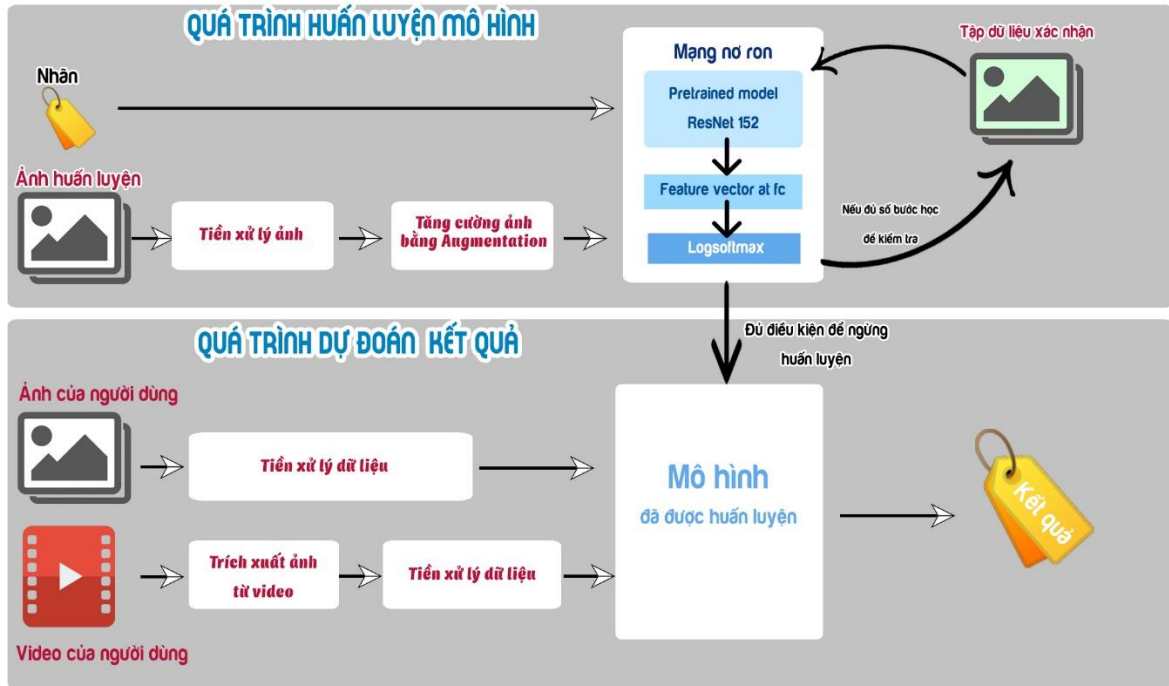
Ưu điểm:

- Mang lại khả năng debug dễ dàng hơn theo hướng interactively, rất nhiều nhà nghiên cứu và engineer đã dùng cả pytorch và tensorflow đều đánh giá cao pytorch hơn trong vấn đề debug và visualize.
- Hỗ trợ tốt đồ thị động.
- Kết hợp cả các API cấp cao và cấp thấp.

Nhược điểm:

- Pytorch vẫn chưa được hoàn thiện trong việc deploy, áp dụng cho các hệ thống lớn như framework ra đời trước nó như tensorflow.
- Ngoài tài liệu chính từ pytorch thì vẫn còn khá hạn chế các nguồn tài liệu bên ngoài.

4.1.3. Tổng quan quá trình huấn luyện mô hình và dự đoán kết quả



Hình 4.1 Tổng quan quá trình huấn luyện mô hình và dự đoán kết quả

Dữ liệu sau khi được tìm kiếm và thu thập sẽ được tiền xử lý và tăng cường ảnh bằng kỹ thuật Augmentation với đúng nhãn được gán ở ảnh gốc. Sau khi được tăng cường, bộ dữ liệu cùng với nhãn được gán sẽ đi vào mạng nơ ron được xây dựng để huấn luyện cho ra mô hình. Trong quá trình huấn luyện, mô hình sẽ được xác thực cùng với tập dữ liệu xác thực khi đã được huấn luyện đủ 70 bước. Công việc xác thực mức độ học tập của mô hình sẽ giúp chúng ta quan sát được quá trình huấn luyện có đi đúng hướng và giúp ta có các điều chỉnh phù hợp.

Mô hình sẽ được ngừng huấn luyện và xuất mô hình khi đủ epoch = 500 định nghĩa lúc ban đầu, đề phòng trường hợp bị ngắt kết nối thì mô hình sẽ được lưu trữ tạm thời vào các mốc epoch = 300 và epoch = 400.

Để dự đoán kết quả cho người dùng thì tùy thuộc vào loại dữ liệu người dùng tải lên mà tiền xử lý loại dữ liệu đó. Sau đó sẽ cho vào mô hình đã được huấn luyện và xuất ra kết quả dự đoán.

4.1.4. Thông số huấn luyện

Để huấn luyện mô hình đạt hiệu quả cao thì việc cấu hình các thông số để huấn luyện rất quan trọng, sau đây là các thông số cho việc huấn luyện mô hình:

- Tổng số ảnh huấn luyện: 14406 ảnh

- Batch_size cho tập dữ liệu huấn luyện: 32
- Num_worker: 8 worker
- Số lượng epochs: 500
- Số bước để kiểm tra mô hình: 70
- Mô hình được huấn luyện trước được sử dụng: ResNet 152 trên Pytorch
- Các lớp được thêm vào sau cùng để phù hợp với đầu ra của mô hình:

```
nn.Sequential(nn.Linear(2048, 1024),  
              nn.ReLU(),  
              nn.Dropout(0.2),  
              nn.Linear(1024, 512),  
              nn.ReLU(),  
              nn.Dropout(0.2),  
              nn.Linear(512, 7),  
              nn.LogSoftmax(dim=1))
```

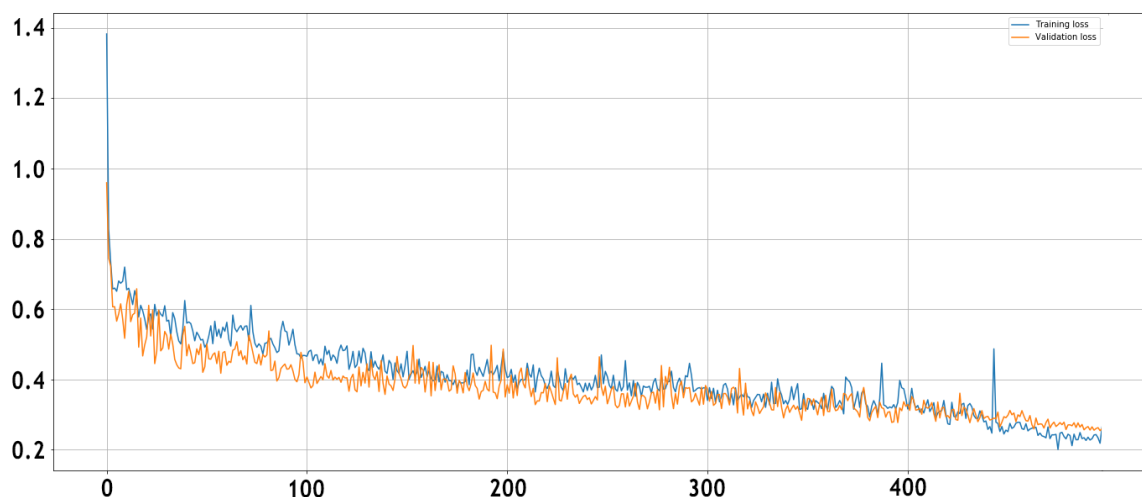
- Hàm tối ưu lựa chọn là hàm Adam được khởi tạo với tham số learn_rate đầu vào là 0.003:

```
optimizer = optim.Adam(model.fc.parameters(), lr=0.003)
```

- Hàm mất mát được lựa chọn là negative log likelihood loss:

```
criterion = nn.NLLLoss()
```

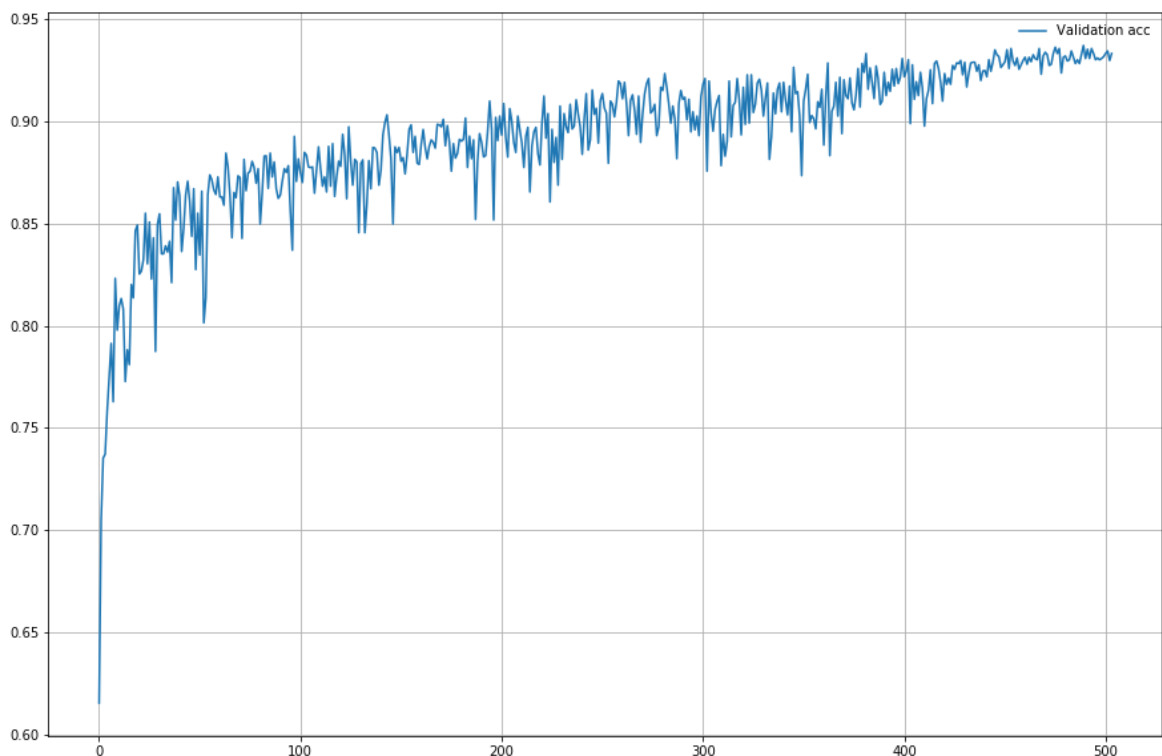
4.1.3. Quá trình huấn luyện mô hình



Hình 4.2 Biểu đồ giá trị mất mát trong quá trình huấn luyện và xác nhận

Trong quá trình huấn luyện mô hình, theo quan sát thì tại các thời điểm epoch bằng 50, 100, 150, 170, 220, 300 thì giá trị mất mát của hàm mất mát có xu hướng không giảm hoặc có khả năng tăng lên và không ổn định. Chính vì thế cần phải có sự điều chỉnh cho hàm tối ưu hóa. Chi tiết của sự điều chỉnh như sau:

- Epoch = 50: Điều chỉnh hàm tối ưu Adam với tham số $\text{learn_rate} = 0.0025$
- Epoch = 90: Điều chỉnh hàm tối ưu Adam với tham số $\text{learn_rate} = 0.002$
- Epoch = 150: Điều chỉnh hàm tối ưu Adam với tham số $\text{learn_rate} = 0.0015$
- Epoch = 170: Điều chỉnh hàm tối ưu Adam với tham số $\text{learn_rate} = 0.001$
- Epoch = 220: Điều chỉnh hàm tối ưu Adam với tham số $\text{learn_rate} = 0.0001$
- Epoch = 300: Điều chỉnh hàm tối ưu Adam với tham số $\text{learn_rate} = 0.00001$



Hình 4.3 Biểu đồ tỉ lệ chính xác trên tập xác nhận trong quá trình huấn luyện

Sau khi có các điều chỉnh kịp thời thì tỉ lệ chính xác của mô hình được đánh giá trên tập Validation được cải thiện một cách đáng kể và giúp cho độ chính xác của mô hình có xu hướng tăng lên dần theo số lượng các epoch.

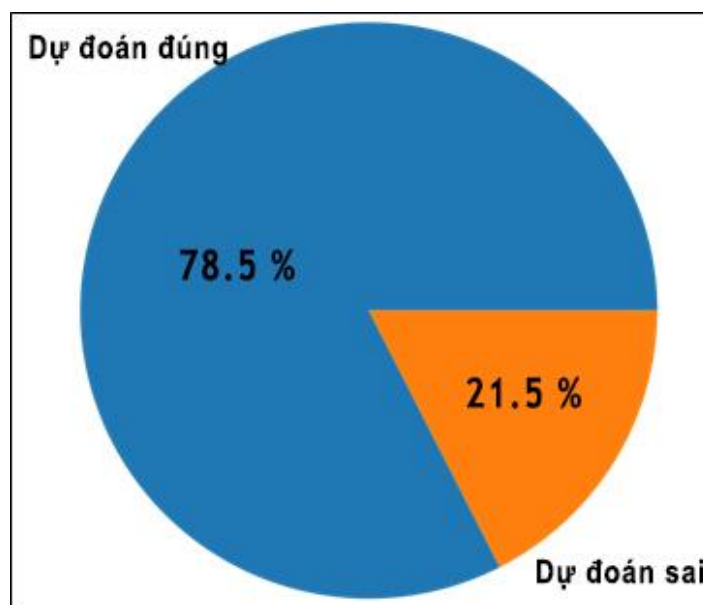
Biên độ chính xác của mô hình dựa trên tập Validation nằm trong vùng giá trị từ 0,615 đến 0.937. Giá trị mất mát cho tập huấn luyện giao động từ 1.508 về 0.231

4.1.4. Kiểm nghiệm mô hình

Tập dữ liệu kiểm nghiệm gồm có 200 ảnh bao gồm 7 nhãn trong đó có 6 nhãn bệnh đã huấn luyện và một nhãn không phải bệnh. Kết quả dự đoán đối với tập kiểm nghiệm có độ chính xác 82,5%.



Hình 4.4 Hình ảnh bộ dữ liệu được kiểm nghiệm



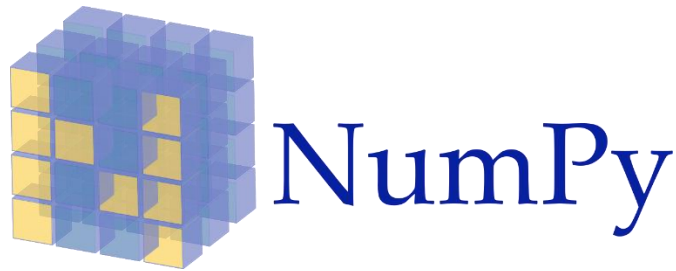
Hình 4.5 Tỷ lệ dự đoán trên tập kiểm nghiệm

4.2. Triển khai hệ thống Back-End

4.2.1. Thư viện cài đặt

Để dễ dàng trong việc triển khai hệ thống Back-End thì môi trường phát triển cần được cài đặt các thư viện hỗ trợ bên dưới:

- flask: Là thư viện chính để xây dựng hệ thống API phục vụ cho ứng dụng di động.
- numpy: Thư viện giúp chuyển đổi các trọng số và tham số kết quả trong quá trình dự đoán.



Hình 4.6 Thư viện NumPy

- pillow: Thư viện này hỗ trợ nhiều định dạng tập tin, và cung cấp khả năng xử lý hình ảnh trước và sau khi dự đoán.
- torch: Là framework được chọn xây dựng mô hình. Mô hình được sử dụng là mô hình ResNet 152 đã được huấn luyện từ trước.
- torchvision
- opencv-python: Thư viện giúp xử lý các Video thành từng ảnh để đưa vào dự đoán.
- psycopg2: Kết nối đến cơ sở dữ liệu
- Flask-SQLAlchemy
- Flask-Migrate
- python-dotenv

4.2.2. Danh sách API

Hệ thống gồm có 2 API chính:

- Tải dữ liệu người dùng để dự đoán:
 - URL: `http://167.71.199.248/app/upload`
 - Phương thức: POST
 - Dạng dữ liệu đưa lên: form-data/files
 - Dữ liệu nhận về:
JSON {
 "data": [
 {
 "class": type(str),
 "id": type(Long),
 "percent": type(float)
 }
],
 "success": type(boolean)
}
- Lấy thông tin về bệnh theo ID:
 - URL: `http://167.71.199.248/app/get-info/<id>`
 - Phương thức: GET
 - Dữ liệu nhận về:
JSON {
 "data": {
 "id": type(Long),
 "name": type(str),
 "expression": type(str),
 "prevention": type(str),
 "reason": type(str)
 },
 "success": type(boolean)
}

4.3. Triển khai hệ thống Front-End

Các lớp cơ sở đã được triển khai để hỗ trợ cho việc sử dụng các API được cung cấp từ phía máy chủ như sau:

- **AbstractHttpApi**: Đây là lớp trừu tượng để các lớp gọi Api thừa kế lại các chức năng chính bao gồm:
 - **executeHttpPost** : Gọi Api với phương thức POST
 - **executeHttpPut**: Gọi Api với phương thức PUT
 - **executeHttpGet**: Gọi Api với phương thức GET
- **BaseTask** : Đây là lớp có chức năng tạo ra một api task cơ bản trong đó có phương thức trừu tượng **callAiMethod()** để các task còn lại sử dụng và ghi đè lên.

```
public BaseTask(Context context, @Nullable ApiListener<Output> listener) {
    mListener = listener;
    mApi = new TaskApi(context);

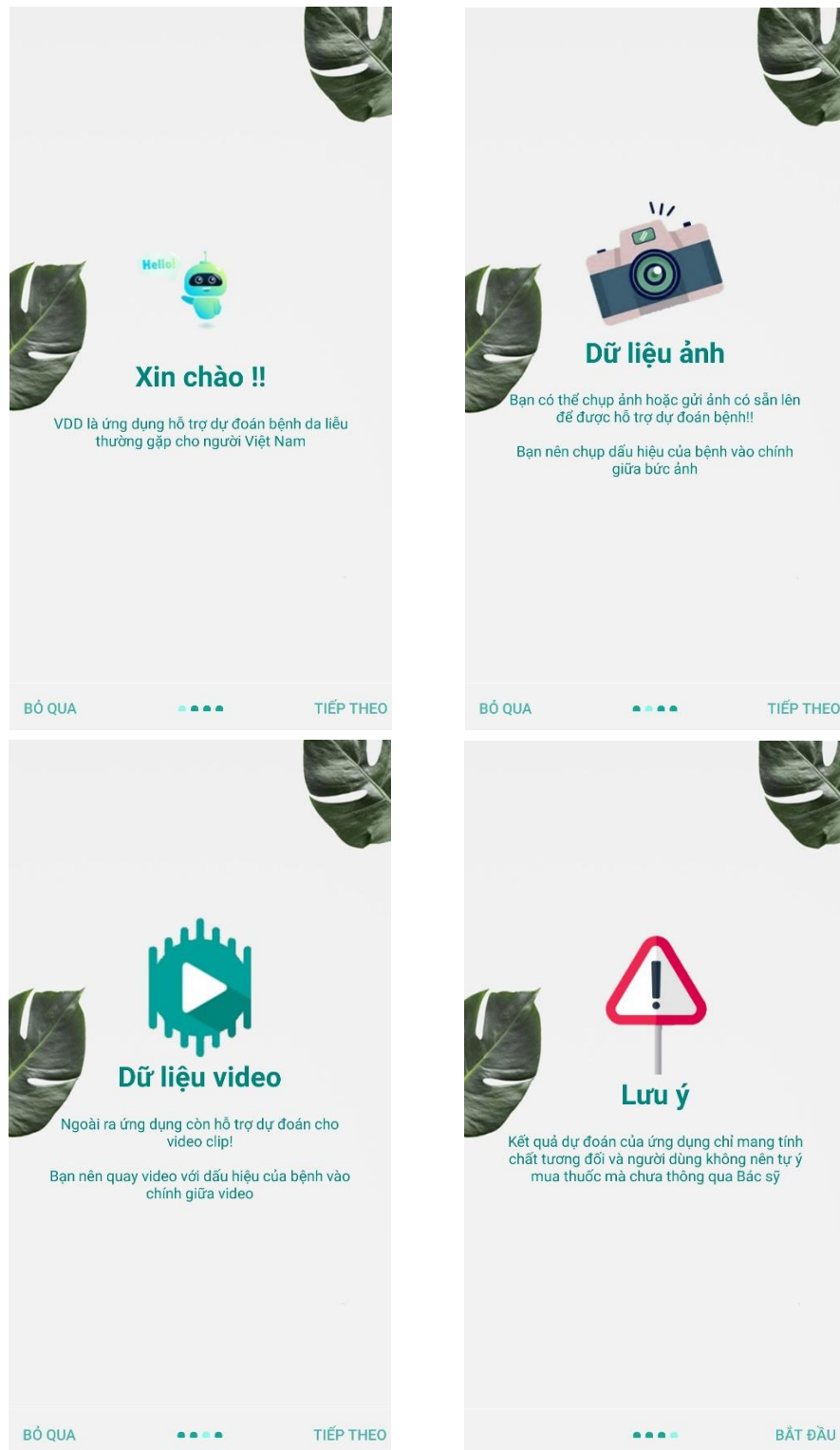
    if(SharedPreferencesHelper.getInstance(context).get(Constants.EXTRAX_TOKE
N_CODE) != null &&
    SharedPreferencesHelper.getInstance(context).get(Constants.EXTRAX_TOKEN_
CODE).length() > 0) {

        mApi.setCredentials(SharedPreferencesHelper.getInstance(context).get(Constants.EXTRAX_TOKEN_CODE));
    }
}
```

- **BaseOutput**: Là lớp cha cho các kết quả sau khi gọi đến Api, trong lớp này sẽ có các biến thông báo về việc gọi Api có thành công hay không, lỗi gửi về từ server là gì.

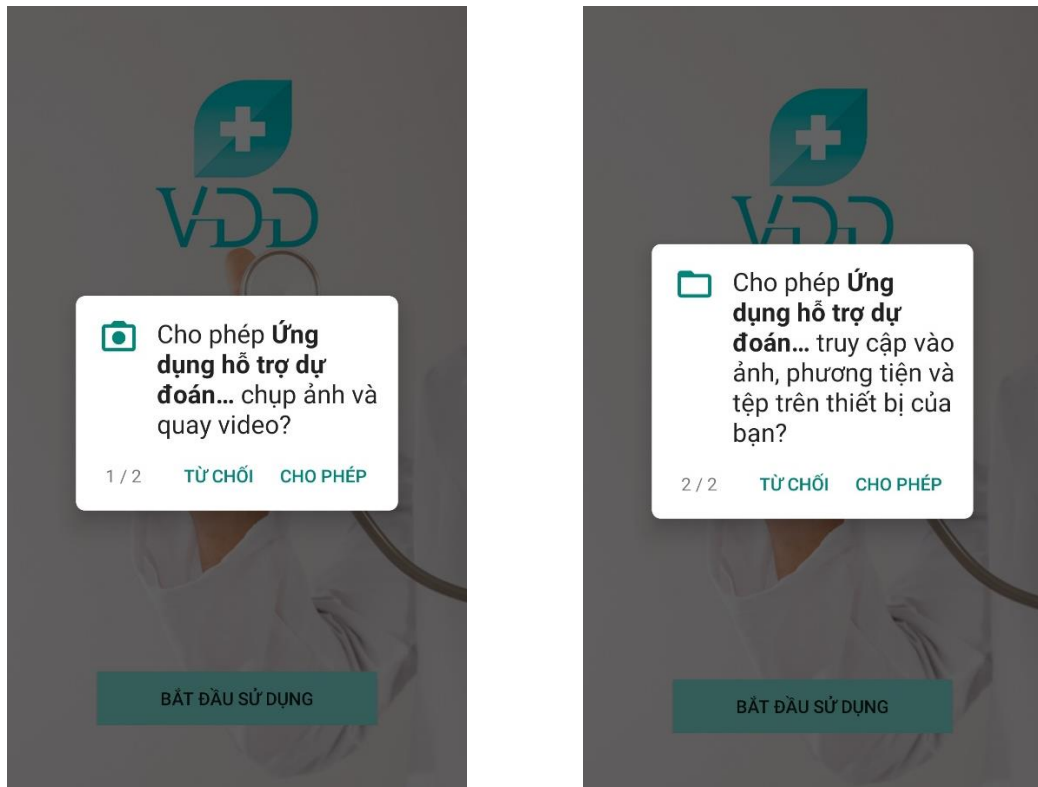
```
public class BaseOutput implements Serializable {
    @SerializedName("success")
    public boolean success;
    @SerializedName("errorCode")
    public String ErrorCode;
    @SerializedName("errorMessage")
    public String errorMessage;
}
```

4.4. Kết quả sản phẩm



Hình 4.7 Màn hình hướng dẫn sử dụng ứng dụng

Màn hình hướng dẫn: Màn hình hướng dẫn người sử dụng sẽ xuất hiện khi người dùng lần đầu cài đặt ứng dụng, trong màn hình này sẽ có các lưu ý, các loại dữ liệu ứng dụng sẽ hỗ trợ dự đoán cũng như cách thức sử dụng ứng dụng làm sao để được nhận hỗ trợ dự đoán bệnh da liễu đang gặp phải.



Hình 4.8 Màn hình cấp phép quyền truy cập

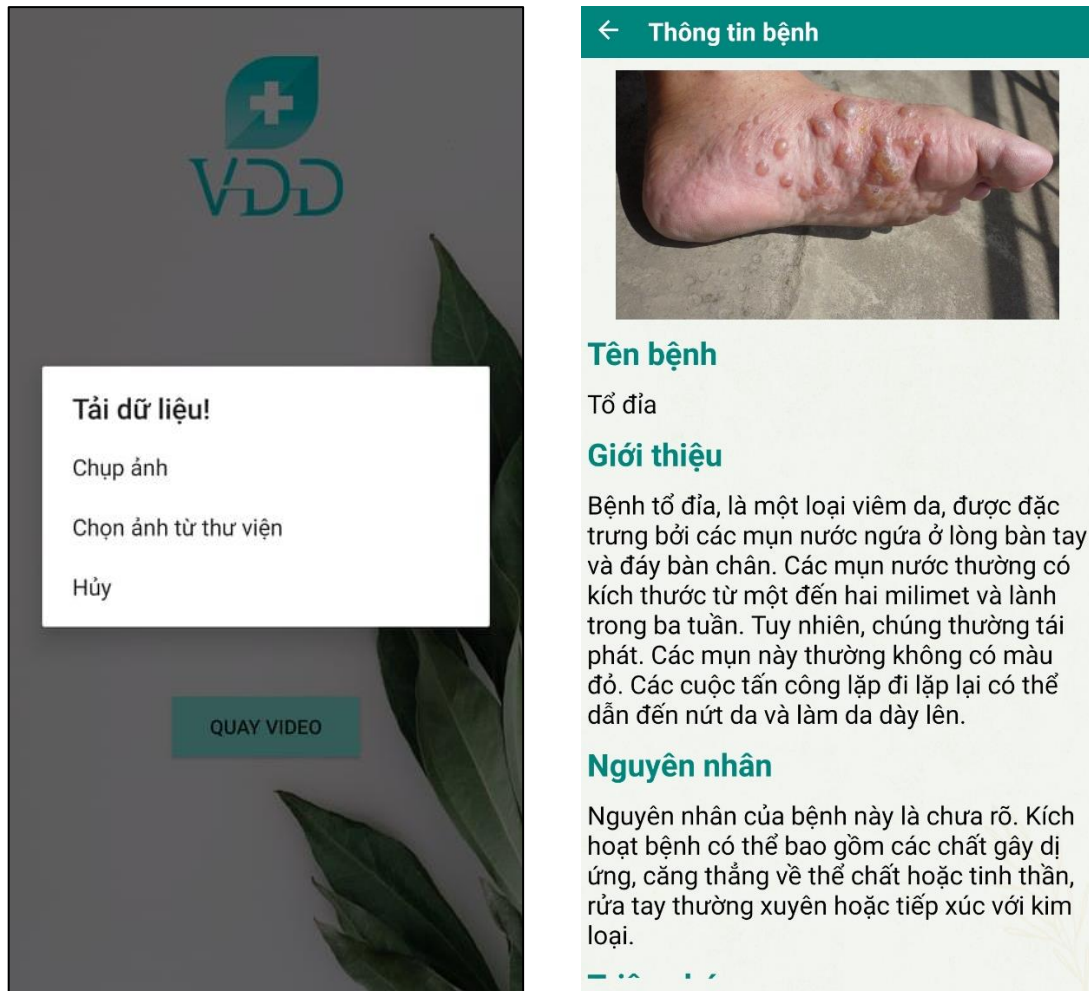
Ứng dụng có sử dụng đến các chức năng chụp ảnh, quay video, tải ảnh lên từ thư viện. Chính vì thế để ứng dụng chạy được thì cần phải có sự cấp phép đến từ người dùng cho phép ứng dụng được quyền sử dụng máy ảnh và truy cập vào nơi lưu trữ để tải ảnh lên từ thư viện.



Hình 4.9 Màn hình chính cho người sử dụng



Hình 4.10 Màn hình kết quả dự đoán



Hình 4.11 Màn hình tùy chọn tải dữ liệu và thông tin bệnh

Người dùng có thể tải dữ liệu lên theo nhiều cách thức và theo 2 loại dữ liệu chính là ảnh và video. Sau khi được dự đoán bệnh thì người dùng có thể đọc thêm chi tiết về bệnh da liễu được dự đoán. Thông tin bệnh bao gồm giới thiệu, nguyên nhân, triệu chứng và cách phòng tránh bệnh da liễu đó.

4.5. Kết luận và hướng phát triển

4.5.1. Đạt được

Thông qua đồ án thì đã giúp cá nhân sinh viên thực hiện:

- Hiểu rõ hơn về học các khái niệm cơ bản trong học máy và biết áp dụng để giải quyết các vấn đề thực tế đang gặp phải
- Thu thập và xây dựng bộ dữ liệu về các bệnh da liễu thường gặp ở Việt Nam cũng như đóng góp một phần xây dựng dữ liệu cho việc học tập cũng như thử nghiệm
- Phân triển khai hệ thống Back- End một cách linh hoạt bằng Flask python và xây dựng cơ sở dữ liệu thông tin về các bệnh da liễu thường gặp.
- Nắm rõ được kiến thức nghiệp vụ về triệu chứng, nguyên nhân và cách phòng ngừa cho các bệnh da liễu được lựa chọn tìm hiểu.
- Hiểu rõ được cấu trúc mạng nơ-ron sâu thông qua mô hình đã được huấn luyện.
- Biết được cách thức giải quyết các vấn đề gặp phải trong quá trình huấn luyện mô hình để dự đoán, hiểu rõ cấu trúc của mô hình ResNet cũng như các kỹ thuật học chuyển đổi và tăng cường dữ liệu ảnh.
- Tận dụng lượng dữ liệu của người dùng tải lên để tăng cường cho bộ dữ liệu của ứng dụng bằng cách phân loại gán nhãn và tiền xử lý.
- Xây dựng ứng dụng di động với giao diện dễ nhìn và dễ sử dụng cho người dùng.
- Đây là một dự án thực hiện đúng quy trình phát triển một sản phẩm công nghệ.
- Cá nhân sinh viên thực hiện đúng tiến độ và làm việc trên tinh thần tự giác cao.

4.5.2. Chưa đạt

- Giao diện ứng dụng chưa thực sự bắt mắt và thu hút người dùng.
- Ứng dụng di động còn đơn giản và ít các chức năng đi kèm theo
- Số lượng kiểu dữ liệu nhận vào của hệ thống còn hạn chế.
- Số lượng bệnh hỗ trợ dự đoán còn ít và khá hạn chế về dữ liệu
- Ứng dụng chưa có nhiều số lượng người sử dụng

4.5.3. Hướng phát triển

Đồ án tuy đã giải quyết được những yêu cầu đặt ra ban đầu, song vẫn có thể phát triển thêm nhiều tính năng cũng như giải quyết một số vấn đề còn tồn đọng, chẳng hạn như:

- Tăng cường cơ sở dữ liệu ảnh và các nhãn bệnh để hỗ trợ cho việc tăng tỉ lệ chính xác khi dự đoán
- Người dùng có thể liên hệ với các trung tâm y tế gần nhất sau khi được dự đoán.
- Hỗ trợ cho các trung tâm y tế trong việc khám và chẩn đoán bệnh da liễu.
- Đóng góp bộ dữ liệu lớn cho việc học tập cũng như nghiên cứu y khoa về da liễu
- Tối ưu hóa phần giao diện hơn, giúp tăng trải nghiệm người dùng.
- Cải thiện mô hình huấn luyện để tăng tỉ lệ chính xác.
- Tích hợp nhiều tính năng về như tin tức, nghiên cứu khoa học liên quan đến y học cho người dùng thêm trải nghiệm về ứng dụng.
- Kết hợp thêm nhiều yếu tố để tăng cường tỉ lệ dự đoán của mô hình như các mô tả từ người dùng, các thông tin về lịch sử bệnh lý...
- Áp dụng và cải thiện mô hình huấn luyện bằng các thuật toán mới với độ chính xác cao hơn.

4.6. Kết chương

Chương 4 trình bày một số yêu cầu đối với môi trường triển khai việc huấn luyện mô hình, xây dựng ứng dụng và quá trình vận hành ứng dụng di động trong thực tế. Giúp người đọc có cái nhìn tổng quan và cụ thể hơn về ứng dụng di động hỗ trợ dự đoán bệnh da liễu thường gặp ở người Việt Nam. Ngoài ra còn là việc đánh giá kết quả đạt được và hướng phát triển ứng dụng trong tương lai.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Nguyễn Thanh Tuấn, Deep Learning cơ bản, Tài liệu lưu hành nội bộ, 2019.
- [2] Võ Tuấn Duy, Python cơ bản, Ebook.
- [3] GS. TS Nguyễn Văn Út, Giáo trình bài giảng bệnh da liễu, 2015
- [4] BS. Việt Hà, BS. Phan Hoa, BS. Bích Thủy, BS. Hải Yến, Atlas các bệnh da liễu thường gặp, 2011

Tiếng Anh

- [5] Birian Driscoll, Nitin Gupta, Robert Vettor, Zeeshan Hirani, Larry Tenny, Entity Framework 6 Recipes, 2012.
- [6] Chistian Nagel, Jay Glynn, Morgan Skinner, Bill Evjen, Karli Watson, Proffesional C# 2012 And .NET 4.5, 2012.
- [7] Aurélien Géron, Hands-Machine-Learning-Scikit-Learn-TensorFlow, 2017.
- [8] Towards Data Science, < <https://towardsdatascience.com>>.
- [9] Li-sheng Wei, Quan Gan, Tao Ji, Skin Disease Recognition Method Based on Image Color and Texture Features, <<https://doi.org/10.1155/2018/8145713>>, Research article, 2018.
- [10] Manu Goyal, Saeed Hassanpour , Moi Hoon Yap, Region of Interest Detection in Dermoscopic Images for Natural Data-augmentation, Article, 2019.
- [11] Haofu Liao, A Deep Learning Approach to Universal Skin Disease Classification, Project Report
- [12] Klaus Wolff, Richard Allen Johnson, Atlas Fitzpatrick's color atlas and synopsis of clinical dermatology, 2009
- [13] University of Birmingham, University Hospitals Birmingham NHS Foundation Trust., The Lancet Digital Health, 2019

PHỤ LỤC 1

Các bệnh được lựa chọn trong tập dữ liệu bệnh cần phân loại

1. Bệnh hắc bào

1.1. Giới thiệu

Hắc bào là tình trạng được gây ra bởi các loại nấm khác nhau. Những loại nấm khác nhau ảnh hưởng đến những vùng khác nhau của cơ thể. Các chứng bệnh nhiễm trùng nấm này được đặt tên theo các bộ phận mà bệnh nấm xuất hiện, chẳng hạn như nấm da toàn thân, nấm da đầu, nấm da chân, nấm da đùi, và nấm móng tay



Hình phụ lục 1.1 Dấu hiệu của bệnh hắc bào

1.2. Nguyên nhân

Nguyên nhân gây ra các chứng bệnh nấm da là do một số loại nấm nhỏ chỉ nhìn được dưới kính hiển vi có tên gọi chung là dermatophytes. Các loại nấm phổ biến là malassezia furfur, trichophyton, microsporum và epidermophyton. Nấm gây ra chứng bệnh nấm da thường rất nhỏ, chỉ có thể nhìn thấy được dưới kính hiển vi và có khả năng phát triển trong môi trường nóng ẩm và ẩm ướt. Những người có da dầu hoặc có thay đổi về hormone trong cơ thể làm cho hệ miễn dịch yếu đi cũng dễ bị nấm da hơn bình thường.

1.3. Triệu chứng:

Triệu chứng thường gặp là ngứa. Đôi khi có xuất hiện tình trạng tróc vảy hoặc bong tróc ở da.

Trên cơ thể, bệnh hắc bào bắt đầu dưới dạng các mảng nổi nhẹ, có hình vòng hoặc bầu dục, có màu đỏ hoặc nâu, xuất hiện ở da và gây ngứa. Vùng da này có thể xuất hiện thành từng mảng, bề mặt còn có các vảy có cạnh sắc cứng hoặc các mụn nước nhỏ phòng rộp.

Trong thời gian xuất hiện mẩn đỏ và vảy da, bệnh có thể lây truyền cho những người khác. Việc cào hoặc gãi ngứa có thể gây ra tình trạng sưng, chảy nước và tạo điều kiện cho nhiễm trùng.

2. Bệnh lang ben

2.1. Giới thiệu

Lang ben là một bệnh nhiễm nấm *Pityrosporum ovale* ngoài da thường gặp. Bệnh có xu hướng lây lan dễ dàng từ người này sang người khác trong môi trường nhiệt đới nóng ẩm thông qua tiếp xúc hoặc dùng chung đồ dùng sinh hoạt cá nhân (quần áo, khăn tắm...). Bệnh không gây nguy hiểm nhưng có thể ảnh hưởng đến thẩm mỹ bởi các mảng da sáng màu (mất sắc tố)



Hình phụ lục 1.2 Dấu hiệu của bệnh lang ben

2.1. Nguyên nhân

Nấm *Pityrosporum ovale* phát triển trên bề mặt da. Nấm *Pityrosporum ovale* tác động vào lớp biểu bì làm sắc tố dưới da thay đổi, tạo nên các vùng da giảm hoặc mất sắc tố (trắng hơn hẳn so với các vùng da xung quanh).

2.2. Triệu chứng

Xuất hiện các dát từ từ trên da, tăng dần về số lượng và kích thước. Da có màu khác so với vị trí xung quanh (có thể sáng hoặc tối hơn), có thể màu trắng, hồng hoặc nâu. Vị trí thường gặp: cổ, ngực, lưng và hai cánh tay. Tuy nhiên có thể gặp ở bất cứ vị trí nào trên cơ thể. Da có thể gây ngứa, tăng lên khi ra nắng hoặc đổ mồ hôi. Nhiễm nấm men làm da không tiếp xúc được với ánh nắng mặt trời.

3. Tổ đũa

3.1. Giới thiệu

Bệnh tổ đỉa là một loại viêm da được đặc trưng bởi các mụn nước ngứa trên lòng bàn tay và lòng bàn chân. Các mụn nước thường có kích thước là 1-2 mm và lành sau hơn ba tuần. Da ít khi xuất hiện nhưng thường tái phát. Bệnh tổ đỉa xuất hiện lặp đi lặp lại có thể gây ra những vết nứt và dày da.



Hình phụ lục 1.3 Dấu hiệu của bệnh tổ đỉa

3.2. Nguyên nhân

Nguyên nhân của bệnh này là chưa rõ. Kích hoạt bệnh có thể bao gồm các chất gây dị ứng, căng thẳng về thể chất hoặc tinh thần, rửa tay thường xuyên hoặc tiếp xúc với kim loại. Chẩn đoán thường dựa trên bề ngoài và các triệu chứng.

3.3. Triệu chứng:

Ngứa ở lòng bàn tay hoặc lòng bàn chân, theo sau sự phát triển đột ngột của mụn nước nhỏ ngứa dữ dội ở hai bên ngón tay, lòng bàn tay hoặc bàn chân. Những mụn nước này thường được mô tả là có hình dạng "bột sắn". Sau một vài tuần, các mụn nước nhỏ cuối cùng biến mất khi lớp da trên cùng rơi ra. Những nốt mụn trồi lên này không xảy ra ở nơi nào khác trên cơ thể. Các nốt sần có thể là đối xứng.

4. Bệnh vẩy nến

4.1. Giới thiệu

Bệnh vẩy nến là một bệnh viêm da mãn tính rất phổ biến. Ở trạng thái bình thường, các tế bào da cũ sau khi chết đi sẽ bong ra và được thay thế bởi các tế bào da mới. Nhưng đối với bệnh nhân mắc vẩy nến, quá trình trên diễn ra nhanh gấp 10 lần do hiện tượng tăng sinh tế bào, khiến các tế bào da cũ và mới không kịp thay đổi, tích tụ lại một chỗ tạo thành những mảng dày, có vảy trắng hoặc bạc.



Hình phụ lục 1.4 Dấu hiệu của bệnh vẩy nến

4.2. Nguyên nhân

Đến nay vẫn chưa được các nhà khoa học chứng minh rõ ràng nhưng có một điều chắc chắn là bệnh này có liên quan đến rối loạn đáp ứng miễn dịch qua trung gian tế bào và dấu ấn của cytokine. Theo đó, các tế bào lympho T trong cơ thể bệnh nhân có thể nhầm lẫn các tế bào khỏe mạnh là kẻ thù và tấn công, làm chúng bị tổn thương

4.3. Triệu chứng

Xuất hiện những mảng dày, đỏ được bao phủ bởi các vảy trắng hoặc bạc. Bên cạnh đó, tùy thuộc vào vị trí xuất hiện và đặc điểm của các tổn thương

5. Bệnh zona (giời leo)

5.1. Giới thiệu

Bệnh zona là một bệnh thường kèm theo những đau đớn kéo dài từ 6 tháng đến vài năm. Để hạn chế nguy cơ mắc bệnh, hiện nay đã có vắc xin tiêm phòng có thể phòng được cả bệnh thủy đậu và bệnh giời leo.



Hình phụ lục 1.5 Dấu hiệu của bệnh zona

5.2. Nguyên nhân

Giời leo là bệnh viêm dây thần kinh do virus nhóm Herpes gây ra, có thể xuất hiện ở bất cứ đâu trên cơ thể con người nhưng thường gặp nhất là ở vùng liên sườn,

gần tai và đuôi trong. Ngoài ra, bệnh còn xuất hiện ở bụng, cổ, vai, mặt, lưng, nguy hiểm nhất và khó điều trị nhất là ở hố mắt.

5.3. Triệu chứng

Tổn thương da đau rất như bị trầy xước hay bỏng, ngứa râm ran giống bị kim châm, thường xuất hiện ở các vùng da bị hở hoặc có nhiều trường hợp rải rác khắp người. Xuất hiện mụn nước cấp tính ở những vùng nhiễm bệnh. Thời gian đầu các mụn sẽ nhỏ li ti sau đó lan rộng thành từng mảng. Sốt nhẹ do mệt mỏi vì phải chịu đau đớn cả bên trong và bên ngoài. Ngoài ra còn có các triệu chứng khác như: giảm thính lực một bên tai, mất vị giác phần trước lưỡi, chóng mặt, hoa mắt, ù tai và có thể yếu một bên mắt. Tình trạng này khiến người bệnh bị chảy nước mũi, thức ăn sẽ bị mắc kẹt ở nửa bên bị ảnh hưởng và gây ra tình trạng khô mắt.

6. Bệnh mụn cóc

6.1. Giới thiệu

Mụn cóc là bệnh da liễu thường gặp do virus HPV (Human Papilloma Virus) gây ra. Virus HPV này xâm nhập vào da qua những vết trầy xước bên ngoài, tạo thành những u nhỏ lành tính, bề mặt sần sùi, gọi là mụn cóc.



Hình phụ lục 1.6 Dấu hiệu của bệnh mụn cóc

6.2. Nguyên nhân

Virus human papillomavirus (HPV) xâm nhập vào cơ thể qua các vết trầy xước hay vết rách trên da. Khi vào trong cơ thể, các virus phát triển và kích thích các tế bào trên bề mặt da, gây ra mụn cóc.

6.3. Triệu chứng

Mụn cóc thường gây khó chịu trên da. Đôi khi mụn cóc gây ra chảy máu nếu mụn xuất hiện ở trên mặt hay đầu. Mụn cóc bàn chân thường rộp và sưng lên, có thể gây đau và dễ vỡ khi bước đi. Mụn cóc quanh móng chân có thể làm nứt nẻ và đau.