

BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC SAO ĐỎ



VŨ THẾ ANH - PHẠM PHONG BẰNG

**ĐỒ ÁN TỐT NGHIỆP**

Ngành: CÔNG NGHỆ THÔNG TIN

Họ và tên sinh viên: Vũ Thế Anh  
Phạm Phong Bằng

Lớp, khoá: DK10 – CNTT, 2019

NGÀNH: CÔNG NGHỆ THÔNG TIN

HẢI DƯƠNG – NĂM 2023

BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC SAO ĐỎ



**ĐỒ ÁN TỐT NGHIỆP**

Ngành: CÔNG NGHỆ THÔNG TIN

TÊN ĐỀ TÀI: XÂY DỰNG MỘT MÔ HÌNH DỮ LIỆU BIG DATA DÙNG CHO  
PHÂN TÍCH BỆNH Ở NGƯỜI

Họ và tên sinh viên: Vũ Thế Anh

Phạm Phong Bằng

Lớp, khóa: DK10 – CNTT, 2019

Giảng viên hướng dẫn: Nguyễn Thị Ánh Tuyết

HẢI DƯƠNG – NĂM 2023

### LỜI CAM ĐOAN

Tôi xin cam đoan các kết quả đưa ra trong đồ án tốt nghiệp này là các kết quả thu được trong quá trình nghiên cứu, thực nghiệm của tôi dưới sự hướng dẫn của cô Nguyễn Thị Ánh Tuyết, không sao chép bất kỳ kết quả nghiên cứu nào của các tác giả khác.

Nội dung nghiên cứu có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn tài liệu đã được liệt kê trong danh mục các tài liệu tham khảo.

Nếu sai tôi xin chịu mọi hình thức kỷ luật theo quy định.

Hải Dương, ngày 30 tháng 11 năm 2023

Sinh viên thực hiện

(Ký, ghi rõ họ tên)

Vũ Thé Anh

Phạm Phong Bằng

## LỜI CẢM ƠN

Qua quá trình học tập và rèn luyện tại Trường Đại học Sao Đỏ, dưới sự dạy bảo của các thầy cô giáo trong trường nói chung và các thầy cô giáo khoa Công Nghệ Thông Tin nói riêng, và trải qua thời gian nghiên cứu thực hiện đồ án tốt nghiệp với sự nỗ lực tìm hiểu của bản thân chúng em đã hoàn thành đồ án của mình.

Chúng em xin gửi lời cảm ơn đến cô Nguyễn Thị Ánh Tuyết đã trực tiếp hướng dẫn tận tình và giúp đỡ em có thể hoàn thành tốt đồ án tốt nghiệp này đúng thời hạn.

Chúng em cũng xin chân thành bày tỏ lòng biết ơn sâu sắc đến các thầy cô trong trường Đại học Sao Đỏ, đặc biệt là các thầy cô trong bộ môn Công Nghệ Thông Tin đã truyền thụ những kiến thức quý báu cho em trong những năm học qua.

Do thời gian thực tập và nghiên cứu làm đồ án còn hạn chế nên đồ án của em khó tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và góp ý chân thành của quý thầy cô và các bạn để chúng em có thể được hoàn thiện hơn nữa đồ án của mình.

Chúng em xin chân thành cảm ơn!

Kính chúc các thầy cô sức khỏe và thành công!

Hải Dương, ngày 05 tháng 12 năm 2023

Sinh viên thực hiện

(Ký, ghi rõ họ tên)

Vũ Thé Anh

Phạm Phong Bằng

**MỤC LỤC**

<b>LỜI CAM ĐOAN</b> .....	i
<b>LỜI CẢM ƠN</b> .....	ii
<b>MỤC LỤC</b> .....	iii
<b>DANH MỤC BẢNG BIỂU</b> .....	vi
<b>DANH MỤC HÌNH VẼ</b> .....	vii
<b>MỞ ĐẦU</b> .....	1
<b>CHƯƠNG 1. CƠ SỞ LÝ THUYẾT</b> .....	1
<b>1.1. Tổng quan về Big data</b> .....	3
1.1.1. Khái niệm .....	3
1.1.2. Các đặc trưng của Big data .....	5
1.1.3. Các thành phần của Big data .....	6
1.1.4. Tầm quan trọng của Big data .....	7
1.1.5. Ứng dụng Big data.....	8
<b>1.2. Sơ lược về mạng nơron</b> .....	17
1.2.1. Giới thiệu về mạng nơron nhân tạo .....	17
1.2.2. Cấu trúc mạng nơron nhân tạo .....	18
1.2.3. Phân loại các mạng nơron .....	19
1.2.4. Huấn luyện mạng .....	20
1.2.5. Mạng neural tích chập .....	21
a, Cách thức hoạt động của CNN .....	21
1.2.6. Mô hình học máy ResNet-50 .....	27
<b>1.3. Tổng quan về bệnh Alzheimer</b> .....	30
1.3.1. Khái niệm bệnh Alzheimer .....	30
1.3.2. Các triệu chứng của bệnh Alzheimer .....	31
1.3.3. Quy trình chẩn đoán bệnh trong y học .....	32
1.3.4. Quy trình điều trị .....	34
<b>CHƯƠNG 2. MÔ HÌNH PHÂN TÍCH DỮ LIỆU BỆNH ALZHEIMER</b> .....	37
<b>2.1. Kiến trúc mô hình</b> .....	37
2.1.1. Dữ liệu .....	37
2.1.2. Tiền xử lý dữ liệu .....	37
2.1.3. Đánh giá hiệu suất .....	37
2.1.4. Kiểm soát Overfitting và Underfitting .....	37
2.1.5. Phân tích đặc trưng .....	37

2.1.7. Phiên bản và triển khai .....	38
<b>2.2. Mô hình ResNet-50 .....</b>	<b>38</b>
2.2.1. Ý tưởng .....	38
2.2.2. Kiến trúc mô hình ResNet-50 .....	38
2.2.3. Tính toán Loss, IoU và DSC của mô hình ResNet-50 .....	38
<b>2.3. Chuẩn bị mô hình .....</b>	<b>40</b>
2.3.1. Chuẩn bị dữ liệu .....	40
2.3.2. Mô hình học máy .....	40
2.3.3. Hàm dự đoán .....	40
2.3.4. Giao diện người dùng với gradio .....	40
2.3.5. Quản lý trạng thái mô hình .....	40
<b>2.6. Cách thiết lập và vận hành ResNet-50 .....</b>	<b>41</b>
2.6.1. Thiết lập ResNet-50 .....	41
2.6.2. Vận hành ResNet-50 .....	41
<b>CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG .....</b>	<b>43</b>
<b>3.1. Công cụ .....</b>	<b>43</b>
3.1.1. Anaconda .....	43
3.1.2. Python .....	44
3.1.3. PyCharm .....	45
3.1.4. Gradio .....	46
<b>3.2. Quy trình phân tích bệnh Alzheimer .....</b>	<b>48</b>
<b>3.3. Cơ sở dữ liệu .....</b>	<b>48</b>
<b>3.4. Xây dựng công dụng chức năng của code .....</b>	<b>49</b>
<b>3.5. Xây dựng chương trình học máy .....</b>	<b>50</b>
3.5.1. Phân dữ liệu .....	50
3.5.2. Mô hình .....	52
3.5.3. Trọng số .....	52
3.5.4. Dự đoán .....	52
<b>3.6. Thử nghiệm chương trình và kết quả thử nghiệm .....</b>	<b>53</b>
<b>KẾT LUẬN .....</b>	<b>58</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>59</b>

## **DANH MỤC BẢNG BIỂU**

Bảng 1.1. So sánh các phương pháp Zero-padding .....	<b>24</b>
Bảng 1.2. So sánh độ phức tạp của mô hình .....	<b>25</b>
Bảng 1.3. Biến thể của ReLU .....	<b>26</b>

Đại học Sao Đỏ

## DANH MỤC HÌNH VẼ

Hình 1.1. Mô hình Big data .....	3
Hình 1.2. Đặc trưng 6V của Big data .....	5
Hình 1.3. Kiến trúc Big data .....	6
Hình 1.4. Mô hình cơ sở hạ tầng cần thiết để hỗ trợ Big data .....	7
Hình 1.5. Hệ thống phân tích bigdata trong chính trị .....	8
Hình 1.6. Hệ thống phân tích tử vong của Cơ quan An toàn Giao thông đường cao tốc quốc gia .....	9
Hình 1.7. Big data trong phân tích hệ thống ngân hàng .....	10
Hình 1.8. Big data phân tích trong mô hình thương mại điện tử .....	10
Hình 1.9. Ảnh dự báo thời tiết .....	11
Hình 1.10. Hình ảnh phân tích liệt kê thu thập dữ liệu trong y tế .....	11
Hình 1.11. Tìm kiếm thông tin trong mảng dữ liệu Big data .....	12
Hình 1.12. Thảo luận bệnh án của bệnh nhân bằng dữ liệu đã lưu trữ .....	14
Hình 1.13. Đưa ra những dự đoán cho việc chăm sóc sức khỏe .....	15
Hình 1.14. Phân tích, dự đoán góp phần nâng cao tỷ lệ điều trị thành công bệnh .. lý	16
Hình 1.15. Cấu trúc của nơron sinh học ..	17
Hình 1.16. Kiến trúc mạng nơron ..	18
Hình 1.17. Sơ đồ cấu trúc một số mạng nơron ..	19
Hình 1.18. Kiến trúc truyền thông của một mạng CNN ..	21
Hình 1.19. Ví dụ mô hình CNN ..	21
Hình 1.20. Minh họa Convolution ..	21
Hình 1.21. Minh họa tầng kết nối đầy đủ ..	22
Hình 1.22. Minh họa mạng CNN ..	22
Hình 1.23. Minh họa các bộ lọc có kích thước khác nhau ..	23
Hình 1.24. Minh họa độ trượt S ..	23
Hình 1.25. Kiến trúc residual Block ..	27
Hình 1.26. Mô hình Giao quan hệ với hiện đại với Deep learning ..	28
Hình 1.27. Mô hình ResNet-50 ..	29
Hình 1.28. Biểu đồ học máy của ResNet-50 ..	29
Hình 1.29. Minh họa bệnh nhân Alzheimer ..	30
Hình 1.30. 10 dấu hiệu của bệnh Alzheimer ..	31

Hình 1.31. Đối tượng có nguy cơ mắc bệnh Alzheimer .....	31
Hình 1.32. Các bước chẩn đoán bệnh .....	32
Hình 1.33. Minh họa hệ thần kinh .....	33
Hình 1.34. Liệt kê các loại thuốc úc chế bệnh .....	35
Hình 2.1. Khai báo hàm tính toán Loss .....	38
Hình 2.2. Hàm tính toán IoU .....	39
Hình 2.3. Hàm tính toán DSC .....	39
Hình 2.4. Xây dựng mô hình tuỳ chỉnh ResNet-50 .....	41
Hình 2.5. Câu lệnh Load Pretrained .....	41
Hình 2.6. Câu lệnh thay đổi lớp Fully .....	42
Hình 3.1. Biểu tượng Anaconda .....	43
Hình 3.2. Biểu tượng Python .....	44
Hình 3.3. Biểu tượng PyCharm .....	45
Hình 3.4. Biểu tượng Gradio .....	46
Hình 3.5. Sơ đồ quy trình phân tích bệnh Alzheimer .....	48
Hình 3.6. Bộ dữ liệu hình ảnh MRI não người bệnh Alzheimer .....	49
Hình 3.7. Kho dữ liệu class MildDemented .....	50
Hình 3.8. Kho dữ liệu class ModerateDemented .....	51
Hình 3.9. Kho dữ liệu class NonDemented .....	51
Hình 3.10. Kho dữ liệu class VeryMildDemented .....	52
Hình 3.11. Một số ảnh thử sai cho chương trình .....	52
Hình 3.12. Màn hình cửa sổ CMD .....	53
Hình 3.13. Nhập lệnh truy cập tới torch-gpu .....	54
Hình 3.14. Xuất hiện đoạn host để chạy giao diện nhận dạng .....	54
Hình 3.15. Giao diện người dùng .....	55
Hình 3.16. Kết quả sau khi nhận dạng .....	55
Hình 3.17. Kết quả nhận dạng ảnh não người bị bệnh Alzheimer (1) .....	56
Hình 3.18. Kết quả nhận dạng ảnh não người không bị bệnh Alzheimer .....	56
Hình 3.19. Kết quả nhận dạng ảnh não người bị xuất huyết não thùy trái trên .....	56
Hình 3.20. Kết quả nhận dạng ảnh não người bị xuất huyết não thùy trái dưới .....	56
Hình 3.21. Kết quả nhận dạng ảnh não người bị xuất huyết não chụp X quang cắt ngang .....	57

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Hiện nay việc già hoá dân số đang là vấn đề gây ra nhiều khó khăn cho các nước trên thế giới do nó sẽ dẫn đến việc thiếu nguồn lao động cũng như gián tiếp ảnh hưởng đến giá trị kinh tế đất nước. Và song song với việc già hoá dân số thì các loại bệnh liên quan cũng ngày càng gia tăng. Đặc biệt là những loại bệnh được liệt vào danh sách những căn bệnh tuổi già khó có thể tránh khỏi. Trong số đó có căn bệnh **Alzheimer** – Căn bệnh gây mất trí nhớ, không kiểm soát được hành vi của bản thân được xếp vào những căn bệnh có triệu chứng nguy hiểm. Tuy nhiên hiện nay bệnh nhân mắc căn bệnh **Alzheimer** đang có xu hướng trẻ hoá khi mà những người trẻ mới ngoài 20 tuổi đã mắc phải. Và với việc xử lý dữ liệu cho căn bệnh đang ngày càng lớn thì việc áp dụng công nghệ và y học cũng đang rất phát triển, lưu tâm.

**Big data** trong y tế là một thuật ngữ được sử dụng để mô tả khối lượng thông tin khổng lồ được tạo ra từ việc áp dụng công nghệ kỹ thuật số để thu thập hồ sơ của bệnh nhân và giúp quản lý hoạt động của bệnh viện, nếu không thì quá lớn và phức tạp đối với các công nghệ truyền thống.

Việc áp dụng phân tích dữ liệu lớn trong y tế mang lại rất nhiều kết quả tích cực và cũng là cứu sống con người. Về bản chất, **Big data** đề cập đến lượng thông tin khổng lồ được tạo ra bởi quá trình số hóa mọi thứ, được tổng hợp và phân tích bằng các công nghệ cụ thể. Được áp dụng cho y tế, nó sẽ sử dụng dữ liệu sức khỏe cụ thể của một dân số (hoặc của một cá nhân cụ thể) và có khả năng giúp ngăn ngừa dịch bệnh, chữa bệnh, cắt giảm chi phí, ...

Trong nhiều năm, việc thu thập một lượng lớn dữ liệu cho mục đích y tế rất tốn kém và mất thời gian. Với công nghệ luôn cải tiến ngày nay, việc thu thập dữ liệu đó trở nên dễ dàng hơn, tạo báo cáo chăm sóc sức khỏe toàn diện và chuyển đổi chúng thành những thông tin chi tiết quan trọng có liên quan, sau đó có thể sử dụng để cung cấp dịch vụ chăm sóc tốt hơn. Đây là mục đích của phân tích dữ liệu y tế: sử dụng các phát hiện dựa trên dữ liệu để dự đoán và giải quyết vấn đề trước khi quá muộn, nhưng cũng đánh giá các phương pháp điều trị nhanh hơn, theo dõi tốt hơn hàng tồn kho, thu hút bệnh nhân nhiều hơn trong sức khỏe của họ và trao quyền cho họ với các công cụ để làm như vậy.

### 2. Mục đích nghiên cứu

Nghiên cứu ra một chương trình có thể phân tích, nhận dạng được bệnh Alzheimer. Xử lý được những dữ liệu lớn liên quan đến căn bệnh Alzheimer.

### 3. Phương pháp nghiên cứu

Sử dụng các dữ liệu đã nghiên cứu kết hợp với ngôn ngữ lập trình để xây dựng một ứng dụng phân tích bệnh Alzheimer.

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

- Đối tượng nghiên cứu: Là những hình ảnh được tổng hợp từ nhiều nguồn thông tin trên mạng cũng như những báo cáo có hình ảnh chụp về bộ não con người.

- Phạm vi nghiên cứu: Trên đây chúng tôi chỉ giới hạn được với những cơ sở dữ liệu mẫu. Nghiên cứu các quy trình học máy và xử lý Bigdata. Chưa được thực hiện trên các kho dữ liệu có quy mô lớn hay trên các trang thông tin mật.

#### 5. Ý nghĩa đề tài

- Về khoa học: Kết quả nghiên cứu của đề tài góp phần vào công cuộc phát hiện bệnh lý sớm ở những người có khả năng mắc căn bệnh Alzheimer. Ứng dụng các kỹ thuật xử lý Bigdata vào bài toán nhận dạng

- Về thực tiễn: Đề tài sẽ góp phần nâng cao cũng như nhận thức về tính nguy hiểm của căn bệnh Alzheimer đến bệnh nhân. Giúp người bệnh sớm phát hiện cũng như điều trị căn bệnh được sớm nhất.

#### 6. Cấu trúc đề tài

Cấu trúc của đồ án gồm 3 chương:

Chương 1: Cơ sở lý thuyết.

Chương 2: Mô hình phân tích dữ liệu bệnh Alzheimer.

Chương 3: Xây dựng ứng dụng.

## **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

## 1.1. Tổng quan về Big data

### **1.1.1. Khái niệm**

*Dữ liệu lớn* (Big data) là một thuật ngữ cho việc xử lý một tập hợp dữ liệu rất lớn và phức tạp mà các ứng dụng xử lý dữ liệu truyền thống không xử lý được. Dữ liệu lớn bao gồm các thách thức như phân tích, thu thập, giám sát dữ liệu, tìm kiếm, chia sẻ, lưu trữ, truyền nhận, trực quan, truy vấn và tính riêng tư. Thuật ngữ này thường chỉ đơn giản để cập đến việc sử dụng các phân tích dự báo, phân tích hành vi người dùng, hoặc một số phương pháp phân tích dữ liệu tiên tiến khác trích xuất giá trị từ dữ liệu mà ít khi đề cập đến kích thước của bộ dữ liệu.



*Hình 1.1. Mô hình Big data*

Phân tích tập dữ liệu có thể tìm ra tương quan mới tới "xu hướng kinh doanh hiện tại, phòng bệnh tật, chống tội phạm...". Các nhà khoa học, điều hành doanh nghiệp, y bác sĩ, quảng cáo và các chính phủ cũng thường xuyên gặp những khó khăn với các tập hợp dữ liệu lớn trong các lĩnh vực bao gồm tìm kiếm Internet, thông tin tài chính doanh nghiệp. Các nhà khoa học gặp giới hạn trong công việc cần tính toán rất lớn, bao gồm khí tượng học, bộ gen, mạng thần kinh, các mô phỏng vật lý phức tạp, sinh vật học và nghiên cứu môi trường.

Dữ liệu lớn được định nghĩa là tập hợp các bộ dữ liệu có khối lượng, tốc độ hoặc sự đa dạng lớn đến mức khó lưu trữ, quản lý, xử lý và phân tích dữ liệu bằng cơ sở dữ liệu và công cụ xử lý dữ liệu truyền thống. Trong những năm gần đây, đã có sự tăng trưởng theo cấp số nhân về cả dữ liệu có cấu trúc và phi cấu trúc do công nghệ thông tin, công nghiệp, chăm sóc sức khỏe, Internet vạn vật và các hệ thống khác tạo ra.

Theo ước tính của IBM có 2,5 triệu byte dữ liệu được tạo ra mỗi ngày. Một báo cáo gần đây của DOMO ước tính lượng dữ liệu được tạo ra mỗi phút trên các nền tảng trực tuyến phổ biến. Dưới đây là một số dữ liệu chính từ báo cáo:

- Người dùng Facebook chia sẻ gần 4.16 triệu nội dung.

- Người dùng Twitter gửi gần 300.000 tweet.
- Người dùng Instagram thích gần 1,73 triệu bức ảnh.
- Người dùng YouTube tải lên 300 giờ nội dung video mới.
- Người dùng Apple tải gần 51.000 ứng dụng.
- Người dùng Skype thực hiện gần 110.000 cuộc gọi mới.
- Amazon nhận được 4300 khách truy cập mới.
- Hành khách Uber đi 694 chuyến.
- Người đăng ký Netflix phát gần 77.000 giờ video.

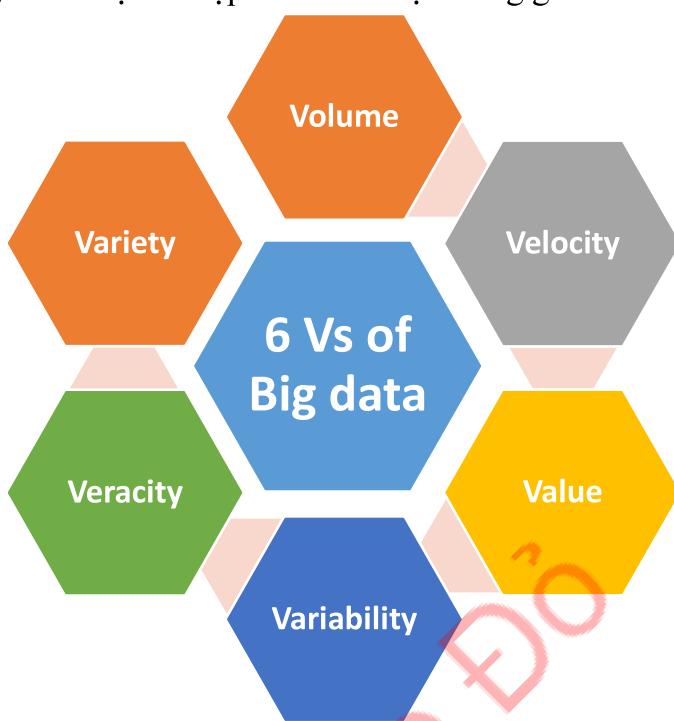
Dữ liệu lớn có tiềm năng cung cấp năng lượng cho các ứng dụng thông minh thế hệ tiếp theo sẽ tận dụng sức mạnh của dữ liệu để làm cho các ứng dụng trở nên thông minh. Các ứng dụng của dữ liệu lớn trải rộng trên nhiều lĩnh vực như web, bán lẻ và tiếp thị, ngân hàng và tài chính, công nghiệp, chăm sóc sức khỏe, môi trường, Internet of Things và các hệ thống thực - ảo.

Phân tích dữ liệu lớn liên quan đến việc thu thập, lưu trữ, xử lý và phân tích dữ liệu quy mô lớn này. Cần có các công cụ và khung chuyên dụng để phân tích dữ liệu lớn khi: Khối lượng dữ liệu liên quan quá lớn nên khó lưu trữ, xử lý và phân tích dữ liệu trên một máy duy nhất, tốc độ truyền dữ liệu rất cao và dữ liệu cần được phân tích theo thời gian thực, có nhiều loại dữ liệu liên quan, có thể có cấu trúc, phi cấu trúc hoặc bán cấu trúc và được thu thập từ nhiều nguồn dữ liệu, cần có nhiều loại phân tích khác nhau được thực hiện để trích xuất giá trị từ dữ liệu, chẳng hạn như phân tích mô tả, chẩn đoán, dự đoán và quy định. Các công cụ và khung dữ liệu lớn có kiến trúc xử lý song song và phân tán, đồng thời có thể tận dụng tài nguyên lưu trữ và tính toán của một cụm máy lớn.

Phân tích dữ liệu lớn bao gồm một số bước bắt đầu từ làm sạch dữ liệu, trộn dữ liệu (hoặc sắp xếp lại dữ liệu), xử lý dữ liệu và trực quan hóa. Vòng đời phân tích dữ liệu lớn bắt đầu từ việc thu thập dữ liệu từ nhiều nguồn dữ liệu. Các công cụ và khung chuyên dụng được yêu cầu để nhập dữ liệu từ các nguồn khác nhau vào phần phụ trợ phân tích dữ liệu tìm kiếm. Dữ liệu được lưu trữ trong các giải pháp lưu trữ chuyên dụng (chẳng hạn như hệ thống tệp phân tán và cơ sở dữ liệu không liên quan) được thiết kế để mở rộng quy mô. Dựa trên các yêu cầu phân tích (theo lô hoặc thời gian thực) và loại phân tích được thực hiện (mô tả, chẩn đoán, hoặc dự đoán), các khung chuyên biệt được sử dụng. Phân tích dữ liệu lớn được kích hoạt bởi một số công nghệ như điện toán đám mây, khung xử lý song song và phân tán, cơ sở dữ liệu phi quan hệ, điện toán trong bộ nhớ chia sẻ.

### 1.1.2. Các đặc trưng của Big data

- Hiện nay, Big data được đề cập đến với 6 đặc trưng gồm:



Hình 1.2. Đặc trưng 6V của Big data

**Volume:** Khối lượng dữ liệu được tạo ra, lưu trữ và xử lý. Dữ liệu bao nhiêu thì là được gọi là lớn? Ít nhất nó nên được tính bằng hàng trăm Terabyte, còn thông thường thì nó là Petabyte hoặc Exabyte, thậm chí Zettabyte. Một số ví dụ: Facebook tạo ra khoảng 500TB dữ liệu mỗi ngày, con số này ở Twitter là khoảng 8TB.

**Velocity:** Tốc độ dữ liệu được tạo ra. Câu hỏi cũ, bao nhiêu thì là lớn? Câu trả lời thì rất đa dạng, vì nó phụ thuộc vào loại dữ liệu mà ta đang xử lý, nhưng có thể lấy một vài ví dụ trực quan của các tên tuổi lớn như 90 triệu bức ảnh được upload lên Facebook mỗi ngày, con số cho Twitter là 500 triệu tweets được post, 0.4 triệu giờ video được upload lên Youtube hay 3.5 tỷ lượt tìm kiếm được thực hiện mỗi ngày trên Google.

**Variety:** Tính đa dạng của dữ liệu. Điều này dễ hiểu, Big data là tổng hợp các dạng dữ liệu, từ dữ liệu có cấu trúc (structure) như các bảng nơi có hàng và cột trong cơ sở dữ liệu quan hệ RDBMS (Relational Database Management System) hay bảng tính excel; đến dữ liệu phi cấu trúc (unstructured) như văn bản (text), ảnh (pictures), video, audio, ...; và thậm chí cả dữ liệu bán cấu trúc (semi-structure) như file json hay file xml.

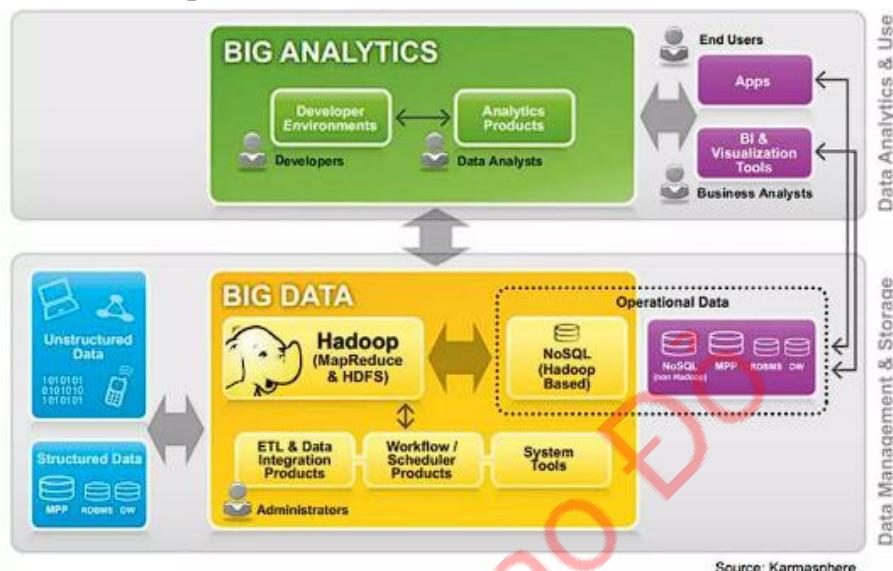
**Variability:** Tính biến thiên của dữ liệu phản ánh sự không nhất quán trong dữ liệu, đặc biệt là trong xử lý ngôn ngữ tự nhiên, cùng một từ vựng nhưng trong các ngữ cảnh khác nhau nó sẽ mang các ý nghĩa khác nhau. Tính biến thiên của dữ liệu cũng còn được thể hiện qua sự biến thiên về kích thước và tốc độ dữ liệu được sinh ra, do dữ liệu được thu thập từ nhiều nguồn khác nhau.

**Veracity:** Mức độ tin cậy của dữ liệu. Đặc tính này đi ngược chiều với các đặc tính khác của Big data, khi khối lượng dữ liệu ngày càng tăng, tính đa dạng của dữ liệu

ngày càng phong phú và tính biến thiên của dữ liệu ngày càng lớn thì mức độ tin cậy của dữ liệu ngày càng giảm xuống.

**Value:** Giá trị của dữ liệu. Sẽ không thể nhắc đến Big data nếu không thể get value từ dữ liệu. Hàng loạt các công ty đã khai thác “mỏ vàng mới” dữ liệu và phát triển mạnh mẽ: Google, Facebook, Amazon, ...

### 1.1.3. Các thành phần của Big data



Hình 1.3. Kiến trúc Big data

### Quản lý dữ liệu

Quản lý cơ sở hạ tầng lưu trữ dữ liệu, và nguồn để thao tác nó. Có 3 hệ cơ sở dữ liệu Big data: Hệ dữ liệu có cấu trúc, hệ dữ liệu không có cấu trúc, hệ dữ liệu bán cấu trúc.

### Phân tích dữ liệu

Phân tích công nghệ và các công cụ để phân tích các dữ liệu và thu thập hiểu biết sâu sắc từ nó.

- Là nơi mà các công ty bắt đầu trích xuất giá trị dữ liệu lớn.
- Liên quan tới việc phát triển các ứng dụng và sử dụng các ứng dụng để đạt được cái nhìn sâu sắc vào dữ liệu lớn.
- Xây dựng các công cụ phân tích dữ liệu.

Phân tích dữ liệu mang lại giá trị thực sự từ các tổ chức dữ liệu lớn. Nếu không có sự phân tích thì đây chỉ là những dữ liệu được sử dụng vô cùng hạn chế trong kinh doanh. Với việc phân tích những dữ liệu lớn, các công ty có thể thu về cho mình những lợi ích như dịch vụ khách hàng được cải thiện, từ đó mang lại hiệu quả cao hơn, tăng doanh thu và tăng khả năng cạnh tranh.

### Sử dụng dữ liệu

Sử dụng dữ liệu là hoạt động trên dữ liệu được phân tích.

Đưa dữ liệu lớn đã phân tích để phục vụ trong kinh doanh thông minh và các ứng dụng của người dùng cuối.

### Cơ sở hạ tầng cần thiết để hỗ trợ Big data

Đối với khái niệm Big data để có thể làm việc, các tổ chức cần phải xây dựng được cơ sở hạ tầng để thu thập và chứa dữ liệu, cung cấp quyền truy cập và đảm bảo thông tin trong khi chuyển tiếp và lưu trữ. Cấp độ cao hơn, bao gồm hệ thống lưu trữ và máy chủ được thiết kế cho Big data, tích hợp dữ liệu và phần mềm quản lý, phần mềm phân tích dữ liệu, thông tin kinh doanh và các ứng dụng Big data.



Hình 1.4. Mô hình cơ sở hạ tầng cần thiết để hỗ trợ Big data

Thu thập dữ liệu đòi hỏi bắt buộc là phải có nguồn. Một số ứng dụng như các ứng dụng web, ứng dụng di động, các kênh truyền thông xã hội và lưu trữ email đã được cài đặt sẵn.

#### 1.1.4. Tầm quan trọng của Big data

Tầm quan trọng của dữ liệu lớn không nằm ở lượng dữ liệu mà chúng ta có, nó nằm ở việc chúng ta làm gì với những dữ liệu đó. Ta có thể sử dụng nguồn dữ liệu lớn phân tích để tìm ra câu trả lời cho các câu hỏi: giảm chi phí, giảm thời gian, phát triển sản phẩm mới và dịch vụ tối ưu, ra quyết định thông minh. Khi việc phân tích nguồn dữ liệu lớn được hỗ trợ tối đa ta có thể hoàn thành tốt một số tác vụ như: xác định nguyên nhân gốc rễ của những thất bại, tạo các chương trình khuyến mại hợp lý dựa trên thói quen của khách hàng đối với công việc kinh doanh, tính toán được những rủi ro gặp phải, phát hiện hành vi gian lận trước khi nó có ảnh hưởng đến chúng ta.

Big data thu thập các thông tin quy mô lớn từ website, các doanh nghiệp có thể dùng công cụ phân tích dữ liệu này để phục vụ cho công việc phân tích thị trường giúp đưa ra các chiến lược nâng cao chất lượng sản phẩm, dịch vụ hiện tại hay tìm hiểu về hành vi khách hàng để ra mắt các sản phẩm mới

Các tổ chức, doanh nghiệp có thể tận dụng Big data để xây dựng một nội dung trang web thu hút người truy cập hơn, có được cái nhìn toàn cảnh và sâu sắc về hành vi mua hàng. Dữ liệu càng nhiều thì những phân tích càng chính xác. Các doanh nghiệp cũng nên cung cấp nội dung trên nhiều kênh xã hội để thu thập dữ liệu từ nhiều nguồn hơn.

Xa hơn một chút Big data có thể giúp các cơ quan Chính phủ dự đoán được tỉ lệ thất nghiệp, xu hướng nghề nghiệp của tương lai để tập trung đầu tư cho các hạng mục

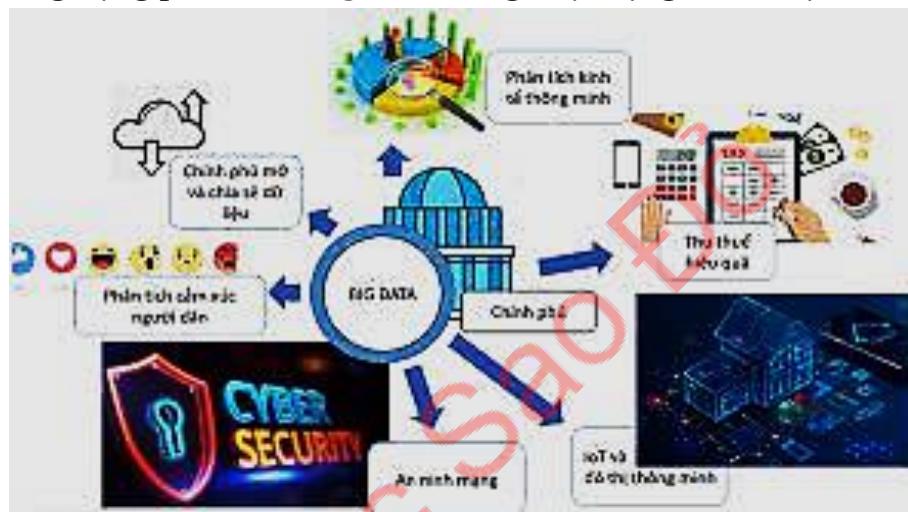
đó, cắt giảm chi phí, kích thích tăng trưởng kinh tế, thậm chí là ra phương án phòng ngừa trước một dịch bệnh nào đó.

Tóm lại, Big data chính là thách thức đặt ra cho các doanh nghiệp trong thời đại công nghệ số. Một khi tận dụng được tối đa nguồn dữ liệu lớn thì có hội thành công sẽ lớn hơn nhiều lần. Tuy vẫn còn những chỉ trích về Big data nhưng đây là một lĩnh vực còn rất mới nên chúng ta hãy chờ đón sự tiến hóa của Big data trong tương lai.

### 1.1.5. Ứng dụng Big data

Do tầm quan trọng của Big data, việc biết, hiểu và ứng dụng tốt Dữ liệu lớn sẽ tạo ra lợi thế rất lớn cho bất kỳ doanh nghiệp, tổ chức nào. Dưới đây là 1 số ví dụ:

#### ❖ Ứng dụng phân tích Big data trong hoạt động chính trị



Hình 1.5. Hệ thống phân tích bigdata trong chính trị

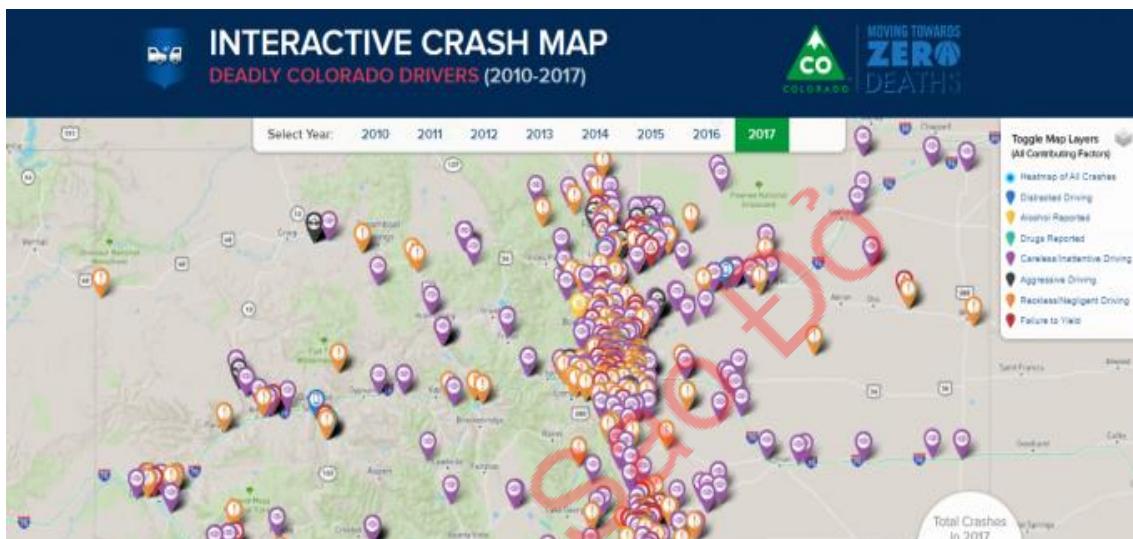
Cựu Tổng thống Obama đã sử dụng dữ liệu dữ liệu lớn để phục vụ cho cuộc tranh cử Tổng thống của mình. Ông xây dựng một đội ngũ nhân viên chuyên đi thu thập thông tin và phân tích dữ liệu thu được trong dự án triển khai về dữ liệu lớn. Đội ngũ nhân viên này thu thập tất cả thông tin về người dân ở các khu vực, sau đó phân tích và chỉ ra một số thông tin quan trọng về người dân Mỹ như: Thích đọc sách gì, thích mua loại thuốc gì, thích sử dụng phương tiện gì,... Thậm chí còn biết được cả thông tin về người đó đã bỏ phiếu tín nhiệm ai ở lần bầu cử trước. Trên cơ sở những thông tin này, Obama đã đưa ra kế hoạch vận động phù hợp, giúp ông tái đắc cử Tổng thống lần 2 của nước Mỹ.

### Ứng dụng phân tích Big data để quản lý xã hội

Hiện nay Chính phủ đang ứng dụng số hóa vào quản lý xã hội bằng cách tạo ra những kho Big data Dữ liệu lớn để cho người dân và các cơ quan hành chính thuận tiện trong việc nhập, tìm kiếm, kiểm tra và lưu trữ dữ liệu bằng cách triển khai Căn cước công dân gắn chip, sổ hộ khẩu điện tử, hộ chiếu điện tử, bảo hiểm xã hội điện tử, ... Việc này giúp giảm các thủ tục hành chính giấy tờ rườm rà cho người dân, đề cao sự minh bạch, trong suốt và tin cậy.

## Ứng dụng Big data trong an toàn giao thông

Sử dụng số liệu CDR trong quá khứ để ước lượng các dòng giao thông trong thành phố vào các giờ cao điểm, từ đó có những bản đồ tai nạn giao thông, kế hoạch phân luồng giao thông ... chi tiết, hợp lý giúp giảm thiểu tai nạn và kẹt xe. Ngoài ra còn đưa ra thông tin cho người tham gia giao thông được biết nếu muốn đi từ nơi này đến nơi khác thì nên đi vào giờ nào để tránh kẹt xe, hoặc đi đường nào là ngắn nhất... Ngoài ra dữ liệu lớn còn giúp phân tích định vị người dùng thiết bị di động, ghi nhận chi tiết cuộc gọi trong thời gian thực; và giảm thiểu tình trạng ùn tắc giao thông.



Hình 1.6. Hệ thống phân tích tử vong của Cơ quan An toàn Giao thông đường cao tốc quốc gia

### Giảm gian lận và tăng cường an ninh

Luôn có những rủi ro liên quan đến việc sử dụng dữ liệu lớn. Tuy nhiên, những lợi ích, tiềm năng mà công nghệ này mang lại vượt xa hơn nhiều so với rủi ro.

Các tổ chức chăm sóc sức khỏe nắm giữ lượng lớn dữ liệu nhạy cảm về bệnh nhân, điều này khiến họ trở thành mục tiêu hàng đầu của những cuộc tấn công mạng. Nhiều nghiên cứu đã chỉ ra rằng, 93% tổ chức chăm sóc sức khỏe đã gặp phải sự cố vi phạm dữ liệu. Lý do rất đơn giản: Dữ liệu cá nhân cực kỳ có giá trị và sinh lời trên thị trường chợ đen.

Do đó, nhiều tổ chức sử dụng công cụ phân tích để giúp ngăn chặn mối đe dọa bảo mật. Họ xác định những thay đổi trong lưu lượng truy cập mạng hoặc bất kỳ hành vi nào khác phản ánh một cuộc tấn công mạng.

Ngoài ra, những tiến bộ về bảo mật, chẳng hạn như công nghệ mã hóa, tường lửa, phần mềm chống vi-rút,... đáp ứng nhu cầu bảo mật cao hơn. Công cụ phân tích dựa trên Big data cũng hỗ trợ phòng chống gian lận. Những công cụ này hợp lý hóa quy trình yêu cầu bảo hiểm, bệnh nhân được chi trả bảo hiểm nhanh chóng, minh bạch.

### Ứng dụng Big data trong tài chính

Từ những dữ liệu chính xác, định kỳ, kịp thời thu thập được thông qua các giao dịch của khách hàng giúp tiến hành phân tích, xếp hạng và quản lý các rủi ro trong đầu tư tài chính, tín dụng.



Hình 1.7. Big data trong phân tích hệ thống ngân hàng

### Ứng dụng Big data trong thương mại



Hình 1.8. Big data phân tích trong mô hình thương mại điện tử

Trong thương mại dữ liệu lớn giúp cho chúng ta thực hiện được một số công việc sau: Phân khúc thị trường và khách hàng; phân tích hành vi khách hàng tại cửa hàng; tiếp thị trên nền tảng định vị; phân tích tiếp thị chéo kênh, tiếp thị đa kênh; quản lý các chiến dịch tiếp thị và khách hàng thân thiết; So sánh giá; Phân tích và quản lý chuỗi cung ứng; Phân tích hành vi, thói quen người tiêu dùng.

### Ứng dụng Big data trong dự báo thời tiết

Trong bối cảnh khí hậu đang thay đổi chóng mặt vì con người, một hệ thống có khả năng dự đoán chính xác và đưa ra cảnh báo sớm có khả năng cứu sống hàng triệu sinh linh. Bằng cách sử dụng Big data, các nhà khoa học dự báo với độ chính xác tương

đối trong vòng 1 tuần cho đến 1 tháng. Và gần đây, các nhà khoa học của Cục Khí tượng thủy văn Úc đang nghiên cứu đưa ra dự đoán chính xác thời tiết trước vài năm thậm chí vài chục năm.



Hình 1.9. Ảnh dự báo thời tiết

Từ những ứng dụng trên đây, có thể nói, Big data tạo nên ảnh hưởng rộng lớn đến đời sống của tất cả con người trong thời đại số. Nhiều ngành kinh tế như ngân hàng, giao thông vận tải, thương mại điện tử hay chăm sóc sức khỏe đã có những bước nhảy vọt, cải tiến công nghệ, sản phẩm của mình.

### Ứng dụng Big data trong Y tế

Trong y học các bác sĩ dựa vào số liệu trong các bệnh án để đưa ra dự đoán về nguy cơ mắc bệnh. Đồng thời cũng đưa ra được xu hướng lây lan của bệnh.



Hình 1.10. Hình ảnh phân tích liệt kê thu thập dữ liệu trong y tế

Dữ liệu lớn trong chăm sóc sức khỏe là thuật ngữ dùng để mô tả khái niệm lượng thông tin khổng lồ được tạo ra bằng cách áp dụng công nghệ kỹ thuật số nhằm thu thập hồ sơ bệnh nhân và giúp quản lý hoạt động của cơ sở y tế.

Ứng dụng phân tích dữ liệu lớn trong chăm sóc sức khỏe có nhiều kết quả tích cực, nâng cao công tác khám chữa bệnh. Về bản chất, khái niệm dữ liệu lớn đề cập đến

lượng thông tin khổng lồ được tạo ra bằng cách số hóa mọi thứ, sau đó sử dụng công nghệ để hợp nhất và phân tích.

Áp dụng vào lĩnh vực y tế, các công nghệ sử dụng dữ liệu sức khỏe cụ thể của một cộng đồng (hoặc của một cá nhân cụ thể) để phân tích, đưa ra hành động, quyết định hỗ trợ ngăn ngừa dịch bệnh, chữa bệnh, cắt giảm chi phí,...

Số lượng nguồn mà chuyên gia y tế có thể thu được thông tin chi tiết từ bệnh nhân cũng không ngừng tăng lên. Những dữ liệu này thường có định dạng và kích cỡ khác nhau, do đó có thể gây khó khăn trong quá trình lưu trữ, phân loại, phân tích thủ công.

Điều quan trọng không chỉ là khối lượng dữ liệu “lớn” như thế nào mà là nó được quản lý thông minh như thế nào. Với sự trợ giúp của công nghệ phù hợp, dữ liệu ngành sức khỏe có thể được trích xuất từ một số nguồn sau một cách nhanh chóng:

- Công thông tin bệnh nhân
- Nghiên cứu
- Hồ sơ sức khỏe điện tử (EHR)
- Thiết bị đeo thông minh
- Công cụ tìm kiếm
- Cơ sở dữ liệu dùng chung
- Cơ quan chính phủ
- Hồ sơ khách hàng
- Lịch trình nhân sự
- Phòng chờ bệnh nhân
- ...

Trước đây, việc thu thập lượng dữ liệu khổng lồ cho mục đích y tế rất tốn kém và mất thời gian. Với các công nghệ luôn cải tiến ngày nay, việc thu thập dữ liệu trở nên dễ dàng hơn, đồng thời tạo ra báo cáo chăm sóc sức khỏe toàn diện và chuyển đổi chúng thành những hiểu biết quan trọng để có thể cung cấp dịch vụ chăm sóc tốt hơn.



Hình 1.11. Tìm kiếm thông tin trong mảng dữ liệu Big data

Dữ liệu lớn cung cấp thông tin phục vụ tối ưu cho việc chẩn đoán, điều trị

Đây cũng chính là mục đích của phân tích dữ liệu chăm sóc sức khỏe: Tận dụng các phát hiện dựa trên dữ liệu để dự đoán và giải quyết vấn đề trước khi quá muộn, đồng thời đánh giá phương pháp điều trị nhanh hơn, theo dõi hàng tồn kho tốt hơn, khuyến khích bệnh nhân chăm sóc sức khỏe chủ động và cung cấp cho họ với những công cụ để làm việc đó.

### **Ứng dụng của Big data trong chăm sóc sức khỏe**

Dưới đây là một số minh chứng về lợi ích nhận được từ việc ứng dụng dữ liệu lớn trong lĩnh vực y tế.

#### **Sắp xếp nhân sự**

Nếu sử dụng quá nhiều nhân sự, bệnh viện sẽ có nguy cơ phải tăng thêm chi phí lao động không cần thiết. Tuy nhiên, nếu lượng nhân sự không đạt mức cần thiết tối thiểu, điều này có thể dẫn đến việc cung cấp dịch vụ khách hàng kém, thậm chí ảnh hưởng lớn đến bệnh nhân nếu không được chăm sóc kịp thời.

Dữ liệu lớn đang giúp giải quyết vấn đề này. Để sắp xếp nhân sự một cách phù hợp nhất, bệnh viện sử dụng dữ liệu từ nhiều nguồn khác nhau, đưa ra dự đoán hàng ngày và hàng giờ về số lượng bệnh nhân dự kiến có mặt tại mỗi cơ sở.

Một trong những bộ dữ liệu quan trọng là hồ sơ nhập viện trong 10 năm trở lại mà các nhà khoa học dữ liệu đã xử lý bằng cách sử dụng kỹ thuật “phân tích chuỗi thời gian”. Những phân tích này cho phép nhà nghiên cứu thấy được mô hình có liên quan trong tỷ lệ nhập viện. Sau đó, họ có thể sử dụng công nghệ máy học (ML) để tìm ra thuật toán chính xác nhất nhằm dự đoán xu hướng thăm khám, điều trị trong tương lai.

Nhóm khoa học dữ liệu cũng phát triển giao diện người dùng dựa trên web để dự báo lượng bệnh nhân và lập kế hoạch phân bổ nguồn lực nhờ trực quan hóa dữ liệu trực tuyến, từ đó đạt được mục tiêu cải thiện dịch vụ chăm sóc tổng thể cho bệnh nhân.

#### **Hồ sơ sức khỏe điện tử (EHRs)**

Đây là ứng dụng rộng rãi nhất của dữ liệu lớn trong y học. Mỗi người đều có hồ sơ sức khỏe kỹ thuật số của riêng mình, bao gồm thông tin nhân khẩu học, tiền sử bệnh, dị ứng, kết quả xét nghiệm,...

Hồ sơ được chia sẻ qua hệ thống thông tin an toàn, có sẵn cho nhà cung cấp dịch vụ từ cả khu vực công và tư nhân. Mỗi bản ghi bao gồm một tệp có thể sửa đổi, có nghĩa là bác sĩ, chuyên gia có quyền cập nhật theo thời gian mà không cần giấy tờ và không có nguy cơ sao chép dữ liệu.

EHR mang lại lợi ích cho cả bệnh nhân và nhà cung cấp dịch vụ y tế. Đầu tiên, EHR giúp bệnh nhân dễ dàng nhận được sự chăm sóc mà họ cần bằng cách cung cấp cho bác sĩ một bức tranh toàn cảnh về sức khỏe cá nhân.

Công cụ này có thể kích hoạt cảnh báo và nhắc nhở khi bệnh nhân cần làm xét nghiệm định kỳ hoặc chuyên gia sẽ theo dõi đơn thuốc để chắc chắn rằng bệnh nhân có tuân theo chỉ dẫn.

Đối với nhà cung cấp dịch vụ y tế, EHR cải thiện chất lượng chăm sóc thông qua việc theo dõi tiến triển của bệnh nhân dễ dàng hơn và xác định sớm vấn đề tiềm ẩn. Ngoài ra, EHR giúp giảm chi phí nhờ ngăn chặn xét nghiệm trùng lặp và loại bỏ nhu cầu về hồ sơ giấy.

Nhìn chung, hồ sơ sức khỏe điện tử là một phần quan trọng trong quá trình chuyển đổi sang y học dựa trên bằng chứng. Thông qua cung cấp thông tin chính xác và cập nhật về sức khỏe của bệnh nhân, EHR cho phép bác sĩ đưa ra chẩn đoán và phương pháp điều trị chính xác hơn. Đổi lại, điều này giúp cải thiện kết quả của bệnh nhân và giảm chi phí trên toàn hệ thống chăm sóc sức khỏe.



Hình 1.12. Thảo luận bệnh án của bệnh nhân bằng dữ liệu đã lưu trữ

Chuyên gia y tế dễ dàng theo dõi, nắm bắt thông tin người bệnh qua hồ sơ sức khỏe điện tử.

### Cảnh báo thời gian thực

Mặc dù cảnh báo theo thời gian thực không phải là một tiện ích mới nhưng tiềm năng của nó trong ngành chăm sóc sức khỏe hiện chỉ vừa bắt đầu được hiện thực hóa.

Nhằm hỗ trợ công dân chăm sóc sức khỏe chủ động ngay tại nhà, những thiết bị đeo thông minh đã được tận dụng. Thiết bị đeo trở thành xu hướng, đặc biệt là trong giai đoạn dịch bệnh lây lan nhanh chóng, không thuận tiện cho việc tiếp xúc và tập trung.

Thiết bị đeo liên tục thu thập dữ liệu sức khỏe của bệnh nhân và gửi dữ liệu này lên đám mây. Từ đó, trí tuệ nhân tạo (AI) tiến hành phân tích, cảnh báo xu hướng tiêu cực và sự bất thường, nguy cơ tiềm ẩn.

Kiểu tiếp cận dựa trên dữ liệu này có khả năng thay đổi hoạt động, thói quen chăm sóc sức khỏe của người dân, chủ động hơn và mang tính phòng ngừa, nâng cao khả năng điều trị thành công, dứt điểm nhiều bệnh lý.

Thông qua thiết bị đeo thông minh, phía bác sĩ có thể truy cập cơ sở dữ liệu theo từng cấp độ để xem xét tình trạng sức khỏe của công chúng nói chung như thế nào. Ngoài ra, việc thu thập những thông tin liên quan như thu nhập, khu vực sống,... giúp cơ sở y tế đưa ra những chiến lược chăm sóc được cá nhân hóa.

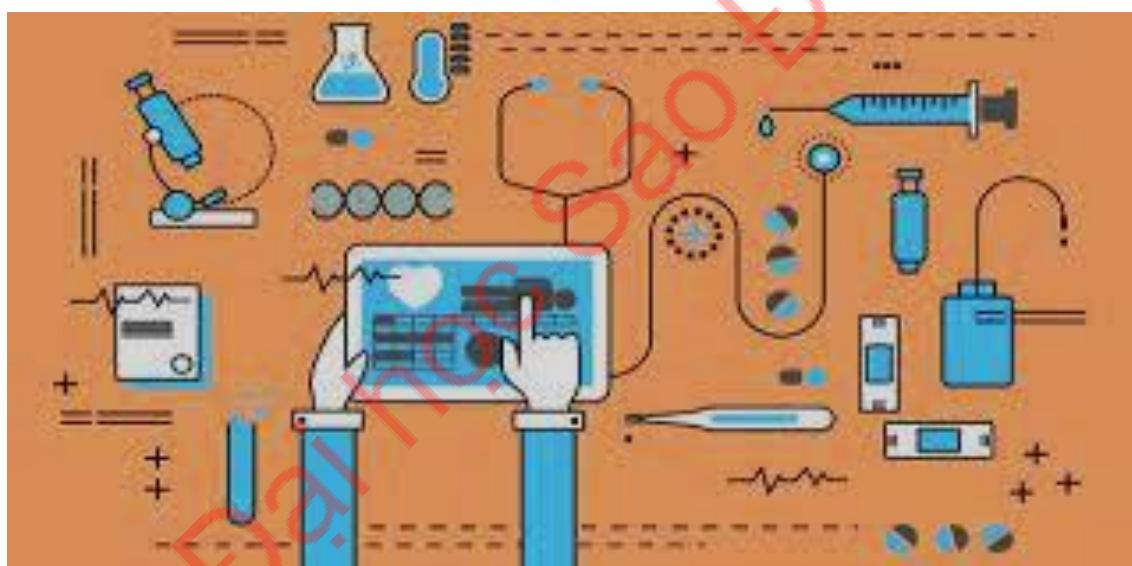
### **Ngăn ngừa lạm dụng nhóm thuốc giảm đau Opioid**

Sử dụng quá liều opioid hiện đã trở thành nguyên nhân hàng đầu gây tử vong ngoài ý muốn ở Hoa Kỳ, vượt qua cả tai nạn giao thông. Vấn đề đã trở nên nghiêm trọng đến mức Canada phải tuyên bố đây là “cuộc khủng hoảng sức khỏe quốc gia”.

Các nhà cung cấp dịch vụ chăm sóc có thể sử dụng dữ liệu để giải quyết vấn đề này bằng cách nghiên cứu mô hình lạm dụng, từ đó phát triển biện pháp can thiệp chính xác, hiệu quả.

Ví dụ: Dữ liệu lớn được ứng dụng nhằm mục đích theo dõi việc phân phối opioid và xác định “điểm nóng”, nơi có dấu hiệu lạm dụng thuốc.

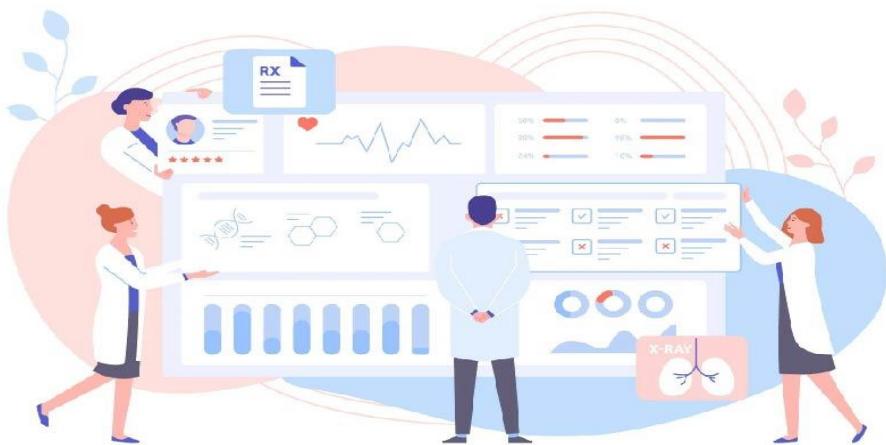
### **Phân tích dự đoán trong chăm sóc sức khỏe**



Hình 1.13. Đưa ra những dự đoán cho việc chăm sóc sức khỏe

Thông tin kinh doanh trực tuyến về chăm sóc sức khỏe được thu thập nhằm mục đích giúp bác sĩ đưa ra quyết định dựa trên dữ liệu và cải thiện việc điều trị cho bệnh nhân. Điều này đặc biệt hữu ích trong trường hợp bệnh nhân có tiền sử bệnh phức tạp.

Các giải pháp và công cụ BI cũng có thể dự đoán, chẳng hạn như ai có nguy cơ mắc bệnh lý nào và từ đó tư vấn sử dụng biện pháp sàng lọc bổ sung hoặc quản lý sức khỏe ngay lập tức. Ngoài ra, kế hoạch điều trị cũng được tạo ra và điều chỉnh riêng cho từng bệnh nhân để mang lại kết quả tốt nhất.



Hình 1.14. Phân tích, dự đoán góp phần nâng cao tỷ lệ điều trị thành công bệnh lý

### Phân tích dữ liệu lớn và hình ảnh y tế

Hình ảnh y tế đóng vai trò quan trọng trong quá trình chẩn đoán và điều trị. Mỗi năm, các bác sĩ ở Mỹ yêu cầu thực hiện khoảng 600 triệu thủ thuật chẩn đoán hình ảnh. Tuy nhiên, việc phân tích và lưu trữ những hình ảnh này rất tốn kém về thời gian và tiền bạc. Bác sĩ X-quang cần kiểm tra từng hình ảnh riêng lẻ và bệnh viện cần có trách nhiệm lưu trữ chúng trong vài năm.

Phân tích dữ liệu lớn ứng dụng cho chăm sóc sức khỏe có thể giúp thuật toán phân tích hàng trăm nghìn hình ảnh. Chúng tự động xác định mẫu cụ thể trong pixel và chuyển đổi chúng thành chỉ số để giúp bác sĩ đưa ra chẩn đoán.

Thậm chí, nhiều dự đoán cho rằng trong tương lai, bác sĩ X-quang sẽ không cần phải nhìn vào hình ảnh nữa mà thay vào đó là sử dụng thuật toán để phân tích kết quả. Những thuật toán này chắc chắn sẽ nghiên cứu và ghi nhớ nhiều hình ảnh hơn con người. Thông qua đó, phân tích Big data có tiềm năng cách mạng hóa hình ảnh y tế và tạo ra hiệu quả đáng kể trong hệ thống chăm sóc sức khỏe.

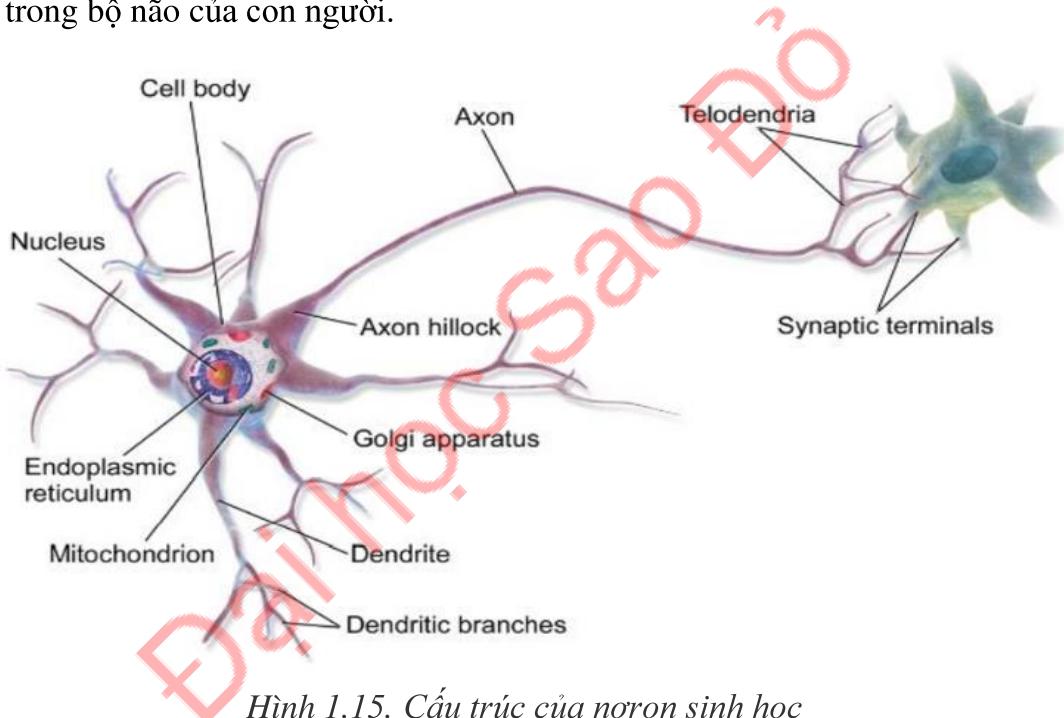
Big data có lợi ích đối với nhiều ngành công nghiệp trọng điểm, bao gồm cả y tế. Trên đây là một số ví dụ về ứng dụng của dữ liệu lớn trong y tế - chăm sóc sức khỏe và sẽ còn được mở rộng, phát triển mạnh mẽ hơn nữa trong tương lai.

## 1.2. Sơ lược về mạng nơron

### 1.2.1. Giới thiệu về mạng nơron nhân tạo

Theo Wikipedia, mạng nơron nhân tạo – Artificial Neural network (viết tắt là ANN) là các mô hình toán học hay mô hình tính toán được xây dựng dựa trên các mạng nơron sinh học. Những ý tưởng xây dựng các mô hình mạng nơron nhân tạo này bắt nguồn từ việc khám phá ra các cơ chế hoạt động đơn giản của mạng nơron sinh học (biological neural network).

Trong hệ thống thần kinh sinh học, nơron là tế bào sống và còn là đơn vị lưu trữ cơ bản trong bộ não của con người. Có khoảng 200 tỉ nơron trong não con người. Mỗi nơron được liên kết với khoảng từ 1.000 đến 10.000 nơron khác thông qua các khớp thần kinh (synapse). Các tín hiệu xung điện được truyền từ tế bào nơron này sang tế bào nơron khác thông qua các khớp thần kinh. Có tất cả khoảng 125 nghìn tỉ khớp thần kinh trong bộ não của con người.



Hình 1.15. Cấu trúc của nơron sinh học

Bộ não của con người lưu trữ thông tin trong những tế bào thần kinh và những khớp thần kinh giữa các tế bào này với nhau. Mỗi khái niệm trong não có thể được tượng trưng bằng một mạng của các kết nối khác nhau giữa những tế bào thần kinh. Sự hình thành các kiến thức mới (learning) xảy ra khi các khớp kết nối giữa các nơron trở nên mạnh hơn và liên kết giữa một cụm tế bào mới được hình thành. Cụ thể hơn là những thông tin này được hình thành từ khả năng thay đổi cường độ liên kết giữa các khớp thần kinh hay còn được gọi là sự mềm dẻo của khớp thần kinh (synaptic plasticity). Giả thuyết này được đưa ra bởi Hebb vào năm 1949, người cho rằng sự mềm dẻo của khớp thần kinh được tạo ra từ sự kích thích lặp đi lặp lại và kéo dài giữa khớp tế bào trước (presynaptic terminal) và khớp tế bào sau (postsynaptic terminal). Điều này có nghĩa là khi hai nơron được kích hoạt cùng lúc, các khớp thần kinh giữa hai tế bào sẽ trở nên mạnh hơn và hai tế bào này sẽ có thể kích hoạt lẫn nhau hiệu quả hơn. Các quá trình

kích hoạt tạo ra sự liên kết này lâu dần sẽ tạo mạng lưới của các kết nối và nó sẽ đại diện cho một khái niệm nào đó.

Các nhà nghiên cứu đã tìm cách chuyển đổi những hiểu biết về cách thức hoạt động của các tế bào thần kinh sinh học thành các mô hình mạng nơron nhân tạo có thể hoạt động được trên máy tính. Một ANN được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất chính là quá trình hiệu chỉnh trong số liên kết giữa các nơron.

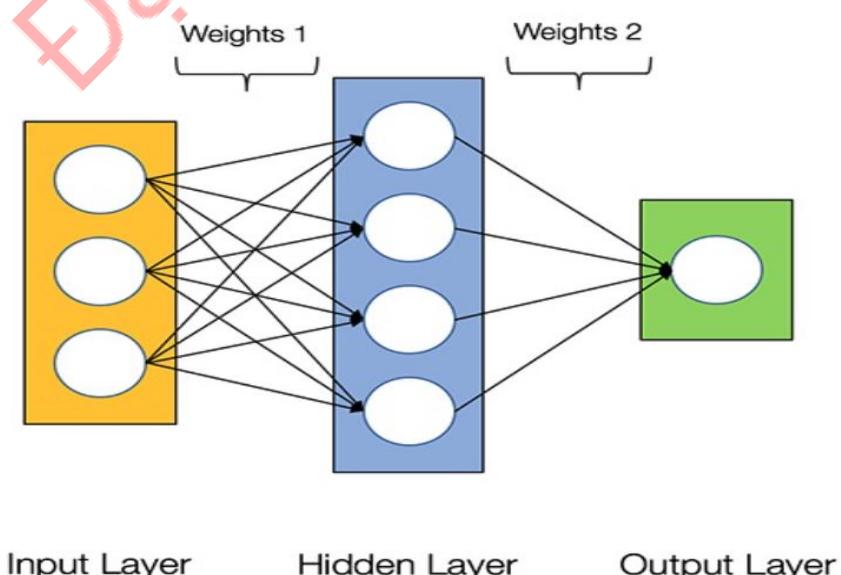
Các nơron là bộ xử lý thông tin đơn giản, bao gồm một thân tế bào (cell body) và các dây kết nối các nơron với nhau. Mỗi một nơron trong mạng là một hàm toán học lấy dữ liệu thông qua đầu vào, xử lý thông tin bằng cách truyền theo các kết nối và tính giá trị mới tại các nút (cách tiếp cận connectionism đối với tính toán) và sau đó đưa kết quả thông qua đầu ra. Trong nhiều trường hợp, ANN 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.

### 1.2.2. Cấu trúc mạng nơron nhân tạo

Mạng nơron có thể coi là một phương thức để miêu tả cách hoạt động trong bộ não của con người. Nói một cách đơn giản hơn, mạng nơron là một hàm nhận một đầu vào  $x$ , xử lý nó và đưa ra một đầu ra  $y$  tương ứng.

Một mạng nơron gồm những thành phần sau:

- Một lớp biểu diễn đầu vào –  $x$  (lớp đầu vào  $x$  thường không được tính vào số lượng lớp của mạng)
- Một số lượng các lớp ẩn - **hidden layers**
- Một lớp biểu diễn đầu ra -  $y$
- Một bộ các chỉ số **weights(W)** và **bias(b)** giữa các lớp
- Một hàm kích hoạt - **activation function** giữa các lớp



Hình 1.16. Kiến trúc mạng nơron

Sơ đồ đơn giản hóa ở trên cung cấp một ý tưởng về cách cấu trúc một mạng lưới thần kinh đơn giản. Trong sơ đồ, mỗi vòng tròn đại diện cho một nơron trong mạng, với các nơron được tổ chức thành các lớp thẳng đứng. Như ta có thể thấy, mỗi nơron được liên kết với mọi nơron ở lớp sau, thể hiện thực tế là mỗi nơron tạo ra một giá trị vào mỗi nơron ở lớp tiếp theo. Một nơron có thể được cho là đã được kích hoạt khi tổng các giá trị được đưa vào nơron này vượt qua ngưỡng đã đặt, có nghĩa là chúng sẽ khuếch đại giá trị khi nó đi qua giữa các lớp. Đổi lại, sự khuếch đại giá trị này có thể giúp kích hoạt tế bào thần kinh mà giá trị đang được đưa vào.

Trong thực tế sử dụng, nhiều mạng nơron 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.

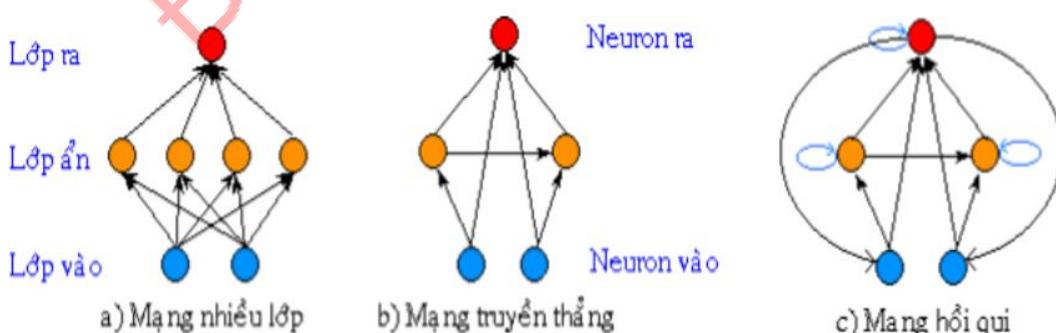
### 1.2.3. Phân loại các mạng nơron

Dựa theo kiểu liên kết giữa các nơron trong mạng mà ta chia mạng nơron thành hai loại chính:

- Mạng nơron truyền thẳng (Feedforward Neural Network): Các liên kết nơron đi theo một hướng nhất định, không tạo thành đồ thị không có chu trình với các đỉnh là các nơron, các cung là các liên kết giữa chúng.

- Mạng nơron quy hồi (Recurrent Neural Network): Các mạng quy hồi cho phép các liên kết nơron tạo thành chu trình. Vì các thông tin ra của các nơron được truyền lại cho các nơron đã góp phần kích hoạt chúng, nên mạng hồi quy còn có khả năng lưu giữ trạng thái trong của nó dưới dạng các ngưỡng kích hoạt ngoài các trọng số liên kết nơron.

Ngoài ra, dựa theo số lớp của mạng, các nơron có thể tổ chức lại thành các lớp sao cho mỗi nơron của lớp này chỉ được nối với các nơron ở lớp tiếp theo, không cho phép các liên kết giữa các nơron trong cùng một lớp, hoặc từ nơron lớp dưới lên nơron lớp trên. Hay còn gọi là mạng MLP (Multilayer Perceptrons Networks)



Hình 1.17. Sơ đồ cấu trúc một số mạng nơron

### 1.2.4. Huấn luyện mạng

Chức năng của một mạng nơron được quyết định bởi các nhân tố như: hình trạng mạng (số lớp, số đơn vị trên mỗi tầng, và cách mà các lớp được liên kết với nhau) và các trọng số của các liên kết bên trong mạng.

Hình trạng của mạng thường là cố định, và các trọng số được quyết định bởi một thuật toán huấn luyện (training algorithm). Tiến trình điều chỉnh các trọng số để mạng “nhận biết” được quan hệ giữa đầu vào và đích mong muốn được gọi là học (learning) hay huấn luyện (training).

Rất nhiều thuật toán học đã được phát minh để tìm ra tập trọng số tối ưu làm giải pháp cho các bài toán. Các thuật toán đó có thể chia làm ba nhóm chính:

- Học có giám sát (Supervised learning).

Là giải thuật điều chỉnh các trọng số kết nối dựa vào sự khác biệt giữa ngõ ra thực tế của mạng (actual network output) và ngõ ra mong muốn (target hoặc desired network output) ứng với một tập tín hiệu vào. Học giám sát đòi hỏi phải có một teacher hay supervisor cung cấp ngõ ra mong muốn, vì thế người ta gọi là học có thầy hay học có giám sát.

- Học không giám sát (Unsupervised Learning).

Trong phương pháp học không giám sát, không đòi hỏi tập ngõ ra mong muốn, vì thế người ta gọi là học không thầy hay học không giám sát. Trong quá trình huấn luyện, chỉ có tập dữ liệu vào được đưa vào mạng, mạng tự điều chỉnh theo nguyên tắc gộp các mẫu dữ liệu vào có đặc điểm tương tự thành từng nhóm.

Mạng neural không được giám sát được huấn luyện bằng cách để mạng neural liên tục tự điều chỉnh theo các đầu vào mới. Chúng được sử dụng để rút ra các suy luận từ các tập dữ liệu bao gồm dữ liệu đầu vào mà không có các phản hồi được gắn nhãn.

- Học tăng cường (reinforcement learning).

Học tăng cường là một giải thuật đặc biệt của học có giám sát. Thay vì phải do thầy cung cấp ngõ ra mong muốn thì giải thuật này sẽ nhờ một “chuyên gia” (critic) ước lượng ngõ ra tốt nhất ứng với một ngõ vào cho trước. Tiêu biểu cho học tăng cường là giải thuật di truyền (Genetic Algorithm - GA)

Học tăng cường là một loại kỹ thuật học máy trong đó tác nhân máy tính học cách thực hiện một tác vụ thông qua các tương tác thử và lỗi lặp đi lặp lại với môi trường động. Cách tiếp cận học tập này cho phép người đại diện đưa ra một loạt quyết định nhằm tối đa hóa số liệu thưởng cho nhiệm vụ mà không cần sự can thiệp của con người và không được lập trình rõ ràng để đạt được nhiệm vụ.

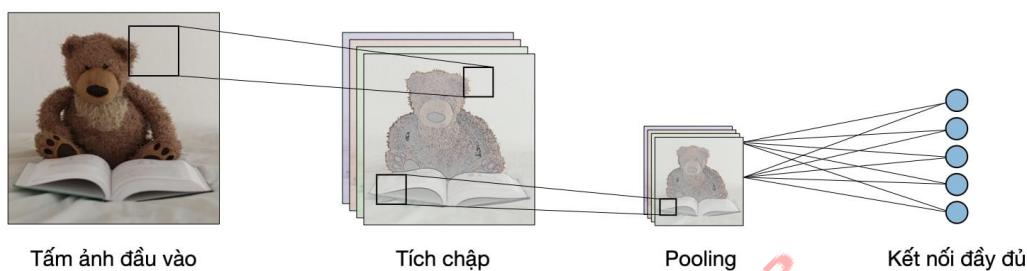
Các chương trình AI được huấn luyện với học tập cũng cố đánh bại người chơi con người trong các trò chơi trên bàn cờ như cờ vây và cờ vua, cũng như trò chơi điện tử. Mặc dù học tăng cường không phải là một khái niệm mới, nhưng tiến bộ gần đây trong học sâu và sức mạnh tính toán đã giúp nó có thể đạt được một số kết quả đáng chú ý trong lĩnh vực trí tuệ nhân tạo.

### 1.2.5. Mạng neural tích chập

#### a, Cách thức hoạt động của CNN

Một mạng neural tích chập có thể có hàng chục hoặc hàng trăm lớp mà mỗi lớp học để phát hiện các đặc điểm khác nhau của hình ảnh.

CNN bao gồm một lớp đầu vào, một lớp đầu ra và nhiều lớp ẩn ở giữa. Các lớp này thực hiện các hoạt động làm thay đổi dữ liệu với mục đích tìm hiểu các đặc trưng cụ thể cho dữ liệu. Một số lớp phổ biến nhất là: Convolution, ReLU, Pooling, ReLU.

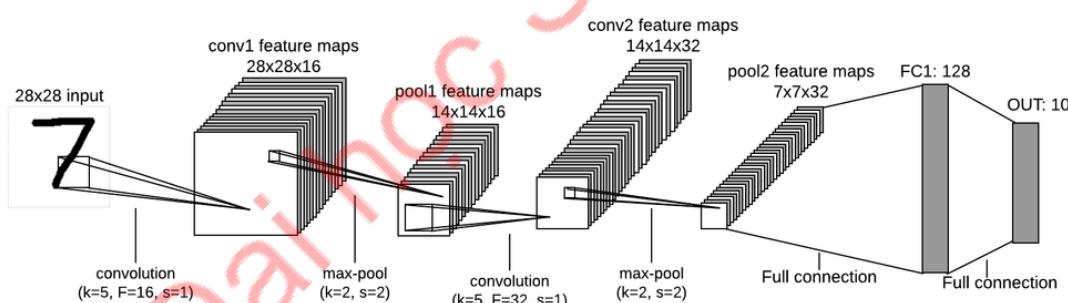


Hình 1.18. Kiến trúc truyền thống của một mạng CNN

#### b, Mô hình tổng quát của CNN

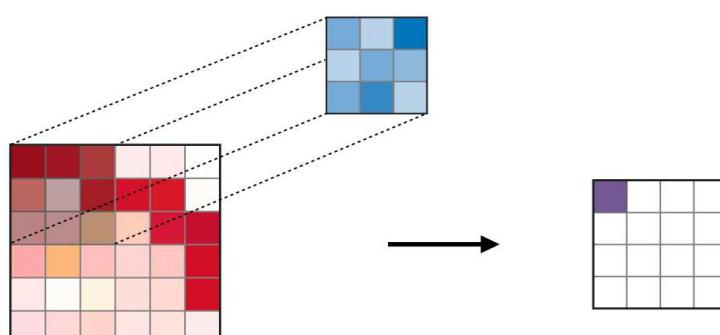
Mô hình tổng quát của một mạng CNN:

Input image → Convolutional layer (Conv) + Pooling layer (Pool) → Fully connected layer (FC) → Output.



Hình 1.19. Ví dụ mô hình CNN

**Convolution:** Đặt các hình ảnh đầu vào thông qua một tập hợp các bộ lọc tích chập, mỗi bộ lọc trong số đó sẽ kích hoạt một số đặc trưng nhất định từ hình ảnh.

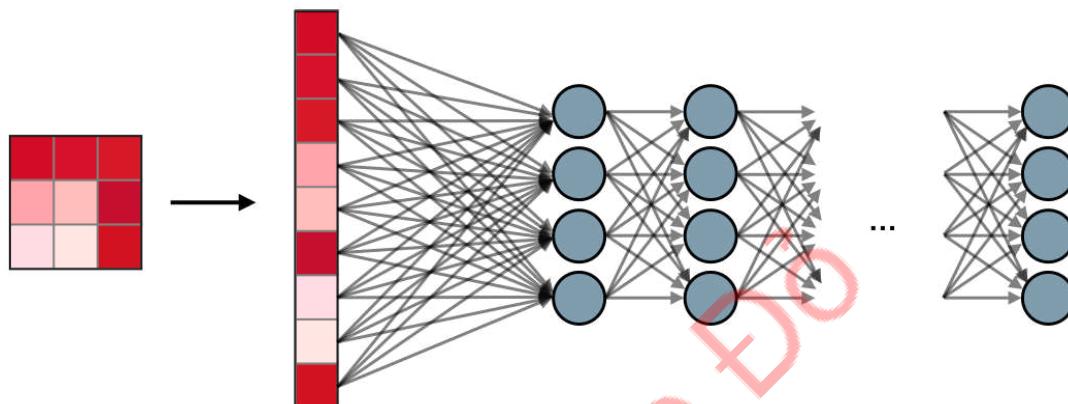


Hình 1.20. Minh họa Convolution

**ReLU:** Cho phép huấn luyện nhanh hơn và hiệu quả hơn bằng cách ánh xạ các giá trị âm về 0 và duy trì các giá trị dương. Điều này đôi khi được gọi là **kích hoạt**, bởi vì chỉ các đặc trưng đã được kích hoạt mới được chuyển sang lớp tiếp theo.

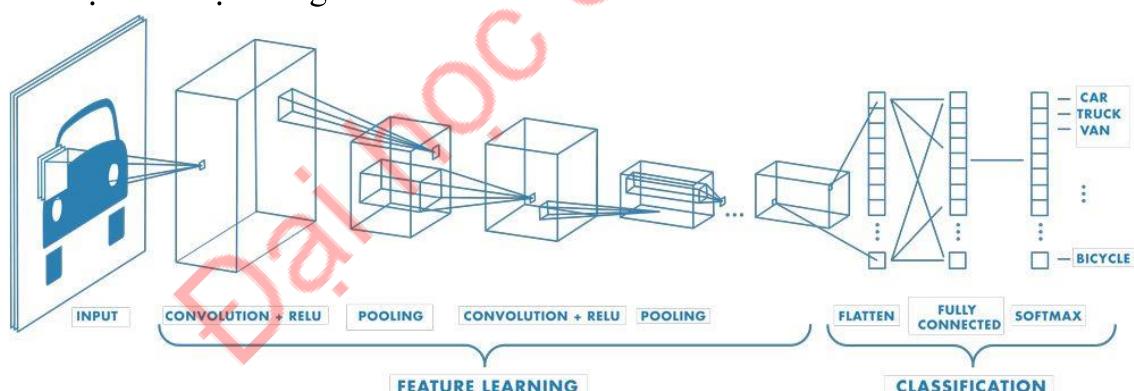
**Pooling:** Đơn giản hóa đầu ra bằng cách thực hiện lấy mẫu xuống phi tuyến, giảm số lượng các tham số mà mạng cần tìm hiểu.

Đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNNs, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.



Hình 1.21. Minh họa tầng kết nối đầy đủ

Các thao tác này được lặp lại trên hàng chục hoặc hàng trăm lớp, với mỗi lớp học để xác định các đặc trưng khác nhau.



Hình 1.22. Minh họa mạng CNN

Hình 1.14 là ví dụ về một mạng có nhiều lớp chập. Các bộ lọc được áp dụng cho mỗi hình ảnh huấn luyện ở các độ phân giải khác nhau và đầu ra của mỗi hình ảnh biến đổi được sử dụng làm đầu vào cho lớp tiếp theo.

### Trọng số và độ lệch chia sẻ

Giống như một mạng neural truyền thống, CNN có trọng số và độ lệch. Mô hình học các giá trị này trong quá trình huấn luyện và nó liên tục cập nhật chúng với mỗi ví dụ huấn luyện mới. Tuy nhiên, trong trường hợp của CNN, trọng số và độ lệch là giống nhau đối với tất cả các neuron ẩn trong một lớp nhất định.

Điều này có nghĩa là tất cả các neuron ẩn đang phát hiện cùng một đặc điểm, chẳng hạn như một cạnh hoặc một đốm màu, ở các vùng khác nhau của hình ảnh. Điều này làm cho mạng có khả năng dịch các đối tượng trong một hình ảnh. Ví dụ: một mạng được huấn luyện để nhận dạng ô tô sẽ có thể làm như vậy ở bất kỳ nơi nào có ô tô trong hình ảnh.

### Các lớp phân loại

Sau khi tìm hiểu các đặc trưng trong nhiều lớp, kiến trúc của CNN chuyển sang phân loại.

Lớp tiếp theo đến lớp cuối cùng là lớp được kết nối dày đủ tạo ra một vector có K chiều trong đó K là số lớp mà mạng có thể dự đoán. Vector này chứa các xác suất cho mỗi lớp của bất kỳ hình ảnh nào đang được phân loại.

Lớp cuối cùng của kiến trúc CNN sử dụng một lớp phân loại như softmax để cung cấp đầu ra phân loại.

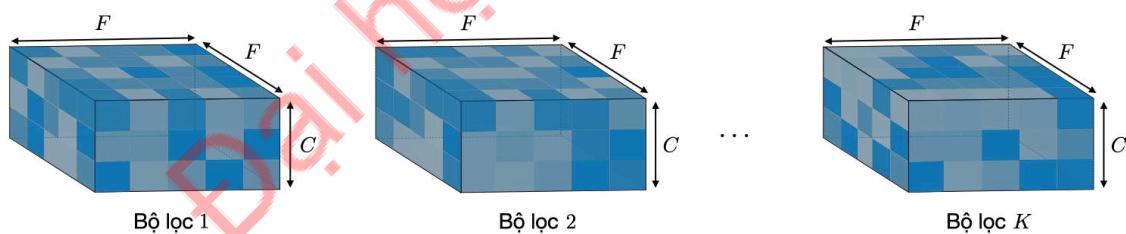
c, Điều chỉnh các siêu tham số

- Các siêu tham số của bộ lọc

Tầng tích chập chứa các bộ lọc mà rất quan trọng cho ta khi biết ý nghĩa đằng sau các siêu tham số của chúng.

#### - Các chiều của một bộ lọc

Một bộ lọc kích thước  $F \times F$  áp dụng lên đầu vào chứa C kênh (channels) thì có kích thước tổng thể là  $F \times F \times C$  thực hiện phép tích chập trên đầu vào kích thước  $I \times I \times C$  và cho ra một feature map (hay còn gọi là activation map) có kích thước  $O \times O \times 1$ .

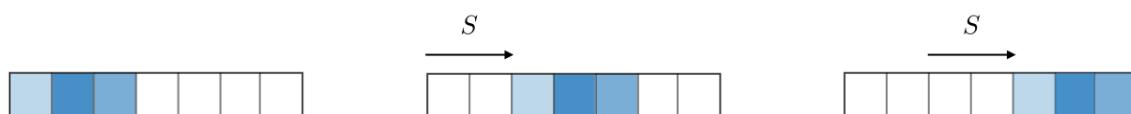


Hình 1.23. Minh họa các bộ lọc có kích thước khác nhau

Lưu ý: Việc áp dụng K bộ lọc có kích thước  $F \times F$  cho ra một feature map có kích thước  $O \times O \times K$ .

#### - Stride

Đối với phép tích chập hoặc phép pooling, độ trượt  $S$  ký hiệu số pixel mà cửa sổ sẽ di chuyển sau mỗi lần thực hiện phép tính.

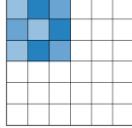
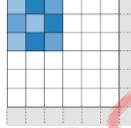
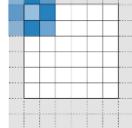


Hình 1.24. Minh họa độ trượt  $S$

### - Zero-padding

Zero-padding là tên gọi của quá trình thêm  $P$  số khống vào các biên của đầu vào. Giá trị này có thể được lựa chọn thủ công hoặc một cách tự động bằng một trong ba những phương pháp mô tả bên dưới:

Bảng 1.1. So sánh các phương pháp Zero-padding

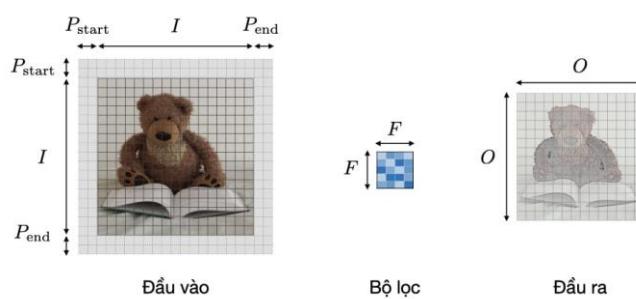
Phương pháp	Valid	Same	Full
Giá trị	$P = 0$	$P_{start} = \left\lceil \frac{S \left[ \frac{I}{S} \right] - I + F - S}{2} \right\rceil$ $P_{end} = \left\lceil \frac{S \left[ \frac{I}{S} \right] - I + F - S}{2} \right\rceil$	$P_{start} \in [0, F - 1]$ $P_{end} = F - 1$
Minh họa			
Mục đích	<ul style="list-style-type: none"> <li>Không sử dụng padding</li> <li>Bỏ phép tích chập cuối nếu số chiều không khớp</li> </ul>	<ul style="list-style-type: none"> <li>Sử dụng padding để làm cho feature map có kích thước <math>\left[ \frac{I}{S} \right]</math></li> <li>Kích thước đầu ra thuận lợi về mặt toán học</li> <li>Còn được gọi là 'half padding'</li> </ul>	<ul style="list-style-type: none"> <li>Padding tối đa sao cho các phép tích chập có thể được sử dụng tại các rìa của đầu vào</li> <li>Bộ lọc 'thầy' được đầu vào từ đầu đến cuối</li> </ul>

#### ▪ Điều chỉnh siêu tham số

#### - Tính tương thích của tham số trong tầng tích chập

Bằng cách ký hiệu  $I$  là độ dài kích thước đầu vào,  $F$  là độ dài của bộ lọc,  $P$  là số lượng zero padding,  $S$  là độ trượt, ta có thể tính được độ dài  $O$  của feature map theo một chiều bằng công thức:

$$O = \frac{I - F + P_{start} + P_{end}}{S} + 1$$



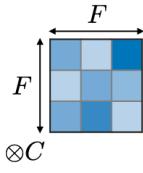
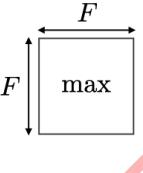
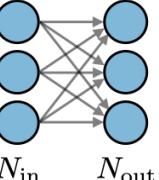
*Hình Error! No text of specified style in document..1: Minh họa tính tương thích của tham số*

Lưu ý: Trong một số trường hợp,  $P_{start}=P_{end} \triangleq P$ , ta có thể thay thế  $P_{start}+P_{end}$  bằng  $2P$  trong công thức trên.

### - Hiểu về độ phức tạp của mô hình

Để đánh giá độ phức tạp của một mô hình, cách hữu hiệu là xác định số tham số mà mô hình đó sẽ có. Trong một tầng của mạng neural tích chập, nó sẽ được tính toán như sau:

Bảng 1.2. So sánh độ phức tạp của mô hình

	CONV	POOL	FC
Minh họa			
Kích thước đầu vào	$I \times I \times C$	$I \times I \times C$	$N_{in}$
Kích thước đầu ra	$O \times O \times K$	$O \times O \times C$	$N_{out}$
Số lượng tham số	$(F \times F \times C + 1) \cdot K$	0	$(N_{in} + 1) \times N_{out}$
Lưu ý	<ul style="list-style-type: none"> <li>Một tham số bias với mỗi bộ lọc</li> <li>Trong đa số trường hợp, <math>S &lt; F</math></li> <li>Một lựa chọn phổ biến cho <math>K</math> là <math>2C</math></li> </ul>	<ul style="list-style-type: none"> <li>Phép pooling được áp dụng lên từng kênh (channel-wise)</li> <li>Trong đa số trường hợp, <math>S = F</math></li> </ul>	<ul style="list-style-type: none"> <li>Đầu vào được làm phẳng</li> <li>Mỗi neuron có một tham số bias</li> <li>Số neuron trong một tầng FC phụ thuộc vào ràng buộc kết cấu</li> </ul>

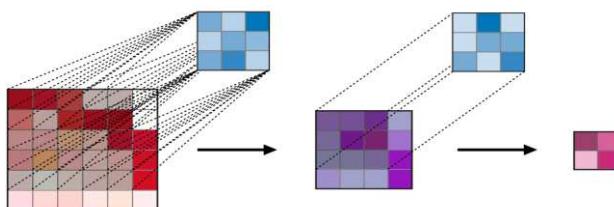
### - Trường thu cảm

Trường thu cảm (receptive field) tại tầng k là vùng được ký hiệu  $R_k \times R_k$  của đầu vào mà những pixel của activation map thứ k có thể "nhìn thấy".

Gọi  $F_j$  là kích thước bộ lọc của tầng j và  $S_i$  là giá trị độ trượt của tầng i và để thuận tiện, ta mặc định  $S_0=1$ , trường thu cảm của tầng k được tính toán bằng công thức:

$$R_k = 1 + \sum_{j=1}^k (F_j - 1) \prod_{i=0}^{j-1} S_i \quad (2.1)$$

Trong ví dụ hình 1.27, ta có  $F_1=F_2=3$  và  $S_1=S_2=1$ , nên cho ra được  $R_2=1+2 \cdot 1+2 \cdot 1=5$ .



Hình Error! No text of specified style in document..2: Minh họa tính toán thuỷ cám tầng k

d, Các hàm kích hoạt thường gặp

- Rectified Linear Unit

Tầng rectified linear unit (ReLU) là một hàm kích hoạt gg được sử dụng trên tất cả các thành phần. Mục đích của nó là tăng tính phi tuyến tính cho mạng. Những biến thể khác của ReLU được tổng hợp ở bảng dưới:

Bảng 1.3. Biến thể của ReLU

ReLU	Leaky ReLU	ELU
$g(z)=\max(0,z)$	$g(z)=\max(\epsilon z, z)$ với $\epsilon \ll 1$	$g(z)=\max(\alpha(e^z-1), z)$ với $\alpha \ll 1$
• Độ phức tạp phi tuyến tính có thể thông dịch được về mặt sinh học	• Gán ván đề ReLU chết cho những giá trị âm	• Khả vi tại mọi nơi

e, Softmax

Bước softmax có thể được coi là một hàm logistic tổng quát lấy đầu vào là một vector chứa các giá trị  $x \in R^n$  và cho ra là một vector gồm các xác suất  $p \in R^n$  thông qua một hàm softmax ở cuối kiến trúc

Huấn luyện mạng phức tạp từ đầu hay sử dụng các mạng được huấn luyện trước để nhanh chóng học các nhiệm vụ mới.

Tạo mạng sâu mới cho các nhiệm vụ phân loại và hồi quy hình ảnh bằng cách xác định kiến trúc mạng và huấn luyện mạng từ đầu. Việc tinh chỉnh mạng phân loại hình ảnh được huấn luyện trước bằng cách học chuyển tiếp thường nhanh hơn và dễ dàng hơn nhiều so với việc huấn luyện từ đầu. Sử dụng mạng sâu được huấn luyện trước

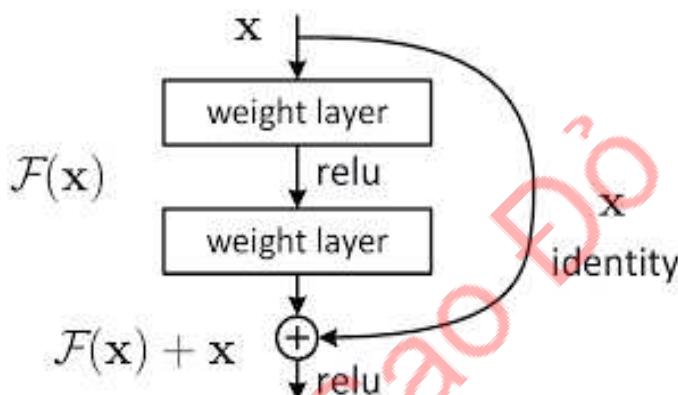
cho phép nhanh chóng tìm hiểu các tác vụ mới mà không cần xác định và huấn luyện mạng mới, có hàng triệu hình ảnh hoặc có GPU mạnh mẽ.

Sau khi xác định kiến trúc mạng, ta phải xác định các tham số huấn luyện bằng cách sử dụng hàm trainingOptions. Sau đó, có thể huấn luyện mạng bằng cách sử dụng trainNetwork. Sử dụng mạng được huấn luyện để dự đoán nhãn lớp hoặc phản hồi số.

Ta có thể huấn luyện mạng neural phức hợp trên một CPU, một GPU, nhiều CPU hoặc GPU hoặc song song trên một cụm hoặc trong đám mây. Chỉ định môi trường thực thi bằng cách sử dụng chức năng trainingOptions.

### 1.2.6. Mô hình học máy ResNet-50

#### a, Kiến Trúc Residual Block



Hình 1.25. Kiến trúc residual Block

- ResNet-50 sử dụng các khối còn được gọi là "residual blocks" để xây dựng kiến trúc của nó.
- Mỗi residual block bao gồm các layers thực hiện các phép tích chập, batch normalization, và activation functions như ReLU.

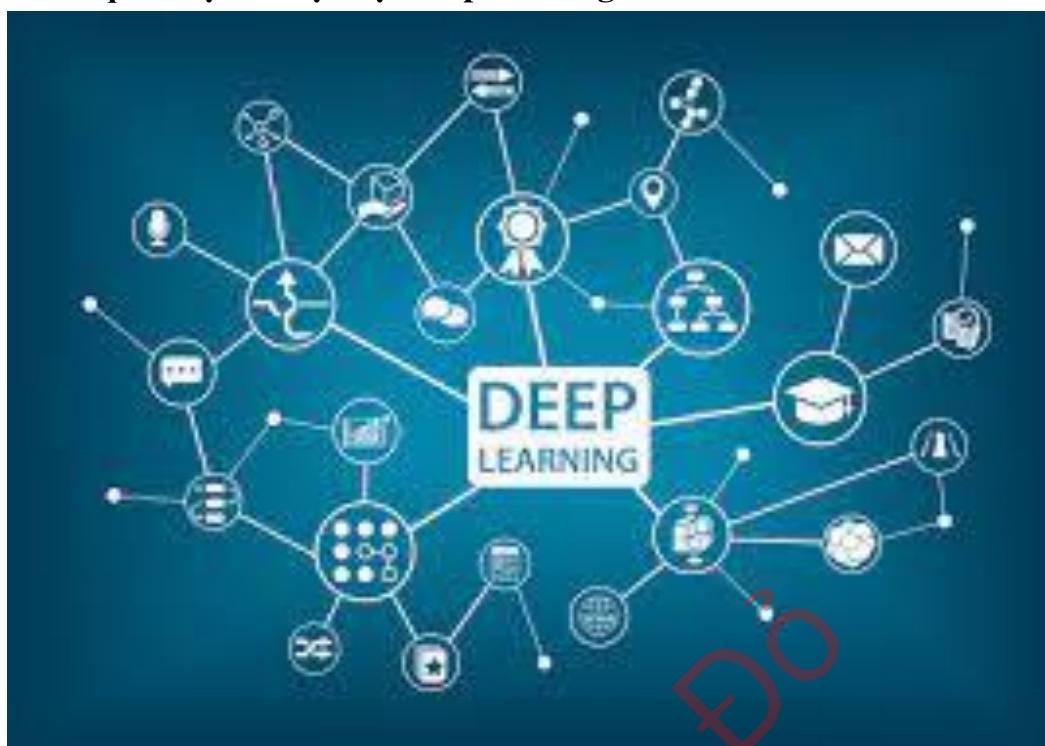
#### Ý Tưởng Chính - Residual Learning:

- Ý tưởng chính của ResNet là sử dụng "residual learning" để học sự cộng dồn của các đặc trưng.
- Thay vì học đầu ra y của một lớp theo cách truyền thống, ResNet học sự chênh lệch hoặc "residual" ( $F(x) - x$ ), nơi x là đầu vào và  $F(x)$  là đầu ra dự đoán của lớp.

#### Residual Block Cơ Bản

- Một residual block thường bao gồm hai nhánh: nhánh chính và nhánh "skip connection".
- Nhánh chính thực hiện các phép tích chập và áp dụng batch normalization và ReLU.
- Nhánh "skip connection" chỉ là đầu vào không thay đổi và được cộng vào đầu ra của nhánh chính.

### Giao quan hệ với hiện đại Deep learning



Hình 1.26. Mô hình Giao quan hệ với hiện đại với Deep learning

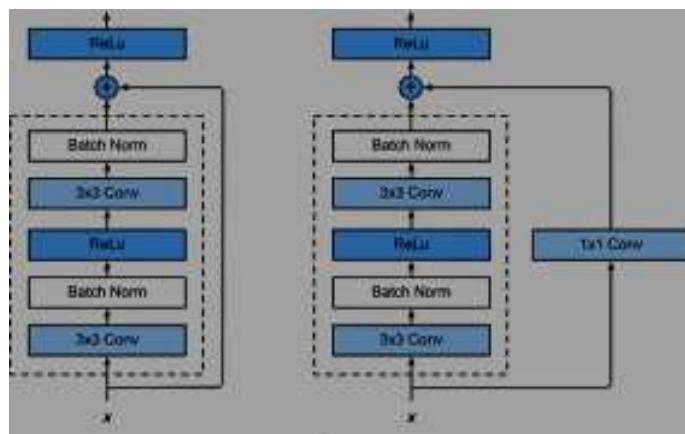
- ResNet đã đóng góp đáng kể cho sự phát triển của deep learning, giúp giải quyết vấn đề của việc huấn luyện các mô hình sâu.
- "Skip connections" giúp tránh được vấn đề của việc mất thông tin (vanishing gradient) trong quá trình lan truyền ngược.

ResNet-50 và các biến thể của nó đã trở thành một trong những kiến trúc quan trọng trong lĩnh vực thị giác máy tính và nó thường được sử dụng cho nhiều nhiệm vụ như nhận diện đối tượng và phân loại hình ảnh.

#### Kiến trúc ResNet-50

- ResNet-50 là một phiên bản cụ thể của ResNet với 50 lớp, bao gồm nhiều residual blocks.
  - Kiến trúc của ResNet-50 có các lớp tích chập, batch normalization, và activation functions, với "skip connections" được thêm vào mỗi một số lớp để tăng cường sự chuyển đổi thông tin qua mạng.

## b, Giới thiệu mô hình ResNet-50



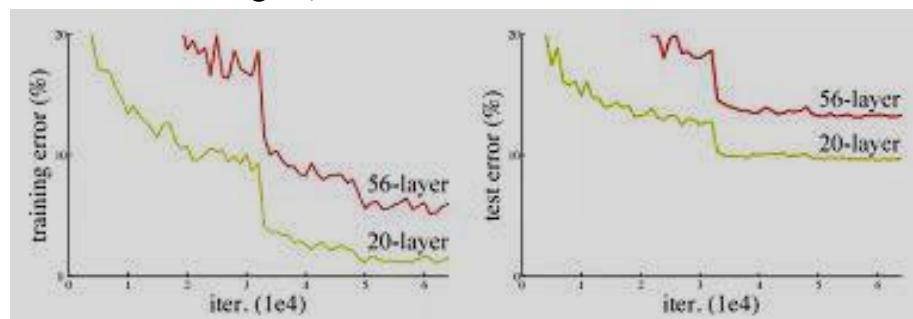
Hình 1.27. Mô hình ResNet-50

**ResNet-50** là một kiến trúc mô hình học máy sử dụng trong thị giác máy tính, nổi tiếng với cơ chế ‘residual learning’ hoặc ‘skip connections’ giúp giải quyết vấn đề vanishing gradient khi huấn luyện mô hình sâu. Được giới thiệu bởi Kaiming He, Xiangyu Zhang, Shaoqing Ren và Jian Sun trong bài báo khoa học có tiêu đề “Deep Residual Learning for Image Recognition” (2016), ResNet đã đạt được thành công lớn trong nhiều nhiệm vụ như phân loại ảnh và nhận diện đối tượng.

**ResNet-50** được thiết kế để giải quyết các vấn đề của việc huấn luyện các mô hình sâu. Tên gọi “ResNet” là viết tắt của “Residual Network”, mô hình dễ dàng học được các biểu diễn phức tạp thông qua các lớp sâu mà không gặp vấn đề vanishing gradient.

Trong ResNet-50, số 50 thường chỉ số lượng lớp (layers) trong mô hình. Mô hình này thường bao gồm các lớp convolutional, batch normalization, activation, và các khối residual.

**Công dụng:** ResNet-50 thường được sử dụng trong các ứng dụng nhận diện hình ảnh, phân loại ảnh, và các nhiệm vụ liên quan đến thị giác máy tính. Đối với các loại bài toán yêu cầu mô hình có khả năng học được các biểu diễn phức tạp, ResNet-50 là một lựa chọn phổ biến do khả năng hiệu suất cao của nó.



Hình 1.28. Biểu đồ học máy của ResNet-50

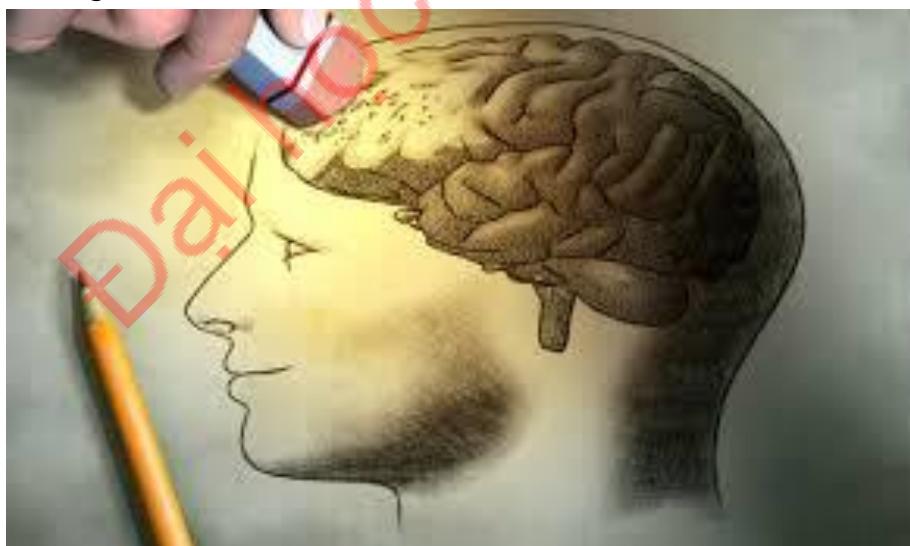
### 1.3. Tổng quan về bệnh Alzheimer

#### 1.3.1. Khái niệm bệnh Alzheimer

Cấu trúc não có độ phức tạp lớn và khó khăn trong việc chẩn đoán. Cùng với đó, ở các tuyến bệnh viện có chuyên môn và kinh nghiệm của các bác sĩ là khác nhau. Chính vì vậy điều này dẫn đến trường hợp kết quả chẩn đoán có thể sai sót và gây ra hậu quả rất nghiêm trọng hoặc những bệnh nhân ở các tuyến vùng sâu vùng xa có xu hướng dồn về các bệnh viện tuyến trên để chẩn đoán và điều trị gây áp lực lớn cho các bác sĩ và cơ sở vật chất ở các bệnh viện này. Hơn nữa, bệnh nhân phải di chuyển một quãng đường xa để tới các bệnh viện này dẫn tới bệnh trở nặng hoặc tăng nguy cơ tử vong. Vì những lí do trên, bài toán đặt ra là làm sao tăng độ chính xác của kết quả chẩn đoán tại các tuyến bệnh viện, đồng thời giảm áp lực cho các bác sĩ và tiết kiệm thời gian chẩn đoán, điều trị bệnh cho bệnh nhân.

Bài toán nhận dạng và phân loại ảnh bất thường ở não đang trở thành nhu cầu cấp thiết và quan trọng trong lĩnh vực y tế hiện nay và việc áp dụng công nghệ thông tin vào các hệ thống nhận dạng đem lại những hiệu quả thiết thực góp phần nâng cao khả năng chẩn đoán và cứu sống thêm nhiều bệnh nhân đột quy kịp thời. Các phương pháp sử dụng: thu thập dữ liệu và rút trích đặc trưng.

Trong lĩnh vực chăm sóc sức khỏe, bạn có thể thấy thuật ngữ "Big Data" khi xem hình ảnh MRI hoặc báo cáo y tế. Nhiều bệnh viện đang chuyển sang sử dụng Dữ liệu lớn (Big Data) để phát triển các phương pháp điều trị cho những căn bệnh từng được coi là không thể chữa khỏi.



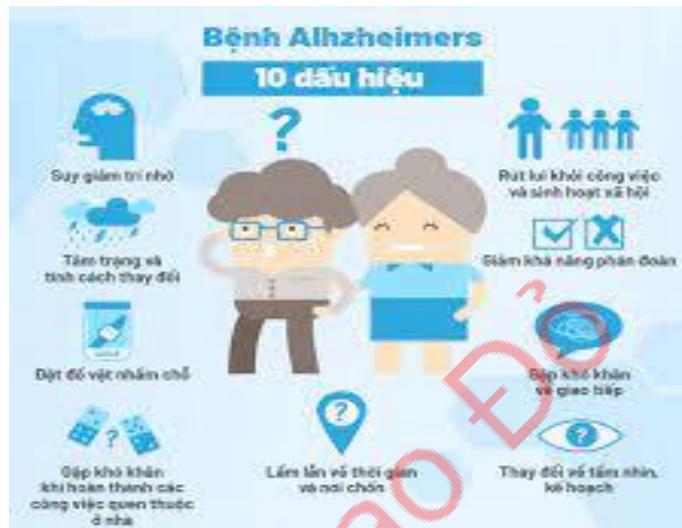
Hình 1.29. Minh họa bệnh nhân Alzheimer

Bệnh Alzheimer là một trong những căn bệnh về não gây ảnh hưởng đến suy nghĩ, hành vi và trí nhớ của người bệnh. Đây không phải là căn bệnh lão khoa hoặc bệnh thần kinh thông thường. Bệnh Alzheimer gây ảnh hưởng không chỉ đối với người bệnh mà còn tác động đến cả người đi chăm sóc bệnh nhân bởi lẽ chăm sóc bệnh nhân Alzheimer rất dễ dẫn đến trầm cảm hoặc có cảm xúc vô cùng căng thẳng. Hiện nay, bệnh Alzheimer ở Việt Nam đã ảnh hưởng ngày càng nhiều trong giới trẻ.

## Nguyên nhân

Mặc dù chưa tìm ra được nguyên nhân cụ thể nên các nhà khoa học chỉ có thể xác định, khi bị bệnh Alzheimer là lúc các tế bào não lưu trữ và xử lý thông tin bắt đầu bị suy yếu và chết. Ngoài ra, các protein bất thường được sản sinh ra, tạo nên những mảng bám và tích tụ quanh và bên trong các tế bào gây cản trở đến quá trình truyền tải thông tin. ảnh hưởng đến sự vận động thần kinh não bộ của người bệnh.

### 1.3.2. Các triệu chứng của bệnh Alzheimer

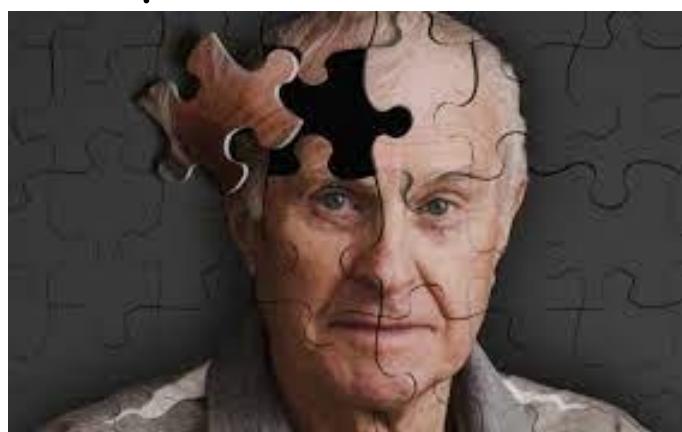


Hình 1.30. 10 dấu hiệu của bệnh Alzheimer

## Dấu hiệu, triệu chứng bệnh Alzheimer

- Triệu chứng đầu tiên là đăng trí, bị quên tên hoặc nơi vừa đặt đồ vật, hay còn gọi là bệnh mất trí nhớ Alzheimer
- Trí nhớ và tư duy bắt thường, có thể quên tên người quen, hỏi cùng một câu hỏi, kể cùng một câu chuyện nhiều lần, khó khăn trong ghi nhớ mọi thứ trong cuộc sống.
- Ở những giai đoạn sau, người bệnh sẽ cần giúp đỡ nhiều hơn từ người xung quanh và cần có một chế độ chăm sóc đặc biệt toàn diện bởi lẽ bệnh nhân bị Alzheimer giai đoạn cuối thường đi lang thang hoặc bị lạc, thay đổi tính cách cảm xúc.

## Đối tượng nguy cơ mắc bệnh Alzheimer



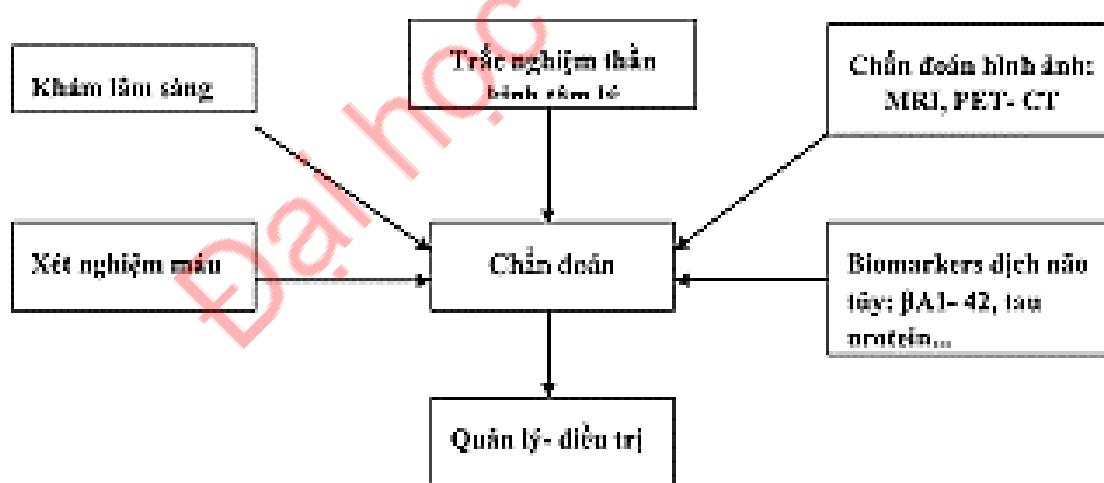
Hình 1.31. Đối tượng có nguy cơ mắc bệnh Alzheimer

- Là một căn bệnh khá phổ biến, bệnh Alzheimer thường thấy ở những người cao tuổi. Một bộ phận những người có rối loạn não bẩm sinh hoặc bị chấn thương cũng có nguy cơ cao bị bệnh. Hiện nay, một bộ phận giới trẻ cũng đang bị đe dọa bởi bệnh Alzheimer.

- Những yếu tố sau sẽ làm tăng nguy cơ mắc bệnh Alzheimer:
  - Tuổi tác: những người cao tuổi, đặc biệt sau 65 tuổi sẽ có nguy cơ cao mắc bệnh Alzheimer;
  - Gia đình có người từng có tiền sử mắc bệnh Alzheimer. Các nhà khoa học đã nghiên cứu và chứng minh gen làm tăng nguy cơ mắc bệnh Alzheimer nhưng không chứng minh chắc chắn rằng cá thể mang gen đó sẽ tiến triển thành bệnh
  - Những người có tiền căn chấn thương đầu hoặc bị suy giảm nhận thức nhẹ;
  - Những người có lối sống không khoa học như: sử dụng các chất kích thích, chế độ ăn thiếu rau xanh và trái cây, ít vận động;
  - Những người trong quá trình học tập và giao tiếp gặp một số vấn đề về mức độ giáo dục chính quy thấp, công việc thiêu các hoạt động cần thử thách trí não hoặc ít giao tiếp xã hội.

Trong các nguy cơ trên, yếu tố nguy cơ được nhắc đến nhiều nhất là tuổi tác. Bệnh Alzheimer ở người cao tuổi trở nên rất phổ biến

### 1.3.3. Quy trình chẩn đoán bệnh trong y học



Hình 1.32. Các bước chẩn đoán bệnh

#### Tìm đúng bác sĩ

Vì không có bác sĩ chuyên về chẩn đoán và điều trị chứng mất trí nhớ hay bệnh Alzheimer's. Nhiều người liên lạc với bác sĩ chăm sóc chính hoặc bác sĩ nội khoa đã được cấp phép để trình bày về lo ngại của họ. Bác sĩ chăm sóc chính thường giám sát quá trình chẩn đoán và đưa ra phương pháp điều trị.

Trong nhiều trường hợp, bác sĩ chăm sóc chính sẽ chuyển người bệnh đến một chuyên gia, như:

- Bác sĩ chuyên khoa thần kinh, chuyên về bệnh lý não và hệ thần kinh.
- Bác sĩ tâm thần, chuyên về các rối loạn ảnh hưởng đến tính cách hay hoạt động của trí não.
- Nhà tâm lý học được đào tạo chuyên sâu về các thử nghiệm kiểm tra trí nhớ, độ tập trung, khả năng giải quyết vấn đề, ngôn ngữ và các chức năng thần kinh khác.

### **Đánh giá tâm trạng và tình trạng tâm thần**

Kiểm tra tình trạng tâm thần giúp bác sĩ có ý niệm chung về việc trí não hoạt động tốt đến mức nào. Sự kiểm tra này cung cấp cái nhìn tổng quát, cho biết liệu người bệnh có thể:

- Nhận thức được các triệu chứng hay không.
- Biết được ngày, giờ và nơi họ đang ở hay không.
- Nhớ được một ít từ ngữ, theo kịp hướng dẫn và thực hiện được các phép tính đơn giản hay không.
- Bác sĩ có thể hỏi người bệnh về địa chỉ của họ, năm hiện tại hay ai đang làm tổng thống. Người bệnh cũng có thể được yêu cầu đánh và ngược 1 từ, vẽ đồng hồ hay bắc trước 1 hành động nào đó. Từ đó bác sĩ đánh giá được tâm trạng tinh thần sức khỏe nhằm phát hiện sự suy nhược hoặc các bệnh khác dẫn đến tình trạng mất trí nhớ.

### **Khám lâm sàng và xét nghiệm chẩn đoán**

Bác sĩ sẽ thực hiện các thủ thuật để kiểm tra sức khỏe tổng quát của người bệnh như đánh giá chế độ dinh dưỡng, đo huyết áp hay nhịp tim. Mẫu máu và nước tiểu sẽ được thu thập, một số xét nghiệm trong phòng thí nghiệm sẽ được tiến hành.

Thông tin từ các xét nghiệm này giúp xác định các rối loạn như bệnh thiếu máu, tiêu đường, bệnh gan hay thận, sự thiếu hụt một loại vitamin nào đó, sự bất thường ở tuyến giáp và các vấn đề tim mạch. Tất cả các tình trạng này có thể dẫn đến sự suy nghĩ lẩn lộn, vấn đề về trí nhớ hoặc các triệu chứng khác tương tự bệnh suy giảm trí nhớ.

### **Khám thần kinh**



Hình 1.33. Minh họa hệ thần kinh

Một bác sĩ, đôi khi là bác sĩ chuyên khoa thần kinh, người chuyên về bệnh lý não và hệ thần kinh, sẽ tiến hành đánh giá cẩn trọng để xác định rằng các vấn đề của người bệnh liệu có phải là dấu hiệu của một chứng rối loạn về não khác với bệnh Alzheimer's hay không.

Bác sĩ cũng kiểm tra phản xạ, khả năng cân bằng, chuyển động mắt, ngôn từ và cảm giác của người bệnh. Bác sĩ tìm kiếm những dấu hiệu của chứng đột quy nhẹ hoặc nặng, bệnh Parkinson, u não, tích tụ dịch trong não và các bệnh khác có thể làm suy giảm trí nhớ hay suy nghĩ.

Quá trình khám thần kinh có thể bao gồm việc chụp cộng hưởng từ (MRI) hay chụp cắt lớp điện toán (CT). MRI và CT có thể phát hiện các khối u, bằng chứng của chứng đột quy nhẹ hoặc nặng, tổn thương từ các chấn thương đầu nghiêm trọng hay sự tích tụ dịch trong não. Bảo hiểm Medicare sẽ chi trả cho xét nghiệm chụp cắt lớp nhò phát xạ positron (PET) để hỗ trợ chẩn đoán trong một số trường hợp nhất định.

### **Kết quả chẩn đoán khẳng định là bệnh Alzheimer's**

Khi hoàn tất các xét nghiệm, bác sĩ sẽ xem xét kết quả và cho biết kết luận. Kết quả chẩn đoán bệnh Alzheimer's phản ánh sự đánh giá tốt nhất của bác sĩ về các triệu chứng của người bệnh.

Bạn có thể yêu cầu bác sĩ giải thích:

- Tại sao kết quả chẩn đoán là bệnh Alzheimer's
- Người bệnh đang ở giai đoạn nào của căn bệnh
- Có thể mong đợi điều gì trong tương lai
- Tìm hiểu xem liệu bác sĩ đó có tiếp tục điều trị không, nếu không, ai sẽ là bác sĩ chăm sóc chính. Bác sĩ chẩn đoán có thể sắp xếp cuộc hẹn tái khám hay giới thiệu cho bạn một bác sĩ khác.

#### **1.3.4. Quy trình điều trị**

##### **Triệu chứng liên quan đến nhận thức: Điều trị dùng thuốc**

Triệu chứng liên quan đến nhận thức ảnh hưởng đến trí nhớ, khả năng nhận biết, ngôn ngữ, óc phán đoán và các quá trình tư duy khác. Hiệp hội Quản lý Thuốc và Thực phẩm (FDA) Hoa Kỳ đã phê chuẩn lưu hành hai loại thuốc dùng để điều trị các triệu chứng liên quan đến nhận thức của bệnh Alzheimer's.

Chất ức chế Cholinesterase (KOH-luh-NES-ter-ays) giúp ngăn ngừa sự giảm hàm lượng acetylcholine (a-SEA-til-KOHlean), một chất dẫn truyền thần kinh quan trọng cho quá trình ghi nhớ và học hỏi. Bằng cách giữ cho hàm lượng acetylcholine ở mức cao, những loại thuốc này sẽ hỗ trợ cho quá trình truyền tín hiệu giữa các tế bào thần kinh.

Ba loại chất ức chế cholinesterase thường được kê trong đơn thuốc là:

- Donepezil (Aricept), được cấp phép dùng để điều trị trong tất cả các giai đoạn của bệnh Alzheimer's.

- Rivastigmine (Exelon), được cấp phép dùng để điều trị bệnh Alzheimer's trong giai đoạn nhẹ và vừa.
- Galantamine (Razadyne), được cấp phép dùng để điều trị bệnh Alzheimer's trong giai đoạn nhẹ và vừa.
- Memantine (Namenda) giúp điều hòa hoạt động của glutamate, một loại chất chuyển dẫn truyền thần kinh khác liên quan đến quá trình ghi nhớ và học hỏi. Loại thuốc này được cấp phép dùng để điều trị bệnh Alzheimer's trong giai đoạn vừa và nặng.



Hình 1.34. Liệt kê các loại thuốc ức chế bệnh

### **Triệu chứng liên quan đến hành vi: Nguyên nhân và phương pháp điều trị**

Nhiều người cho rằng sự thay đổi trong hành vi của người bệnh là điều khó khăn và đáng lo lắng nhất. Sự thay đổi này bao gồm bối rối, lo âu, gây hấn và xáo trộn giấc ngủ. Nguyên nhân cơ bản nhất gây nên các triệu chứng về mặt tâm thần và hành vi này là do các tế bào não bị tổn thương không ngừng. Các nguyên nhân khác gây ra các triệu chứng liên quan đến hành vi bao gồm:

- Tác dụng phụ của thuốc:
  - Tác dụng phụ của thuốc từ việc uống thuốc theo đơn có thể xảy ra. Trong một vài trường hợp, tương tác thuốc có thể phát sinh do sử dụng nhiều loại thuốc.
  - Tình trạng sức khỏe.
  - Triệu chứng của sự nhiễm trùng hay bệnh tật, những loại bệnh có thể chữa khỏi, có thể ảnh hưởng đến hành vi của người bệnh. Viêm phổi hay nhiễm trùng đường tiết niệu có thể làm cho người bệnh khó chịu. Việc điều trị không dứt điểm viêm tai hay xoang có thể gây choáng váng và đau nhức.
- Tác động của môi trường:
  - Các tình huống ảnh hưởng đến hành vi của người bệnh bao gồm di chuyển đến nơi cư ngụ hay cơ sở chăm sóc mới; nhận thức không đúng về các mối đe dọa;
  - Hoặc sợ hãi và mệt mỏi khi cố gắng nhận biết thế giới hỗn độn xung quanh.

Điều trị dùng thuốc đối với các triệu chứng liên quan đến hành vi  
Có hai loại điều trị cho các triệu chứng liên quan đến hành vi: không sử dụng thuốc và sử dụng thuốc. Điều trị không dùng thuốc nên áp dụng trước.

Điều trị không dùng thuốc (đối với các triệu chứng liên quan đến hành vi)  
Các bước để điều trị không dùng thuốc bao gồm:

- Xác định triệu chứng.
- Tìm hiểu nguyên nhân.
- Thay đổi môi trường chăm sóc để loại bỏ các khó khăn hoặc trở ngại.

Việc xác định nguyên nhân dẫn đến hành vi của người bệnh có thể giúp xác định hướng điều trị phù hợp nhất. Thông thường, nguyên nhân có thể là do sự thay đổi môi trường của người bệnh, như:

- Người chăm sóc mới.
- Nhập viện.
- Có người lạ trong nhà.
- Được yêu cầu tắm hay thay quần áo.

Việc uống thuốc theo đơn có thể có hiệu quả trong việc giúp kiểm soát các triệu chứng liên quan đến hành vi, nhưng phải được sử dụng cẩn trọng và sẽ đạt hiệu quả cao nhất khi kết hợp với các liệu pháp không sử dụng thuốc khác. Các loại thuốc cần chú trọng đến những triệu chứng cụ thể của bệnh để có thể theo dõi được kết quả điều trị.

Việc kê đơn thuốc cho bệnh nhân Alzheimer's phải được tiến hành vô cùng cẩn trọng. Việc sử dụng thuốc để điều trị các triệu chứng liên quan đến hành vi và tâm thần phải được giám sát chặt chẽ bởi bác sĩ và gia đình người bệnh.

## CHƯƠNG 2. MÔ HÌNH PHÂN TÍCH DỮ LIỆU BỆNH ALZHEIMER

Phân tích một mô hình học máy được sử dụng để phân loại hoặc dự đoán bệnh Alzheimer là một quy trình phức tạp và có thể liên quan đến nhiều yếu tố. Sau đây là kiến trúc mô hình để xem xét khi phân tích mô hình học máy trong ngữ cảnh của ứng dụng dự đoán bệnh Alzheimer.

Từ nghiên cứu chúng em đã lựa chọn tạo một mô hình học máy sử dụng ResNet - 50 phân tích dữ liệu Big data của bệnh Alzheimer.

### 2.1. Kiến trúc mô hình

- Loại mô hình: Mô hình được sử dụng có thể là một mô hình học sâu như CNN (Convolutional Neural Network) cho xử lý hình ảnh, hoặc có thể là mô hình học máy truyền thống khác như SVM (Support Vector Machine) hoặc RandomForest.

- Độ sâu của mô hình: Đối với mô hình học sâu, việc kiểm tra độ sâu của mô hình có thể là quan trọng để hiểu cách nó học các đặc trưng từ dữ liệu.

#### 2.1.1. *Dữ liệu*

- Tính đa dạng của dữ liệu: Kiểm tra loại dữ liệu được sử dụng, có phải là hình ảnh MRI, dữ liệu lâm sàng hay các biến khác.

- Kích thước và số lượng dữ liệu: Đánh giá kích thước và số lượng mẫu dữ liệu. Trong nhiều trường hợp, việc có một lượng dữ liệu đủ lớn và đa dạng có thể là yếu tố quyết định hiệu suất của mô hình.

#### 2.1.2. *Tiền xử lý dữ liệu*

- Chuẩn hóa và chuyển đổi: Đảm bảo dữ liệu được chuẩn hóa và chuyển đổi đúng cách trước khi đưa vào mô hình.

- Xử xý Imbalance: Nếu có sự mất cân bằng giữa các lớp (bệnh và không bệnh), cần xem xét các chiến lược để xử lý vấn đề này.

#### 2.1.3. *Đánh giá hiệu suất*

- Metrics đánh giá: Xác định các chỉ số đánh giá hiệu suất như độ chính xác (accuracy), độ nhạy (sensitivity), độ đặc specificity), và F1-score.

- ROC Curve và AUC: Đối với bài toán phân loại, nếu có thể, xem xét đồ thị ROC và diện tích dưới đường cong (AUC).

#### 2.1.4. *Kiểm soát Overfitting và Underfitting*

- Kiểm soát Overfitting: Sử dụng các kỹ thuật như dropout, regularization để giảm overfitting.

- Kiểm soát Underfitting: Đảm bảo mô hình có độ phức tạp đủ để học từ dữ liệu.

#### 2.1.5. *Phân tích đặc trưng*

- Sử dụng Shapley Values hoặc LRP: Để hiểu cách mô hình quan sát và đánh giá đặc trưng của dữ liệu.

- Grid search hoặc Random search: Tinh chỉnh tham số để tìm kiếm bộ tham số tốt nhất cho mô hình.

### 2.1.7. Phiên bản và triển khai

- Lưu phiên bản mô hình: Lưu lại phiên bản mô hình đã được huấn luyện để có thể tái sử dụng và so sánh.
- Triển khai mô hình: Nếu có kế hoạch triển khai mô hình, đảm bảo kiểm tra cơ sở hạ tầng và xác minh tính ổn định của mô hình trên môi trường triển khai.

## 2.2. Mô hình ResNet-50

### 2.2.1. Ý tưởng

Ý tưởng chính đằng sau **ResNet-50** là giảm vấn đề vanishing gradient khi lan truyền ngược qua các lớp sâu trong mạng nơ-ron. Trong quá trình huấn luyện, khi mô hình trở nên sâu, gradient có thể giảm đáng kể khi lan truyền ngược, làm cho quá trình học trở nên khó khăn và thậm chí dẫn đến việc mô hình không học được gì cả.

Cơ chế chính để giải quyết vấn đề này là sử dụng các “skip connections” hoặc “shortcuts”. Thay vì trực tiếp ánh xạ đầu vào thành đầu ra, ResNet sử dụng các “residual blocks” để ánh xạ đầu vào thành sự khác biệt giữa đầu ra và đầu vào, sau đó thêm sự khác biệt này vào ban đầu.

### 2.2.2. Kiến trúc mô hình ResNet-50

- Convolutional Layer và Pooling Layer: Mô hình bắt đầu bằng một lớp convolutional và lớp pooling để trích xuất đặc trưng từ ảnh đầu vào.
- Residual Blocks: Kiến trúc chính của ResNet-50 bao gồm nhiều residual blocks. Mỗi khối bao gồm nhiều lớp convolutional và các kết nối "skip connections".
- Global Average Pooling (GAP): Thay vì sử dụng fully connected layers ở cuối mô hình, ResNet-50 sử dụng lớp GAP để giảm số lượng tham số và tránh overfitting.
- Fully Connected Layer và Softmax: Mô hình kết thúc bằng một lớp fully connected và hàm kích thích softmax để xuất ra xác suất của các lớp đối tượng.

### 2.2.3. Tính toán Loss, IoU và DSC của mô hình ResNet-50

#### Tính Toán Loss (Hàm mất mát)

```

import torch
import torch.nn as nn
import torch.optim as optim

# Khởi tạo mô hình ResNet-50 và hàm mất mát
resnet50 = torchvision.models.resnet50(pretrained=True)
criterion = nn.BCELoss()

# Giả sử bạn có một đầu ra dự đoán của mô hình (output) và nhãn thực tế
output = resnet50(input_data)
target = ground_truth_labels

# Tính toán hàm mất mát
loss = criterion(output, target)

```

Hình 2.1. Khai báo hàm tính toán Loss

Hàm mất mát thường được sử dụng tùy thuộc vào loại nhiệm vụ mà bạn đang giải quyết. Đối với bài toán phân loại ảnh, thường sử dụng hàm mất mát như Cross Entropy Loss. Đối với bài toán học máy khác như segmentation, bạn có thể sử dụng hàm mất mát như Dice Loss hoặc Binary Cross Entropy Loss.

### Tính toán IoU (Intersection over Union):

IoU là một độ đo đánh giá độ chính xác của mô hình trong bài toán segmentation. IoU được tính bằng cách chia diện tích phần chồng lấp giữa dự đoán và nhãn thực tế cho tổng diện tích của cả hai.

```
def calculate_iou(predicted, target):
    intersection = (predicted & target).sum().float()
    union = (predicted | target).sum().float()

    iou = (intersection + 1e-6) / (union + 1e-6)
    return iou.item()
```

Hình 2.2. Hàm tính toán IoU

Trong đoạn mã trên, “predicted” và “target” là các tensor nhị phân (0 và 1) biểu diễn dự đoán của mô hình và nhãn thực tế.

### Tính toán DSC (Dice Similarity Coefficient):

Dice Similarity Coefficient cũng là một độ đo thường được sử dụng trong bài toán segmentation. Nó được tính bằng cách lấy gấp đôi phần chồng lấp giữa dự đoán và nhãn thực tế chia cho tổng diện tích của cả hai.

```
def calculate_dice_coefficient(predicted, target):
    intersection = (predicted & target).sum().float()
    dice_coefficient = (2.0 * intersection + 1e-6) / ((predicted.sum() +
    target.sum() + 1e-6))
    return dice_coefficient.item()
```

Hình 2.3. Hàm tính toán DSC

Trong cả hai đoạn mã trên, “predicted” và “target” đều là các tensor nhị phân, và “&” và “|” là phép toán AND và OR nhị phân. Cả hai độ đo IoU và DSC đều dao động từ 0 đến 1, với giá trị cao hơn đại diện cho hiệu suất tốt hơn của mô hình.

## 2.3. Chuẩn bị mô hình

Từ các gợi ý trên chúng ta sẽ tạo một chương trình học máy sử dụng mô hình ResNet-50 sau đó lần lượt cho các ảnh từ các class vào để training cho chương trình (4 class).

### 2.3.1. Chuẩn bị dữ liệu

- Sử dụng thư viện torchvision để tải một phiên bản được huấn luyện sẵn của mô hình ResNet-50.

- Chuẩn bị một đối tượng biến đổi (`transform`) để xử lý dữ liệu đầu vào. Trong trường hợp này, hình ảnh được chuyển đổi thành tensor và được resize về kích thước mong muốn (224x224).

### 2.3.2. Mô hình học máy

- Sử dụng mô hình ResNet-50, một kiến trúc mạng nơ-ron sâu (deep neural network) có khả năng học từ các hình ảnh.

- Thay đổi lớp cuối cùng của mô hình để phù hợp với số lượng lớp của bài toán cụ thể (ở đây là 4 lớp).

- Tải trọng số đã được fine-tune vào mô hình.

### 2.3.3. Hàm dự đoán

- Hàm predict nhận một hình ảnh đầu vào.

- Sử dụng mô hình để dự đoán lớp của hình ảnh.

- Trả về kết quả dự đoán bao gồm tên lớp và độ tin cậy.

### 2.3.4. Giao diện người dùng với gradio

- Sử dụng thư viện Gradio để tạo một giao diện người dùng đơn giản.

- Đặt hàm predict làm hàm dự đoán cho giao diện.

- Giao diện cho phép người dùng tải lên hình ảnh và hiển thị kết quả dự đoán.

### 2.3.5. Quản lý trạng thái mô hình

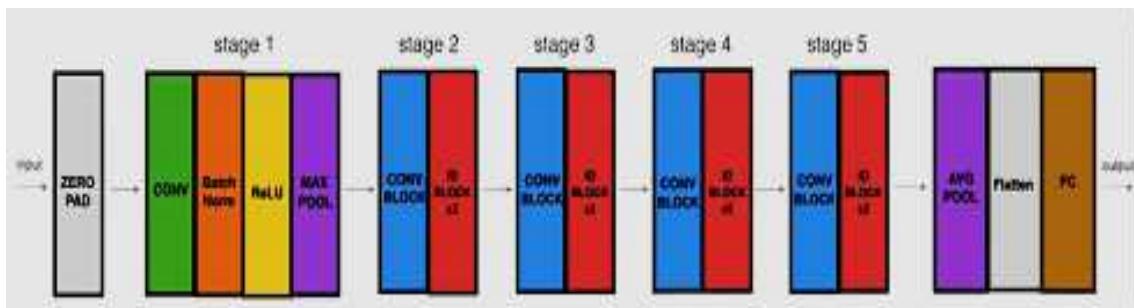
- Mô hình được đặt ở chế độ đánh giá (`model.eval()`) sau khi tải trọng số.

- Tất cả các trọng số của mô hình đã được tải từ tệp `fine_tuned_resnet50.pth`.

## 2.6. Cách thiết lập và vận hành ResNet-50

### 2.6.1. Thiết lập ResNet-50

#### a) Thiết lập mô hình



Hình 2.4. Xây dựng mô hình tùy chỉnh ResNet-50

- Sử dụng mã nguồn mở hoặc pre-trained weights: Tải các mã nguồn mở của ResNet-50 từ các thư viện như TensorFlow hoặc PyTorch, hoặc sử dụng trọng số đã được huấn luyện trước đó.
- Xây dựng mô hình tùy chỉnh: Nếu cần, có thể xây dựng một phiên bản tùy chỉnh của ResNet-50 bằng cách điều chỉnh số lượng lớp hoặc thay đổi cấu trúc.

#### b) Chuẩn bị dữ liệu

Đảm bảo dữ liệu huấn luyện và kiểm thử đã được chuẩn bị đúng cách và tuân thủ định dạng đầu vào của ResNet-50.

#### c) Quá trình huấn luyện:

- Chọn các siêu tham số (hyperparameters) phù hợp như learning rate, batch size, số lượng epoch, v.v...
- Sử dụng tập dữ liệu huấn luyện để đào tạo mô hình. Có thể sử dụng các kỹ thuật như data augmentation để tăng cường dữ liệu.

#### d) Đánh giá và tinh chỉnh

- Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm thử để đảm bảo tính tổng quát.
- Nếu cần tiến trình tinh chỉnh mô hình bằng cách thay đổi các siêu tham số hoặc thay đổi kiến trúc.

#### e) Triển khai

Sau khi mô hình đã được huấn luyện và đánh giá, triển khai nó trong môi trường thực tế để thực hiện dự đoán trên dữ liệu mới.

### 2.6.2. Vận hành ResNet-50

- Load Pretrained ResNet-50 Model:

```
model = models.resnet50(pretrained=True)
```

Hình 2.5. Câu lệnh Load Pretrained

Ở đây, “models.resnet50(pretrained=True)” tải một phiên bản của mô hình ResNet-50 từ thư viện torchvision với trạng thái pretrained (đã được huấn luyện trước). Điều này có nghĩa là mô hình đã được huấn luyện trên tập dữ liệu ImageNet và có thể chứa các trọng số đã học được từ hàng triệu hình ảnh.

- Thay đổi Fully Connected Layer (fc):

```
num_classes = len(classes)
model.fc = nn.Linear(model.fc.in_features, num_classes)
```

Hình 2.6. Câu lệnh thay đổi lớp Fully

Trong mạng ResNet-50, lớp fully connected layer cuối cùng (fc) thường có số lượng đầu ra bằng số lớp trong tập dữ liệu ImageNet (đối với phiên bản pretrained). Đoạn mã trên thay đổi lớp fc này để phù hợp với số lượng lớp trong tập dữ liệu mới được sử dụng. “model.fc.in\_features” là số lượng đầu ra của lớp fc trước đó, và ‘nn.Linear’ tạo một lớp fc mới với số lượng đầu ra là ‘num\_classes’.

- Load Fine-Tuned Weights:

```
model.load_state_dict(torch.load("fine_tuned_resnet50.pth",
map_location=torch.device('cpu')))
```

Đoạn mã này tải trọng số đã được fine-tuned từ tệp "fine\_tuned\_resnet50.pth" và áp dụng chúng lên mô hình. Fine-tuning thường được thực hiện bằng cách tiếp tục quá trình huấn luyện trên một tập dữ liệu nhỏ hoặc tập dữ liệu mới, để mô hình có thể được điều chỉnh tốt hơn cho nhiệm vụ cụ thể hoặc tập dữ liệu đó.

Như vậy, mô hình ResNet-50 được tải và điều chỉnh để phù hợp với bài toán cụ thể mà bạn đang làm việc thông qua việc thay đổi số lượng lớp đầu ra và việc tải trọng số đã được fine-tuned.

## CHƯƠNG 3. XÂY DỰNG ỦNG DỤNG

### 3.1. Công cụ

#### 3.1.1. Anaconda

**Anaconda** là một nền tảng phân phối miễn phí của ngôn ngữ lập trình Python và R cho tính toán khoa học (khoa học dữ liệu, machine learning, xử lý dữ liệu lớn, phân tích dự đoán, v.v.), nhằm mục đích đơn giản hóa việc quản lý và triển khai gói.



Hình 3.1. Biểu tượng Anaconda

#### ❖ Quản lý môi trường (Environment Management)

- Anaconda cho phép tạo và quản lý các môi trường ảo, nơi bạn có thể cài đặt các phiên bản cụ thể của Python và các gói phần mềm khác mà bạn cần cho dự án cụ thể.

- Điều này giúp tránh xung đột giữa các phiên bản của các thư viện và đảm bảo tính nhất quán của môi trường làm việc.

#### ❖ Quản lý gói và thư viện (Package Management)

- Anaconda sử dụng conda, một hệ thống quản lý gói, để cài đặt, cập nhật và xóa bỏ các gói và thư viện.

- Người sử dụng có thể dễ dàng cài đặt các gói như NumPy, Pandas, TensorFlow, PyTorch, và nhiều thư viện học máy khác chỉ bằng cách sử dụng câu lệnh conda.

#### ❖ Hỗ trợ đa nền tảng (Cross-Platform Support)

Anaconda hỗ trợ đa nền tảng, điều này có nghĩa là bạn có thể chuyển đổi giữa các hệ điều hành như Windows, macOS và Linux mà không gặp vấn đề tương thích với gói và thư viện.

#### ❖ Jupyter notebooks và JupyterLab

- Anaconda đi kèm với Jupyter Notebooks, một môi trường tương tác mạnh mẽ cho phân tích dữ liệu và thực hiện các thực nghiệm máy học một cách linh hoạt.

- JupyterLab là một phiên bản cải tiến của Jupyter Notebooks, mang lại một giao diện người dùng hiện đại và nhiều tính năng hữu ích khác.

**❖ Quản lý dự án (Project Management)**

- Anaconda giúp tổ chức dự án và mã nguồn của bạn thông qua quy trình làm việc với môi trường và gói.

- Điều này làm giảm sự phức tạp trong việc quản lý dependencies và giúp giữ cho dự án của bạn có thể tái tạo được.

**❖ Tích hợp công cụ (Intergated Tools)**

Anaconda có tích hợp nhiều công cụ hữu ích khác như Spyder (một IDE cho Python), VSCode, và nhiều công cụ hỗ trợ phân tích dữ liệu và học máy.

**❖ Phân phối và triển khai dự án (Project Distribution and Deployment)**

- Anaconda giúp quản lý quá trình triển khai mô hình học máy, bao gồm việc đóng gói dự án và phân phối môi trường làm việc.

- Anaconda đóng vai trò quan trọng trong học máy bằng cách cung cấp một môi trường làm việc hiệu quả, linh hoạt, và dễ quản lý cho các nhà khoa học dữ liệu và nhà phân tích dữ liệu. Nó giúp giải quyết nhiều thách thức quản lý môi trường và gói trong quá trình phát triển và triển khai các dự án học máy.

### **3.1.2. Python**



Hình 3.2. Biểu tượng Python

Python là một ngôn ngữ lập trình thông dịch, đa mục đích và ngôn ngữ lập trình hợp đồng (high-level programming language). Được tạo ra bởi Guido van Rossum và lần đầu tiên xuất hiện vào năm 1991, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

- Đơn giản và dễ đọc: Python được thiết kế để có cú pháp đơn giản và dễ đọc, giúp người lập trình viết mã một cách nhanh chóng và hiệu quả.

- Đa mục đích: Python là một ngôn ngữ đa mục đích, có thể sử dụng cho nhiều loại ứng dụng khác nhau, từ phát triển web đến khoa học dữ liệu, trí tuệ nhân tạo, máy học, và nhiều lĩnh vực khác.

- Cộng đồng lớn: Python có một cộng đồng lập trình viên lớn và tích cực. Cộng đồng này cung cấp nhiều tài nguyên, thư viện, và framework hữu ích.

- Thư viện phong phú: Python có nhiều thư viện chuẩn và bên thứ ba mạnh mẽ, giúp người lập trình giảm thiểu công sức khi phát triển ứng dụng.
- Tương tác và giáo dục: Python thường được sử dụng trong việc dạy lập trình vì tính đơn giản và dễ hiểu của nó. Nó cũng là một ngôn ngữ tương tác tốt, cho phép người dùng thử nghiệm và kiểm tra mã một cách dễ dàng.
- Hỗ trợ đối tượng Python: Hỗ trợ lập trình hướng đối tượng, cho phép người lập trình tổ chức mã của họ theo cách hợp lý và tái sử dụng mã một cách linh hoạt.
- Khả năng tích hợp tốt: Python có thể tích hợp dễ dàng với các ngôn ngữ khác như C và C++, và có thể sử dụng trong các hệ thống lớn và phức tạp.
- Di động: Python có thể chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS và các phiên bản khác nhau của Linux.
- Các phiên bản hiện tại của Python bao gồm Python 2 và Python 3, nhưng Python 2 đã chính thức ngừng hỗ trợ vào ngày 1 tháng 1 năm 2020 và Python 3 được khuyến khích sử dụng. Python tiếp tục phát triển với các phiên bản mới được phát hành đều đặn để cải thiện hiệu suất và tính năng.

### 3.1.3. PyCharm



Hình 3.3. Biểu tượng PyCharm

PyCharm là một IDE (Integrated Development Environment) dành cho ngôn ngữ lập trình Python, được phát triển bởi JetBrains. Đây là một công cụ mạnh mẽ và linh hoạt, cung cấp nhiều tính năng hỗ trợ cho quá trình phát triển ứng dụng Python. Dưới đây là một tổng quan về PyCharm:

Giao diện người dùng thân thiện: PyCharm cung cấp một giao diện người dùng sáng tạo và dễ sử dụng. Các tính năng và công cụ được tổ chức một cách có logic, giúp người lập trình tập trung vào việc viết mã mà không bị làm phiền bởi quá trình quản lý dự án.

Quản lý dự án và Phiên bản kiểm soát: PyCharm hỗ trợ quản lý dự án một cách hiệu quả, giúp bạn tổ chức mã nguồn, tài nguyên và các thành phần khác của dự án một cách dễ dàng. Nó cũng tích hợp với các hệ thống kiểm soát phiên bản như Git.

Kiểm thử và gỡ lỗi: PyCharm cung cấp các công cụ kiểm thử tích hợp và chế độ gỡ lỗi (debugging) mạnh mẽ. Bạn có thể kiểm thử từng đoạn mã một và theo dõi biến, giúp nhanh chóng xác định và sửa lỗi.

Intelligent Code Assistance: PyCharm sử dụng hệ thống thông minh để cung cấp gợi ý mã, kiểm tra lỗi ngay trong quá trình nhập mã, và tự động hoàn thiện mã. Điều này giúp tăng tốc độ phát triển và giảm thiểu lỗi.

Refactoring: Có các công cụ tái cấu trúc mã nguồn (refactoring) giúp bạn cải thiện cấu trúc mã một cách dễ dàng mà không làm thay đổi hành vi của ứng dụng.

Hỗ trợ Ngôn ngữ và Frameworks: PyCharm hỗ trợ nhiều ngôn ngữ và frameworks phổ biến trong cộng đồng Python, bao gồm Django, Flask, Pyramid và nhiều hơn nữa.

Plugins và tích hợp: PyCharm có khả năng mở rộng thông qua việc cài đặt plugins, giúp tối ưu hóa trải nghiệm làm việc của người lập trình theo nhu cầu cụ thể.

Hỗ trợ Cộng đồng: PyCharm có một cộng đồng lớn và tích cực, với các diễn đàn, tài liệu và nguồn thông tin hữu ích từ người dùng khác.

PyCharm có cả phiên bản miễn phí (Community Edition) và phiên bản trả phí (Professional Edition). Phiên bản Professional cung cấp nhiều tính năng cao cấp hơn và thích hợp cho các dự án lớn và chuyên nghiệp.

### 3.1.4. Gradio



Hình 3.4. Biểu tượng Gradio

Gradio là một thư viện Python dùng để tạo giao diện người dùng tùy chỉnh cho các mô hình học máy. Quá trình trải qua các bước thu thập và chuẩn bị dữ liệu đầu vào và tích hợp một mô hình phân loại đã xây dựng sẵn vào giao diện Gradio. Dự án cung cấp một thực tế về cách Gradio có thể được sử dụng để tạo ứng dụng học máy thân thiện với người dùng.

### ❖ Tổng quan về dự đoán Churn khách hàng

Dự đoán Churn khách hàng liên quan đến việc sử dụng các thuật toán học máy để dự đoán những khách hàng có khả năng chuyển đổi. Bằng cách phân tích hành vi và dữ liệu dân số của khách hàng, có thể xác định các mẫu và xu hướng để dự đoán sự chuyển đổi. Các mô hình dự đoán có thể được đào tạo trên dữ liệu lịch sử của khách hàng để dự đoán khả năng chuyển đổi của khách hàng mới. Việc xác định khách hàng có rủi ro chuyển đổi cao giúp các công ty thực hiện biện pháp tích cực để giữ họ, như cung cấp chương trình trung thành, giảm giá hoặc gợi ý cá nhân hóa.

### ❖ Hiểu về Gradio

Gradio là một giao diện web thân thiện người dùng giúp đơn giản hóa quá trình triển khai các mô hình học máy. Nó cho phép các nhà khoa học dữ liệu tạo giao diện người dùng trực quan và tương tác cho mô hình của họ mà không cần kỹ năng phát triển web chuyên sâu.

### ❖ Tổng quan về Gradio như một công cụ xây dựng ứng dụng học máy

Gradio là một framework web mã nguồn mở giúp đơn giản hóa quá trình xây dựng ứng dụng học máy (ML). Nó cho phép nhà phát triển nhanh chóng tạo giao diện cho các mô hình ML mà không cần viết bất kỳ mã frontend nào. Gradio hỗ trợ các mô hình dựa trên văn bản, hình ảnh và âm thanh, và các framework ML phổ biến như TensorFlow, PyTorch và Scikit-learn. Giao diện kéo và thả của Gradio giúp tạo ra các trường đầu vào và đầu ra tùy chỉnh, và hỗ trợ tích hợp và chia sẻ mô hình ML giúp đơn giản hóa quá trình triển khai.

### ❖ Các khả năng và tính năng của Gradio

- Giao diện đơn giản: Gradio cung cấp giao diện kéo và thả đơn giản cho việc xây dựng giao diện người dùng thân thiện cho các mô hình ML.
- Hỗ trợ cho các framework ml phổ biến: Gradio hỗ trợ TensorFlow, PyTorch và Scikit-learn.
- Triển khai và lưu trữ: Hỗ trợ tích hợp sẵn cho việc lưu trữ và chia sẻ các mô hình ML.
- Các trường đầu vào và đầu ra tương tác: Gradio cho phép tạo ra các trường đầu vào và đầu ra tương tác cho các mô hình ML, bao gồm thanh trượt, menu thả xuống và hộp văn bản.
- Giao diện có thể tùy chỉnh: Giao diện của Gradio có thể tùy chỉnh cao, cho phép nhà phát triển thay đổi bố cục, chủ đề màu sắc và kiểu dáng của ứng dụng theo ý muốn cá nhân hoặc nhãn hiệu của họ.

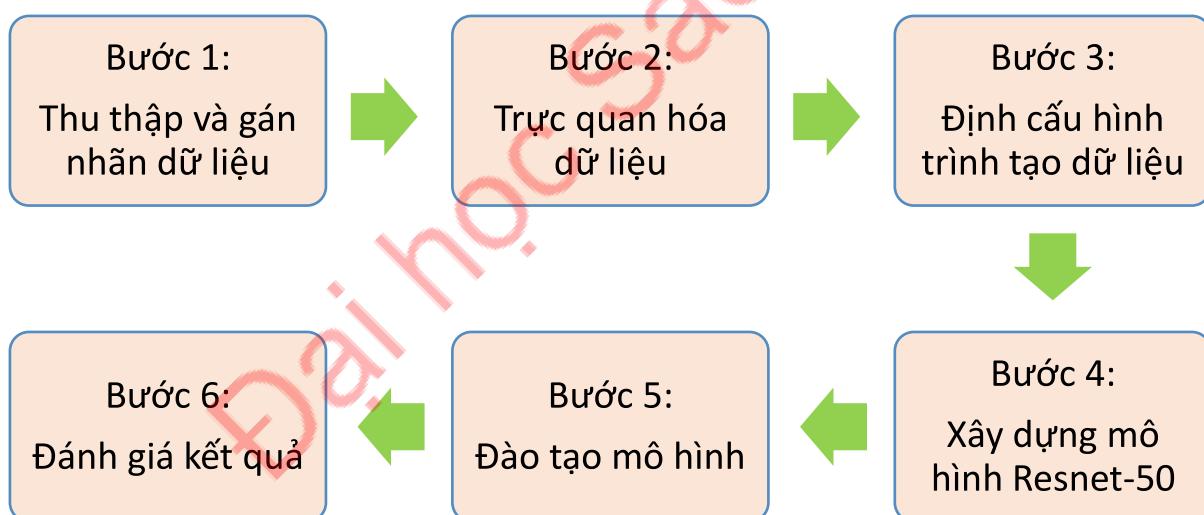
### ❖ Lợi ích của việc sử dụng Gradio

- Triển khai nhanh chóng: Gradio có hỗ trợ tích hợp cho việc lưu trữ và chia sẻ các mô hình ML.
- Hỗ trợ đa mô hình: Gradio cho phép tạo giao diện cho nhiều mô hình ML trong một ứng dụng, giúp so sánh và đối chiếu các mô hình khác nhau.
- Tiết kiệm thời gian: Gradio đơn giản hóa quá trình xây dựng ứng dụng ML bằng cách loại bỏ nhu cầu viết bất kỳ mã frontend nào. Điều này có thể giúp nhà phát triển tiết kiệm thời gian và công sức, giúp họ tập trung vào việc xây dựng và tinh chỉnh mô hình ML của mình.

### 3.2. Quy trình phân tích bệnh Alzheimer

Quy trình phân tích bệnh Alzheimer gồm 6 bước:

- Bước 1: Thu thập và gán nhãn dữ liệu
- Bước 2: Trực quan hóa dữ liệu
- Bước 3: Định cấu hình trình tạo dữ liệu
- Bước 4: Xây dựng mô hình Resnet-50
- Bước 5: Đào tạo mô hình
- Bước 6: Đánh giá kết quả



Hình 3.5. Sơ đồ quy trình phân tích bệnh Alzheimer

### 3.3. Cơ sở dữ liệu

Bộ dữ liệu chứa hình ảnh MRI não cùng với mặt nạ phân đoạn bất thường flair thủ công. Các hình ảnh đã được lấy từ kho lưu trữ hình ảnh ung thư (TCIA). Chúng tương ứng với 110 bệnh nhân được bao gồm trong bộ sưu tập Glioma cấp thấp hơn của ung thư (TCGA) với ít nhất là trình tự phục hồi đảo ngược (Flair) và dữ liệu bộ gen có sẵn.

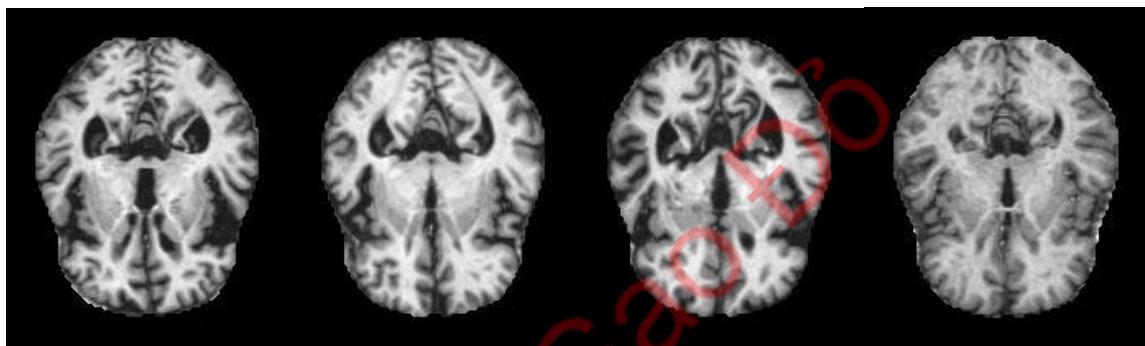
Bộ dữ liệu này chứa hình ảnh MR não cùng với mặt nạ phân đoạn bất thường FLAIR thủ công.

Các hình ảnh được lấy từ cơ quan lưu trữ hình ảnh ung thư (TCIA). Chúng tương ứng với 110 bệnh nhân được đưa vào bộ sưu tập u thần kinh đệm cấp thấp hơn của The Cancer Genome Atlas (TCGA) với ít nhất trình tự phục hồi đảo ngược giảm độc tố chất lỏng (FLAIR) và dữ liệu cụm gen có sẵn.

Các cụm gen khối u và dữ liệu bệnh nhân được cung cấp trong tệp “data.csv”. Tất cả hình ảnh được cung cấp ở định dạng “.tif” với 3 kênh trên mỗi hình ảnh.

Đối với 101 trường hợp, 3 chuỗi có sẵn, tức là trước tương phản, FLAIR, sau tương phản (theo thứ tự kênh này). Đối với 9 trường hợp, trình tự sau tương phản bị thiếu và đối với 6 trường hợp, thiếu trình tự tương phản trước.

Các trình tự bị thiếu được thay thế bằng trình tự FLAIR để làm cho tất cả hình ảnh thành 3 kênh. Mặt nạ là hình ảnh nhị phân, 1 kênh. Họ phân đoạn bất thường FLAIR có trong trình tự FLAIR (có sẵn cho mọi trường hợp).



Hình 3.6. Bộ dữ liệu hình ảnh MRI não người bệnh Alzheimer

### 3.4. Xây dựng công dụng chức năng của code

Thiết kế và khai báo các thư viện và các chức năng của code

#### ❖ Import các thư viện

- Gradio: Thư viện giúp tạo giao diện người dùng cho các mô hình máy học.
- PIL.Image: Được sử dụng để xử lý hình ảnh.
- Torch và torch.nn: Thư viện PyTorch cho việc xây dựng và huấn luyện mô hình học máy.

- Torchvision.transforms và torchvision.models: Cung cấp các chức năng xử lý ảnh và mô hình học máy đa nhiệm.

#### ❖ Chuẩn bị dữ liệu

- Transform: Một đối tượng Compose của PyTorch, định nghĩa một chuỗi các biến đổi hình ảnh được áp dụng cho hình ảnh đầu vào. Ở đây, nó thay đổi kích thước hình ảnh về (224, 224) và chuyển đổi nó thành tensor.

- Classes: Một danh sách các lớp mà mô hình có thể dự đoán.

#### ❖ Load mô hình

- Models.resnet50(pretrained=True): Tải một mô hình ResNet-50 được huấn luyện trước từ thư viện torchvision.

- Nn.Linear: Thay đổi lớp fully connected cuối cùng để phù hợp với số lượng lớp của bài toán (ở đây là num\_classes).

- Model.load\_state\_dict(...): Tải trọng số đã được fine-tune vào mô hình.
- Model.eval(): Đặt mô hình vào chế độ đánh giá (không huấn luyện).

#### ❖ Hàm dự đoán (Predict)

- Nhận một hình ảnh đầu vào (img).
- Chuyển đổi hình ảnh thành tensor và thêm chiều batch.
- Sử dụng mô hình để dự đoán lớp của hình ảnh.
- Trả về kết quả dự đoán với tên lớp và độ tin cậy.

#### ❖ Tạo Gradio Interface

- Gr.Interface(fn=predict, inputs="image", outputs="text"): Tạo một giao diện Gradio với hàm predict làm hàm dự đoán, đầu vào là hình ảnh và đầu ra là văn bản.
- Iface.launch(): Khởi động giao diện Gradio để chạy mô hình dự đoán từ hình ảnh đầu vào.

### 3.5. Xây dựng chương trình học máy

Chương trình sử dụng mô hình học máy Resnet-50 để phân loại hình ảnh. Đây là một số điểm chính về mã nguồn:

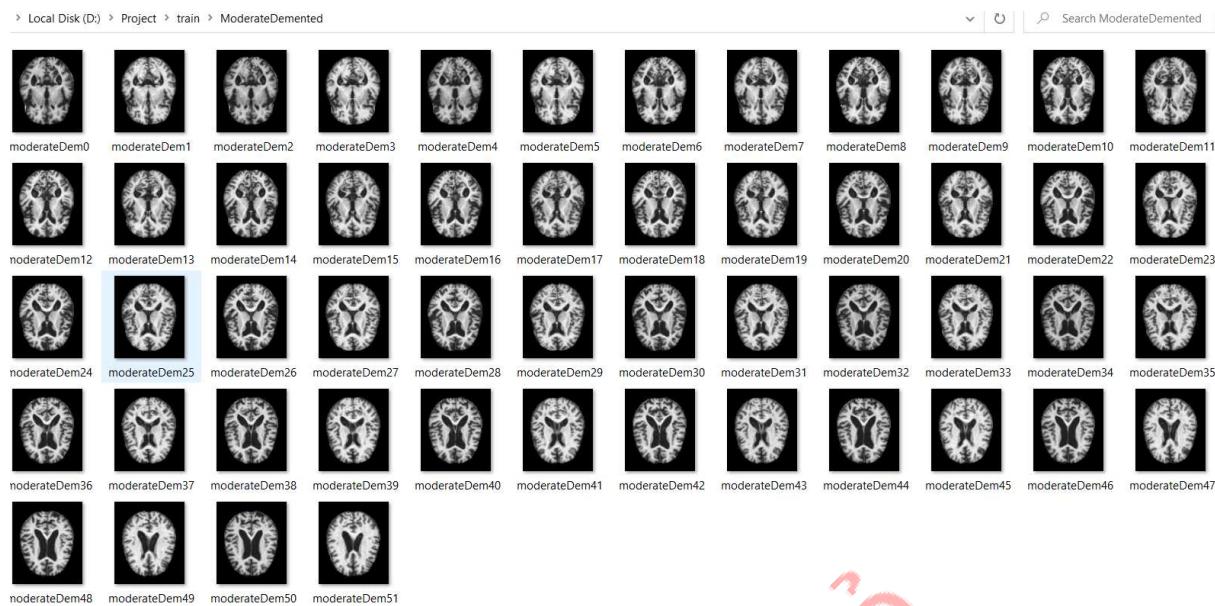
#### 3.5.1. Phân dữ liệu

##### ❖ Class 1: MildDemented



Hình 3.7. Kho dữ liệu class MildDemented

### ❖ Class 2: ModerateDemented



Hình 3.8. Kho dữ liệu class ModerateDemented

### ❖ Class 3 : NonDemented



Hình 3.9. Kho dữ liệu class NonDemented

### ❖ Class 4: VeryMildDemented



Hình 3.10. Kho dữ liệu class VeryMildDemented



Hình 3.11. Một số ảnh thử sai cho chương trình

#### 3.5.2. Mô hình

- Sử dụng mô hình “ResNet-50” từ thư viện “torchvision.models”.
- Thay đổi lớp fully connected cuối cùng để phù hợp với số lượng lớp của bài toán cụ thể.

```
model = models.resnet50(pretrained=True)
num_classes = len(classes)
model.fc = nn.Linear(model.fc.in_features, num_classes)
```

#### 3.5.3. Trọng số

Load trọng số đã được fine-tune từ tệp ‘fine\_tuned\_resnet50.pth’ vào mô hình.

```
model.load_state_dict(torch.load("fine_tuned_resnet50.pth",
map_location=torch.device('cpu')))
model.eval()
```

#### 3.5.4. Dự đoán

Hàm “predict” thực hiện dự đoán cho một hình ảnh đầu vào bằng cách chuyển đổi nó thành tensor và sử dụng mô hình.

```
def predict(img):
```

```

# Preprocess the input image
img_tensor = transform(Image.fromarray(img))

# Add a batch dimension
img_tensor = img_tensor.unsqueeze(0)

# Make prediction
with torch.no_grad():
    outputs = model(img_tensor)
    _, predicted = torch.max(outputs, 1)

# Get the predicted class and its confidence score
predicted_class = classes[predicted.item()]
confidence = torch.nn.functional.softmax(outputs,
dim=1)[0][predicted].item()

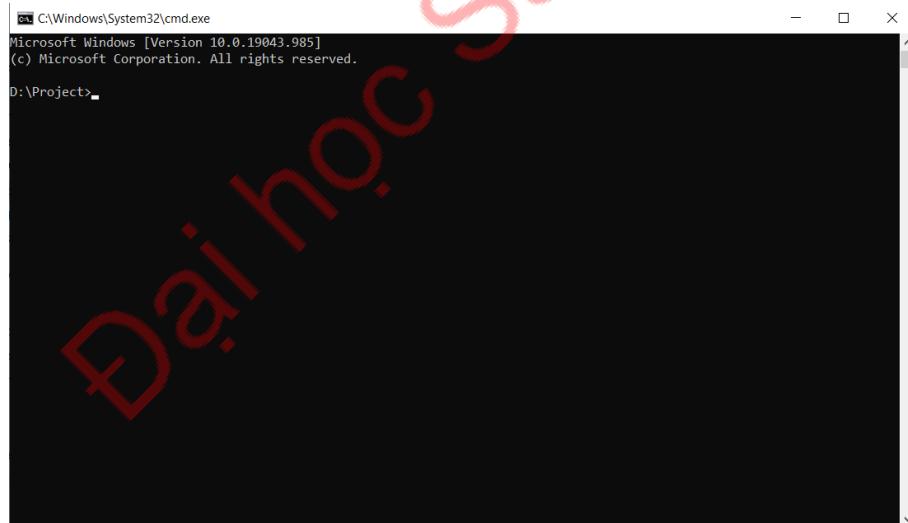
return f"Predicted Class: {predicted_class}, Confidence: {confidence:.2f}"

```

### 3.6. Thử nghiệm chương trình và kết quả thử nghiệm

❖ **Khởi chạy chương trình**

- Bước 1: Mở cửa sổ CMD



Hình 3.12. Màn hình cửa sổ CMD

- Bước 2: Nhập câu lệnh “conda activate torch-gpu” để có thể truy cập tới (torch-gpu).

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.985]
(c) Microsoft Corporation. All rights reserved.

D:\Project>conda activate torch-gpu
(torch-gpu) D:\Project>
```

Hình 3.13. Nhập lệnh truy cập tới torch-gpu

- Bước 3: Tiếp theo nhập câu lệnh “python test\_demen1.py” sẽ xuất hiện một dòng host.

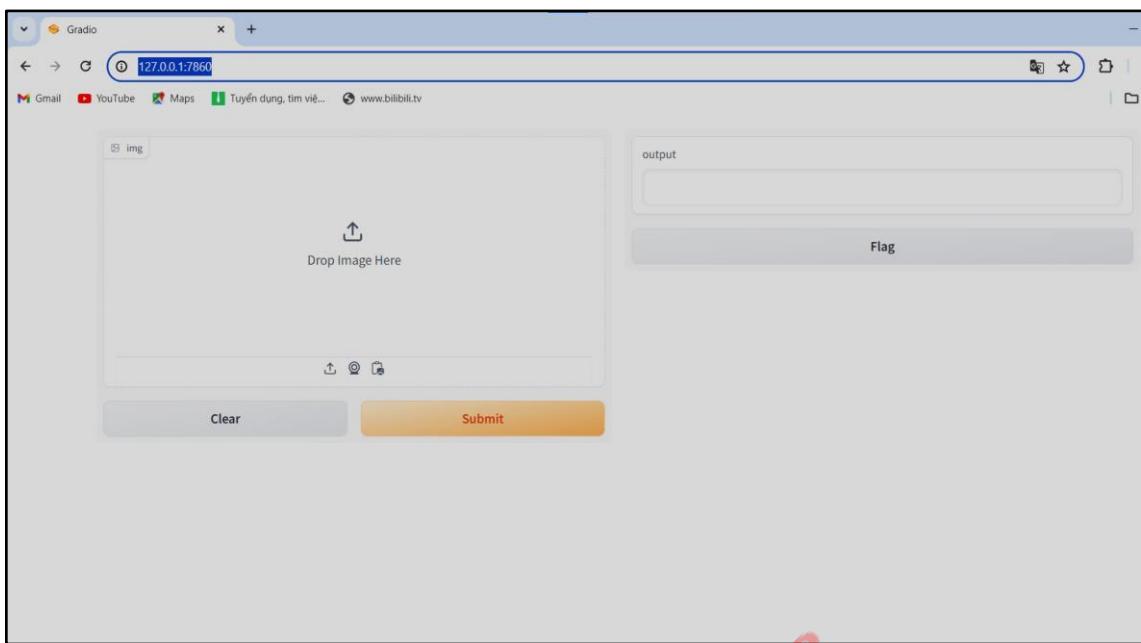
```
Select C:\Windows\System32\cmd.exe - python test_demen1.py
Microsoft Windows [Version 10.0.19043.985]
(c) Microsoft Corporation. All rights reserved.

D:\Project>python test_demen1.py
C:\Users\Anh\anaconda3\envs\torch-gpu\lib\site-packages\torchvision\models\_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.
  warnings.warn(
C:\Users\Anh\anaconda3\envs\torch-gpu\lib\site-packages\torchvision\models\_utils.py:223: UserWarning: Arguments other than a weight enum or 'None' for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing `weights=ResNet50_Weights.IMAGENET1K_V1` . You can also use `weights=ResNet50_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
Running on local URL: http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.
```

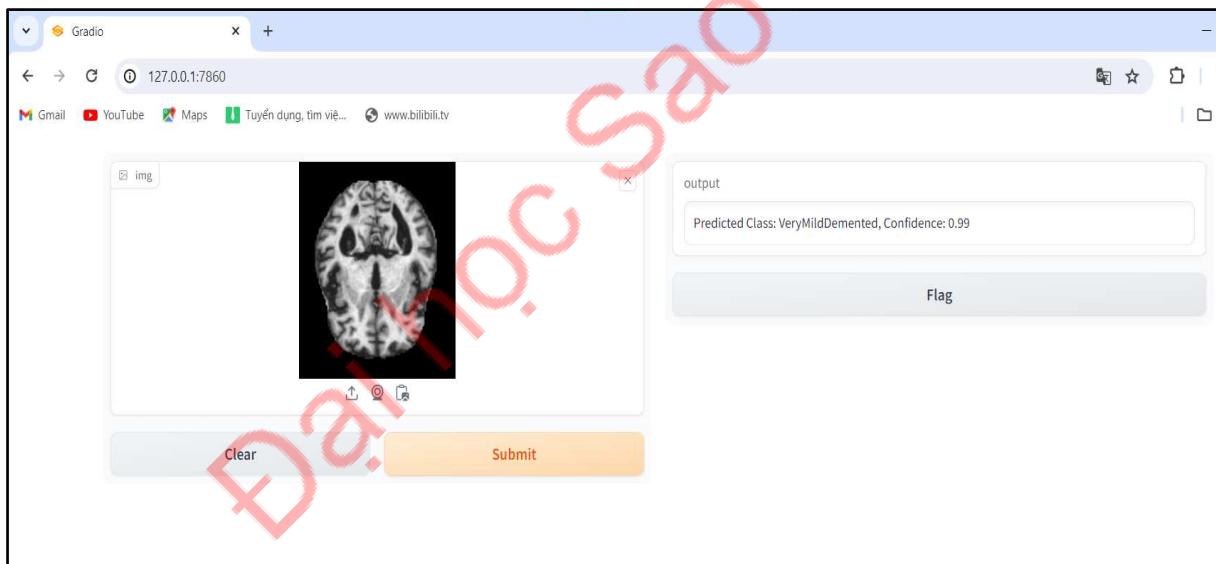
Hình 3.14. Xuat hiện đoạn host để chạy giao diện nhận dạng

- Bước 4: Sao chép dòng host và dán lên trình duyệt mạng sẽ xuất hiện giao diện thao tác người dùng.



Hình 3.15. Giao diện người dùng

- Bước 5: Chọn những ảnh cần kiểm tra đưa lên giao diện sẽ trả về một trong 4 class đã tạo trước đó với một độ tin cậy nhất định.

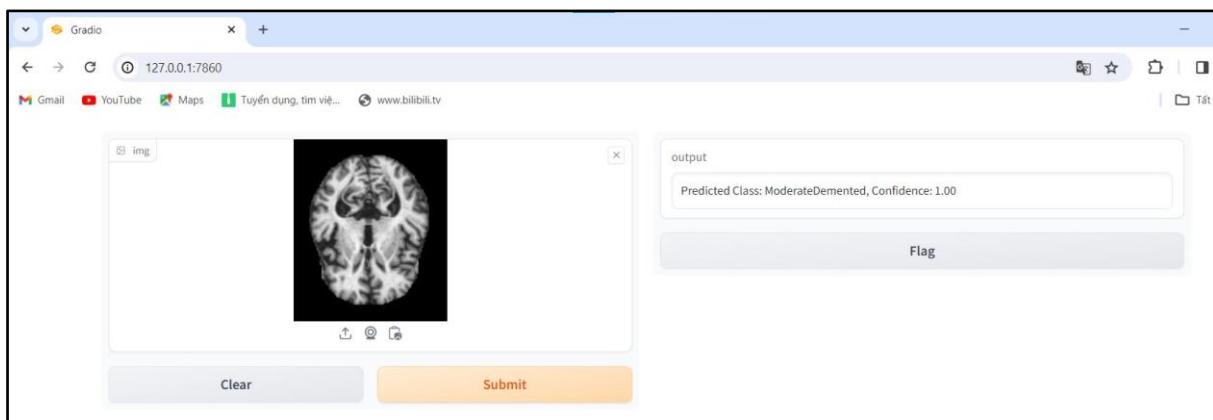


Hình 3.16. Kết quả sau khi nhận dạng

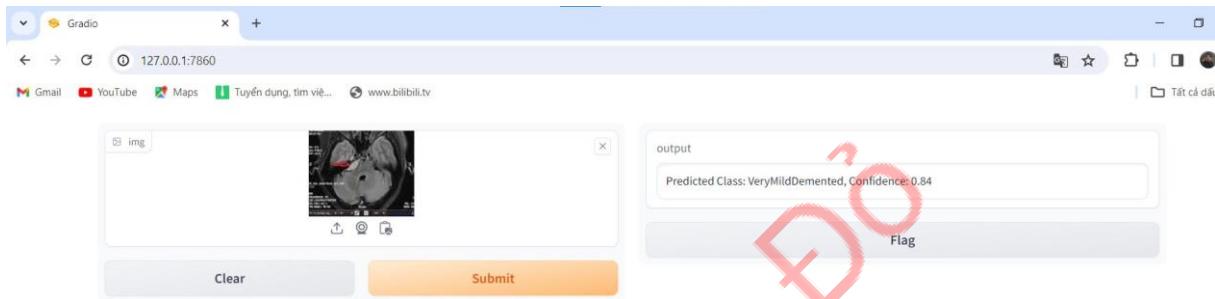
### Kết quả thử nghiệm

Độ chính xác của chương trình phân tích dữ liệu ảnh não đưa vào đạt khoảng 80%. Dữ liệu ảnh não đưa vào phân tích với output có Confidence là 1.00 thì cho thông báo kết quả là não của bệnh nhân bị Alzheimer. Các trường hợp có Confidence nhỏ hơn 1.00 trả kết quả là ảnh não của bệnh nhân không bị Alzheimer.

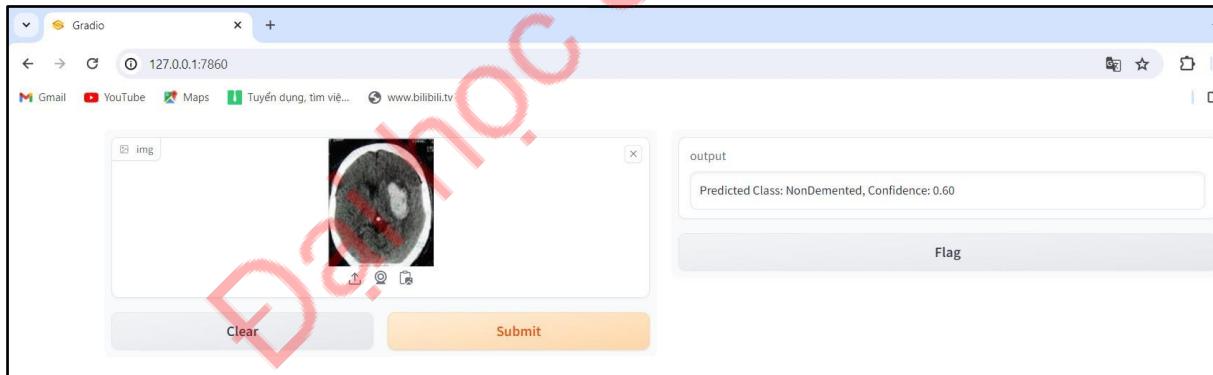
Một số hình ảnh kết quả của chương trình phân tích dữ liệu bệnh Alzheimer:



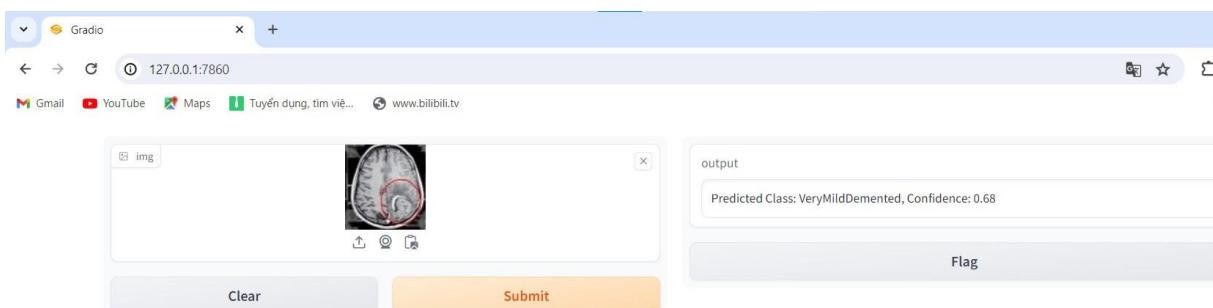
Hình 3.17. Kết quả nhận dạng ảnh não người bị bệnh Alzheimer (1)



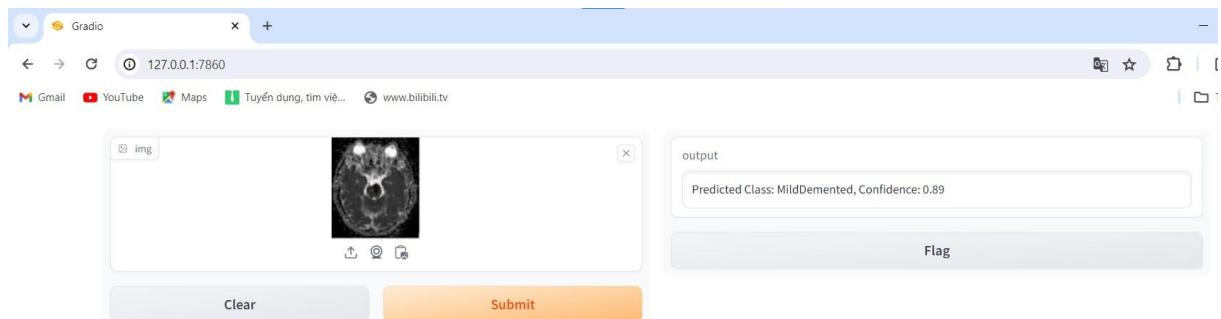
Hình 3.18. Kết quả nhận dạng ảnh não người không bị bệnh Alzheimer



Hình 3.19. Kết quả nhận dạng ảnh não người bị xuất huyết não thùy trái trên



Hình 3.20. Kết quả nhận dạng ảnh não người bị xuất huyết não thùy trái dưới



Hình 3.21. Kết quả nhận dạng ảnh não người bị xuất huyết não chụp X quang cắt ngang

Đại học Sao Đỏ

## KẾT LUẬN

### 1. Kết quả đạt được

- Nhóm tác giả đã đạt được các yêu cầu của đề tài:
  - Xây dựng được bộ dữ liệu Big data về bệnh Alzheimer dựa trên kỹ thuật phân tích dữ liệu Big data và kỹ thuật học máy.
  - Xây dựng được quy trình phân tích bệnh Alzheimer sử dụng học máy.
  - Xây dựng được chương trình phân tích bệnh Alzheimer.
- Chương trình đã tạo một giao diện dễ sử dụng cho người dùng có thể kiểm tra cách mô hình dự đoán lớp của hình ảnh. Đây là một cách tiếp cận hữu ích để triển khai mô hình học máy cho người dùng không chuyên nghiệp.
- Bên cạnh đó chương trình còn giúp người dùng kiểm tra mô hình trên hình ảnh một cách dễ dàng mà không cần kiến thức sâu về lập trình hoặc học máy. Nó có thể sử dụng như một công cụ thực hành cho người mới bắt đầu hoặc để kiểm tra hiệu suất của mô hình trước khi triển khai nó trong môi trường thực tế.

#### ➤ Ưu điểm

- Dễ dàng sử dụng, lập trình, giao diện đơn giản giúp người sử dụng tiếp cận đến việc đào tạo học máy.
- Khả năng tự học, cũng như đưa ra kết quả nhanh chóng thuận tiện cho việc nhận dạng các hình ảnh khác.
- Kết quả đưa ra có các mức độ đánh giá độ tin cậy vào hình ảnh được chọn.
- Việc sử dụng thư viện ảnh tĩnh giúp chương trình và người sử dụng luôn chủ động trong việc khai thác sử dụng, có độ tin cậy cao với kết quả nhận được.

#### ➤ Hạn chế

Chưa khai thác việc sử dụng các thư viện dữ liệu trực tuyến. Việc sử dụng thư viện ảnh tĩnh cần load nhiều ảnh về để chương trình có thể tự học được hoàn chỉnh hơn chiếm dụng nhiều bộ nhớ và ảnh hưởng đến tốc độ phân tích dữ liệu của chương trình, hiệu suất làm việc của máy tính, thời gian trả kết quả cho chương trình.

- Giao diện người dùng chưa được bắt mắt, khó gây hứng thú với người dùng mới khi sử dụng giao diện

### 2. Hướng phát triển của đề tài

- Phát triển thêm chức năng phân tích hình ảnh, dữ liệu trên các trang có liên kết không cần tải dữ liệu về thư viện máy.
- Thêm những đánh giá về độ tin cậy có sức thuyết phục hơn.
- Hoàn thành giao diện người dùng bắt mắt hơn để thu hút được ngay cả những người dùng mới.

## **TÀI LIỆU THAM KHẢO**

- [1] Đại học Sao Đỏ (2022), *Giáo trình Big data*.
- [2] Đại học Sao Đỏ (2021), *Giáo trình môn Mạng Nơron*.
- [3] <https://docs.anaconda.com/>
- [4] <https://docs.python.org/3/>
- [5] Python for Data Science For Dummies, Data Science for Beginners.
- [6] <https://scikit-learn.org/stable/index.html>
- [7] <https://towardsdatascience.com/>
- [8] <https://www.youtube.com/watch?v=g5BdrxPhQU0>
- [9] <https://vinbigdata.com/kham-pha/04-mo-hinh-pre-trained-cnn-giup-ban-giai-quyetcac-bai-toan-thi-giac-may-tinh-voi-transfer-learning.html>

Đại học Sao Đỏ