

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC TẬP CƠ SỞ

Giảng viên hướng dẫn	: TRẦN ĐÌNH QUẾ
Họ và tên sinh viên	: BUI MẠU VĂN
Mã sinh viên	: B22DCCN887
Lớp	: D22CQCN11_B

Hà Nội – 2025

Mục lục

CHƯƠNG 1: TRÍ TUỆ NHÂN TẠO VÀ ỨNG DỤNG CỦA TRÍ TUỆ NHÂN TẠO	5
1.1. Giới thiệu chung về trí tuệ nhân tạo	5
1.1.1. Đặt vấn đề	5
1.1.2. Lý do nghiên cứu và ý nghĩa thực tiễn	5
1.2. Khái niệm và lịch sử phát triển trí tuệ nhân tạo.....	5
1.2.1. Định nghĩa trí tuệ nhân tạo	5
1.2.2. Lịch sử phát triển của AI	6
1.3. Phân loại trí tuệ nhân tạo	7
1.3.1. Theo mức độ thông minh	7
1.3.2. Theo chức năng hoạt động	8
1.4. Các kỹ thuật và công nghệ nền tảng của trí tuệ nhân tạo	8
1.4.1. Học máy (Machine Learning)	8
1.4.2. Học sâu (Deep Learning)	9
1.4.3. Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP).....	9
1.4.4. Thị giác máy tính (Computer Vision)	10
1.4.5. Các kỹ thuật khác (Expert Systems, Robotics, v.v.).....	10
1.5. Ứng dụng của trí tuệ nhân tạo trong các lĩnh vực.....	10
1.5.1. Y tế và chăm sóc sức khỏe	10
1.5.2. Giáo dục	11
1.5.3. Tài chính – ngân hàng	11
1.5.4. Giao thông vận tải	11
1.5.5. Công nghiệp và sản xuất	11
1.5.6. Nông nghiệp	12
1.5.7. An ninh mạng	12
1.5.8. Dịch vụ khách hàng và đời sống	12
1.5.9. Một số ứng dụng nổi bật khác	12
1.6. Lợi ích và thách thức khi ứng dụng trí tuệ nhân tạo	13
1.6.1. Lợi ích.....	13
1.6.2. Thách thức và rủi ro.....	13
1.7. Xu hướng phát triển và triển vọng tương lai của trí tuệ nhân tạo.....	14
1.7.1. Các xu hướng công nghệ mới.....	14
1.7.2. Triển vọng ứng dụng trong 5-10 năm tới.....	14
1.8. Kết luận chương	14
CHƯƠNG 2: CÁC KỸ THUẬT HỌC SÂU	15

2.1. Giới thiệu chung về Học sâu	15
2.1.1. Định nghĩa Học sâu (Deep Learning).....	15
2.1.2. Mối quan hệ giữa Học sâu, Học máy và Trí tuệ nhân tạo	15
2.1.3. Vai trò và tầm quan trọng của Học sâu trong AI hiện đại	15
2.1.4. Lịch sử phát triển và các cột mốc quan trọng trong Học sâu.....	15
2.2. Cơ sở lý thuyết của học sâu	16
2.2.1. Mạng nơ-ron nhân tạo (Artificial Neural Networks – ANN).....	16
2.2.2. Quá trình huấn luyện mạng nơ-ron.....	17
2.2.3. Vấn đề quá khớp (overfitting), thiếu khớp (underfitting) và các phương pháp khắc phục.....	18
2.3. Các kiến trúc mạng học sâu tiêu biểu.....	19
2.3.1. Mạng nơ-ron truyền thẳng (Feedforward Neural Networks – FNN).....	19
2.3.2. Mạng nơ-ron tích chập (Convolutional Neural Networks – CNN).....	19
2.3.3. Mạng nơ-ron hồi tiếp (Recurrent Neural Networks – RNN)	20
2.3.4. Mạng Transformer và cơ chế Attention.....	21
2.3.5. Mạng đối nghịch sinh (Generative Adversarial Networks – GAN).....	22
2.3.6. Mạng nơ-ron đồ thị (Graph Neural Networks – GNN)	22
2.4. Các kỹ thuật và thuật toán hỗ trợ trong Học sâu.....	22
2.4.1. Hàm mất mát (Loss functions) và các loại phổ biến	22
2.4.2. Các thuật toán tối ưu hóa (SGD, Adam, RMSProp....).....	23
2.4.3. Regularization.....	24
2.4.4. Data Augmentation và Preprocessing	25
2.4.5. Transfer Learning và Fine-tuning.....	25
2.5. Đánh giá, ưu nhược điểm và thách thức của Học sâu.....	26
2.5.1. Ưu điểm vượt trội.....	26
2.5.2. Hạn chế và thách thức	26
2.6. Xu hướng phát triển và các hướng nghiên cứu mới trong Học sâu	27
2.6.1. Mô hình lớn (Foundation Models, Large Language Models)	27
2.6.2. Học sâu đa phương thức (Multimodal Deep Learning)	27
2.6.3. Học tăng cường sâu (Deep Reinforcement Learning)	27
2.6.4. Học ít mẫu (Few-shot, One-shot Learning)	28
2.6.5. Học bán giám sát, tự giám sát (Semi-supervised, Self-supervised Learning).....	28
2.6.6. Học sâu lượng tử (Quantum Deep Learning)	28
2.6.7. Tối ưu hóa kiến trúc tự động (Neural Architecture Search - NAS)	28
2.6.8. Học sâu tiết kiệm năng lượng, mô hình nhẹ (Efficient Deep Learning).....	29
2.6.9. Giải thích và kiểm soát Học sâu (Explainable & Controllable Deep Learning - XAI)	29
2.7. Kết luận chương	29
2.7.1. Tóm tắt các nội dung chính	29

2.7.2. Vai trò của Học sâu trong phát triển AI hiện đại	30
2.7.3. Định hướng nghiên cứu và ứng dụng trong các chương tiếp theo	30
CHƯƠNG 3: ỨNG DỤNG HỌC SÂU ĐỂ NHẬN DIỆN U NÃO QUA ẢNH MRI	30
3.1. Giới thiệu tổng quan về Ứng dụng và Bài toán	30
3.1.1. Bối cảnh và Ý nghĩa của Ứng dụng	31
3.1.2. Phát biểu Bài toán cụ thể của Ứng dụng	31
3.1.3. Mục tiêu của Chương 3	31
3.2. Cơ sở Dữ liệu và Phương pháp Chuẩn bị	31
3.2.1. Mô tả Bộ dữ liệu 1	31
3.2.2. Mô tả Bộ dữ liệu 2	32
3.2.3. Mô tả Bộ dữ liệu 3	33
3.3. Phương pháp Nghiên cứu và Xây dựng Mô hình	34
3.3.1. Quy trình Tiền xử lý Dữ liệu	34
3.3.2. Lựa chọn và Thiết kế Kiến trúc Mô hình Học sâu	38
3.3.3. Thiết lập Môi trường và Thông số Huấn luyện	41
3.3.4. Các Chỉ số Đánh giá Hiệu năng Mô hình	43
3.4. Kết quả Thực nghiệm và Phân tích Chi tiết	44
3.4.1. Kết quả trên tập dataset_3	44
3.4.2. Kết quả trên tập dataset_1	47
3.4.3. Kết quả trên tập dataset_2	50
3.5. Triển khai Ứng dụng Minh họa	52
3.5.1. Lựa chọn Nền tảng và Công nghệ Triển khai	53
3.5.2. Kiến trúc và Luồng hoạt động của Ứng dụng	53
3.5.3. Hướng dẫn Sử dụng và Minh họa Giao diện	54
3.6. Kết luận cho Chương 3 và Đề xuất Hướng Phát triển cho Ứng dụng	56
3.6.1. Tóm tắt những Kết quả Chính của Chương	57
3.6.2. Đánh giá Chung về Giải pháp và Mô hình đã Xây dựng	57
3.6.3. Đề xuất các Hướng Phát triển Tiếp theo cho Ứng dụng và Mô hình	57

CHƯƠNG 1: TRÍ TUỆ NHÂN TẠO VÀ ỨNG DỤNG CỦA TRÍ TUỆ NHÂN TẠO

1.1. Giới thiệu chung về trí tuệ nhân tạo

1.1.1. Đặt vấn đề

Trong bối cảnh cuộc Cách mạng công nghiệp lần thứ tư (Công nghiệp 4.0) đang diễn ra mạnh mẽ trên toàn cầu, trí tuệ nhân tạo (Artificial Intelligence - AI) nổi lên như một trong những công nghệ cốt lõi, đóng vai trò trung tâm thúc đẩy sự chuyển đổi số trong mọi lĩnh vực của đời sống và sản xuất. AI không chỉ góp phần tự động hóa các quy trình truyền thống mà còn mở ra những khả năng hoàn toàn mới, từ việc xử lý dữ liệu lớn, dự báo xu hướng, đến hỗ trợ ra quyết định thông minh và sáng tạo vượt trội.

Sự phát triển nhanh chóng của AI đã tạo ra những thay đổi mang tính cách mạng trong các ngành như y tế, giáo dục, tài chính, giao thông, sản xuất, nông nghiệp, an ninh mạng và dịch vụ khách hàng. Ngày nay, AI hiện diện rõ nét trong đời sống hàng ngày thông qua các ứng dụng như trợ lý ảo, hệ thống đề xuất sản phẩm, nhận diện khuôn mặt, xe tự lái, dịch máy tự động, chẩn đoán hình ảnh y tế, v.v. AI giúp tối ưu hóa quy trình làm việc, nâng cao hiệu quả sản xuất, tiết kiệm chi phí và mang lại nhiều tiện ích cho con người.

1.1.2. Lý do nghiên cứu và ý nghĩa thực tiễn

Việc nghiên cứu và ứng dụng trí tuệ nhân tạo ngày càng trở nên cấp thiết trong bối cảnh cạnh tranh toàn cầu và nhu cầu đổi mới sáng tạo không ngừng. AI không chỉ góp phần nâng cao năng suất lao động, chất lượng sản phẩm, dịch vụ mà còn giúp giải quyết nhiều vấn đề phức tạp của xã hội như chăm sóc sức khỏe, bảo vệ môi trường, quản lý đô thị thông minh, phòng chống tội phạm công nghệ cao, v.v.

1.2. Khái niệm và lịch sử phát triển trí tuệ nhân tạo

1.2.1. Định nghĩa trí tuệ nhân tạo

- Các khái niệm cơ bản

Trí tuệ nhân tạo (Artificial Intelligence – AI) là lĩnh vực của khoa học máy tính nghiên cứu và phát triển các hệ thống, phần mềm hoặc máy móc có khả năng thực hiện các nhiệm vụ đòi hỏi trí thông minh của con người như học tập, suy luận, giải quyết vấn đề, nhận thức, hiểu ngôn ngữ tự nhiên và ra quyết định. AI cho phép máy móc mô phỏng các hành vi

thông minh, xử lý dữ liệu ở quy mô lớn, tự động hóa các quy trình phức tạp và thích nghi với môi trường thay đổi.

Nói một cách đơn giản, AI là trí tuệ của máy móc được con người tạo ra, cho phép máy móc tư duy, học hỏi và giải quyết vấn đề tương tự như trí tuệ con người, nhưng với tốc độ, quy mô và độ chính xác vượt trội hơn.

- **Phân biệt AI với các lĩnh vực liên quan (Machine Learning, Deep Learning)**

AI là khái niệm rộng, bao gồm mọi hệ thống có khả năng thực hiện các hành vi thông minh. Bên trong AI, Machine Learning (học máy) là một nhánh tập trung vào việc phát triển các thuật toán cho phép máy tính tự học hỏi từ dữ liệu mà không cần lập trình cụ thể cho từng tác vụ. Machine Learning gồm hai nhóm chính: học có giám sát (supervised learning) và học không giám sát (unsupervised learning), ứng dụng trong phân loại, dự báo, phân cụm, v.v.

Deep Learning (học sâu) là một nhánh con của Machine Learning, sử dụng các mạng nơ-ron nhân tạo nhiều lớp (deep neural networks) để xử lý dữ liệu phức tạp, đặc biệt hiệu quả với dữ liệu phi cấu trúc như hình ảnh, âm thanh, văn bản. Deep Learning tự động trích xuất đặc trưng từ dữ liệu qua nhiều lớp, giảm sự can thiệp thủ công của con người, và đã tạo ra bước tiến lớn trong các lĩnh vực như nhận diện khuôn mặt, xử lý ngôn ngữ tự nhiên, dịch máy, và xe tự lái.

Tóm lại, AI là khái niệm tổng quát nhất, trong đó Machine Learning là một nhánh, và Deep Learning là một nhánh nhỏ hơn nằm trong Machine Learning.

1.2.2. Lịch sử phát triển của AI

- **Các mốc quan trọng**

- 1943: Warren McCulloch và Walter Pitts giới thiệu mô hình mạng nơ-ron nhân tạo đầu tiên, đặt nền móng cho học sâu sau này.
- 1950: Alan Turing đề xuất bài kiểm tra Turing, đặt ra câu hỏi liệu máy móc có thể suy nghĩ như con người hay không.
- 1955: Herbert Simon và Allen Newell phát triển chương trình trí tuệ nhân tạo đầu tiên mang tên “Logic Theorist”, có khả năng chứng minh các định lý toán học.
- 1956: Thuật ngữ “Artificial Intelligence” (Trí tuệ nhân tạo) lần đầu tiên được John McCarthy sử dụng tại Hội nghị Dartmouth, đánh dấu sự ra đời chính thức của lĩnh vực AI.
- 1957: Frank Rosenblatt phát triển Perceptron – mạng nơ-ron nhân tạo đầu tiên có khả năng học từ dữ liệu, mở đường cho các mô hình học sâu hiện đại.

- **Sự phát triển của công nghệ và thuật toán**

- Sau thời kỳ khởi đầu, AI trải qua các giai đoạn phát triển và thách thức:

- ❖ 1960-1970: Sự ra đời của các hệ chuyên gia đầu tiên và ngôn ngữ lập trình AI như LISP, PROLOG. Tuy nhiên, kỳ vọng quá cao dẫn đến “mùa đông AI” đầu tiên khi nguồn tài trợ bị cắt giảm do tiến bộ chậm.
- ❖ 1980-1990: Nhật Bản khởi động dự án “Máy tính thế hệ thứ 5”, thúc đẩy nghiên cứu AI toàn cầu. Các hệ thống AI bắt đầu được ứng dụng vào kinh doanh, sản xuất và robot thông minh.
- ❖ 1997: Máy tính Deep Blue của IBM đánh bại nhà vô địch cờ vua Garry Kasparov, chứng minh sức mạnh của AI trong các lĩnh vực đòi hỏi chiến lược cao.
- ❖ Từ năm 2000: Sự xuất hiện của Big Data và sức mạnh tính toán tăng lên đã thúc đẩy sự phát triển mạnh mẽ của các thuật toán học máy và học sâu. Các mô hình Deep Learning đạt thành tựu vượt trội trong nhận diện hình ảnh, giọng nói, xử lý ngôn ngữ tự nhiên và tự động hóa.
- ❖ Ngày nay, AI đã trở thành công nghệ trọng tâm trong chuyển đổi số, hiện diện rộng rãi trong các ứng dụng thực tiễn như trợ lý ảo, hệ thống đề xuất, xe tự lái, chẩn đoán y tế và nhiều lĩnh vực khác. Sự phát triển liên tục của AI không chỉ nhờ vào các tiến bộ về thuật toán mà còn nhờ vào sự bùng nổ của dữ liệu và khả năng xử lý ngày càng mạnh mẽ của phần cứng máy tính.

Tóm lại, trí tuệ nhân tạo là lĩnh vực khoa học máy tính nghiên cứu phát triển các hệ thống thông minh, với lịch sử phát triển hơn 70 năm, trải qua nhiều giai đoạn thăng trầm và hiện nay đang bùng nổ mạnh mẽ, tạo ra những thay đổi sâu rộng trong đời sống và sản xuất hiện đại.

1.3. Phân loại trí tuệ nhân tạo

Trí tuệ nhân tạo (AI) là một lĩnh vực rộng lớn với nhiều hướng phát triển khác nhau. Để thuận tiện cho việc nghiên cứu, ứng dụng và phát triển, AI thường được phân loại dựa trên hai tiêu chí chính: mức độ thông minh và chức năng hoạt động của hệ thống.

1.3.1. Theo mức độ thông minh

- **AI hẹp (Narrow AI):**

Đây là loại AI phổ biến nhất hiện nay, còn gọi là AI yếu (Weak AI). AI hẹp được thiết kế để thực hiện một nhiệm vụ cụ thể, ví dụ như nhận diện khuôn mặt, dịch ngôn ngữ, chơi cờ vua, hay đề xuất sản phẩm trên các sàn thương mại điện tử. Các hệ thống này có thể vượt trội con người ở các tác vụ chuyên biệt, nhưng không có khả năng tự nhận thức hay hiểu biết ngoài phạm vi được lập trình.

- **AI tổng quát (General AI):**

AI tổng quát, hay còn gọi là AI mạnh (Strong AI), là hệ thống có khả năng hiểu, học hỏi và thực hiện bất kỳ nhiệm vụ trí tuệ nào mà con người có thể làm được. AI

tổng quát có thể tự thích nghi, suy luận và giải quyết vấn đề trong nhiều lĩnh vực khác nhau mà không cần lập trình trước. Hiện nay, AI tổng quát vẫn chỉ tồn tại ở mức ý tưởng và nghiên cứu lý thuyết, chưa có sản phẩm thực tiễn.

- **Siêu trí tuệ nhân tạo (Super AI):**

Đây là cấp độ AI vượt xa trí tuệ con người trong mọi lĩnh vực, kể cả sáng tạo, ra quyết định và kỹ năng xã hội. Siêu trí tuệ nhân tạo có thể tự phát triển, tự cải tiến và đưa ra các giải pháp mà con người khó có thể tưởng tượng. Tuy nhiên, Super AI vẫn là khái niệm lý thuyết và là chủ đề tranh luận về đạo đức, an toàn và tương lai của nhân loại.

1.3.2. Theo chức năng hoạt động

- Máy phản ứng (Reactive Machines)
- Trí nhớ hạn chế (Limited Memory)
- Lý thuyết về tâm trí (Theory of Mind)
- Tự nhận thức (Self-Awareness)

1.4. Các kỹ thuật và công nghệ nền tảng của trí tuệ nhân tạo

Trí tuệ nhân tạo (AI) là một lĩnh vực đa ngành, kết hợp nhiều kỹ thuật và công nghệ nền tảng nhằm xây dựng các hệ thống thông minh có khả năng tự động hóa, học hỏi và thích nghi với môi trường. Trong số đó, nổi bật nhất là các kỹ thuật học máy, học sâu, xử lý ngôn ngữ tự nhiên, thị giác máy tính cùng với các hệ chuyên gia, robot và nhiều công nghệ hỗ trợ khác. Dưới đây là tóm tắt các kỹ thuật quan trọng nhất tạo nên nền tảng cho sự phát triển của AI hiện đại.

1.4.1. Học máy (Machine Learning)

Học máy (Machine Learning - ML) là nhánh quan trọng của AI, tập trung vào việc phát triển các thuật toán cho phép máy tính tự học hỏi từ dữ liệu mà không cần lập trình cụ thể cho từng tác vụ. Các mô hình học máy có thể nhận diện mẫu, dự đoán xu hướng và cải thiện hiệu suất thông qua quá trình huấn luyện với dữ liệu thực tế.

Các phương pháp học máy phổ biến bao gồm:

- Học có giám sát (Supervised Learning): Máy học từ dữ liệu đã được gán nhãn để dự đoán hoặc phân loại dữ liệu mới. Ví dụ: phân loại email spam, nhận diện chữ viết tay.
- Học không giám sát (Unsupervised Learning): Máy tự tìm kiếm cấu trúc hoặc mẫu ẩn trong dữ liệu chưa gán nhãn. Ví dụ: phân cụm khách hàng, phát hiện bất thường.

- Học tăng cường (Reinforcement Learning): Máy học thông qua quá trình thử - sai và nhận phản hồi (thưởng/phạt) từ môi trường để tối ưu hóa quyết định. Ứng dụng trong robot, trò chơi, tối ưu hóa logistics.

Machine Learning đã và đang được ứng dụng rộng rãi trong các lĩnh vực như tài chính (dự báo thị trường), y tế (phân tích hình ảnh y khoa), thương mại điện tử (đề xuất sản phẩm), và nhiều lĩnh vực khác.

1.4.2. Học sâu (Deep Learning)

Học sâu (Deep Learning - DL) là một nhánh con của học máy, sử dụng các mạng nơ-ron nhân tạo nhiều lớp (deep neural networks) để mô hình hóa các mối quan hệ phức tạp trong dữ liệu, đặc biệt hiệu quả với dữ liệu phi cấu trúc như hình ảnh, âm thanh, và văn bản.

Một số kiến trúc học sâu tiêu biểu:

- Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN): Được sử dụng rộng rãi trong nhận diện và phân loại hình ảnh, phát hiện vật thể, nhận diện khuôn mặt.
- Mạng nơ-ron hồi tiếp (Recurrent Neural Networks - RNN) và LSTM: Phù hợp với xử lý chuỗi thời gian, văn bản, dịch máy, nhận diện giọng nói.
- Transformer: Kiến trúc hiện đại, nổi bật với cơ chế attention, đã cách mạng hóa xử lý ngôn ngữ tự nhiên và nhiều lĩnh vực khác.

Deep Learning đã tạo ra bước tiến vượt bậc trong các ứng dụng như xe tự lái, phân tích video, tổng hợp giọng nói, và các trợ lý ảo thông minh.

1.4.3. Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP)

Xử lý ngôn ngữ tự nhiên (NLP) là lĩnh vực nghiên cứu cách máy tính hiểu, phân tích, tạo ra và phản hồi ngôn ngữ của con người. NLP kết hợp kiến thức từ ngôn ngữ học, khoa học máy tính và trí tuệ nhân tạo để xây dựng các hệ thống có khả năng giao tiếp tự nhiên với con người.

Các tác vụ chính của NLP bao gồm:

- Phân tích cú pháp và ngữ nghĩa: Hiểu cấu trúc câu, ý nghĩa của từ/ngữ trong ngữ cảnh.
- Nhận diện thực thể (Named Entity Recognition): Xác định tên người, địa điểm, tổ chức trong văn bản.
- Tóm tắt văn bản, trả lời câu hỏi, dịch máy tự động.
- Sinh văn bản tự động (text generation), chatbot, trợ lý ảo.

1.4.4. Thị giác máy tính (Computer Vision)

Thị giác máy tính là lĩnh vực giúp máy tính “nhìn” và hiểu được thế giới hình ảnh và video giống như con người. Các hệ thống thị giác máy tính sử dụng các thuật toán để trích xuất thông tin, nhận diện đối tượng, phân tích cảnh vật, và thậm chí hiểu được cảm xúc từ hình ảnh.

Các ứng dụng tiêu biểu của thị giác máy tính:

- Nhận diện khuôn mặt, vật thể, biển số xe.
- Phát hiện bất thường trong hình ảnh y tế (X-quang, MRI).
- Phân tích hành vi, giám sát an ninh, xe tự lái.
- Tự động hóa kiểm tra chất lượng sản phẩm trong công nghiệp.

1.4.5. Các kỹ thuật khác (Expert Systems, Robotics, v.v.)

Bên cạnh các lĩnh vực chủ đạo trên, AI còn bao gồm nhiều kỹ thuật và công nghệ nền tảng khác:

- **Hệ chuyên gia (Expert Systems)**
- **Robot thông minh (Robotics)**
- **Tìm kiếm và tối ưu hóa (Search & Optimization)**
- **Học tăng cường (Reinforcement Learning)**
- **Hệ thống lai (Hybrid Systems)**

Tóm lại, các kỹ thuật và công nghệ nền tảng của trí tuệ nhân tạo đã và đang tạo nên sức mạnh vượt trội cho AI trong việc giải quyết các vấn đề thực tiễn, thúc đẩy đổi mới sáng tạo và nâng cao chất lượng cuộc sống. Sự kết hợp giữa học máy, học sâu, xử lý ngôn ngữ tự nhiên, thị giác máy tính cùng với các hệ chuyên gia, robot và các công nghệ liên quan đã mở ra kỷ nguyên mới cho trí tuệ nhân tạo, đồng thời đặt nền móng vững chắc cho sự phát triển bền vững của AI trong tương lai.

1.5. Ứng dụng của trí tuệ nhân tạo trong các lĩnh vực

Trí tuệ nhân tạo (AI) đang tạo ra những thay đổi sâu rộng trên nhiều lĩnh vực, từ y tế, giáo dục, tài chính đến công nghiệp, nông nghiệp và đời sống. Dưới đây là tổng quan các ứng dụng tiêu biểu của AI trong thực tiễn hiện nay.

1.5.1. Y tế và chăm sóc sức khỏe

AI được ứng dụng mạnh mẽ trong chẩn đoán bệnh, phát hiện sớm ung thư, bệnh tim, tiểu đường nhờ khả năng phân tích hình ảnh y học (X-quang, MRI) và dữ liệu lâm sàng với

độ chính xác cao. Các hệ thống AI hỗ trợ bác sĩ nhận diện sớm dấu hiệu bất thường, cá nhân hóa phác đồ điều trị và dự báo nguy cơ bệnh tật. AI còn giúp quản lý hồ sơ bệnh án, tối ưu hóa quy trình khám chữa bệnh, hỗ trợ tư vấn sức khỏe qua chatbot, trợ lý ảo và tự động hóa chăm sóc bệnh nhân từ xa. Ngoài ra, AI còn đóng vai trò quan trọng trong nghiên cứu phát triển thuốc mới và phân tích dữ liệu y tế quy mô lớn.

1.5.2. Giáo dục

Trong giáo dục, AI giúp cá nhân hóa quá trình học tập, phân tích điểm mạnh - yếu của từng học sinh để đề xuất lộ trình phù hợp. AI tự động hóa chấm điểm, theo dõi tiến độ học tập, gửi báo cáo chi tiết cho giáo viên và học sinh, từ đó nâng cao hiệu quả giảng dạy và hỗ trợ can thiệp kịp thời. Công nghệ AI còn giúp nhà trường tối ưu hóa quản lý tài nguyên, lập lịch học, phân bổ phòng học hợp lý. Ngoài ra, AI tích hợp cùng thực tế ảo (VR) và mô phỏng giúp học sinh thực hành, trải nghiệm kiến thức một cách sinh động, an toàn, đặc biệt hữu ích trong các ngành kỹ thuật, y tế.

1.5.3. Tài chính – ngân hàng

AI đã trở thành công cụ không thể thiếu trong lĩnh vực tài chính – ngân hàng. Các hệ thống AI hỗ trợ đánh giá tín dụng, phát hiện gian lận, chống rửa tiền, tự động hóa quy trình giao dịch và quản lý rủi ro. Chatbot và tư vấn viên ảo giúp khách hàng tra cứu thông tin, thực hiện giao dịch nhanh chóng 24/7. AI còn được ứng dụng vào quản lý danh mục đầu tư, dự báo thị trường, cá nhân hóa dịch vụ và tối ưu hóa trải nghiệm khách hàng. Nhờ AI, các ngân hàng nâng cao hiệu quả hoạt động, giảm chi phí và tăng khả năng cạnh tranh.

1.5.4. Giao thông vận tải

AI góp phần tối ưu hóa tuyến đường vận chuyển dựa trên dữ liệu thời gian thực về giao thông, thời tiết, giúp giảm thời gian di chuyển và tiết kiệm nhiên liệu. Các hệ thống AI còn dự báo, phát hiện sự cố, hỗ trợ bảo trì dự đoán cho phương tiện, giảm thiểu tai nạn và chi phí sửa chữa. Xe tự lái, quản lý đội xe thông minh, kiểm soát giao thông tự động là những ứng dụng nổi bật, góp phần nâng cao an toàn và hiệu quả vận tải.

1.5.5. Công nghiệp và sản xuất

Trong sản xuất, AI tự động hóa quy trình từ quản lý dữ liệu, kiểm soát chất lượng đến dự đoán nhu cầu thị trường và tối ưu hóa chuỗi cung ứng. Hệ thống máy móc thông minh có thể tự điều chỉnh, phát hiện lỗi sớm, đề xuất giải pháp cải tiến, giảm lãng phí và tăng hiệu suất. AI giúp doanh nghiệp giảm chi phí vận hành, nâng cao chất lượng sản phẩm và phản ứng linh hoạt với biến động thị trường.

1.5.6. Nông nghiệp

AI thúc đẩy phát triển nông nghiệp chính xác thông qua phân tích dữ liệu từ cảm biến, camera, vệ tinh để giám sát cây trồng, phát hiện sâu bệnh, dự báo thời tiết và tối ưu hóa lượng nước, phân bón. Các hệ thống AI giúp nông dân đưa ra quyết định kịp thời, tăng năng suất, giảm chi phí và hạn chế rủi ro mất mùa. Ứng dụng AI còn bao gồm tự động hóa thu hoạch, phân tích chất lượng nông sản và dự báo giá cả thị trường.

1.5.7. An ninh mạng

AI đang cách mạng hóa lĩnh vực an ninh mạng nhờ khả năng phát hiện, phân tích các mối đe dọa và hành vi bất thường trong thời gian thực. Các hệ thống AI có thể chủ động dự báo, ngăn chặn tấn công mạng, quản lý lỗ hổng bảo mật và hỗ trợ phòng thủ chủ động. Công nghệ này giúp giảm thiểu rủi ro mất an toàn thông tin, bảo vệ dữ liệu cá nhân và doanh nghiệp trước các mối nguy ngày càng tinh vi.

1.5.8. Dịch vụ khách hàng và đời sống

AI nâng cao trải nghiệm khách hàng qua chatbot, trợ lý ảo, tự động hóa trả lời câu hỏi, xử lý yêu cầu 24/7. Hệ thống AI còn phân tích cảm xúc, cá nhân hóa dịch vụ, tự động phân loại và chuyển tiếp yêu cầu đến bộ phận phù hợp, giúp doanh nghiệp tối ưu hóa quy trình chăm sóc khách hàng, giảm chi phí và tăng sự hài lòng. Ngoài ra, AI còn hiện diện trong các ứng dụng đời sống như trợ lý thông minh, hệ thống đề xuất phim, nhạc, mua sắm, nhà thông minh.

1.5.9. Một số ứng dụng nổi bật khác

AI còn được ứng dụng rộng rãi trong giải trí (tạo nhạc, phim, trò chơi điện tử), thương mại điện tử (đề xuất sản phẩm, phân tích hành vi tiêu dùng), truyền thông (tự động hóa biên tập, kiểm duyệt nội dung), pháp lý (phân tích hồ sơ, dự báo kết quả tranh tụng), môi trường (dự báo thiên tai, giám sát chất lượng không khí), và nhiều lĩnh vực khác. Nhờ AI, các ngành nghề này không chỉ tăng hiệu suất mà còn đổi mới phương thức hoạt động, mang lại giá trị mới cho xã hội.

Tóm lại, trí tuệ nhân tạo đã và đang trở thành động lực quan trọng thúc đẩy đổi mới sáng tạo, tối ưu hóa quy trình, nâng cao chất lượng sản phẩm, dịch vụ và cải thiện đáng kể đời sống con người trên nhiều lĩnh vực khác nhau.

1.6. Lợi ích và thách thức khi ứng dụng trí tuệ nhân tạo

1.6.1. Lợi ích

Trí tuệ nhân tạo (AI) mang lại nhiều lợi ích nổi bật cho xã hội và doanh nghiệp. Trước hết, AI giúp nâng cao hiệu quả và tiết kiệm chi phí nhờ khả năng tự động hóa các công việc lặp đi lặp lại, xử lý khối lượng lớn dữ liệu với tốc độ và độ chính xác vượt trội. Nhờ đó, doanh nghiệp giảm thiểu chi phí lao động, hạn chế sai sót và tăng năng suất sản xuất. AI còn hoạt động liên tục 24/7, giúp các dịch vụ như chăm sóc khách hàng, sản xuất, vận hành không bị gián đoạn.

AI cũng thúc đẩy đổi mới sáng tạo và tạo ra giá trị mới. Với khả năng phân tích dữ liệu chuyên sâu, AI hỗ trợ phát hiện xu hướng, dự báo rủi ro, đưa ra quyết định tối ưu và mở ra các mô hình kinh doanh mới. AI giúp cá nhân hóa trải nghiệm người dùng, từ đề xuất sản phẩm/dịch vụ phù hợp đến hỗ trợ dịch ngôn ngữ, xóa bỏ rào cản giao tiếp giữa các quốc gia. Ngoài ra, AI góp phần bảo vệ con người khỏi các công việc nguy hiểm, tăng tốc nghiên cứu khoa học, phát triển thuốc và cải thiện chất lượng cuộc sống hàng ngày.

1.6.2. Thách thức và rủi ro

Bên cạnh lợi ích, AI cũng đặt ra nhiều thách thức và rủi ro. Vấn đề nổi bật là đạo đức và quyền riêng tư: AI thu thập, phân tích lượng lớn dữ liệu cá nhân, tiềm ẩn nguy cơ rò rỉ thông tin, vi phạm quyền riêng tư nếu không được bảo vệ tốt. Hệ thống AI còn có thể duy trì hoặc khuếch đại các định kiến xã hội, gây ra quyết định thiếu công bằng nếu dữ liệu huấn luyện không đa dạng hoặc bị thiên lệch.

Nguy cơ mất việc làm là một thách thức lớn khi AI thay thế con người ở các công việc lặp lại, đơn giản, đặc biệt trong sản xuất, dịch vụ, tài chính. Điều này có thể dẫn đến thất nghiệp gia tăng và đòi hỏi xã hội phải thích nghi, đào tạo lại nguồn lao động.

An ninh dữ liệu cũng là mối lo ngại khi AI trở thành mục tiêu của tấn công mạng hoặc bị khai thác cho mục đích xấu. Ngoài ra, sự phụ thuộc vào công nghệ khiến doanh nghiệp và xã hội dễ bị ảnh hưởng nghiêm trọng nếu hệ thống AI gặp sự cố, bị lỗi hoặc bị kiểm soát ngoài ý muốn. Chi phí đầu tư, vận hành và duy trì AI cũng là rào cản đối với nhiều tổ chức nhỏ và vừa.

1.7. Xu hướng phát triển và triển vọng tương lai của trí tuệ nhân tạo

1.7.1. Các xu hướng công nghệ mới

Trong những năm tới, AI sẽ tiếp tục phát triển mạnh mẽ với nhiều xu hướng nổi bật như AI đa phương thức (Multimodal AI) – kết hợp xử lý văn bản, hình ảnh, âm thanh để tạo ra các hệ thống thông minh toàn diện hơn. AI tự giám sát (Self-supervised AI) đang mở ra khả năng học từ dữ liệu chưa gán nhãn, giúp giảm chi phí và tăng hiệu quả huấn luyện mô hình. AI có thể giải thích được (Explainable AI) cũng trở thành xu hướng quan trọng, giúp người dùng hiểu rõ cách AI đưa ra quyết định, tăng tính minh bạch, tin cậy và giảm rủi ro pháp lý.

Bên cạnh đó, các công nghệ như AI tạo sinh (Generative AI), AI lượng tử (Quantum AI) và Agentic AI (AI tác nhân tự chủ) sẽ tiếp tục định hình lại cách AI được ứng dụng trong kinh doanh, nghiên cứu và đời sống.

1.7.2. Triển vọng ứng dụng trong 5-10 năm tới

Trong 5-10 năm tới, AI dự kiến sẽ trở thành thành phần cốt lõi của hầu hết các ngành kinh tế - xã hội. AI sẽ hỗ trợ ra quyết định chiến lược, quản lý vận hành thông minh, tối ưu hóa chuỗi cung ứng, cá nhân hóa dịch vụ và nâng cao trải nghiệm khách hàng. Các hệ thống AI sẽ ngày càng tự động hóa nhiều quy trình phức tạp, từ chăm sóc sức khỏe, giáo dục, tài chính đến quản lý đô thị thông minh và bảo vệ môi trường.

1.8. Kết luận chương

Chương này đã trình bày tổng quan về trí tuệ nhân tạo, từ khái niệm, lịch sử phát triển, phân loại, các kỹ thuật nền tảng đến các ứng dụng thực tiễn, lợi ích, thách thức và xu hướng phát triển trong tương lai. AI đang và sẽ tiếp tục là động lực quan trọng thúc đẩy đổi mới sáng tạo, tối ưu hóa hoạt động và nâng cao chất lượng cuộc sống.

CHƯƠNG 2: CÁC KỸ THUẬT HỌC SÂU

2.1. Giới thiệu chung về Học sâu

2.1.1. Định nghĩa Học sâu (Deep Learning)

Học sâu (Deep Learning) là một lĩnh vực thuộc học máy (Machine Learning), tập trung vào việc xây dựng và huấn luyện các mạng nơ-ron nhân tạo nhiều lớp (deep neural networks). Các mô hình học sâu có khả năng tự động học và trích xuất các đặc trưng phức tạp từ dữ liệu lớn, giúp máy tính thực hiện các nhiệm vụ nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, nhận diện giọng nói, và nhiều tác vụ trí tuệ khác mà trước đây chỉ con người mới làm được.

2.1.2. Mối quan hệ giữa Học sâu, Học máy và Trí tuệ nhân tạo

Trí tuệ nhân tạo (AI) là lĩnh vực rộng lớn nghiên cứu phát triển các hệ thống có khả năng mô phỏng trí thông minh của con người. Học máy (Machine Learning) là một nhánh của AI, tập trung vào việc phát triển các thuật toán giúp máy tính tự học từ dữ liệu mà không cần lập trình cụ thể cho từng tác vụ. Học sâu (Deep Learning) là một tập con chuyên biệt của học máy, sử dụng các mạng nơ-ron nhiều lớp để giải quyết các bài toán phức tạp. Có thể hình dung mối quan hệ này qua sơ đồ Venn: AI bao hàm học máy, và học máy bao hàm học sâu.

2.1.3. Vai trò và tầm quan trọng của Học sâu trong AI hiện đại

Học sâu được xem là động lực chính thúc đẩy sự phát triển vượt bậc của AI trong thập kỷ qua. Nhờ khả năng tự động học đặc trưng từ dữ liệu thô mà không cần can thiệp thủ công, học sâu đã giúp AI giải quyết thành công các bài toán phức tạp như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, dịch máy, tổng hợp giọng nói, và xe tự lái. Khả năng xử lý dữ liệu lớn, phi cấu trúc và mô hình hóa các mối quan hệ phi tuyến đã đưa học sâu trở thành nền tảng của nhiều sản phẩm và dịch vụ thông minh hiện đại.

2.1.4. Lịch sử phát triển và các cột mốc quan trọng trong Học sâu

Lịch sử học sâu gắn liền với sự phát triển của các mạng nơ-ron nhiều lớp. Năm 1986, mô hình lan truyền ngược (backpropagation) được giới thiệu, mở đường cho việc huấn luyện các mạng nơ-ron nhiều tầng. Năm 1998, LeNet-5 ra đời, đánh dấu bước tiến lớn trong nhận diện chữ viết tay. Đến năm 2006, Geoffrey Hinton và cộng sự công bố phương pháp pre-training cho mạng nơ-ron sâu, giúp giải quyết vấn đề tiêu biến gradient. Năm 2012, sự kiện AlexNet chiến thắng cuộc thi ImageNet với độ chính xác vượt trội đã thúc đẩy làn sóng

nghiên cứu và ứng dụng học sâu trên toàn cầu. Các kiến trúc đột phá tiếp theo như VGG, GoogLeNet, ResNet, LSTM, GAN, và Transformer đã liên tục mở rộng khả năng của học sâu, đưa công nghệ này trở thành trụ cột của AI hiện đại.

2.2. Cơ sở lý thuyết của học sâu

2.2.1. Mạng nơ-ron nhân tạo (Artificial Neural Networks – ANN)

Cấu trúc cơ bản của một neuron nhân tạo

Mạng nơ-ron nhân tạo (ANN) là mô hình tính toán lấy cảm hứng từ hoạt động của não bộ sinh học, gồm các đơn vị cơ bản gọi là neuron nhân tạo (artificial neuron). Mỗi neuron thực hiện bốn bước chính:

- **Nhận đầu vào (Input):** Các giá trị đầu vào có thể là đặc trưng của dữ liệu (ví dụ: giá trị pixel của ảnh, đặc trưng âm thanh, v.v.).
- **Trọng số (Weights):** Mỗi đầu vào được nhân với một trọng số, thể hiện mức độ quan trọng của đặc trưng đó đối với quyết định của neuron. Trọng số là tham số được học trong quá trình huấn luyện mạng.
- **Bias và tổng hợp (Bias, Summation):** Một giá trị bias được cộng vào tổng các đầu vào đã nhân trọng số, giúp dịch chuyển hàm kích hoạt và tăng tính linh hoạt cho mô hình. Tổng hợp này thường là phép cộng tuyến tính: $z = \sum_{i=1}^n w_i x_i + b$
- **Hàm kích hoạt (Activation Function):** Đầu ra tổng hợp z được đưa qua một hàm kích hoạt để tạo ra đầu ra cuối cùng của neuron. Hàm kích hoạt quyết định neuron có “bật” hay không và đưa tính phi tuyến vào mạng, giúp mô hình hóa các quan hệ phức tạp.

Các lớp trong mạng nơ-ron

Một ANN điển hình gồm ba loại lớp chính:

- **Lớp đầu vào (Input layer):** Nhận dữ liệu gốc từ bên ngoài.
- **Lớp ẩn (Hidden layers):** Thực hiện các phép biến đổi phức tạp, trích xuất đặc trưng từ dữ liệu. Số lượng lớp ẩn càng nhiều, mạng càng “sâu” và có khả năng học các đặc trưng trừu tượng hơn.
- **Lớp đầu ra (Output layer):** Đưa ra dự đoán cuối cùng (phân loại, hồi quy, v.v.).

Các neuron giữa các lớp được kết nối với nhau, tạo thành một mạng lưới phức tạp có khả năng học và tổng quát hóa thông tin.

Hàm kích hoạt (activation functions)

Hàm kích hoạt là thành phần không thể thiếu, quyết định tính phi tuyến cho mạng nơ-ron. Nếu không có hàm kích hoạt, toàn bộ mạng chỉ là một phép biến đổi tuyến tính, không thể học các quan hệ phức tạp trong dữ liệu.

Một số hàm kích hoạt phổ biến:

- **Sigmoid:** $\sigma(x) = \frac{1}{1+e^{-x}}$ Cho đầu ra trong khoảng (0, 1), phù hợp cho phân loại nhị phân, nhưng dễ gây bão hòa gradient ở giá trị lớn/nhỏ.
- **Tanh:** $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ Đầu ra từ (-1, 1), trung tâm hóa dữ liệu tốt hơn sigmoid, nhưng vẫn gặp vấn đề bão hòa gradient.
- **ReLU (Rectified Linear Unit):** $f(x) = \max(0, x)$ Được sử dụng rộng rãi nhờ tính đơn giản, hiệu quả tính toán và giảm hiện tượng bão hòa gradient, giúp mạng học nhanh và sâu hơn. Các biến thể như Leaky ReLU, ELU cũng được dùng để khắc phục nhược điểm “chết” neuron của ReLU.

2.2.2. Quá trình huấn luyện mạng nơ-ron

Lan truyền tiến (forward propagation)

Lan truyền tiến là quá trình tính toán đầu ra của mạng dựa trên dữ liệu đầu vào. Dữ liệu được truyền qua từng lớp, mỗi lớp thực hiện phép biến đổi tuyến tính (nhân trọng số, cộng bias), sau đó áp dụng hàm kích hoạt. Kết quả cuối cùng là dự đoán của mạng cho một mẫu dữ liệu. Trong quá trình này, các giá trị trung gian (output của từng lớp) được lưu lại để phục vụ cho bước lan truyền ngược sau này.

Lan truyền ngược (backpropagation)

Lan truyền ngược là thuật toán cốt lõi giúp mạng nơ-ron học từ dữ liệu bằng cách cập nhật các trọng số nhằm giảm sai số dự đoán. Ý tưởng chính:

- Sau khi tính toán đầu ra và hàm mất mát (loss) qua lan truyền tiến, mạng sẽ tính gradient (đạo hàm) của loss đối với từng trọng số bằng quy tắc chuỗi (chain rule).
- Quá trình này bắt đầu từ lớp đầu ra, lan truyền ngược từng lớp về lớp đầu vào, tính toán mức độ ảnh hưởng của từng trọng số đến sai số cuối cùng.
- Các gradient này cho biết cần điều chỉnh trọng số theo hướng nào để giảm sai số dự đoán.

Lan truyền ngược là nền tảng để các thuật toán tối ưu hóa cập nhật trọng số, giúp mạng dần học được các mẫu quan trọng trong dữ liệu mà không cần can thiệp thủ công.

Tối ưu hóa (gradient descent và các biến thể)

Tối ưu hóa là quá trình điều chỉnh các tham số của mạng (trọng số, bias) để giảm giá trị hàm mất mát. Thuật toán cơ bản nhất là **Gradient Descent**:

- Ở mỗi vòng lặp, các trọng số được cập nhật theo hướng ngược lại với gradient của hàm mất mát, nhằm tiến dần về điểm cực tiểu (minimize loss).
- Công thức cập nhật đơn giản: $w_{new} = w_{old} - \eta \frac{\partial L}{\partial w}$ với η là tốc độ học (learning rate).

Các biến thể nổi bật:

- **Stochastic Gradient Descent (SGD)**: Cập nhật trọng số sau mỗi mẫu hoặc mini-batch, giúp hội tụ nhanh, tránh mắc kẹt tại cực tiểu cục bộ.
- **Momentum**: Thêm “quán tính” vào quá trình cập nhật, giúp vượt qua các vùng phẳng hoặc cực tiểu cục bộ.
- **RMSProp**: Tự động điều chỉnh tốc độ học cho từng tham số dựa trên độ lớn gradient gần đây.
- **Adam (Adaptive Moment Estimation)**: Kết hợp ưu điểm của Momentum và RMSProp, tự động điều chỉnh tốc độ học, phổ biến trong thực tiễn nhờ hiệu quả và ổn định.

Nhờ các thuật toán tối ưu hóa hiện đại, mạng nơ-ron có thể học hiệu quả trên dữ liệu lớn, đa dạng và phức tạp.

2.2.3. Vấn đề quá khớp (overfitting), thiếu khớp (underfitting) và các phương pháp khắc phục

Quá khớp (Overfitting):

Xảy ra khi mô hình học quá kỹ các chi tiết và nhiễu trong tập huấn luyện, dẫn đến khả năng tổng quát hóa kém trên dữ liệu mới. Biểu hiện: mô hình đạt kết quả tốt trên tập huấn luyện nhưng kém trên tập kiểm tra.

Thiếu khớp (Underfitting):

Xảy ra khi mô hình quá đơn giản, không đủ khả năng học các đặc trưng quan trọng của dữ liệu, dẫn đến kết quả kém trên cả tập huấn luyện lẫn kiểm tra.

Các phương pháp khắc phục chi tiết như regularization, data augmentation, dropout... sẽ được trình bày ở mục 2.4. Ở đây, cần lưu ý rằng việc cân bằng giữa độ phức tạp mô hình và khả năng tổng quát hóa là yếu tố quyết định thành công của hệ thống học sâu.

Tóm lại, cơ sở lý thuyết của học sâu dựa trên mô hình mạng nơ-ron nhiều lớp, quá trình lan truyền tiến – lan truyền ngược, tối ưu hóa hiện đại và kiểm soát tốt hiện tượng quá khớp/thiếu

khớp. Hiểu rõ các nguyên lý này là nền tảng để xây dựng và ứng dụng hiệu quả các mô hình học sâu trong thực tiễn.

2.3. Các kiến trúc mạng học sâu tiêu biểu

2.3.1. Mạng nơ-ron truyền thẳng (Feedforward Neural Networks – FNN)

Mạng nơ-ron truyền thẳng (FNN) là kiến trúc mạng nơ-ron nhân tạo cơ bản nhất, trong đó dữ liệu di chuyển một chiều từ lớp đầu vào, qua các lớp ẩn, đến lớp đầu ra mà không có vòng lặp hay kết nối ngược. FNN phù hợp cho các bài toán phân loại, hồi quy với dữ liệu dạng bảng hoặc đặc trưng trích xuất sẵn. Ưu điểm của FNN là cấu trúc đơn giản, dễ xây dựng và huấn luyện, phù hợp cho người mới bắt đầu học về học sâu. Tuy nhiên, FNN có những giới hạn rõ rệt: không tận dụng được cấu trúc không gian (như ảnh) hay quan hệ tuần tự (như chuỗi), không có cơ chế ghi nhớ thông tin quá khứ, và không hiệu quả với dữ liệu phức tạp như hình ảnh, chuỗi thời gian hoặc ngôn ngữ tự nhiên. Chính vì vậy, các kiến trúc mạng chuyên biệt hơn đã ra đời để giải quyết các hạn chế này.

2.3.2. Mạng nơ-ron tích chập (Convolutional Neural Networks – CNN)

Cấu trúc, nguyên lý hoạt động

CNN là kiến trúc mạng học sâu chủ đạo cho xử lý dữ liệu dạng hình ảnh và video, tận dụng các đặc tính không gian của dữ liệu. Cấu trúc CNN gồm bốn thành phần chính:

- **Tầng tích chập (Convolutional Layer):**
Đây là tầng cốt lõi, sử dụng các bộ lọc (kernel) nhỏ quét qua ảnh đầu vào, thực hiện phép tích chập để phát hiện các đặc trưng cục bộ như cạnh, góc, hoa văn. Mỗi bộ lọc học một đặc trưng khác nhau, tạo ra các bản đồ đặc trưng (feature maps).
- **Tầng phi tuyến (Activation Layer):**
Thường sử dụng hàm ReLU, giúp mạng học các quan hệ phi tuyến giữa các đặc trưng, tăng khả năng biểu diễn của mô hình.
- **Tầng gộp (Pooling Layer):**
Thực hiện giảm kích thước bản đồ đặc trưng (thường dùng max pooling hoặc average pooling), giúp giảm số lượng tham số, tăng tính bất biến không gian và chống quá khớp.
- **Tầng kết nối đầy đủ (Fully Connected Layer):**
Nằm ở cuối mạng, kết nối toàn bộ các neuron với nhau, thực hiện phân loại hoặc hồi quy dựa trên đặc trưng đã trích xuất.

CNN khai thác kết nối cục bộ (local connectivity) và chia sẻ trọng số, giúp giảm mạnh số lượng tham số và tăng hiệu quả học tập cho dữ liệu lớn như ảnh.

Ứng dụng trong xử lý ảnh và video

CNN đã tạo ra bước tiến vượt bậc trong các bài toán như:

- Phân loại hình ảnh (image classification)
- Nhận diện và định vị đối tượng (object detection)
- Phân đoạn ảnh (image segmentation)
- Nhận diện khuôn mặt, chữ viết tay, biển số xe
- Phân tích video, nhận diện hành động, tạo ảnh mới (image generation)

Các ứng dụng thực tiễn bao gồm: chẩn đoán hình ảnh y tế, giám sát an ninh, xe tự lái, kiểm tra chất lượng sản phẩm công nghiệp, phục chế ảnh cũ, tạo ảnh nghệ thuật.

Các kiến trúc CNN nổi bật

- LeNet: CNN hiệu quả đầu tiên cho nhận diện chữ viết tay.
- AlexNet: Đột phá với việc sử dụng ReLU, Dropout, huấn luyện trên GPU, chiến thắng ImageNet 2012, mở ra kỷ nguyên học sâu hiện đại.
- VGG: Mạng sâu đơn giản với nhiều lớp tích chập 3x3, cho phép học đặc trưng phức tạp hơn.
- GoogLeNet/Inception: Sử dụng các module song song với nhiều kích thước kernel, tối ưu hóa hiệu suất và số tham số.
- ResNet: Giới thiệu skip connections (kết nối tắt), giải quyết vấn đề tiêu biến gradient, cho phép huấn luyện mạng cực sâu.
- DenseNet: Kết nối dày đặc giữa các lớp, tăng khả năng truyền thông tin và giảm số tham số.

2.3.3. Mạng nơ-ron hồi tiếp (Recurrent Neural Networks – RNN)

Nguyên lý hoạt động, ưu nhược điểm

RNN là kiến trúc mạng học sâu dành cho dữ liệu chuỗi (sequence data) như văn bản, âm thanh, tín hiệu thời gian. RNN có các kết nối ngược, cho phép thông tin từ bước trước ảnh hưởng đến bước sau, giúp mô hình ghi nhớ trạng thái quá khứ. Tại mỗi bước thời gian, đầu vào mới và trạng thái ẩn trước đó được kết hợp để tính toán trạng thái ẩn hiện tại và đầu ra.

Ưu điểm: Phù hợp cho bài toán xử lý chuỗi, ngôn ngữ tự nhiên, dịch máy, dự báo thời gian.

Hạn chế: Dễ gặp vấn đề tiêu biến/bùng nổ gradient khi chuỗi quá dài, dẫn đến khó học các phụ thuộc xa.

Các biến thể: LSTM, GRU

- **LSTM (Long Short-Term Memory):**
Giải quyết hạn chế của RNN truyền thống bằng các “cổng” (gates): cổng quên, cổng đầu vào, cổng đầu ra. Các cổng này kiểm soát luồng thông tin, giúp lưu trữ hoặc loại bỏ thông tin không cần thiết, nhờ đó học được các phụ thuộc dài hạn.
- **GRU (Gated Recurrent Unit):**
Đơn giản hóa kiến trúc LSTM với hai cổng (cổng cập nhật và cổng đặt lại), giảm số tham số nhưng vẫn giữ khả năng ghi nhớ dài hạn.
- LSTM/GRU có cấu trúc các cổng kiểm soát trạng thái, giúp mô hình “nhớ” hoặc “quên” thông tin qua từng bước thời gian.

Ứng dụng trong xử lý chuỗi, ngôn ngữ tự nhiên

- Dịch máy tự động (machine translation)
- Tổng hợp văn bản, dự báo chuỗi thời gian
- Nhận diện giọng nói, phân tích cảm xúc, chatbot, phân tích chuỗi gen

2.3.4. Mạng Transformer và cơ chế Attention

Kiến trúc Transformer

Transformer là kiến trúc mạng học sâu hiện đại, thay thế hoàn toàn RNN trong nhiều bài toán chuỗi. Kiến trúc gồm hai thành phần chính: Encoder (mã hóa đầu vào) và Decoder (giải mã đầu ra). Điểm khác biệt là Transformer không xử lý tuần tự mà xử lý song song toàn bộ chuỗi, nhờ đó tăng tốc huấn luyện và khai thác tốt hơn các phụ thuộc xa.

Cơ chế Self-Attention và Multi-Head Attention

- **Self-Attention:**
Cho phép mô hình đánh giá mức độ quan trọng của từng phần tử trong chuỗi so với các phần tử khác, từ đó tập trung vào các thông tin quan trọng nhất. Ví dụ, khi dịch một câu, mô hình có thể “chú ý” đến các từ liên quan trong toàn câu, thay vì chỉ dựa vào trạng thái trước đó.
- **Multi-Head Attention:**
Sử dụng nhiều “đầu” attention song song, mỗi đầu học một khía cạnh khác nhau của chuỗi, tăng khả năng biểu diễn và trích xuất thông tin đa chiều.

Nhờ attention, Transformer xử lý hiệu quả các phụ thuộc dài, song song hóa quá trình huấn luyện, và đạt kết quả vượt trội trong xử lý ngôn ngữ tự nhiên.

Các mô hình nổi bật

- **BERT:** Mô hình tiền huấn luyện hai chiều, xuất sắc trong hiểu ngữ nghĩa văn bản, phân loại, trả lời câu hỏi.
- **GPT:** Mô hình sinh ngôn ngữ mạnh mẽ, tạo văn bản tự nhiên, dịch máy, tóm tắt, hội thoại.

- ViT: Áp dụng Transformer cho xử lý ảnh, cạnh tranh với CNN trong nhiều tác vụ thị giác máy tính.

2.3.5. Mạng đối nghịch sinh (Generative Adversarial Networks – GAN)

Nguyên lý hoạt động

GAN gồm hai mạng nơ-ron: Generator (sinh dữ liệu giả) và Discriminator (phân biệt thật/giả), huấn luyện theo cơ chế “đối nghịch” như một trò chơi hai người:

- Generator cố gắng tạo ra dữ liệu giả giống dữ liệu thật nhất.
- Discriminator cố gắng phân biệt dữ liệu thật và dữ liệu do Generator tạo ra.

Hai mạng này cạnh tranh nhau, đến khi Generator tạo ra dữ liệu mà Discriminator không thể phân biệt được. Ý tưởng “trò chơi đối kháng” này giúp GAN học sinh dữ liệu mới rất thực tế.

Các ứng dụng tiêu biểu

- Tạo ảnh chân thực, ảnh nghệ thuật, video, giọng nói giả lập
- Tăng cường dữ liệu huấn luyện khi dữ liệu thật hạn chế
- Chuyển đổi phong cách ảnh, phục chế ảnh cũ, tạo ảnh động

2.3.6. Mạng nơ-ron đồ thị (Graph Neural Networks – GNN)

Kiến trúc, nguyên lý

GNN là kiến trúc chuyên biệt cho dữ liệu có cấu trúc dạng đồ thị (graph), nơi các phần tử (node) liên kết với nhau qua các cạnh (edge). GNN học cách truyền và tích hợp thông tin giữa các node lân cận, giúp khai thác cấu trúc liên kết phức tạp mà các mạng truyền thống không xử lý được.

Ứng dụng với dữ liệu dạng đồ thị

- Mạng xã hội: Phân tích cộng đồng, đề xuất kết bạn, phát hiện hành vi bất thường.
- Hóa học, sinh học: Dự đoán tính chất phân tử, tương tác protein.
- Đề xuất sản phẩm: Xây dựng hệ thống đề xuất dựa trên quan hệ giữa người dùng và sản phẩm.
- Tối ưu hóa mạng lưới giao thông, điện lực, viễn thông.

2.4. Các kỹ thuật và thuật toán hỗ trợ trong Học sâu

Các kỹ thuật và thuật toán hỗ trợ là nền tảng giúp mô hình học sâu đạt hiệu quả cao, tránh các lỗi phổ biến và tối ưu hóa khả năng tổng quát hóa. Dưới đây là các nhóm kỹ thuật quan trọng nhất.

2.4.1. Hàm mất mát (Loss functions) và các loại phổ biến

Vai trò:

Hàm mất mát (loss function) là thước đo định lượng mức độ sai lệch giữa dự đoán của mô hình và giá trị thực tế. Trong quá trình huấn luyện, mục tiêu là điều chỉnh các tham số của mạng để hàm mất mát đạt giá trị nhỏ nhất, giúp mô hình dự đoán chính xác hơn.

Các hàm mất mát phổ biến:

- **Mean Squared Error (MSE):**

Dùng cho các bài toán hồi quy, đo lường bình phương sai số trung bình giữa dự đoán và thực tế.

MSE nhạy cảm với ngoại lệ (outlier), phù hợp khi muốn phạt mạnh các sai số lớn.

- **Mean Absolute Error (MAE):**

Cũng dùng cho hồi quy, đo lường sai số tuyệt đối trung bình. MAE ít nhạy cảm với ngoại lệ hơn MSE.

- **Cross-Entropy Loss:**

Dùng cho bài toán phân loại, đặc biệt là phân loại nhị phân và đa lớp. Cross-entropy đo lường sự khác biệt giữa phân phối xác suất dự đoán và thực tế.

- **Binary Cross-Entropy:** Cho phân loại nhị phân.
- **Categorical Cross-Entropy:** Cho phân loại đa lớp.

Lưu ý lựa chọn:

- Hồi quy: Ưu tiên MSE, MAE.
- Phân loại: Cross-Entropy là lựa chọn chuẩn mực.
- Một số bài toán đặc biệt có thể cần hàm mất mát tùy chỉnh.

2.4.2. Các thuật toán tối ưu hóa (SGD, Adam, RMSProp...)

Vai trò:

Thuật toán tối ưu hóa (optimizer) quyết định cách thức cập nhật trọng số của mạng dựa trên gradient của hàm mất mát. Mục tiêu là tìm bộ tham số giúp hàm mất mát nhỏ nhất có thể, tức là mô hình dự đoán tốt nhất.

Các thuật toán phổ biến:

- **Stochastic Gradient Descent (SGD):**

Là phương pháp cơ bản, cập nhật trọng số sau mỗi mẫu hoặc mini-batch. SGD giúp học nhanh hơn so với batch gradient descent truyền thống, nhưng có thể dao động mạnh quanh điểm tối ưu.

- **Momentum:**

Bổ sung “quán tính” vào SGD, giúp tăng tốc hội tụ và vượt qua các vùng phẳng

hoặc cực tiểu cục bộ. Trọng số được cập nhật không chỉ dựa vào gradient hiện tại mà còn dựa vào hướng di chuyển trước đó.

- **RMSProp:**

Tự động điều chỉnh tốc độ học cho từng tham số dựa trên độ lớn gradient gần đây, giúp mô hình học ổn định ngay cả với dữ liệu không đồng nhất.

- **Adam (Adaptive Moment Estimation):**

Kết hợp ưu điểm của Momentum và RMSProp, tự động điều chỉnh tốc độ học và hướng cập nhật, phổ biến nhờ hiệu quả và ổn định trên nhiều loại bài toán phức tạp.

2.4.3. Regularization

Vai trò:

Regularization là nhóm kỹ thuật giúp giảm nguy cơ quá khớp (overfitting), tăng khả năng tổng quát hóa của mô hình bằng cách kiểm soát độ phức tạp hoặc đa dạng hóa quá trình học.

Các kỹ thuật regularization phổ biến:

- **Dropout:**

Trong quá trình huấn luyện, một tỷ lệ ngẫu nhiên các neuron trong mạng sẽ bị “tắt” (không hoạt động) ở mỗi batch. Điều này buộc mạng không phụ thuộc vào một số đường truyền nhất định, giúp học các đặc trưng đa dạng hơn và giảm quá khớp. Khi kiểm tra, tất cả neuron đều hoạt động, nhưng đầu ra được nhân với xác suất dropout để cân bằng.

- **Batch Normalization:**

Chuẩn hóa đầu ra của mỗi lớp về phân phối chuẩn ($\text{mean} = 0$, $\text{variance} = 1$), giúp tăng tốc huấn luyện, ổn định gradient và giảm phụ thuộc vào khởi tạo trọng số. BatchNorm còn có tác dụng như một regularizer nhẹ, giúp mô hình tổng quát hóa tốt hơn.

- **Early Stopping:**

Dừng huấn luyện khi hiệu suất trên tập kiểm tra (validation) không còn cải thiện, tránh mô hình học quá kỹ dữ liệu huấn luyện. Kỹ thuật này giúp tiết kiệm thời gian và giảm nguy cơ quá khớp.

- **L1/L2 Regularization:**

Thêm điều kiện phạt vào hàm mất mát để kiểm soát độ lớn của trọng số. L1 regularization khuyến khích trọng số thưa (nhiều giá trị bằng 0), L2 regularization giới hạn trọng số lớn, giúp mô hình đơn giản và ổn định hơn.

2.4.4. Data Augmentation và Preprocessing

- **Vai trò:**

Data augmentation và preprocessing giúp mô hình học sâu tận dụng tối đa dữ liệu, tăng khả năng tổng quát hóa và giảm nguy cơ quá khớp.

- **Data Augmentation:**

Tạo thêm dữ liệu huấn luyện từ dữ liệu gốc bằng các phép biến đổi như xoay, lật, phóng to/thu nhỏ, thay đổi độ sáng, thêm nhiễu... Đặc biệt quan trọng với dữ liệu hình ảnh, giúp mô hình học được các đặc trưng đa dạng, tăng hiệu quả khi dữ liệu thực tế hạn chế.

- **Preprocessing:**

Xử lý sơ bộ dữ liệu trước khi đưa vào mô hình, bao gồm chuẩn hóa (scaling), chuẩn hóa về trung bình và phương sai, mã hóa nhãn (encoding), loại bỏ nhiễu, xử lý dữ liệu thiếu... Quá trình này đảm bảo dữ liệu đầu vào phù hợp với yêu cầu của mô hình, giúp tăng hiệu suất và độ ổn định.

2.4.5. Transfer Learning và Fine-tuning

Khái niệm và lợi ích:

Transfer Learning (học chuyển giao) là kỹ thuật tận dụng kiến thức từ các mô hình đã được huấn luyện trước trên tập dữ liệu lớn (ví dụ: ImageNet, Wikipedia) để giải quyết các bài toán mới với dữ liệu hạn chế. Fine-tuning là quá trình tinh chỉnh lại một phần hoặc toàn bộ mô hình tiền huấn luyện trên tập dữ liệu mới, giúp mô hình thích nghi với bài toán cụ thể.

Lợi ích:

- Tiết kiệm thời gian, chi phí huấn luyện.
- Đạt hiệu quả cao ngay cả khi dữ liệu huấn luyện ít.
- Ứng dụng rộng rãi trong thị giác máy tính (dùng các mô hình như VGG, ResNet, Inception đã huấn luyện trên ImageNet) và xử lý ngôn ngữ tự nhiên (BERT, GPT...).

Quy trình Fine-tuning:

- Tải mô hình tiền huấn luyện.
- Thay thế lớp đầu ra phù hợp với bài toán mới.
- Đóng băng (không cập nhật) một số lớp ban đầu, chỉ huấn luyện các lớp cuối hoặc toàn bộ mô hình với tốc độ học nhỏ.
- Đánh giá và điều chỉnh mô hình trên tập dữ liệu mới.

Tóm lại, các kỹ thuật và thuật toán hỗ trợ như lựa chọn hàm mất mát phù hợp, tối ưu hóa hiệu quả, regularization, tăng cường dữ liệu và học chuyển giao là những yếu tố then chốt giúp mô hình học sâu đạt hiệu suất cao, tổng quát hóa tốt và ứng dụng thành công trong thực tiễn.

2.5. Đánh giá, ưu nhược điểm và thách thức của Học sâu

2.5.1. Ưu điểm vượt trội

Học sâu (Deep Learning) đã chứng minh nhiều ưu thế nổi bật so với các phương pháp học máy truyền thống, đặc biệt ở các bài toán xử lý dữ liệu lớn, phi cấu trúc và phức tạp. Một trong những ưu điểm quan trọng nhất là khả năng tự động học đặc trưng: các mạng học sâu tự động trích xuất và tối ưu hóa các đặc trưng quan trọng từ dữ liệu đầu vào, không cần thiết kế thủ công đặc trưng như trước đây. Điều này giúp giảm phụ thuộc vào chuyên gia lĩnh vực và tăng khả năng mở rộng cho nhiều loại dữ liệu khác nhau.

Học sâu cũng thể hiện hiệu suất vượt trội trong các bài toán như nhận diện hình ảnh, phân tích video, xử lý ngôn ngữ tự nhiên, nhận diện giọng nói, tổng hợp dữ liệu, dự báo chuỗi thời gian, v.v. Các mô hình sâu có khả năng xấp xỉ các hàm phi tuyến phức tạp với độ chính xác cao, giúp giải quyết những bài toán mà các phương pháp tuyến tính hoặc mạng nông không thể xử lý hiệu quả. Ngoài ra, học sâu còn dễ dàng thích nghi với dữ liệu đa chiều, đa phương thức (kết hợp ảnh, văn bản, âm thanh), mở rộng phạm vi ứng dụng trong thực tiễn.

2.5.2. Hạn chế và thách thức

Bên cạnh những ưu điểm vượt trội, học sâu cũng tồn tại nhiều hạn chế và thách thức đặc thù:

- Đòi hỏi lượng dữ liệu huấn luyện rất lớn
- Yêu cầu tài nguyên tính toán lớn
- Vấn đề giải thích mô hình (Explainability)
- Nhạy cảm với dữ liệu nhiễu và tấn công adversarial
- Khó khăn trong điều chỉnh siêu tham số (hyperparameter tuning)
- Các vấn đề về bias, fairness, security

2.6. Xu hướng phát triển và các hướng nghiên cứu mới trong Học sâu

2.6.1. Mô hình lớn (Foundation Models, Large Language Models)

Một trong những xu hướng nổi bật nhất là sự phát triển của các mô hình học sâu quy mô lớn, còn gọi là foundation models hoặc large language models (LLMs). Các mô hình như GPT, BERT, T5, PaLM, Gemini... được huấn luyện trên kho dữ liệu khổng lồ, với hàng tỷ tham số, cho phép chúng hiểu, sinh và tổng hợp ngôn ngữ tự nhiên, hình ảnh, mã nguồn, v.v. ở mức độ chưa từng có.

Ưu điểm của foundation models là khả năng thích ứng với nhiều tác vụ khác nhau thông qua fine-tuning hoặc prompt engineering, giảm nhu cầu xây dựng mô hình riêng biệt cho từng bài toán.

Tuy nhiên, việc huấn luyện và triển khai các mô hình lớn đòi hỏi tài nguyên tính toán cực lớn, đặt ra thách thức về chi phí, năng lượng và kiểm soát chất lượng đầu ra.

2.6.2. Học sâu đa phương thức (Multimodal Deep Learning)

Học sâu đa phương thức là hướng nghiên cứu kết hợp xử lý nhiều loại dữ liệu khác nhau (văn bản, hình ảnh, âm thanh, video...) trong cùng một mô hình. Các hệ thống này cho phép AI hiểu và tương tác với thế giới giống con người hơn, ví dụ: mô tả ảnh bằng văn bản, trả lời câu hỏi dựa trên hình ảnh, hoặc tổng hợp video từ mô tả ngôn ngữ.

Các mô hình nổi bật như CLIP, DALL-E, Gemini, Flamingo... đã chứng minh khả năng kết hợp thông tin từ nhiều nguồn, mở rộng phạm vi ứng dụng cho AI trong giáo dục, y tế, truyền thông, giải trí và nhiều lĩnh vực khác.

2.6.3. Học tăng cường sâu (Deep Reinforcement Learning)

Học tăng cường sâu (DRL) kết hợp học sâu với học tăng cường truyền thống, cho phép mô hình học thông qua tương tác với môi trường và nhận phần thưởng hoặc hình phạt dựa trên hành động thực hiện.

DRL đã đạt thành tựu nổi bật trong các trò chơi (AlphaGo, Dota 2, StarCraft II), robot tự động hóa, điều khiển hệ thống phức tạp (giao thông, năng lượng, logistics), và tối ưu hóa quy trình sản xuất.

2.6.4. Học ít mẫu (Few-shot, One-shot Learning)

Học ít mẫu hướng đến xây dựng các mô hình có khả năng học và nhận diện đối tượng mới chỉ với rất ít ví dụ (thậm chí chỉ một mẫu). Điều này đặc biệt quan trọng trong các lĩnh vực khó thu thập dữ liệu hoặc dữ liệu hiếm gặp (y tế, sinh học, an ninh...).

Các kỹ thuật như meta-learning, Siamese Network, Prototypical Network, Matching Network... đang được nghiên cứu để tăng khả năng tổng quát hóa của mô hình với dữ liệu hạn chế, giảm chi phí gán nhãn và mở rộng ứng dụng AI cho các bài toán mới.

2.6.5. Học bán giám sát, tự giám sát (Semi-supervised, Self-supervised Learning)

Trong thực tế, dữ liệu gán nhãn thường rất hạn chế, trong khi dữ liệu chưa gán nhãn lại rất phong phú. Học bán giám sát (kết hợp dữ liệu có nhãn và không nhãn) và tự giám sát (tạo nhiệm vụ phụ để mô hình tự học đặc trưng từ dữ liệu không nhãn) là hai hướng tiếp cận giúp tận dụng tối đa nguồn dữ liệu sẵn có.

Self-supervised learning đã tạo ra bước tiến lớn trong các mô hình ngôn ngữ (BERT, GPT) và thị giác máy tính, giúp mô hình học các biểu diễn mạnh mẽ mà không cần nhiều nhãn thủ công, giảm chi phí và tăng hiệu quả huấn luyện.

2.6.6. Học sâu lượng tử (Quantum Deep Learning)

Học sâu lượng tử là giao thoa giữa điện toán lượng tử và học sâu, tận dụng sức mạnh xử lý song song và không gian trạng thái khổng lồ của máy tính lượng tử để tăng tốc huấn luyện, tối ưu hóa mô hình và giải quyết các bài toán phức tạp vượt quá khả năng của máy tính cổ điển.

2.6.7. Tối ưu hóa kiến trúc tự động (Neural Architecture Search - NAS)

NAS là lĩnh vực nghiên cứu các thuật toán tự động tìm kiếm kiến trúc mạng học sâu tối ưu cho từng bài toán cụ thể, thay vì dựa vào kinh nghiệm và thử nghiệm thủ công. Các phương pháp NAS sử dụng học tăng cường, tiến hóa hoặc gradient để khám phá không gian kiến trúc, giúp phát hiện ra các mô hình hiệu quả về hiệu suất, kích thước, tốc độ và tiêu thụ tài nguyên.

2.6.8. Học sâu tiết kiệm năng lượng, mô hình nhẹ (Efficient Deep Learning)

Trước nhu cầu triển khai AI trên các thiết bị biên (edge devices) như điện thoại, camera, thiết bị IoT, hướng nghiên cứu về học sâu tiết kiệm năng lượng và mô hình nhẹ đang phát triển mạnh mẽ.

Các kỹ thuật như nén mô hình (model compression), lượng tử hóa (quantization), cắt tỉa (pruning), chuyển giao tri thức (knowledge distillation), và thiết kế kiến trúc tối ưu giúp giảm kích thước, độ phức tạp tính toán và tiêu thụ năng lượng mà vẫn giữ được độ chính xác cao.

2.6.9. Giải thích và kiểm soát Học sâu (Explainable & Controllable Deep Learning - XAI)

Giải thích mô hình (Explainable AI - XAI) là hướng nghiên cứu phát triển các phương pháp giúp người dùng và chuyên gia hiểu được lý do đằng sau các quyết định của mô hình học sâu, tăng tính minh bạch, tin cậy và khả năng kiểm soát hệ thống AI.

Các kỹ thuật phổ biến gồm:

- Mô hình tự giải thích (self-interpretable models): Như cây quyết định, hồi quy tuyến tính.
- Giải thích hậu kiểm (post-hoc explanations): Như LIME, SHAP, saliency map, phân tích độ nhạy, giải thích đối chiếu (contrastive explanation)...

XAI đặc biệt quan trọng trong các lĩnh vực nhạy cảm như y tế, pháp lý, tài chính, nơi quyết định của AI cần được giải trình rõ ràng và kiểm soát rủi ro.

Tóm lại, các xu hướng mới trong học sâu đang tập trung vào mở rộng khả năng mô hình hóa (mô hình lớn, đa phương thức), tăng hiệu quả dữ liệu (học ít mẫu, tự giám sát), tối ưu hóa hiệu suất (mô hình nhẹ, NAS), khai thác sức mạnh lượng tử, và tăng cường tính minh bạch, kiểm soát của hệ thống. Những hướng đi này sẽ tiếp tục thúc đẩy AI phát triển mạnh mẽ, ứng dụng rộng rãi và an toàn hơn trong tương lai gần.

2.7. Kết luận chương

2.7.1. Tóm tắt các nội dung chính

Chương này đã trình bày toàn diện về các kỹ thuật, kiến trúc và quy trình cốt lõi của học sâu. Từ nền tảng mạng nơ-ron truyền thẳng (FNN), các kiến trúc chuyên biệt như mạng tích chập (CNN) cho xử lý ảnh, mạng hồi tiếp (RNN, LSTM, GRU) cho dữ liệu chuỗi, Transformer với cơ chế Attention cho ngôn ngữ tự nhiên, đến các mô hình sinh tổng hợp (GAN) và mạng nơ-ron đồ thị (GNN) cho dữ liệu phức tạp, mỗi kiến trúc đều có vai trò và

ứng dụng riêng biệt. Chương cũng đã phân tích chi tiết các quy trình huấn luyện mạng nơ-ron, các thuật toán tối ưu hóa, kỹ thuật regularization, tăng cường dữ liệu, học chuyển giao và các phương pháp hỗ trợ khác nhằm nâng cao hiệu quả và khả năng tổng quát hóa của mô hình học sâu.

2.7.2. Vai trò của Học sâu trong phát triển AI hiện đại

Học sâu đã trở thành động lực chính thúc đẩy làn sóng đổi mới trong trí tuệ nhân tạo hiện đại. Nhờ khả năng tự động học đặc trưng, xử lý dữ liệu lớn và phi cấu trúc, cũng như mô hình hóa các quan hệ phi tuyến phức tạp, học sâu đã giúp AI đạt được những thành tựu vượt trội trong nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, tổng hợp dữ liệu, điều khiển robot và nhiều lĩnh vực khác. Sự phát triển của các mô hình lớn, học đa phương thức và các kỹ thuật tối ưu hóa hiện đại tiếp tục mở rộng phạm vi ứng dụng và sức mạnh của AI trong thực tiễn.

2.7.3. Định hướng nghiên cứu và ứng dụng trong các chương tiếp theo

Các kiến thức, kỹ thuật và kiến trúc học sâu trình bày trong chương này là nền tảng vững chắc cho việc triển khai các ứng dụng AI cụ thể ở các chương tiếp theo của báo cáo. Những nội dung này sẽ được vận dụng để phân tích, thiết kế và đánh giá các hệ thống AI thực tiễn trong các lĩnh vực như y tế, giáo dục, tài chính, công nghiệp, an ninh mạng, v.v. Đồng thời, các xu hướng nghiên cứu mới cũng sẽ là định hướng quan trọng cho các phân tích chuyên sâu, đề xuất giải pháp và phát triển ứng dụng AI trong tương lai.

CHƯƠNG 3: ỨNG DỤNG HỌC SÂU ĐỂ NHẬN DIỆN U NÃO QUA ẢNH MRI

3.1. Giới thiệu tổng quan về Ứng dụng và Bài toán

Chương này tập trung trình bày quá trình xây dựng và đánh giá một ứng dụng cụ thể của học sâu – một lĩnh vực tiên tiến của Trí tuệ Nhân tạo (đã được giới thiệu tổng quan ở Chương 1) và sử dụng các kỹ thuật học sâu (đã phân tích ở Chương 2) – vào việc giải quyết một bài toán thực tiễn trong lĩnh vực y tế: nhận diện u não qua ảnh chụp cộng hưởng từ (MRI).

3.1.1. Bối cảnh và Ý nghĩa của Ứng dụng

U não là một trong những bệnh lý nguy hiểm với tỷ lệ mắc và tử vong đáng kể trên toàn cầu. Việc phát hiện sớm và chẩn đoán chính xác các loại u não đóng vai trò then chốt trong việc đưa ra phác đồ điều trị kịp thời và hiệu quả, từ đó cải thiện tiên lượng cho bệnh nhân. Hình ảnh chụp cộng hưởng từ (MRI) hiện là công cụ không thể thiếu, cung cấp hình ảnh chi tiết về cấu trúc não và các tổn thương. Tuy nhiên, việc phân tích ảnh MRI đòi hỏi chuyên môn cao và thời gian của các bác sĩ X-quang. Trước bối cảnh đó, ứng dụng các kỹ thuật học sâu có tiềm năng to lớn trong việc tự động hóa một phần quy trình, trợ giúp các chuyên gia y tế bằng cách cung cấp một công cụ phân loại sơ bộ các loại u não, đặc biệt là các dạng phổ biến như Glioma, Meningioma và Pituitary, cũng như phân biệt với trường hợp không có khối u. Ứng dụng được trình bày trong chương này nhằm mục đích khai thác tiềm năng đó.

3.1.2. Phát biểu Bài toán cụ thể của Ứng dụng

Bài toán được giải quyết trong khuôn khổ ứng dụng này là **phân loại đa lớp các ảnh chụp cộng hưởng từ (MRI) sọ não thành bốn (04):**

- **Glioma (U thần kinh đệm)**
- **Meningioma (U màng não)**
- **No_Tumor (Không có khối u)**
- **Pituitary (U tuyến yên)**

Đầu vào của hệ thống là các ảnh MRI não đã qua xử lý, và đầu ra mong muốn là nhãn dự đoán tương ứng với một trong bốn lớp nêu trên cho mỗi ảnh đầu vào.

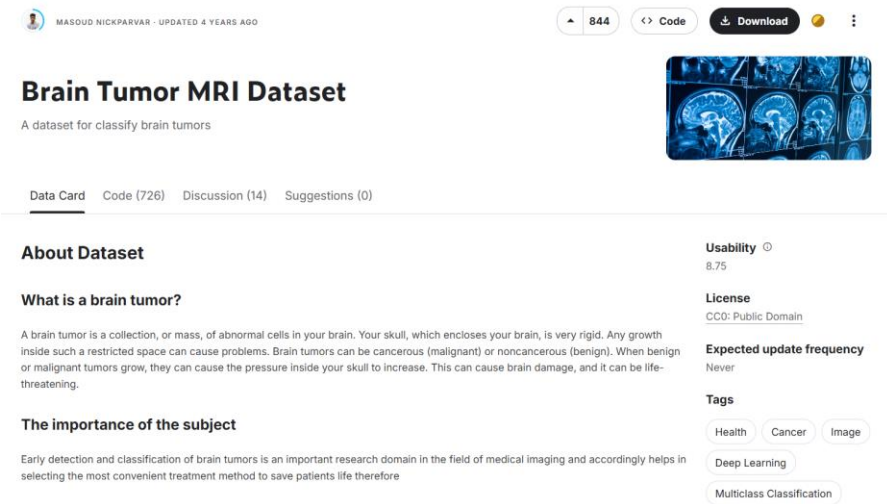
3.1.3. Mục tiêu của Chương 3

- Trình bày chi tiết quá trình xây dựng, từ thu thập dữ liệu, tiền xử lý, lựa chọn kiến trúc mô hình học sâu, đến huấn luyện và tinh chỉnh mô hình cho bài toán phân loại 4 lớp u não đã nêu.
- Đánh giá một cách khách quan hiệu năng của mô hình đã phát triển trên ít nhất ba bộ dữ liệu ảnh MRI khác nhau để kiểm tra tính tổng quát hóa và độ tin cậy.
- Minh họa tính ứng dụng thực tiễn thông qua việc triển khai một nguyên mẫu (prototype) cho phép người dùng tương tác và nhận dự đoán từ mô hình.

3.2. Cơ sở Dữ liệu và Phương pháp Chuẩn bị

3.2.1. Mô tả Bộ dữ liệu 1

- Nguồn: <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>
- Tên bộ dữ liệu: **Brain Tumor MRI Dataset**



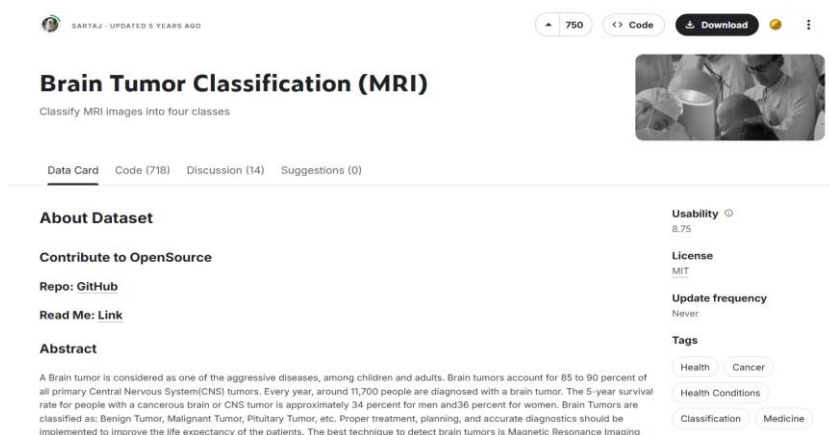
Hình 3. 1. Dataset 1

Đặc điểm chi tiết:

- Tổng số lượng: Bộ dữ liệu bao gồm 7022 ảnh MRI lát cắt sọ não. (Thông tin về số bệnh nhân nếu có).
- Phân bố lớp train:
 - Glioma: 1321 ảnh
 - Meningioma: 1339 ảnh
 - No Tumor: 1595 ảnh
 - Pituitary: 1457 ảnh
 - (Có thể trình bày bảng hoặc nhận xét về sự cân bằng lớp)
- Loại Chuỗi xung MRI: Chủ yếu là ảnh T1-weighted contrast-enhanced (T1-CE).
- Định dạng Ảnh: .jpg.

3.2.2. Mô tả Bộ dữ liệu 2

- Nguồn: <https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>
- Tên: Brain Tumor Classification (MRI)



Hình 3. 2. Dataset 2

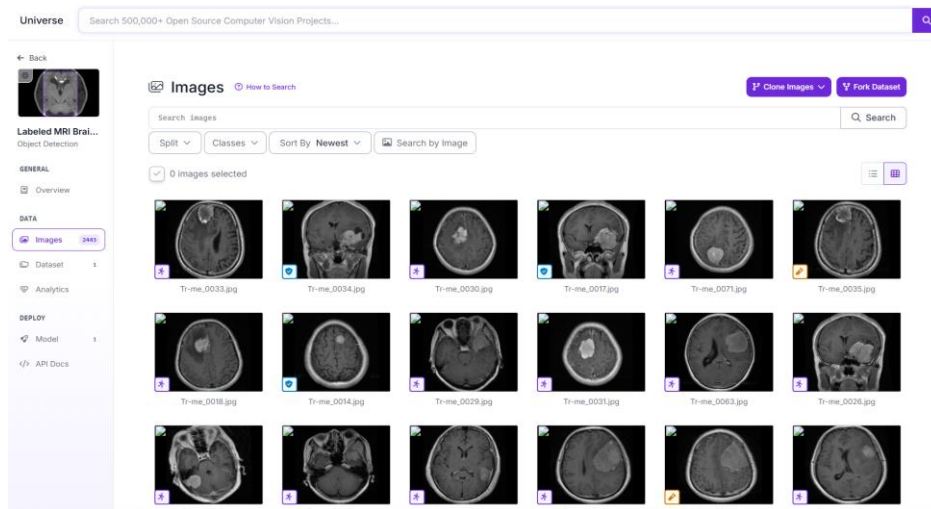
Đặc điểm chi tiết:

- Tổng số lượng: Bộ dữ liệu bao gồm 3264 ảnh MRI lát cắt sọ não. (Thông tin về số bệnh nhân nếu có).
- Phân bố lớp train:
 - Glioma: 826 ảnh
 - Meningioma: 822 ảnh
 - No Tumor: 395 ảnh
 - Pituitary: 827 ảnh
 - (Có thể trình bày bảng hoặc nhận xét về sự cân bằng lớp)
- Loại Chuỗi xung MRI: Chủ yếu là ảnh T1-weighted contrast-enhanced (T1-CE).
- Định dạng Ảnh: .jpg.

3.2.3. Mô tả Bộ dữ liệu 3

Nguồn: <https://universe.roboflow.com/ali-rostami/labeled-mri-brain-tumor-dataset/dataset/1>

Tên: Labeled MRI Brain Tumor Dataset



Hình 3. 3. Dataset 3

Đặc điểm chi tiết:

- Tổng số lượng: Bộ dữ liệu bao gồm 2443 ảnh MRI lát cắt sọ não. (Thông tin về số bệnh nhân nếu có).
- Phân bố lớp train:
 - Glioma: 564 ảnh
 - Meningioma: 358 ảnh
 - No Tumor: 335 ảnh
 - Pituitary: 438 ảnh
 - (Có thể trình bày bảng hoặc nhận xét về sự cân bằng lớp)

- Loại Chuỗi xung MRI: Chủ yếu là ảnh T1-weighted contrast-enhanced (T1-CE).
- Định dạng Ảnh: .jpg.

3.3. Phương pháp Nghiên cứu và Xây dựng Mô hình

3.3.1. Quy trình Tiền xử lý Dữ liệu

- Làm đại diện với 1 tập data cụ thể, ở đây là dataset_3, dataset1 và dataset 2 làm tương tự như vậy.
- Resize ảnh về kích thước thống nhất
- Các thư viện cần thiết:

```
# Tạo thư mục chứa dữ liệu đã tiền xử lý
import os
import cv2
import numpy as np
import shutil
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Hình 3. 4. Thư viện cần dùng

- Tạo thư mục lưu ảnh đã xử lý:

```
# Các thư mục input và output
base_dir = "D:/MRI_4/dataset_3"
processed_dir = "D:/MRI_4/dataset_3_processed"
target_size = (224, 224) # Kích thước phổ biến cho nhiều mô hình CNN
```

```
# Tạo cấu trúc thư mục output
for split in ['train', 'validation', 'test']:
    for cls in ["Glioma", "Meningioma", "No_Tumor", "Pituitary"]:
        os.makedirs(os.path.join(processed_dir, split, cls), exist_ok=True)
```

Hình 3. 5. Chia lại dữ liệu

- Hàm resize ảnh và lưu vào thư mục mới tạo:

```

# Hàm resize ảnh và lưu vào thư mục mới
def resize_and_save(source_dir, target_dir, target_size):
    print(f"Đang xử lý thư mục: {source_dir}")
    files_processed = 0

    for cls in ["Glioma", "Meningioma", "No_Tumor", "Pituitary"]:
        source_cls_dir = os.path.join(source_dir, cls)
        target_cls_dir = os.path.join(target_dir, cls)

        if not os.path.exists(source_cls_dir):
            continue

        files = [f for f in os.listdir(source_cls_dir) if f.endswith((''.jpg', '.png', '.jpeg'))]

        for file in files:
            try:
                img_path = os.path.join(source_cls_dir, file)
                img = cv2.imread(img_path)

                if img is not None:
                    # Resize ảnh
                    resized_img = cv2.resize(img, target_size)
                    # Lưu ảnh đã resize
                    cv2.imwrite(os.path.join(target_cls_dir, file), resized_img)
                    files_processed += 1
                else:
                    print(f"Lỗi: Không thể đọc ảnh {img_path}")
            except Exception as e:
                print(f"Lỗi khi xử lý {file}: {e}")

    print(f"Đã xử lý {files_processed} ảnh trong {source_dir}")

```

Hình 3. 6. Hàm *Resize* và *Save ảnh*

```

# Thực hiện resize cho cả 3 tập
for split in ['train', 'validation', 'test']:
    resize_and_save(
        os.path.join(base_dir, split),
        os.path.join(processed_dir, split),
        target_size
    )

```

Đang xử lý thư mục: [D:/MRI_4/dataset_3\train](#)
 Đã xử lý 1695 ảnh trong [D:/MRI_4/dataset_3\train](#)
 Đang xử lý thư mục: [D:/MRI_4/dataset_3\validation](#)
 Đã xử lý 502 ảnh trong [D:/MRI_4/dataset_3\validation](#)
 Đang xử lý thư mục: [D:/MRI_4/dataset_3\test](#)
 Đã xử lý 246 ảnh trong [D:/MRI_4/dataset_3\test](#)

Hình 3. 7. Chia dữ liệu

- Cân bằng dữ liệu bằng tăng cường
- Lưu lại tập val và test vào thư mục mới

```
# Tạo thư mục cho dữ liệu đã cân bằng
balanced_dir = "D:/MRI_4/dataset_3_balanced"
os.makedirs(balanced_dir, exist_ok=True)
```

```
# Copy cấu trúc từ processed_dir
for split in ['train', 'validation', 'test']:
    for cls in ["Glioma", "Meningioma", "No_Tumor", "Pituitary"]:
        os.makedirs(os.path.join(balanced_dir, split, cls), exist_ok=True)
```

```
# Copy validation và test nguyên vẹn
for split in ['validation', 'test']:
    for cls in ["Glioma", "Meningioma", "No_Tumor", "Pituitary"]:
        source = os.path.join(processed_dir, split, cls)
        dest = os.path.join(balanced_dir, split, cls)
        if os.path.exists(source):
            for file in os.listdir(source):
                if file.endswith((''.jpg', '.png', '.jpeg')):
                    shutil.copy2(os.path.join(source, file), os.path.join(dest, file))
```

Hình 3. 8. Lưu tập val và test trước khi tăng cường dữ liệu

○ Tăng cường data cho tập train

```
# Đếm số ảnh trong từng lớp của tập train
train_counts = {}
for cls in ["Glioma", "Meningioma", "No_Tumor", "Pituitary"]:
    cls_dir = os.path.join(processed_dir, 'train', cls)
    if os.path.exists(cls_dir):
        train_counts[cls] = len([f for f in os.listdir(cls_dir) if f.endswith((''.jpg', '.png', '.jpeg'))])
```

```
# Tìm lớp có nhiều ảnh nhất để làm mục tiêu cân bằng
target_count = max(train_counts.values())
print(f"Số lượng ảnh mỗi lớp sau khi cân bằng: {target_count}")
```

Số lượng ảnh mỗi lớp sau khi cân bằng: 564

Hình 3. 9. Tăng cường dữ liệu cho tập train

```
cv2.imwrite(os.path.join(target_dir, aug_filename), aug_img)

aug_count += 1
if aug_count >= augment_needed:
    break
else:
    # Đối với lớp đã có đủ số lượng ảnh, chỉ cần copy
    source_dir = os.path.join(processed_dir, 'train', cls)
    target_dir = os.path.join(balanced_dir, 'train', cls)

    for file in os.listdir(source_dir):
        if file.endswith((''.jpg', '.png', '.jpeg')):
            shutil.copy2(os.path.join(source_dir, file), os.path.join(target_dir, file))
```

Tạo thêm 206 ảnh cho lớp Meningioma
Tạo thêm 229 ảnh cho lớp No_Tumor
Tạo thêm 126 ảnh cho lớp Pituitary

Hình 3. 10. Kết quả sau khi tăng cường dữ liệu

● Chuẩn hóa giá trị pixel và tạo bộ tải dữ liệu

```

# Tạo các bộ tài dữ liệu để dùng trong huấn luyện
def create_dataset_generators(data_dir, batch_size=32):
    # Tăng cường nhẹ cho tập train khi huấn luyện (có thể bỏ nếu đã làm augmentation ở trên)
    train_datagen = ImageDataGenerator(
        rescale=1./255, # Chuẩn hóa pixel về [0, 1]
        # Thêm augmentation nhẹ khi huấn luyện (tùy chọn)
        rotation_range=10,
        width_shift_range=0.1,
        height_shift_range=0.1,
        zoom_range=0.1,
        horizontal_flip=True
    )

    # Chỉ rescale cho validation và test
    valid_test_datagen = ImageDataGenerator(rescale=1./255)

    # Tạo generators
    train_generator = train_datagen.flow_from_directory(
        os.path.join(data_dir, 'train'),
        target_size=target_size,
        batch_size=batch_size,
        class_mode='categorical',
        shuffle=True
    )

    validation_generator = valid_test_datagen.flow_from_directory(
        os.path.join(data_dir, 'validation'),
        target_size=target_size,
        batch_size=batch_size,
        class_mode='categorical',
        shuffle=False
    )

    test_generator = valid_test_datagen.flow_from_directory(
        os.path.join(data_dir, 'test'),
        target_size=target_size,
        batch_size=batch_size,
        class_mode='categorical',
        shuffle=False
    )

    return train_generator, validation_generator, test_generator

```

Hình 3. 11. Chuẩn hóa và tạo bộ tài dữ liệu

```

# Tạo generators từ dữ liệu đã cân bằng
train_gen, val_gen, test_gen = create_dataset_generators(balanced_dir, batch_size=32)

```

Found 2256 images belonging to 4 classes.
Found 502 images belonging to 4 classes.
Found 246 images belonging to 4 classes.

Hình 3. 12. Kết quả tạo generator

```

print("Class indices (mapping giữa lớp và chỉ số):")
print(train_gen.class_indices)

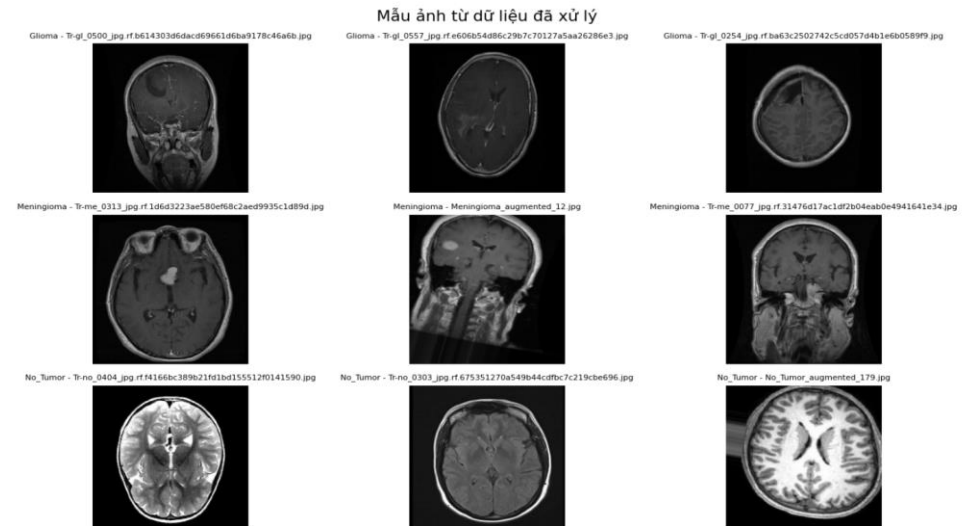
print(f"Số batch trong tập train: {len(train_gen)}")
print(f"Số batch trong tập validation: {len(val_gen)}")
print(f"Số batch trong tập test: {len(test_gen)}")

```

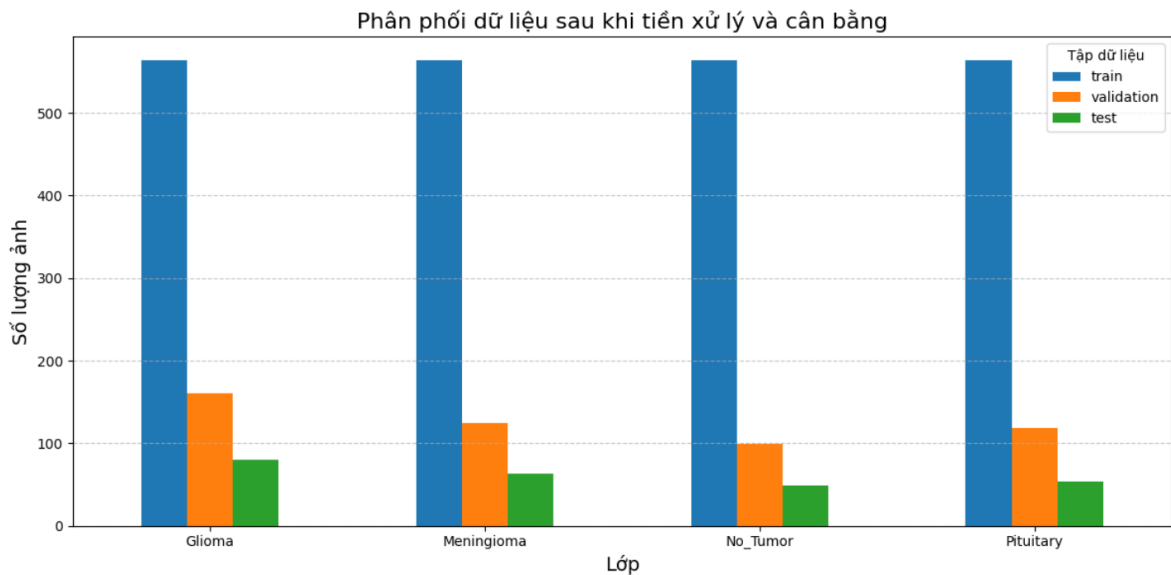
Class indices (mapping giữa lớp và chỉ số):
{'Glioma': 0, 'Meningioma': 1, 'No_Tumor': 2, 'Pituitary': 3}
Số batch trong tập train: 71
Số batch trong tập validation: 16
Số batch trong tập test: 8

Hình 3. 13. Kết quả tạo bộ tài dữ liệu

- Kiểm tra lại dữ liệu đã tiền xử lý



Hình 3. 14. Minh họa dữ liệu sau khi tăng cường



Hình 3. 15. Đồ thị tương quan dữ liệu

3.3.2. Lựa chọn và Thiết kế Kiến trúc Mô hình Học sâu

Sau quá trình tìm hiểu và tham khảo các công trình nghiên cứu liên quan (đã trình bày ở Chương 2), cũng như xem xét đặc thù của bài toán phân loại ảnh MRI não với 4 lớp (Glioma, Meningioma, No_Tumor, Pituitary), kiến trúc **ResNet50V2** kết hợp với cơ chế chú ý (**Attention Mechanism**) đã được lựa chọn để xây dựng mô hình cho ứng dụng này.

a. Lý do lựa chọn kiến trúc ResNet50V2 với Attention:

- **ResNet50V2:**

- **Hiệu quả của Học Chuyển giao (Transfer Learning):** ResNet50V2 là một kiến trúc mạng nơ-ron tích chập (CNN) sâu, đã được tiền huấn luyện trên bộ

dữ liệu ImageNet rất lớn. Việc sử dụng trọng số tiền huấn luyện này giúp mô hình khởi đầu với các đặc trưng hình ảnh tổng quát đã được học tốt, từ đó có khả năng hội tụ nhanh hơn và đạt hiệu suất cao hơn trên các bộ dữ liệu y tế thường có kích thước hạn chế.

- **Khắc phục suy giảm độ dốc (Vanishing Gradient):** Kiến trúc ResNet với các kết nối tắt (skip connections) nổi tiếng với khả năng giải quyết vấn đề suy giảm độ dốc, cho phép huấn luyện các mạng rất sâu một cách hiệu quả. Phiên bản V2 của ResNet cải tiến thứ tự của các lớp Batch Normalization và Activation, giúp cải thiện dòng chảy thông tin và hiệu suất.
- **Khả năng trích xuất đặc trưng mạnh mẽ:** Với 50 lớp, ResNet50V2 có khả năng học được các biểu diễn đặc trưng phân cấp và phức tạp từ ảnh đầu vào, rất cần thiết cho việc phân biệt các khác biệt tinh tế giữa các loại u não và mô não bình thường.
- **Cơ chế Chú ý (Attention Mechanism):**
 - **Tập trung vào vùng quan trọng:** Trong ảnh MRI não, không phải tất cả các vùng đều chứa thông tin quan trọng như nhau để phân loại khối u. Cơ chế chú ý cho phép mô hình tự động học cách "tập trung" vào các vùng ảnh nổi bật và có ý nghĩa nhất (ví dụ: vùng chứa khối u, các đặc điểm bất thường), đồng thời giảm sự ảnh hưởng của các vùng nền hoặc không liên quan.
 - **Cải thiện hiệu suất:** Bằng cách nhấn mạnh các đặc trưng quan trọng, cơ chế chú ý có thể giúp cải thiện độ chính xác và khả năng diễn giải của mô hình, đặc biệt trong các bài toán y tế nơi mà việc xác định chính xác vùng bệnh lý là rất quan trọng.
 - **Tích hợp linh hoạt:** Khối chú ý (attention block) có thể được tích hợp vào nhiều vị trí khác nhau trong kiến trúc CNN. Trong mô hình này, khối chú ý được thêm vào sau các lớp trích xuất đặc trưng của ResNet50V2 để tinh chỉnh các bản đồ đặc trưng trước khi đưa vào các lớp phân loại.

b. Chi tiết Kiến trúc Mô hình:

Mô hình được xây dựng dựa trên nền tảng ResNet50V2 với các tùy chỉnh như sau:

1. Mô hình cơ sở (Base Model):

- Sử dụng `applications.ResNet50V2` từ thư viện Keras với trọng số được tiền huấn luyện trên ImageNet (`weights='imagenet'`).
- Tham số `include_top=False` được thiết lập để loại bỏ lớp phân loại gốc của ResNet50V2 (vốn được thiết kế cho 1000 lớp của ImageNet), chỉ giữ lại các lớp tích chập để trích xuất đặc trưng.
- Đầu vào của mô hình cơ sở được định nghĩa với kích thước `input_shape=(224, 224, 3)`, phù hợp với kích thước ảnh đầu vào tiêu chuẩn của nhiều mô hình tiền huấn luyện và sau khi đã tiền xử lý (resize) ảnh MRI.
- **Đóng băng một phần trọng số (Partial Weight Freezing):** Để tận dụng tối đa kiến thức từ ImageNet và tránh overfitting trên bộ dữ liệu y tế nhỏ hơn, 100

lớp đầu tiên của `base_model` được đóng băng (`layer.trainable = False`), nghĩa là trọng số của chúng không được cập nhật trong quá trình huấn luyện ban đầu. Các lớp sâu hơn sẽ được phép cập nhật trọng số để học các đặc trưng cụ thể hơn của ảnh MRI não.

2. Khối Chú ý (Attention Block):

- Một khối chú ý tùy chỉnh (`attention_block`) được thiết kế và chèn vào sau đầu ra của `base_model`.
- Khối này bao gồm:
 - Một lớp `Conv2D` với `kernel_size=1` để giảm số kênh hoặc biến đổi đặc trưng.
 - `BatchNormalization` và `Activation('relu')`.
 - Một lớp `Conv2D` khác với `kernel_size=1` và một kênh đầu ra, theo sau bởi `BatchNormalization` và `Activation('sigmoid')`. Lớp sigmoid này tạo ra một bản đồ chú ý (attention map) với các giá trị từ 0 đến 1, thể hiện "mức độ quan trọng" của từng vị trí trong bản đồ đặc trưng.
 - Cuối cùng, lớp `Multiply` thực hiện phép nhân theo từng phần tử (element-wise multiplication) giữa bản đồ đặc trưng đầu vào (`inputs`) và bản đồ chú ý (`attention`) đã học được. Kết quả là các đặc trưng ở vùng quan trọng sẽ được giữ nguyên hoặc khuếch đại, trong khi các đặc trưng ở vùng ít quan trọng hơn sẽ bị suy giảm.
- Số lượng bộ lọc (filters) cho lớp `Conv2D` đầu tiên trong khối chú ý được đặt là 2048, phù hợp với số kênh đầu ra của `ResNet50V2` trước lớp `GlobalAveragePooling2D`.

3. Các Lớp Phân loại (Classification Head):

- Sau khối chú ý, các đặc trưng được tinh chỉnh sẽ đi qua các lớp sau để thực hiện việc phân loại:
 - `layers.GlobalAveragePooling2D()`: Giảm chiều dữ liệu không gian của bản đồ đặc trưng, chuyển mỗi bản đồ đặc trưng thành một giá trị duy nhất, giúp giảm số lượng tham số và nguy cơ overfitting.
 - `layers.Dense(1024, activation='relu')`: Lớp kết nối đầy đủ với 1024 đơn vị và hàm kích hoạt ReLU.
 - `layers.Dropout(0.5)`: Lớp Dropout với tỷ lệ 0.5 để điều chuẩn, giảm overfitting.
 - `layers.Dense(512, activation='relu')`: Một lớp kết nối đầy đủ nữa với 512 đơn vị và hàm kích hoạt ReLU.
 - `layers.Dropout(0.3)`: Lớp Dropout với tỷ lệ 0.3.

- `layers.Dense(num_classes, activation='softmax')`: Lớp đầu ra cuối cùng với `num_classes=4` (tương ứng với Glioma, Meningioma, No_Tumor, Pituitary) và hàm kích hoạt `softmax` để đưa ra xác suất cho mỗi lớp.

4. Mô hình hoàn chỉnh:

- Mô hình cuối cùng (`model`) được tạo bằng cách kết hợp đầu vào của `base_model` và lớp `outputs` đã được định nghĩa ở trên.

Tóm lại, Mô hình được xây dựng dựa trên nền tảng ResNet50V2 từ thư viện Keras (sử dụng `tensorflow.keras.applications.ResNet50V2`) với trọng số tiền huấn luyện trên ImageNet.

Kiến trúc này bao gồm ba phần chính:

- Mô hình cơ sở (Base Model): Lõi ResNet50V2 được sử dụng để trích xuất đặc trưng, với các lớp đầu được đóng băng để giữ lại kiến thức từ ImageNet (ví dụ: 100 lớp đầu tiên như đã đề cập [cite: 429]). Đầu ra của khối ResNet50V2 là một tensor đặc trưng (ví dụ, lớp `'post_relu'` trong summary chi tiết, có output shape (None, 7, 7, 2048)).
- Khối Chú ý (Attention Block): Được thêm vào sau mô hình cơ sở. Khối này bao gồm các lớp Conv2D (ví dụ, `'conv2d'` với 2048 bộ lọc và `'conv2d_1'` với 1 bộ lọc để tạo bản đồ chú ý), BatchNormalization, Activation và một lớp `'Multiply'` để áp dụng cơ chế chú ý lên các đặc trưng [cite: 431, 432, 433, 434, 435].
- Các Lớp Phân loại (Classification Head): Bao gồm lớp `'GlobalAveragePooling2D'` để giảm chiều dữ liệu, tiếp theo là các lớp `'Dense'` (ví dụ: 1024 units, 512 units) xen kẽ với các lớp `'Dropout'` (ví dụ: tỷ lệ 0.5 và 0.3) để regularization, và cuối cùng là một lớp `'Dense'` với 4 units sử dụng hàm kích hoạt softmax cho bài toán phân loại 4 lớp [cite: 439, 440, 441, 442, 443]. Mô hình hoàn chỉnh được xây dựng bằng Keras Functional API (tên mô hình trong summary là "functional").

Tổng cộng, mô hình có 30,396,425 tham số, trong đó 27,387,399 tham số có thể huấn luyện (trainable) và 3,009,026 tham số không thể huấn luyện (non-trainable).

3.3.3. Thiết lập Môi trường và Thông số Huấn luyện

Để đảm bảo tính tái lập của các thực nghiệm và tối ưu hóa quá trình huấn luyện mô hình, môi trường làm việc và các thông số huấn luyện được thiết lập một cách cẩn thận như sau:

a. Môi trường Thực nghiệm:

- **Phần cứng:**
 - Các thực nghiệm huấn luyện mô hình được thực hiện chủ yếu trên nền tảng Google Colaboratory Pro, sử dụng tài nguyên GPU hiệu năng cao như NVIDIA Tesla T4 hoặc Tesla P100 (tùy thuộc vào phiên làm việc được cấp phát).
 - Dung lượng RAM được cung cấp bởi Google Colab (thường từ 12GB đến 25GB tùy phiên bản).
- **Phần mềm:**

- **Ngôn ngữ lập trình:** Python 3.10.10.
- **Thư viện Học sâu:** TensorFlow phiên bản 2.12.0 với Keras API được sử dụng làm framework chính để xây dựng, huấn luyện và đánh giá mô hình.
- **Các thư viện phụ trợ:**
 - **NumPy:** Cho các thao tác tính toán số học và xử lý mảng.
 - **Pandas:** Để xử lý và quản lý dữ liệu dạng bảng.
 - **Matplotlib / Seaborn:** Để trực quan hóa dữ liệu và kết quả (ví dụ: vẽ đồ thị quá trình huấn luyện, ma trận nhầm lẫn).
 - **OpenCV (cv2):** Cho các tác vụ tiền xử lý ảnh.
 - **Scikit-learn:** Để tính toán các chỉ số đánh giá chi tiết.

b. Thông số và Chiến lược Huấn luyện:

- **Biên dịch Mô hình (Model Compilation):**
 - **Trình tối ưu hóa (Optimizer):** Thuật toán **Adam** (Adaptive Moment Estimation) được lựa chọn với tốc độ học (learning rate) ban đầu là **1e-4** (0.0001). Adam được chọn vì khả năng hội tụ nhanh và hiệu quả trên nhiều loại bài toán học sâu. (optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4))
- **Hàm mất mát (Loss Function):** **categorical_crossentropy** được sử dụng. Đây là hàm mất mát tiêu chuẩn cho bài toán phân loại đa lớp khi nhãn được biểu diễn dưới dạng one-hot encoding (hoặc khi đầu ra của mô hình sử dụng hàm kích hoạt **softmax**). (loss='categorical_crossentropy')
- **Chỉ số đánh giá trong quá trình huấn luyện (Metrics):** **accuracy** (độ chính xác) được theo dõi trong suốt quá trình huấn luyện và kiểm định để đánh giá hiệu năng của mô hình. (metrics=['accuracy'])
- **Các Cơ chế Gọi lại (Callbacks):** Một số cơ chế gọi lại được sử dụng để kiểm soát và tối ưu hóa quá trình huấn luyện:
 - **ModelCheckpoint:** Lưu lại trọng số của mô hình có hiệu năng tốt nhất trên tập kiểm định (dựa trên **val_accuracy**) vào tệp **best_resnet_attention_model.h5**. Chỉ có phiên bản tốt nhất được giữ lại (**save_best_only=True**). (ModelCheckpoint('best_resnet_attention_model.h5', save_best_only=True, monitor='val_accuracy'))
 - **EarlyStopping:** Dừng quá trình huấn luyện sớm nếu hiệu năng trên tập kiểm định không cải thiện (cụ thể là **val_accuracy** không tăng) sau một số lượng **patience** epochs nhất định (ở đây là 20 epochs). Cơ chế này giúp tránh lãng phí thời gian huấn luyện và giảm nguy cơ overfitting. Trọng số của mô hình tốt nhất sẽ được khôi phục sau khi dừng (**restore_best_weights=True**). (EarlyStopping(patience=20, restore_best_weights=True))

- **ReduceLROnPlateau:** Tự động giảm tốc độ học khi hiệu năng trên tập kiểm định (**val_accuracy**) không còn cải thiện. Nếu **val_accuracy** không tăng sau **patience=7** epochs, tốc độ học sẽ được giảm đi một hệ số **factor=0.2**. Tốc độ học sẽ không giảm xuống dưới một ngưỡng tối thiểu **min_lr=1e-6**. Kỹ thuật này giúp mô hình tiếp cận gần hơn đến điểm tối ưu cục bộ tốt.
(ReduceLROnPlateau(factor=0.2, patience=7, min_lr=1e-6))
- **Số lượng Kỷ nguyên (Epochs):** Mô hình được thiết lập để huấn luyện trong tối đa **50** epochs. Tuy nhiên, với việc sử dụng **EarlyStopping**, quá trình huấn luyện có thể kết thúc sớm hơn nếu đạt điều kiện dừng. (epochs=50)
- **Dữ liệu Huấn luyện và Kiểm định:** Dữ liệu được cung cấp cho mô hình thông qua các trình tạo dữ liệu (data generators) là **train_gen** (cho tập huấn luyện) và **val_gen** (cho tập kiểm định). Các trình tạo này thường thực hiện việc tải dữ liệu theo lô (batch) và áp dụng các phép tăng cường dữ liệu (data augmentation) cho tập huấn luyện một cách "on-the-fly".

3.3.4. Các Chỉ số Đánh giá Hiệu năng Mô hình

Để đánh giá khách quan hiệu năng của mô hình phân loại 4 lớp u não (Glioma, Meningioma, No_Tumor, Pituitary), các chỉ số sau được sử dụng trên tập kiểm tra (test set) của mỗi bộ dữ liệu:

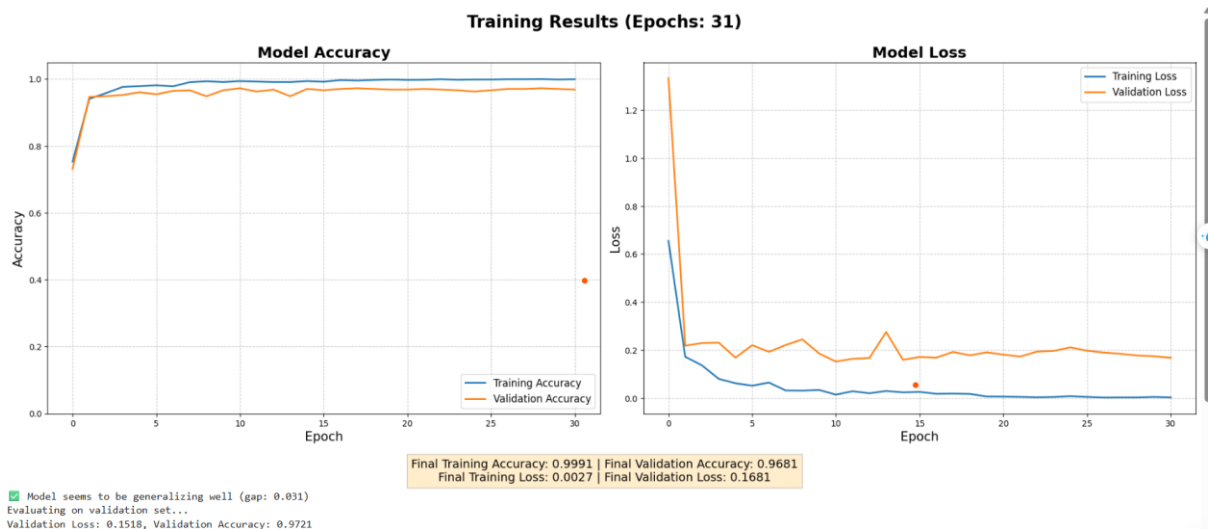
1. Độ chính xác (Accuracy): Tỷ lệ tổng số mẫu được phân loại đúng, cung cấp cái nhìn tổng quan về hiệu suất.
$$\text{Accuracy} = \frac{\text{Số mẫu dự đoán đúng}}{\text{Tổng số mẫu}}$$
2. Độ chuẩn (Precision), Độ nhạy (Recall/Sensitivity), và F1-score: Được tính cho từng lớp riêng biệt để đánh giá chi tiết hơn.
 - Precision: Tỷ lệ mẫu thực sự thuộc lớp X trong số các mẫu được dự đoán là lớp X.
 - Recall: Tỷ lệ mẫu lớp X được mô hình phát hiện đúng trên tổng số mẫu thực sự thuộc lớp X.
 - F1-score: Trung bình điều hòa của Precision và Recall, hữu ích khi cần cân bằng cả hai. Các giá trị trung bình có trọng số (weighted-average) của Precision, Recall, F1-score cũng sẽ được báo cáo để tổng hợp hiệu năng trên tất cả các lớp, đặc biệt khi có sự mất cân bằng dữ liệu.
3. Ma trận Nhầm lẫn (Confusion Matrix): Một bảng trực quan hóa số lượng dự đoán đúng và sai giữa các lớp, giúp xác định các loại lỗi mà mô hình thường mắc phải.

Các chỉ số này cùng nhau cung cấp một đánh giá toàn diện về khả năng phân loại của mô hình, từ độ chính xác chung đến hiệu suất chi tiết trên từng loại u và khả năng xử lý các trường hợp mất cân bằng lớp.

3.4. Kết quả Thực nghiệm và Phân tích Chi tiết

3.4.1. Kết quả trên tập dataset_3

3.4.1.1. Quá trình Huấn luyện

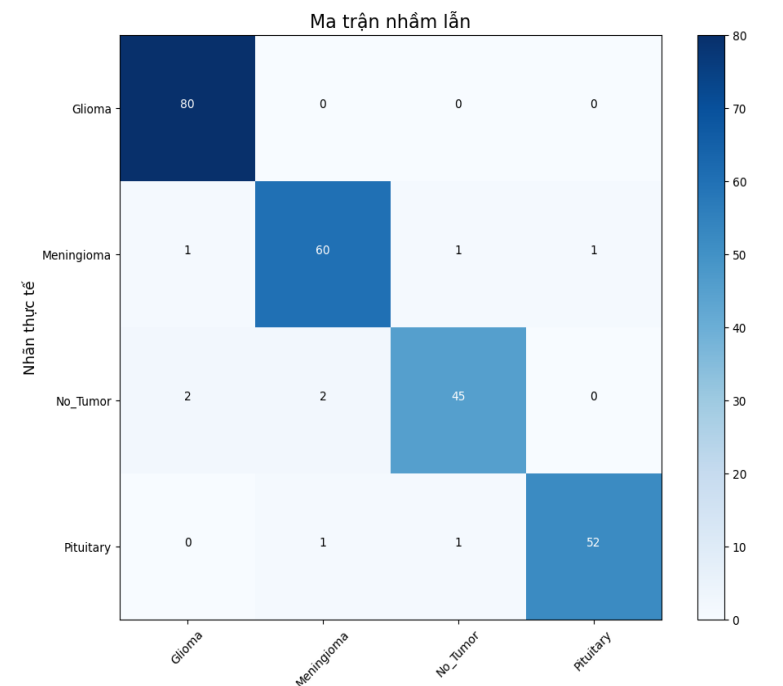


Hình 3. 16. Quá trình huấn luyện

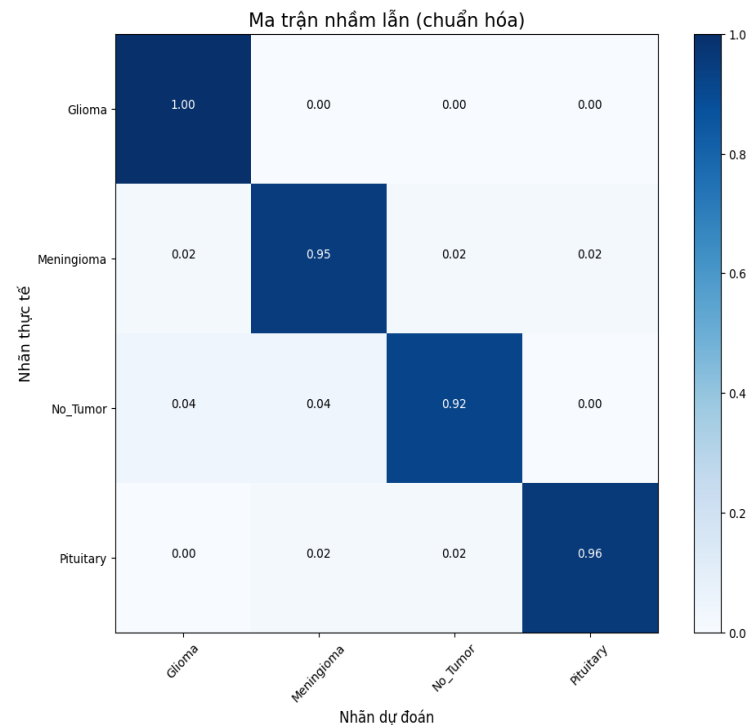
3.4.1.2. Kết quả định lượng

Classification Report (Test Set):

	precision	recall	f1-score	support
Glioma	0.96	1.00	0.98	80
Meningioma	0.95	0.95	0.95	63
No_Tumor	0.96	0.92	0.94	49
Pituitary	0.98	0.96	0.97	54
accuracy			0.96	246
macro avg	0.96	0.96	0.96	246
weighted avg	0.96	0.96	0.96	246



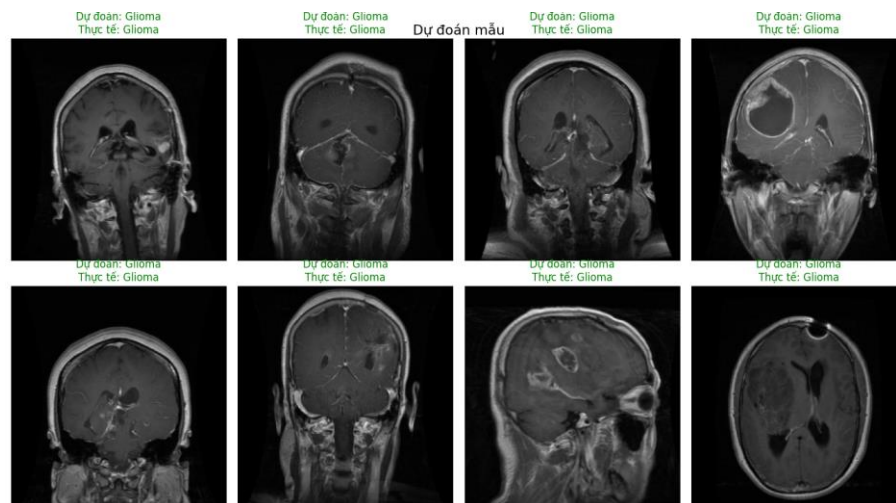
Hình 3. 17. Ma trận nhầm lẫn không chuẩn hóa



Hình 3. 18. Ma trận nhầm lẫn được chuẩn hóa

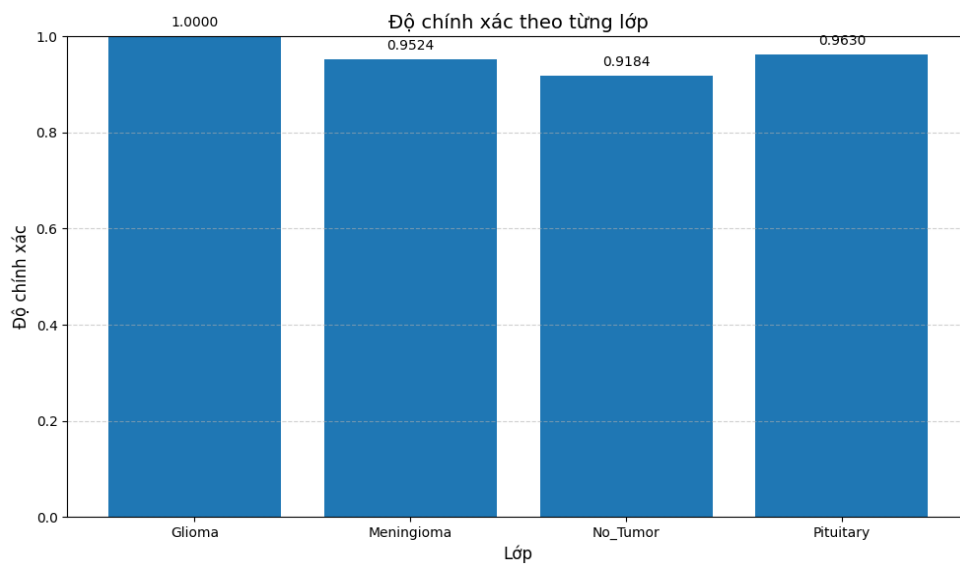
Hình 3.3 Ma trận nhầm lẫn được chuẩn hóa

3.4.1.3 Kết quả định tính



Hình 3. 19. Kết quả dự đoán

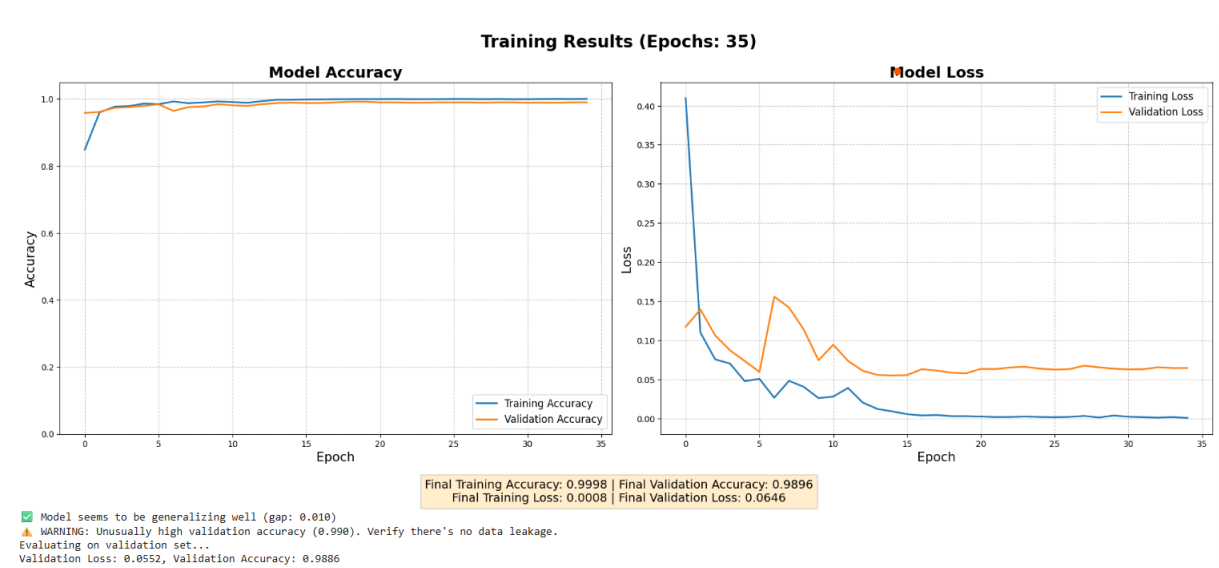
Hình 3.4. Ảnh kết quả dự đoán 1 số mẫu



Hình 3. 20. Hiệu suất nhận diện của mô hình đối với từng lớp trên tập kiểm tra

3.4.2 Kết quả trên tập dataset_1

3.4.2.1. Quá trình Huấn luyện

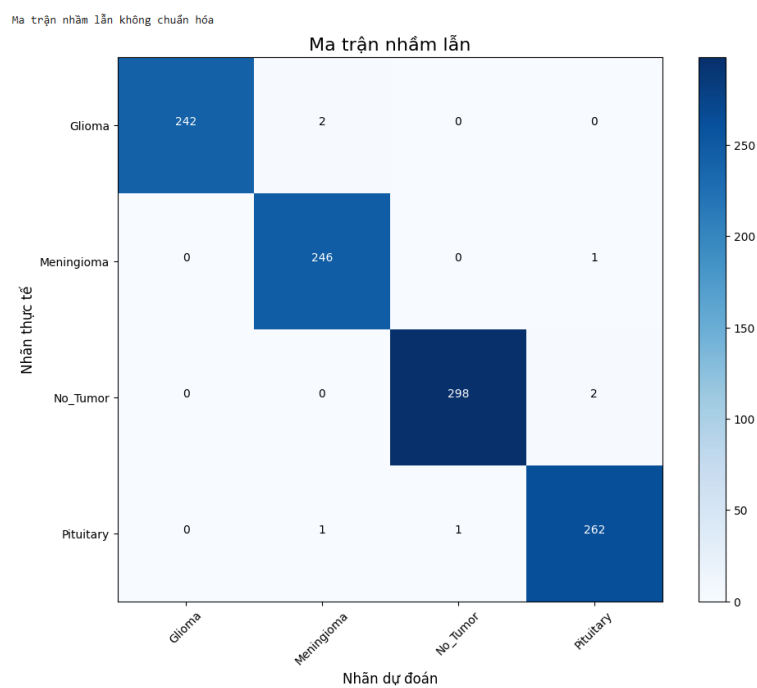


Hình 3. 21. Quá trình train dataset 1

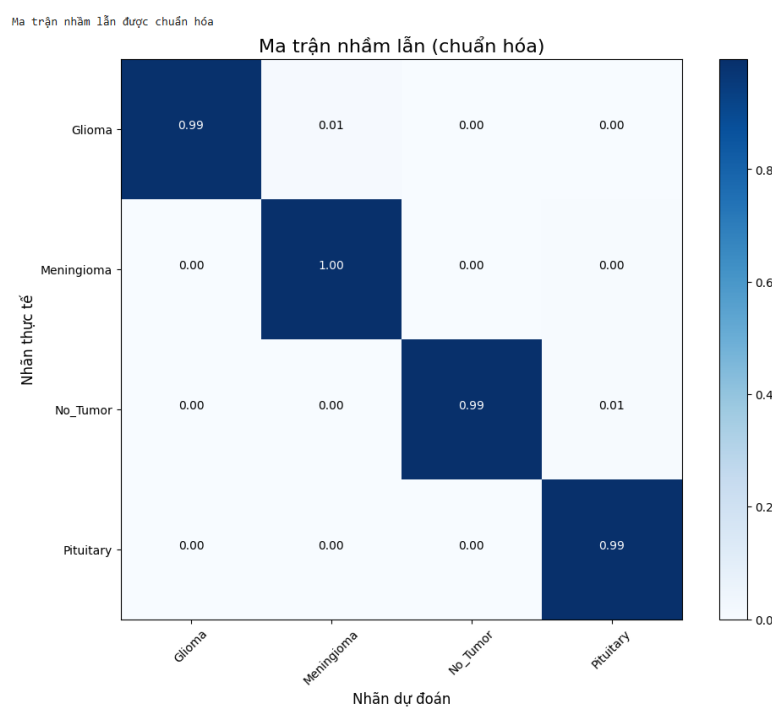
3.4.2.2. Kết quả định lượng

Classification Report (Test Set):

	precision	recall	f1-score	support
Glioma	1.00	0.99	0.99	244
Meningioma	0.99	0.99	0.99	247
No_Tumor	1.00	0.99	0.99	300
Pituitary	0.98	0.99	0.99	264
accuracy			0.99	1055
macro avg	0.99	0.99	0.99	1055
weighted avg	0.99	0.99	0.99	1055

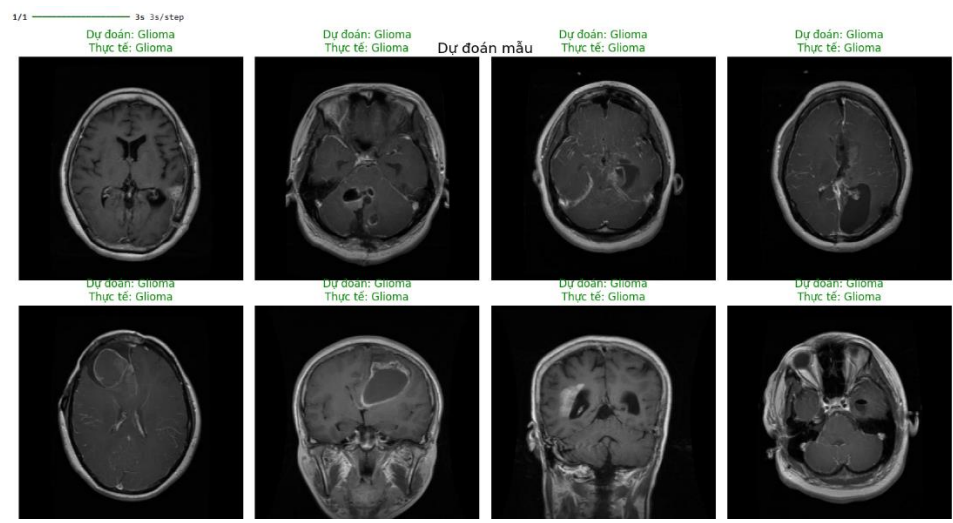


Hình 3. 22. Ma trận nhầm lẫn không chuẩn hóa

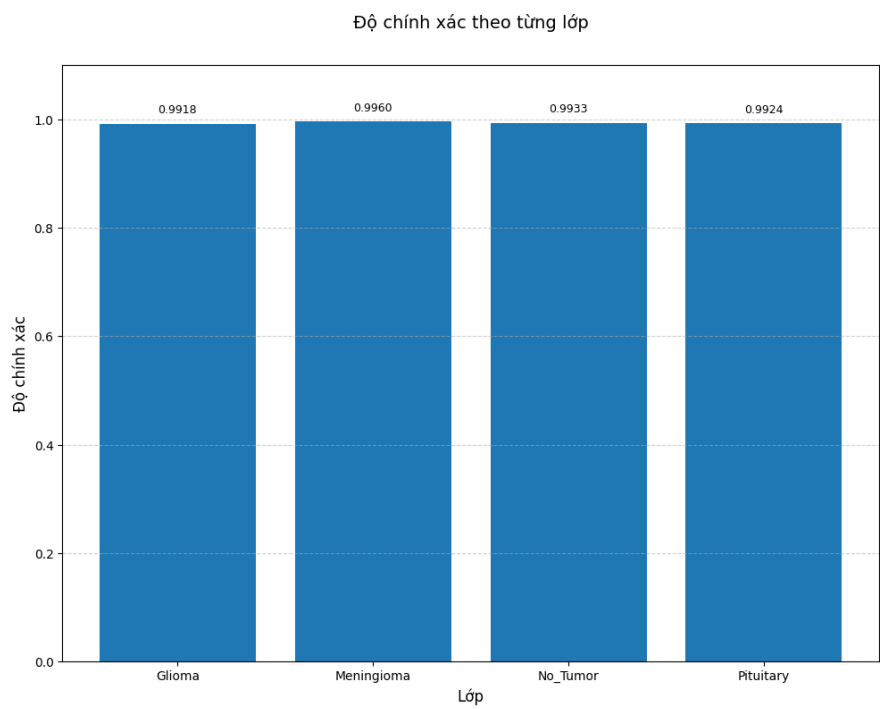


Hình 3. 23. Ma trận nhầm lẫn được chuẩn hóa

3.4.2.3 Kết quả định tính



Hình 3. 24. Kết quả 1 số dự đoán

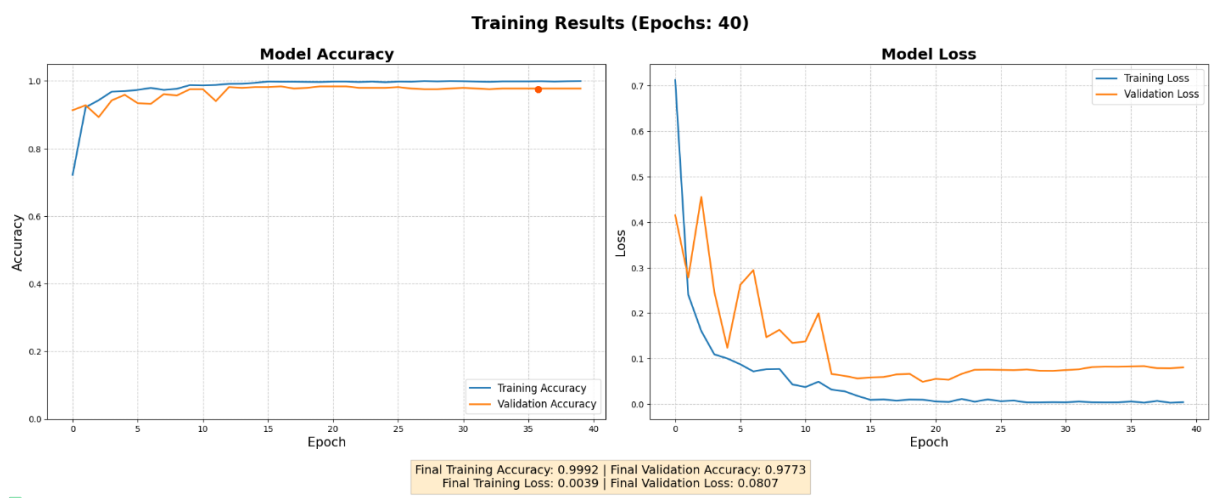


Hình 3. 25. Độ chính xác theo từng lớp dataset 1

Hình 3.9: Hiệu suất nhận diện của mô hình đối với từng lớp trên tập kiểm tra

3.4.3 Kết quả trên tập dataset_2

3.4.3.1. Quá trình Huấn luyện



Hình 3. 26. Quá trình train dataset 2

Validation Loss: 0.0487, Validation Accuracy: 0.9835

3.4.3.2. Kết quả định lượng

Classification Report (Test Set):

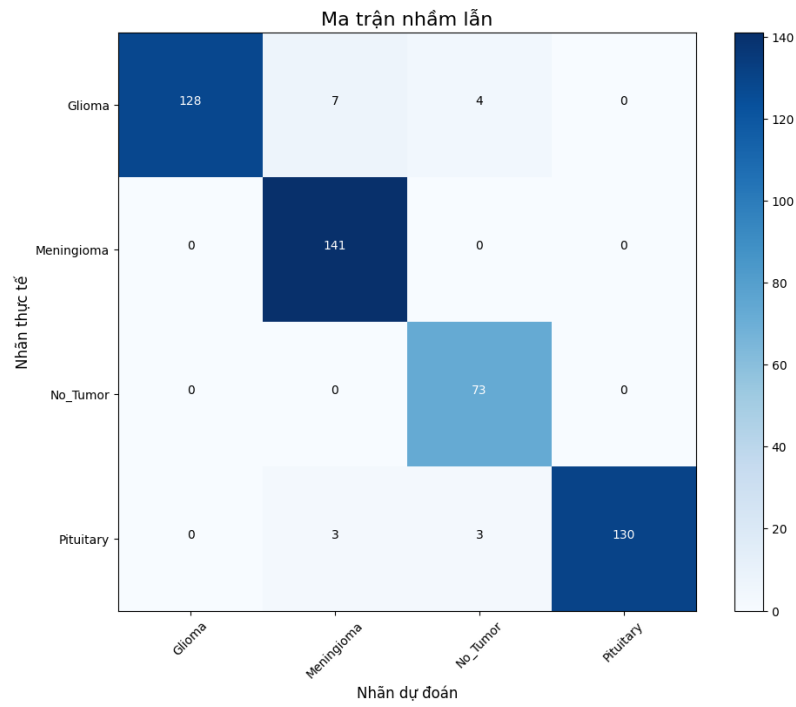
Độ chính xác trên tập test: 0.9652

Loss trên tập test: 0.2680

	precision	recall	f1-score	support
Glioma	1.00	0.92	0.96	139
Meningioma	0.93	1.00	0.97	141
No_Tumor	0.91	1.00	0.95	73
Pituitary	1.00	0.96	0.98	136
accuracy			0.97	489
macro avg	0.96	0.97	0.96	489

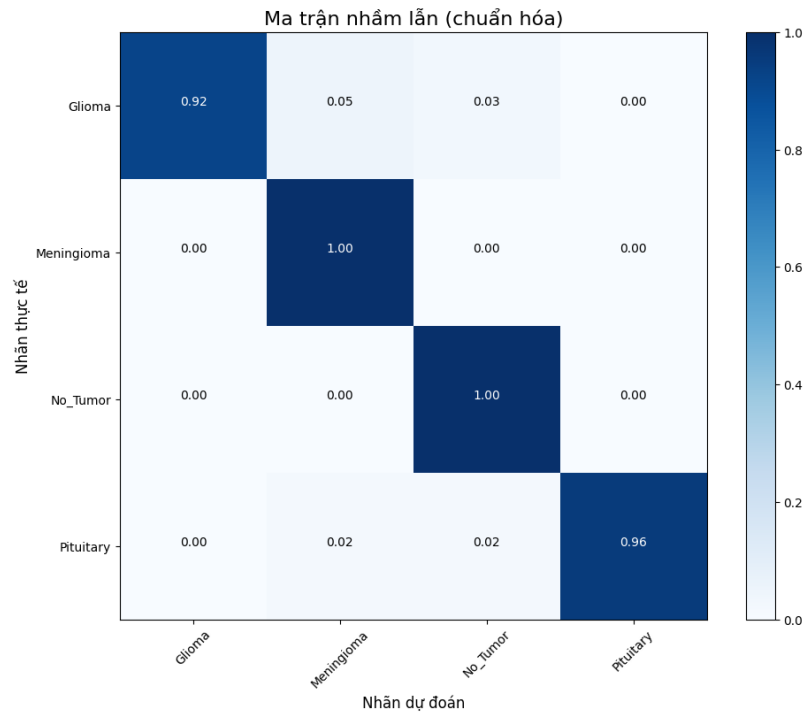
weighted avg	0.97	0.97	0.97	489
---------------------	------	------	------	-----

Ma trận nhầm lẫn không chuẩn hóa



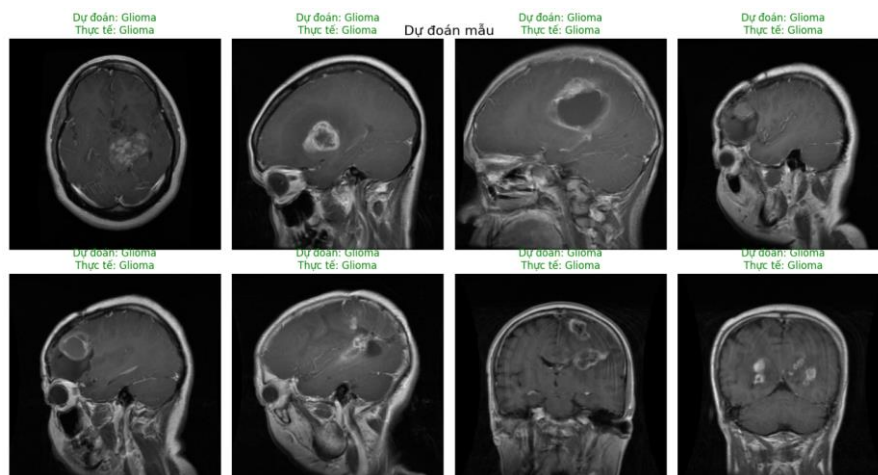
Hình 3. 27. Ma trận nhầm lẫn không chuẩn hóa

Ma trận nhầm lẫn được chuẩn hóa

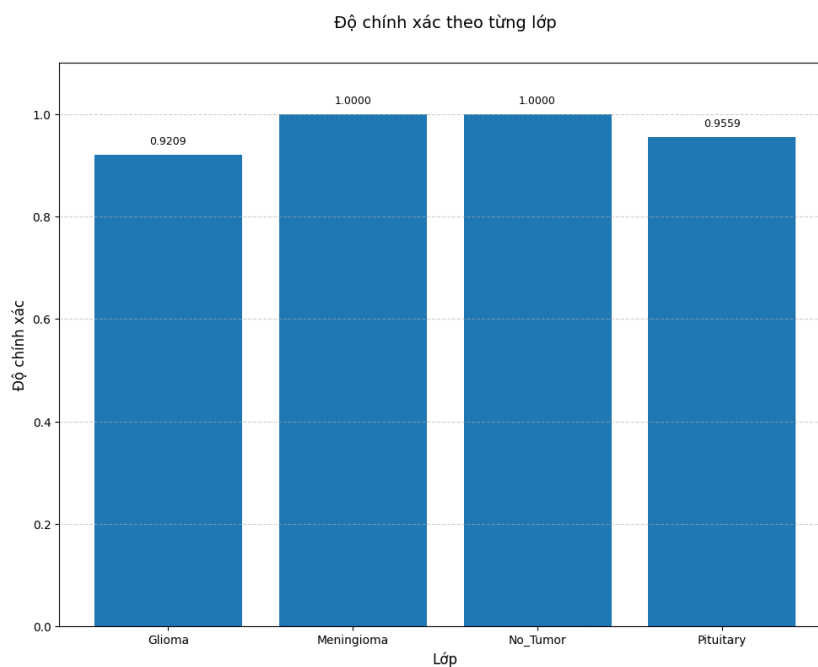


Hình 3. 28. Ma trận nhầm lẫn đã chuẩn hóa

3.4.3.3 Kết quả định tính



Hình 3. 29. Kết quả 1 số dự đoán



Hình 3. 30. Độ chính xác theo từng lớp

Hình 3.9: Hiệu suất nhận diện của mô hình đối với từng lớp trên tập kiểm tra

3.5. Triển khai Ứng dụng Minh họa

Để minh họa tính ứng dụng thực tiễn của mô hình học sâu đã huấn luyện, một ứng dụng web đơn giản đã được xây dựng, cho phép người dùng tải lên ảnh MRI não và nhận kết quả phân loại loại u não (Glioma, Meningioma, No_Tumor, Pituitary) cùng với các thông tin trực quan hóa.

3.5.1. Lựa chọn Nền tảng và Công nghệ Triển khai

- **Nền tảng triển khai:** Ứng dụng được xây dựng dưới dạng một ứng dụng web tương tác.
- **Công nghệ chính:**
 - **Streamlit:** Một framework Python mã nguồn mở, cho phép nhanh chóng xây dựng và chia sẻ các ứng dụng web đẹp mắt cho các dự án khoa học dữ liệu và học máy mà không cần kiến thức sâu về phát triển web frontend. Streamlit được lựa chọn vì tính đơn giản, dễ sử dụng và khả năng tích hợp tốt với các thư viện Python phổ biến.
 - **TensorFlow/Keras:** Được sử dụng để tải mô hình đã huấn luyện ([best_resnet_attention_model.h5](#)) và thực hiện dự đoán.
 - **OpenCV (cv2) và Pillow (PIL):** Dùng cho các tác vụ xử lý và thao tác với hình ảnh đầu vào.
 - **NumPy:** Cho các tính toán mảng hiệu quả.
 - **Matplotlib**

3.5.2. Kiến trúc và Luồng hoạt động của Ứng dụng

Ứng dụng được thiết kế với giao diện người dùng đơn giản, chia thành hai cột chính: một cột để người dùng tải ảnh lên và một cột để hiển thị kết quả.

- **Luồng hoạt động chính:**
 1. **Tải mô hình:** Khi ứng dụng khởi chạy, mô hình ResNet50V2 với Attention đã được huấn luyện ([best_resnet_attention_model.h5](#)) được tải vào bộ nhớ. Cơ chế [@st.cache_resource](#) của Streamlit được sử dụng để đảm bảo mô hình chỉ được tải một lần, giúp tối ưu hiệu suất.
 2. **Người dùng tải ảnh:** Người dùng sử dụng nút "Chọn ảnh MRI" để tải lên một tệp ảnh (định dạng .jpg, .jpeg, .png) từ máy tính cá nhân.
 3. **Hiển thị ảnh gốc:** Ảnh được tải lên sẽ được hiển thị trên giao diện để người dùng xác nhận.
 4. **Xử lý yêu cầu phân loại:** Khi người dùng nhấp vào nút "Phân loại ảnh":
 - Ảnh đầu vào được tiền xử lý (chuyển đổi định dạng nếu cần, đảm bảo ảnh RGB, resize về kích thước 224x224, chuẩn hóa giá trị pixel về khoảng [0,1]) thông qua hàm [preprocess_image](#).
 - Ảnh đã tiền xử lý được đưa vào mô hình để dự đoán thông qua hàm [predict](#). Hàm này cũng tính toán thời gian xử lý.
 - Kết quả dự đoán (lớp có xác suất cao nhất) và xác suất của từng lớp được trích xuất.
 5. **Hiển thị kết quả:**
 - Lớp dự đoán (ví dụ: "Glioma") và thời gian xử lý được hiển thị rõ ràng.

- Xác suất của cả bốn lớp (Glioma, Meningioma, No_Tumor, Pituitary) được trình bày dưới dạng thanh tiến trình (progress bar) và giá trị phần trăm cụ thể, giúp người dùng có cái nhìn chi tiết hơn về mức độ "tự tin" của mô hình cho từng lớp.

6. **(Tùy chọn) Hiển thị Grad-CAM:** Nếu người dùng chọn tùy chọn "Hiển thị bản đồ nhiệt (CAM)", ứng dụng sẽ tính toán và hiển thị bản đồ kích hoạt lớp có trọng số theo gradient (Grad-CAM) thông qua hàm `get_grad_cam`. Bản đồ nhiệt này được phủ lên ảnh gốc, giúp trực quan hóa các vùng trong ảnh mà mô hình tập trung chú ý nhất khi đưa ra quyết định phân loại. Cơ chế này cố gắng tìm lớp `attention_block` hoặc lớp tích chập cuối cùng để tạo CAM.
7. **Thông tin bổ sung:** Ứng dụng cũng cung cấp một mục "Thông tin về các loại khối u" để người dùng có thể tham khảo nhanh về đặc điểm của từng loại.

3.5.3. Hướng dẫn Sử dụng và Minh họa Giao diện

Ứng dụng có giao diện trực quan và dễ sử dụng:

1. Truy cập ứng dụng:

Cách khởi chạy ứng dụng: Tại thư mục chứa code gõ ở terminal: **streamlit run app.py**

```
PS D:\MRI_4> streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.12:8501

2025-05-15 17:01:20.126343: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-05-15 17:01:26.342311: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-05-15 17:01:46.302134: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
```

Hình 3. 31. Khởi chạy ứng dụng

2. **Tải ảnh lên:** Tại cột bên trái, nhấp vào nút "Chọn ảnh MRI" và chọn một tệp ảnh MRI não từ máy tính của bạn. Ảnh bạn chọn sẽ được hiển thị ngay ở cột kết quả.



Hệ Thống Phân Loại Khối U Não MRI

Hệ thống này phân loại ảnh MRI não thành 4 nhóm: Glioma, Meningioma, No_Tumor (Không có u) và Pituitary (U tuyến yên).

Tải lên ảnh MRI để nhận kết quả phân loại.

Tải lên ảnh MRI não

Kết quả phân loại

Chọn ảnh MRI

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

Hình 3. 32. Tải ảnh lên

3. **(Tùy chọn) Chọn hiển thị Grad-CAM:** Trong mục "Tùy chọn nâng cao", người dùng có thể tick vào ô "Hiển thị bản đồ nhiệt (CAM)" nếu muốn xem vùng chú ý của mô hình.

Tải lên ảnh MRI não

Kết quả phân loại

Chọn ảnh MRI

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

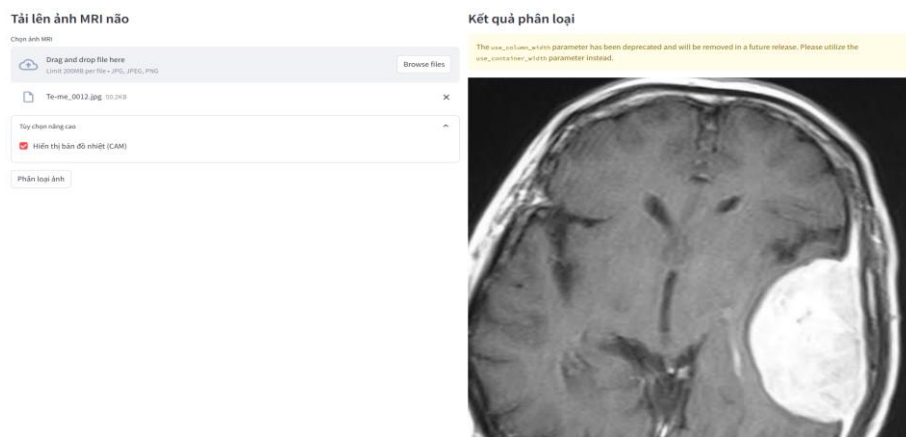
Browse files

Tùy chọn nâng cao

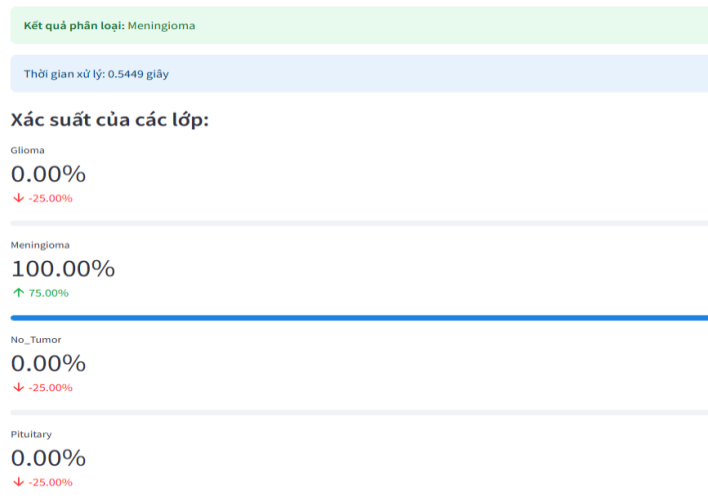
☒ Hiển thị bản đồ nhiệt (CAM)

Hình 3. 33. Tùy chọn nâng cao

4. **Thực hiện phân loại:** Nhấp vào nút "Phân loại ảnh".
5. **Xem kết quả:** Tại cột bên phải, kết quả phân loại sẽ được hiển thị, bao gồm:
 - Ảnh gốc đã tải lên.
 - Lớp dự đoán (ví dụ: "Kết quả phân loại: Glioma").
 - Thời gian xử lý.
 - Xác suất chi tiết cho từng lớp dưới dạng thanh tiến trình và phần trăm.
 - Nếu tùy chọn Grad-CAM được bật, bản đồ nhiệt sẽ được hiển thị bên dưới các thông tin trên.



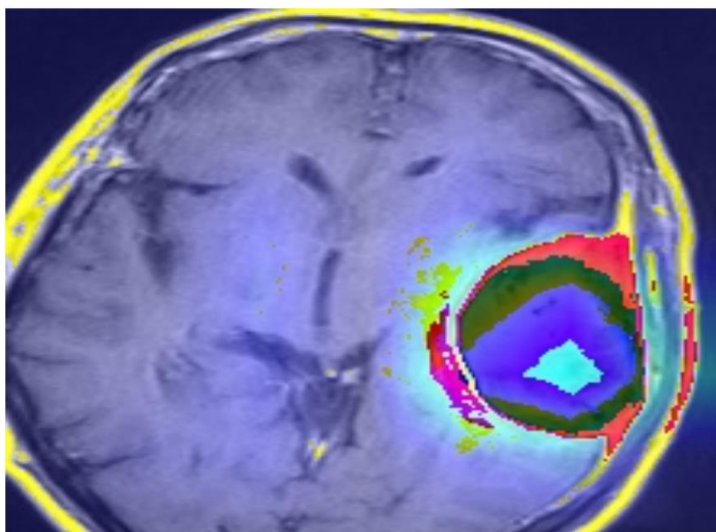
Hình 3. 34. Minh họa khi ảnh được tải lên



Hình 3. 35. Minh họa kết quả dự đoán

Vùng chú ý của mô hình (Grad-CAM):

The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.



Hình 3. 36. Minh họa khi chọn tùy chọn nâng cao

6. **Tham khảo thông tin:** Người dùng có thể mở rộng mục "Thông tin về các loại khối u" để đọc thêm.

Thông tin về các loại khối u

Các loại khối u não:

1. Glioma:

- Là khối u phát sinh từ tế bào hỗ trợ (tế bào đệm) trong não
- Có thể là lành tính hoặc ác tính
- Bao gồm các loại như u sao bào, u thần kinh đệm và u nguyên bào đệm

2. Meningioma:

- Phát triển từ màng não (màng bao quanh não và tủy sống)
- Thường là lành tính và phát triển chậm
- Có thể gây áp lực lên não và các dây thần kinh

3. Pituitary (U tuyến yên):

- Phát triển ở tuyến yên, một tuyến nhỏ nằm ở đáy não
- Thường là lành tính
- Có thể ảnh hưởng đến sự sản xuất hormone và gây ra các vấn đề về nội tiết

Phương pháp phân loại:

Mô hình này sử dụng mạng nơ-ron tích chập (CNN) với kiến trúc ResNet50V2 kết hợp cơ chế Attention để phân loại chính xác các loại khối u não từ ảnh MRI.

Hình 3. 37. Phần thông tin đọc thêm

3.6. Kết luận cho Chương 3 và Đề xuất Hướng Phát triển cho Ứng dụng

Chương này đã trình bày chi tiết quá trình xây dựng, thử nghiệm và triển khai một ứng dụng học sâu nhằm nhận diện bốn loại u não (Glioma, Meningioma, No_Tumor, Pituitary) từ ảnh chụp cộng hưởng từ (MRI).

3.6.1. Tóm tắt những Kết quả Chính của Chương

Ứng dụng được phát triển dựa trên kiến trúc ResNet50V2 kết hợp cơ chế chú ý (Attention), sử dụng kỹ thuật học chuyển giao với trọng số tiền huấn luyện từ ImageNet. Mô hình đã được huấn luyện và đánh giá trên ba bộ dữ liệu ảnh MRI khác nhau.

Kết quả thực nghiệm cho thấy mô hình đạt hiệu suất khả quan trên cả ba bộ dữ liệu, với độ chính xác tổng thể trên tập kiểm tra dao động trong khoảng 95%-97%. Quá trình huấn luyện diễn ra ổn định, mô hình có khả năng tổng quát hóa tốt và không có dấu hiệu overfitting nghiêm trọng. Cuối cùng, một ứng dụng web minh họa sử dụng Streamlit đã được triển khai thành công, cho phép người dùng tải ảnh MRI lên và nhận kết quả phân loại kèm theo bản đồ nhiệt Grad-CAM để tăng tính giải thích.

3.6.2. Đánh giá Chung về Giải pháp và Mô hình đã Xây dựng

Giải pháp sử dụng ResNet50V2 với cơ chế chú ý và học chuyển giao đã chứng tỏ là một hướng tiếp cận hiệu quả cho bài toán phân loại u não đa lớp. Mô hình không chỉ đạt được độ chính xác cao mà còn cho thấy sự ổn định nhất định khi thử nghiệm trên các nguồn dữ liệu khác nhau. Việc tích hợp Grad-CAM vào ứng dụng triển khai là một điểm cộng, giúp tăng cường sự tin cậy và khả năng diễn giải của hệ thống. Tuy nhiên, hiệu suất có thể bị ảnh hưởng bởi chất lượng và đặc điểm của từng bộ dữ liệu cụ thể, cũng như sự mất cân bằng tiềm ẩn giữa các lớp.

3.6.3. Đề xuất các Hướng Phát triển Tiếp theo cho Ứng dụng và Mô hình

Để tiếp tục nâng cao hiệu quả và tính ứng dụng của hệ thống, một số hướng phát triển có thể được xem xét trong tương lai:

- **Cải thiện Mô hình:**
 - Thử nghiệm với các kiến trúc học sâu tiên tiến hơn hoặc các biến thể của cơ chế chú ý.
 - Tinh chỉnh sâu hơn các siêu tham số và chiến lược huấn luyện.
 - Áp dụng các kỹ thuật xử lý mất cân bằng lớp một cách mạnh mẽ hơn nếu cần.
 - Nghiên cứu tích hợp thông tin từ nhiều chuỗi xung MRI khác nhau (multi-modal learning) để tăng cường đặc trưng đầu vào.
- **Mở rộng và Đa dạng hóa Dữ liệu:**
 - Thu thập hoặc sử dụng thêm nhiều bộ dữ liệu từ các nguồn khác nhau, đặc biệt là dữ liệu từ các cơ sở y tế thực tế để tăng tính đa dạng và khả năng tổng quát hóa.

- **Nâng cao Ứng dụng Triển khai:**

- Cải thiện giao diện người dùng, tối ưu hóa tốc độ xử lý.
- Tích hợp thêm các công cụ trực quan hóa và phân tích kết quả chi tiết hơn.
- Phát triển khả năng xử lý ảnh 3D MRI (nếu mô hình được huấn luyện trên dữ liệu 3D) thay vì chỉ từng lát cắt 2D.
- Xem xét các phương án triển khai trên các nền tảng khác.

- **Tăng cường Tính Giải thích và Tin cậy:**

- Nghiên cứu sâu hơn và áp dụng các phương pháp XAI (Explainable AI) khác để cung cấp giải thích rõ ràng hơn về quyết định của mô hình.

Những cải tiến này hứa hẹn sẽ làm tăng độ chính xác, độ tin cậy và khả năng ứng dụng thực tiễn của hệ thống trong việc hỗ trợ chẩn đoán u não.

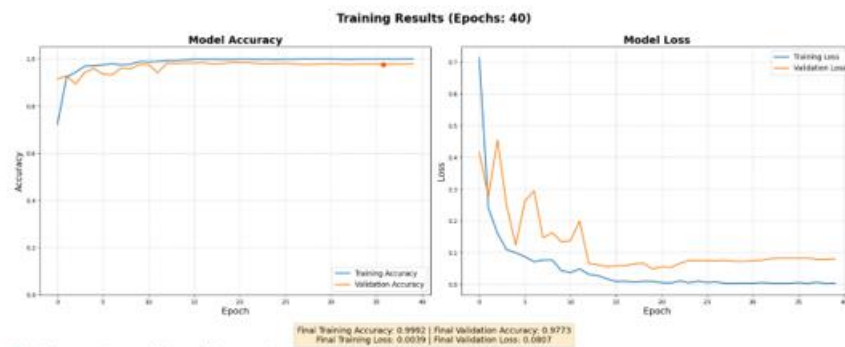
Hết

Nội dung đã chỉnh sửa

- Đã thêm phần huấn luyện cho dataset 2

3.4.3 Kết quả trên tập dataset_2

3.4.3.1. Quá trình Huấn luyện



Hình 3. 26. Quá trình train dataset 2

Validation Loss: 0.0487, Validation Accuracy: 0.9835

3.4.3.2. Kết quả định lượng

Classification Report (Test Set):

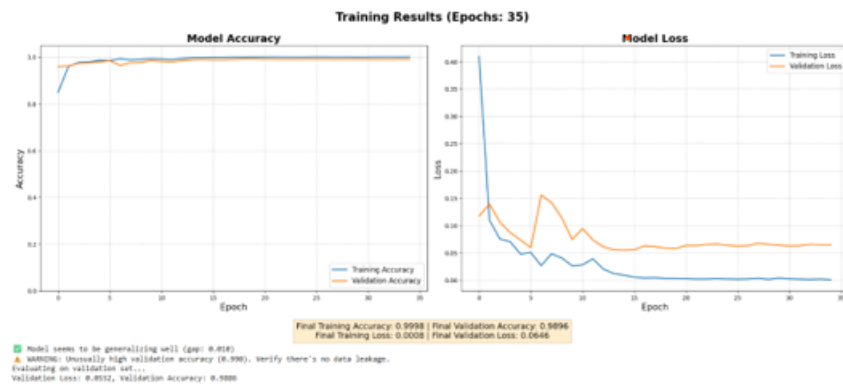
Độ chính xác trên tập test: 0.9652

Loss trên tập test: 0.2680

- Đã thêm phần huấn luyện cho dataset 1:

3.4.2 Kết quả trên tập dataset_1

3.4.2.1. Quá trình Huấn luyện



Hình 3. 21. Quá trình train dataset 1

3.4.2.2. Kết quả định lượng

Classification Report (Test Set):

	precision	recall	f1-score	support
Glioma	1.00	0.99	0.99	244
Meningioma	0.99	0.99	0.99	247
No_Tumor	1.00	0.99	0.99	300
Pituitary	0.98	0.99	0.99	264
accuracy			0.99	1055
macro avg	0.99	0.99	0.99	1055

- Đã thêm chỉ mục, chú thích cho các ảnh



Hình 3. 1. Dataset 1

Cách khởi chạy ứng dụng: Tại thư mục chứa code gõ ở terminal: **streamlit run app.py**

```
PS C:\VRG_4> streamlit run app.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.1.12:8501
2023-05-15 17:02:20.126343: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2023-05-15 17:02:26.342311: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2023-05-15 17:02:46.382134: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
```

Hình 3. 31. Khởi chạy ứng dụng

2. **Tải ảnh lên:** Tại cột bên trái, nhấp vào nút "Chọn ảnh MRI" và chọn một tệp ảnh MRI não từ máy tính của bạn. Ảnh bạn chọn sẽ được hiển thị ngay ở cột kết quả.



Hình 3. 32. Tải ảnh lên

- Đã sửa phần tổng kết kiến trúc mô hình

Pituitary) và hàm kích hoạt **softmax** để đưa ra xác suất cho mỗi lớp.

4. Mô hình hoàn chỉnh:

- o Mô hình cuối cùng (**model**) được tạo bằng cách kết hợp đầu vào của **base_model** và lớp **outputs** đã được định nghĩa ở trên.

Tóm lại, Mô hình được xây dựng dựa trên nền tảng ResNet50V2 từ thư viện Keras (sử dụng `tensorflow.keras.applications.ResNet50V2`) với trọng số tiền huấn luyện trên ImageNet. Kiến trúc này bao gồm ba phần chính:

- Mô hình cơ sở (Base Model): Lõi ResNet50V2 được sử dụng để trích xuất đặc trưng, với các lớp đầu được đóng băng để giữ lại kiến thức từ ImageNet (ví dụ: 100 lớp đầu tiên như đã đề cập [cite: 429]). Đầu ra của khối ResNet50V2 là một tensor đặc trưng (ví dụ, lớp `post_relu` trong summary chi tiết, có output shape (None, 7, 7, 2048)).
- Khối Chú ý (Attention Block): Được thêm vào sau mô hình cơ sở. Khối này bao gồm các lớp Conv2D (ví dụ, `conv2d` với 2048 bộ lọc và `conv2d_1` với 1 bộ lọc để tạo bản đồ chú ý), BatchNormalization, Activation và một lớp `Multiply` để áp dụng cơ chế chú ý lên các đặc trưng [cite: 431, 432, 433, 434, 435].
- Các Lớp Phân loại (Classification Head): Bao gồm lớp `GlobalAveragePooling2D` để giảm chiều dữ liệu, tiếp theo là các lớp `Dense` (ví dụ: 1024 units, 512 units) xen kẽ với các lớp `Dropout` (ví dụ: tỷ lệ 0.5 và 0.3) để regularization, và cuối cùng là một lớp `Dense` với 4 units sử dụng hàm kích hoạt softmax cho bài toán phân loại 4 lớp [cite: 439, 440, 441, 442, 443]. Mô hình hoàn chỉnh được xây dựng bằng Keras Functional API (tên mô hình trong summary là "functional").

Tổng cộng, mô hình có 30,396,425 tham số, trong đó 27,387,399 tham số có thể huấn luyện (trainable) và 3,009,026 tham số không thể huấn luyện (non-trainable).

3.3.3. Thiết lập Môi trường và Thông số Huấn luyện

- Đã chuyển các kết quả classification report thành dạng bảng

3.4.2.2. Kết quả định lượng

Classification Report (Test Set):

	precision	recall	f1-score	support
Glioma	1.00	0.99	0.99	244
Meningioma	0.99	0.99	0.99	247
No_Tumor	1.00	0.99	0.99	300
Pituitary	0.98	0.99	0.99	264
accuracy			0.99	1055
macro avg	0.99	0.99	0.99	1055
weighted avg	0.99	0.99	0.99	1055

- Đã kiểm tra tính nhất quán ngôn ngữ, căn lề, chính tả, font chữ
- Đã chỉ rõ tiền xử lý dataset 1,2 như thế nào
- Đã bổ sung hướng dẫn chạy app

1. Truy cập ứng dụng:

Cách khởi chạy ứng dụng: Tại thư mục chứa code gõ ở terminal: **streamlit run app.py**

```
PS C:\Users\A\streamlit> streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.32:8501

2023-05-15 17:02:20.120343: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2023-05-15 17:02:26.342311: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2023-05-15 17:02:46.382134: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
```

Hình 3. 31. Khởi chạy ứng dụng

2. **Tải ảnh lên:** Tại cột bên trái, nhấp vào nút "Chọn ảnh MRI" và chọn một tệp ảnh MRI nào từ máy tính của bạn. Ảnh bạn chọn sẽ được hiển thị ngay ở cột kết quả.

- Đã tính lược nội dung để đáp ứng yêu cầu độ dài của thầy

- ...

- Tự động hóa kiểm tra chất lượng sản phẩm trong công nghiệp.

1.4.5. Các kỹ thuật khác (Expert Systems, Robotics, v.v.)

Bên cạnh các lĩnh vực chủ đạo trên, AI còn bao gồm nhiều kỹ thuật và công nghệ nền tảng khác:

- **Hệ chuyên gia (Expert Systems)**
- **Robot thông minh (Robotics)**
- **Tìm kiếm và tối ưu hóa (Search & Optimization)**
- **Học tăng cường (Reinforcement Learning)**
- **Hệ thống lai (Hybrid Systems)**

Tóm lại, các kỹ thuật và công nghệ nền tảng của trí tuệ nhân tạo đã và đang tạo nên sức mạnh vượt trội cho AI trong việc giải quyết các vấn đề thực tiễn, thúc đẩy đổi mới sáng tạo và nâng cao chất lượng cuộc sống. Sự kết hợp giữa học máy, học sâu, xử lý ngôn ngữ tự nhiên, thị giác máy tính cùng với các hệ chuyên gia, robot và các công nghệ liên quan đã mở ra kỷ nguyên mới cho trí tuệ nhân tạo, đồng thời đặt nền móng vững chắc cho sự phát triển bền vững của AI trong tương lai.

- ...

Đã lược bớt nội dung

thông minh, xử lý dữ liệu ở quy mô lớn, tự động hóa các quy trình phức tạp và thích nghi với môi trường thay đổi.

Nói một cách đơn giản, AI là trí tuệ của máy móc được con người tạo ra, cho phép máy móc tư duy, học hỏi và giải quyết vấn đề tương tự như trí tuệ con người, nhưng với tốc độ, quy mô và độ chính xác vượt trội hơn.

- **Phân biệt AI với các lĩnh vực liên quan (Machine Learning, Deep Learning)**

AI là khái niệm rộng, bao gồm mọi hệ thống có khả năng thực hiện các hành vi thông minh. Bên trong AI, Machine Learning (học máy) là một nhánh tập trung vào việc phát triển các thuật toán cho phép máy tính tự học hỏi từ dữ liệu mà không cần lập trình cụ thể cho từng tác vụ. Machine Learning gồm hai nhóm chính: học có giám sát (supervised learning) và học không giám sát (unsupervised learning), ứng dụng trong phân loại, dự báo, phân cụm, v.v.

Deep Learning (học sâu) là một nhánh con của Machine Learning, sử dụng các mạng nơ-ron nhân tạo nhiều lớp (deep neural networks) để xử lý dữ liệu phức tạp, đặc biệt hiệu quả với dữ liệu phi cấu trúc như hình ảnh, âm thanh, văn bản. Deep Learning tự động trích xuất đặc trưng từ dữ liệu qua nhiều lớp, giảm sự can thiệp thủ công của con người, và đã tạo ra bước tiến lớn trong các lĩnh vực như nhận diện khuôn mặt, xử lý ngôn ngữ tự nhiên, dịch máy, và xe tự lái.

Tóm lại, AI là khái niệm tổng quát nhất, trong đó Machine Learning là một nhánh, và Deep Learning là một nhánh nhỏ hơn nằm trong Machine Learning.

2.5.2. Hạn chế và thách thức

Bên cạnh những ưu điểm vượt trội, học sâu cũng tồn tại nhiều hạn chế và thách thức đặc thù:

- Đòi hỏi lượng dữ liệu huấn luyện rất lớn:
- Yêu cầu tài nguyên tính toán lớn:
- Vấn đề giải thích mô hình (Explainability):
- Nhạy cảm với dữ liệu nhiễu và tấn công adversarial:
- Khó khăn trong điều chỉnh siêu tham số (hyperparameter tuning):
- Các vấn đề về bias, fairness, security: