|  |  |
| --- | --- |
|  | |
| ĐẠI HỌC THÁI NGUYÊN  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG** | |
|  | |
| D:\DATA\Picture\1, LOGO\TNU-CNTT_2.png | |
|  | |
| **BÁO CÁO CHUYÊN ĐỀ 1** | |
|  | |
| ***Đề tài:*** | |
| **TÌM HIỂU PHƯƠNG PHÁP HỌC MÁY VÀ ỨNG DỤNG  TRONG BÀI TOÁN DỰ ĐOÁN MỨC ĐỘ UNG THƯ PHỔI** | |
|  | |
| **Sinh viên thực hiện:**  **Lớp:**  **Giáo viên hướng dẫn:** | **Nguyễn Văn …**  **CNTT K20 …**  **TS. Ngô Hữu Huy** |
|  | |

**THÁI NGUYÊN, NĂM 2025**

**LỜI CẢM ƠN**

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và sâu sắc tới **TS. Ngô Hữu Huy** – người thầy đã tận tình hướng dẫn, chỉ bảo và hỗ trợ em trong suốt quá trình thực hiện đề tài *"So sánh các mô hình học sâu trong nhận diện viêm phổi từ ảnh X-quang"*.

Em cũng xin chân thành cảm ơn các thầy cô giáo Trường **Đại học Công nghệ Thông tin và Truyền thông – Đại học Thái Nguyên** đã luôn nhiệt tình giảng dạy và truyền đạt cho em những kiến thức quý báu, là nền tảng vững chắc để em hoàn thành tốt đề tài này.

Qua quá trình nghiên cứu và triển khai đề tài, em đã có cơ hội tiếp cận sâu hơn với các kỹ thuật hiện đại trong trí tuệ nhân tạo và thị giác máy tính, đồng thời rèn luyện được kỹ năng làm việc với mô hình học sâu, dữ liệu y tế và xây dựng ứng dụng hỗ trợ chẩn đoán bệnh bằng AI.

Xin chân thành cảm ơn!

*Thái Nguyên, ngày …… tháng …… năm 2025*

|  |  |
| --- | --- |
|  | **Sinh viên** |
|  |  |
|  | **Nguyễn Văn …** |

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc196393514)

[MỤC LỤC 1](#_Toc196393515)

[DANH SÁCH CÁC BẢNG 3](#_Toc196393516)

[DANH SÁCH CÁC HÌNH 4](#_Toc196393517)

[MỞ ĐẦU 5](#_Toc196393518)

[CHƯƠNG 1. CỞ SỞ LÝ THUYẾT 6](#_Toc196393519)

[1.1. Khái niệm học máy, học sâu 6](#_Toc196393520)

[1.1.1. Học máy (Machine Learning) 6](#_Toc196393521)

[1.1.2. Học sâu (Deep Learning) 6](#_Toc196393522)

[1.2. Ứng dụng học máy 6](#_Toc196393523)

[1.2.1. Xử lý ảnh 6](#_Toc196393524)

[1.2.2. Phân tích văn bản 7](#_Toc196393525)

[1.2.3. Khai phá dữ liệu 7](#_Toc196393526)

[1.2.4. Trò chơi điện tử và Robot 8](#_Toc196393527)

[1.3.1. Một số ví dụ thực tế 9](#_Toc196393528)

[1.3.2. Huấn luyện mô hình 10](#_Toc196393529)

[1.4. Phân loại thuật toán Học máy (Machine learning) 10](#_Toc196393530)

[1.4.1. Học có giám sát – Supervised learning 10](#_Toc196393531)

[1.4.2. Học không giám sát – Unsupervised learning 11](#_Toc196393532)

[CHƯƠNG 2. CÔNG CỤ KHAI PHÁ DỮ LIỆU ORANGE DATA MINING 13](#_Toc196393533)

[2.1. Giới thiệu về Data Mining 13](#_Toc196393534)

[2.1.1. Data mining 13](#_Toc196393535)

[2.1.2. Một số tính năng chính của Data Mining 13](#_Toc196393536)

[2.2. Các bước trong Data Mining 13](#_Toc196393537)

[CHƯƠNG 3. BÀI TOÁN CHUẨN ĐOÁN MỨC ĐỘ UNG THƯ PHỔI 15](#_Toc196393538)

[3.1. Mô tả bài toán 15](#_Toc196393539)

[3.2. Thu thập cơ sở dữ liệu 16](#_Toc196393540)

[3.3. Mô hình phân loại 17](#_Toc196393541)

[3.4. Thực hiện chuẩn đoán trên Orange Data Mining 18](#_Toc196393542)

[KẾT LUẬN 20](#_Toc196393543)

[TÀI LIỆU THAM KHẢO 21](#_Toc196393544)

DANH SÁCH CÁC BẢNG

[Bảng 3.1. Minh họa 13 số tính năng trước trong cơ sở dữ liệu 16](#_Toc196318747)

[Bảng 3.2. Minh họa 12 số tính năng sau trong cơ sở dữ liệu 16](#_Toc196318748)

DANH SÁCH CÁC HÌNH

[Hình 1.1. Học máy (Machine learning). 10](#_Toc196318529)

[Hình 1.2. Ví dụ về phân cụm. 12](#_Toc196318530)

[Hình 2.1. Các bước trong Data Mining 15](#_Toc196318531)

[Hình 3.1. Tỷ lệ mắc ung thư toàn cầu năm 2018 17](#_Toc196318532)

[Hình 3.2. Phân phối của tính năng mục tiêu 18](#_Toc196318533)

[Hình 3.3. Mô hình bài toán 19](#_Toc196318534)

MỞ ĐẦU

Trong thời đại hiện nay, sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI) và học máy (Machine Learning) đã mở ra nhiều cơ hội mới trong lĩnh vực chăm sóc sức khỏe. Khi các hệ thống y tế đang đối mặt với áp lực về nhân lực, chi phí và nhu cầu chẩn đoán nhanh chóng, chính xác, việc ứng dụng công nghệ hiện đại vào hỗ trợ chẩn đoán bệnh trở nên vô cùng cấp thiết.

Một trong những căn bệnh hô hấp nguy hiểm cần được phát hiện sớm là **viêm phổi** – nguyên nhân hàng đầu gây tử vong ở trẻ em dưới 5 tuổi và là mối đe dọa lớn đối với người cao tuổi và những người có hệ miễn dịch suy yếu. Quá trình chẩn đoán viêm phổi từ ảnh X-quang truyền thống thường phụ thuộc vào kinh nghiệm bác sĩ, có thể dẫn đến sai sót trong điều kiện thiếu nhân lực chuyên môn hoặc quá tải y tế.

Trong bối cảnh đó, các mô hình học sâu, đặc biệt là **mạng nơ-ron tích chập (CNN)**, đã chứng minh khả năng vượt trội trong việc xử lý và phân loại hình ảnh y tế. Các mô hình CNN có thể tự động học đặc trưng từ ảnh X-quang và đưa ra dự đoán với độ chính xác cao.

Nghiên cứu này tập trung vào việc **xây dựng và đánh giá hiệu quả của các mô hình học sâu khác nhau, bao gồm mô hình truyền thống CNN, mô hình VGG16 và ResNet50** trong nhiệm vụ phát hiện bệnh viêm phổi từ ảnh X-quang ngực.

Bằng cách so sánh hiệu năng giữa các mô hình này, chúng tôi kỳ vọng tìm ra kiến trúc phù hợp nhất cho bài toán xác định viêm phổi, từ đó phát triển một hệ thống hỗ trợ chẩn đoán tự động đáng tin cậy. Nghiên cứu không chỉ góp phần giảm tải công việc cho bác sĩ, mà còn hướng tới xây dựng một nền tảng y tế thông minh, hỗ trợ chăm sóc sức khỏe bền vững và chính xác hơn trong tương lai.

Mặc dù đã cố gắng hết sức nhưng do thời gian ngắn nên đề tài không tránh khỏi nhiều thiếu sót, rất mong được sự quan tâm và góp ý thêm của thầy cô và tất cả các bạn.

Để có thể hoàn thành đề tài này, tôi xin gửi lời cảm ơn sâu sắc giáo viên hướng dẫn đã nhiệt tình hướng dẫn, chỉ bảo và cung cấp cho tôi nhiều kiến thức thực tế rất bổ ích trong suốt quá trình làm đề tài. Nhờ sự giúp đỡ tận tình của thầy nên tôi mới hoàn thành được đề tài này.

Em xin chân thành cảm ơn!

CHƯƠNG 1. CỞ SỞ LÝ THUYẾT

1.1. Khái niệm học máy, học sâu

1.1.1. Học máy (Machine Learning)

Học máy (Machine Learning) là một lĩnh vực của [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o) liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể.

1.1.2. Học sâu (Deep Learning)

Học sâu (Deep Learning) là một lĩnh vực thuộc học máy, trong đó các mô hình mạng nơ-ron nhân tạo được xây dựng với nhiều tầng (layers) nhằm học và trích xuất đặc trưng từ dữ liệu đầu vào một cách tự động. Nhờ vào cấu trúc nhiều lớp, các mô hình học sâu có khả năng biểu diễn và xử lý thông tin phức tạp tốt hơn so với các phương pháp truyền thống.

Deep Learning thường gắn liền với các lĩnh vực như Trí tuệ nhân tạo (AI) và Dữ liệu lớn (Big Data), đặc biệt được thúc đẩy mạnh mẽ nhờ sự phát triển của phần cứng (như GPU) và các thư viện mã nguồn mở như TensorFlow, PyTorch, giúp quá trình xây dựng và huấn luyện mô hình trở nên thuận tiện, dễ tiếp cận hơn.

Với tính linh hoạt cao và khả năng học biểu diễn từ dữ liệu, học sâu hiện đang đóng vai trò quan trọng trong nhiều lĩnh vực, từ xử lý ảnh, ngôn ngữ tự nhiên đến y học, tài chính, và robot. Cũng chính vì lý do này mà cụm từ “Deep Learning” ngày càng được nhắc đến rộng rãi như một biểu tượng nổi bật của công nghệ hiện đại.

1.2. Ứng dụng học máy

1.2.1. Xử lý ảnh

Xử lý ảnh (Image Processing) là một lĩnh vực trong trí tuệ nhân tạo và thị giác máy tính, tập trung vào việc trích xuất thông tin hữu ích từ hình ảnh hoặc thực hiện các thao tác biến đổi ảnh để phục vụ các mục tiêu cụ thể. Các thuật toán học máy hiện nay đã được ứng dụng hiệu quả vào nhiều tác vụ xử lý ảnh trong thực tế, bao gồm:

**Gán nhãn đối tượng (Image Tagging):** Một kỹ thuật giúp tự động phát hiện và gắn nhãn đối tượng trong ảnh, ví dụ như gắn thẻ khuôn mặt người dùng và bạn bè trên các nền tảng mạng xã hội như Facebook. Hệ thống học từ những ảnh đã được gắn nhãn trước đó để tự động hóa quá trình nhận diện.

**Nhận diện ký tự quang học (OCR):** Là kỹ thuật chuyển đổi hình ảnh chứa chữ viết tay hoặc in ấn thành văn bản dạng số. Mô hình học máy cần được huấn luyện để phân biệt và xác định chính xác các ký tự trong ảnh.

**Hỗ trợ phương tiện tự lái:** Trong công nghệ xe tự hành, xử lý ảnh đóng vai trò trọng yếu. Hệ thống cần liên tục phân tích hình ảnh thu được từ camera để xác định làn đường, biển báo giao thông hoặc phát hiện các vật cản, từ đó đưa ra quyết định lái xe phù hợp.

1.2.2. Phân tích văn bản

Phân tích văn bản (Text Analysis) là quá trình xử lý dữ liệu dạng văn bản để trích xuất thông tin có giá trị hoặc phân loại nội dung theo mục đích cụ thể. Nguồn văn bản có thể đến từ nhiều kênh như mạng xã hội, email, hội thoại trực tuyến, tài liệu số hóa, v.v.

Một số ứng dụng tiêu biểu trong phân tích văn bản gồm:

**Lọc thư rác (Spam Filtering):** Hệ thống phân loại email dựa trên nội dung hoặc tiêu đề nhằm xác định liệu email đó có phải là thư rác hay không. Đây là một trong những ứng dụng lâu đời và phổ biến nhất của phân loại văn bản trong học máy.

**Phân tích cảm xúc (Sentiment Analysis):** Phương pháp này dùng để xác định thái độ hoặc quan điểm của người viết, chẳng hạn như phân loại một đánh giá sản phẩm là tích cực, tiêu cực hay trung lập.

**Trích xuất thông tin (Information Extraction):** Hệ thống học cách nhận biết và rút trích các yếu tố quan trọng từ văn bản như tên người, địa chỉ, tổ chức, từ khóa,... phục vụ cho các mục tiêu tìm kiếm, phân tích hoặc tổ chức dữ liệu.

…

1.2.3. Khai phá dữ liệu

Khai phá dữ liệu (Data Mining) là quá trình phát hiện ra các mẫu, quy luật hoặc thông tin giá trị ẩn chứa trong một tập dữ liệu lớn. Thông qua việc áp dụng các thuật toán học máy, khai phá dữ liệu giúp khám phá tri thức và đưa ra dự đoán dựa trên dữ liệu có sẵn.

Mỗi dòng dữ liệu có thể được xem như một đối tượng quan sát, trong đó mỗi cột thể hiện một đặc trưng hoặc thuộc tính. Dựa vào những mẫu đã học, hệ thống có thể thực hiện các tác vụ phân tích hoặc suy luận như sau:

**Phát hiện bất thường (Anomaly Detection):** Tìm ra những điểm dữ liệu khác biệt với phần còn lại, chẳng hạn như giao dịch gian lận trong hệ thống thanh toán.

**Khai thác luật kết hợp (Association Rules):** Phát hiện mối quan hệ đồng xuất hiện giữa các mục dữ liệu, ví dụ: khách hàng mua sản phẩm A thường có xu hướng mua thêm sản phẩm B. Điều này rất hữu ích trong thương mại điện tử và tiếp thị sản phẩm.

**Phân cụm (Clustering):** Nhóm các đối tượng dữ liệu có đặc điểm tương đồng vào cùng một nhóm, ví dụ như phân loại khách hàng theo hành vi tiêu dùng hoặc hồ sơ cá nhân.

**Dự đoán (Prediction):** Ước lượng giá trị của một thuộc tính đầu ra dựa trên các đặc trưng đầu vào, chẳng hạn như dự đoán giá bất động sản dựa trên vị trí, diện tích, số phòng,...

1.2.4. Trò chơi điện tử và Robot

Trò chơi điện tử (Video Games) và robot (Robotics) là hai lĩnh vực điển hình trong việc ứng dụng các kỹ thuật học máy, đặc biệt là học tăng cường (Reinforcement Learning). Trong các trò chơi, một nhân vật ảo cần đưa ra quyết định để di chuyển và tránh chướng ngại vật. Thay vì lập trình thủ công, mô hình học máy có thể tự học hành vi tối ưu thông qua tương tác với môi trường.

Học tăng cường là một phương pháp trong đó mô hình (agent) học cách hành động bằng cách nhận phần thưởng hoặc hình phạt từ môi trường sau mỗi bước tương tác. Mục tiêu là tối ưu hóa tổng phần thưởng nhận được. Một hành động được xem là tích cực nếu nó giúp agent đến gần mục tiêu hơn, và ngược lại nếu nó gây thất bại, như va chạm với vật cản.

Một ví dụ nổi bật cho sự thành công của học tăng cường là hệ thống **AlphaGo** của Google DeepMind, đã đánh bại kỳ thủ cờ vây hàng đầu thế giới. Cờ vây là một trò chơi có không gian trạng thái cực kỳ lớn, và việc học chiến lược chiến thắng từ dữ liệu ván đấu là một thách thức lớn mà AlphaGo đã vượt qua.

[](https://nguyenvanhieu.vn/wp-content/uploads/2018/09/machine-learning.jpg)

Hình 1.1. Học máy (Machine learning).

Có thể mô hình hóa một bài toán học tăng cường với ba yếu tố chính: **nhiệm vụ (T)** là trò chơi cụ thể, **kinh nghiệm (E)** là dữ liệu từ các ván đấu trước hoặc tương tác với môi trường, và **hiệu suất (P)** là tỷ lệ chiến thắng hoặc mức độ hoàn thành mục tiêu.

1.3.1. Một số ví dụ thực tế

Để hiểu rõ hơn về cách hoạt động của các thuật toán học máy, dưới đây là một số tình huống ứng dụng điển hình:

**Ví dụ 1:** Một hệ thống camera giám sát được trang bị khả năng phát hiện chuyển động và phân loại đối tượng. Khi phát hiện có người bước vào khu vực, hệ thống sẽ xác định đó là người, vật nuôi hay xe cộ để đưa ra cảnh báo phù hợp.

**Ví dụ 2:** Một ứng dụng phân tích đánh giá phim từ các trang web như IMDb hay Rotten Tomatoes. Hệ thống được huấn luyện để tự động phân loại cảm xúc trong các nhận xét của người dùng nhằm xác định liệu bộ phim được đón nhận tích cực hay tiêu cực.

**Ví dụ 3:** Một nền tảng thương mại điện tử cá nhân hóa đề xuất sản phẩm cho người dùng. Dựa trên lịch sử mua sắm và hành vi lướt trang, hệ thống sẽ đề xuất các mặt hàng có khả năng cao phù hợp với sở thích của người dùng đó.

Với ví dụ đầu tiên, mô hình học từ các hình ảnh có nhãn xác định loại đối tượng trong video để đưa ra dự đoán. Độ chính xác của hệ thống được đánh giá dựa trên tỷ lệ phát hiện đúng trong tập video kiểm thử.

Ở ví dụ thứ hai, mô hình học từ tập dữ liệu gồm các đoạn đánh giá phim kèm theo nhãn cảm xúc. Hiệu suất của thuật toán được đo bằng tỷ lệ phân loại đúng trên các đánh giá mới chưa từng thấy.

Trong ví dụ cuối cùng, hệ thống học được từ dữ liệu lịch sử người dùng (như sản phẩm đã xem hoặc đã mua) để đưa ra các đề xuất sản phẩm phù hợp. Hiệu suất được đánh giá thông qua tỷ lệ người dùng thực sự chọn mua các sản phẩm được đề xuất.

1.3.2. Huấn luyện mô hình

Để một thuật toán học máy có thể đưa ra kết quả chính xác từ dữ liệu đầu vào, trước tiên cần trải qua quá trình huấn luyện trên một tập dữ liệu có sẵn. Tập dữ liệu này đóng vai trò là “kinh nghiệm” mà mô hình sử dụng để học mối liên hệ giữa đặc trưng đầu vào và nhãn đầu ra.

Trong quá trình huấn luyện, thuật toán sẽ điều chỉnh các tham số bên trong (ví dụ như trọng số của mạng nơ-ron) nhằm tối ưu hóa khả năng dự đoán. Việc lựa chọn và trích xuất các đặc trưng đầu vào phù hợp (feature selection) là một yếu tố then chốt quyết định hiệu quả của mô hình.

Nội dung này sẽ được phân tích kỹ hơn ở các phần sau, trong đó làm rõ các phương pháp tiền xử lý và kỹ thuật rút trích đặc trưng thường dùng trong học máy.

1.4. Phân loại thuật toán Học máy (Machine learning)

Trong học máy, các thuật toán được phân loại dựa trên cách mà mô hình học từ dữ liệu. Bốn nhóm chính thường được đề cập gồm:

**Học có giám sát (Supervised Learning):** Mô hình được huấn luyện từ dữ liệu đã gắn nhãn, nghĩa là mỗi mẫu dữ liệu đầu vào đi kèm với đầu ra mong muốn. Thuật toán học cách ánh xạ giữa đầu vào và đầu ra để dự đoán nhãn cho các dữ liệu mới.

**Học không giám sát (Unsupervised Learning):** Dữ liệu không có nhãn đầu ra. Mục tiêu của mô hình là phát hiện cấu trúc tiềm ẩn trong dữ liệu như các cụm (clusters) hoặc các quy luật liên kết.

**Học bán giám sát (Semi-supervised Learning):** Kết hợp giữa dữ liệu có nhãn và không có nhãn để cải thiện độ chính xác. Phương pháp này đặc biệt hữu ích khi việc gán nhãn thủ công cho toàn bộ dữ liệu là không khả thi hoặc tốn kém.

**Học tăng cường (Reinforcement Learning):** Mô hình học cách ra quyết định thông qua quá trình tương tác với môi trường. Hệ thống sẽ nhận phần thưởng hoặc hình phạt tương ứng với hành động thực hiện, từ đó học được chính sách hành động tối ưu.

1.4.1. Học có giám sát – Supervised learning

Học có giám sát là phương pháp trong đó mô hình học từ dữ liệu đầu vào đã được gắn nhãn. Nghĩa là, mỗi mẫu dữ liệu trong tập huấn luyện đều có thông tin đầu ra tương ứng. Mục tiêu của mô hình là học được một hàm ánh xạ giữa đầu vào và đầu ra, từ đó có thể dự đoán chính xác kết quả cho các dữ liệu chưa thấy trước đó.

Về mặt toán học, giả sử ta có một tập dữ liệu đầu vào X={x1,x2,...,xN}X = \{x\_1, x\_2, ..., x\_N\}X={x1​,x2​,...,xN​} và một tập nhãn tương ứng Y={y1,y2,...,yN}Y = \{y\_1, y\_2, ..., y\_N\}Y={y1​,y2​,...,yN​}. Học có giám sát nhằm mục tiêu xây dựng một hàm ánh xạ fff sao cho f(xi)≈yif(x\_i) \approx y\_if(xi​)≈yi​ với mọi i=1,2,...,Ni = 1, 2, ..., Ni=1,2,...,N.

Một trong những hạn chế của phương pháp này là yêu cầu dữ liệu được gán nhãn đầy đủ, trong khi quá trình gán nhãn thủ công thường tốn nhiều thời gian và công sức, đặc biệt đối với các bài toán phức tạp hoặc dữ liệu lớn.

Học có giám sát thường được ứng dụng trong hai dạng bài toán chính:

**Phân loại (Classification):** Dự đoán nhãn đầu ra rời rạc (ví dụ: phân loại email thành spam hoặc không spam).

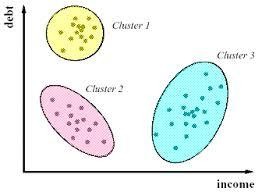
**Hồi quy (Regression):** Dự đoán giá trị liên tục (ví dụ: dự đoán giá nhà dựa trên diện tích, vị trí,...).

1.4.2. Học không giám sát – Unsupervised learning

Học không giám sát là phương pháp mà trong đó thuật toán chỉ có dữ liệu đầu vào mà không có nhãn đầu ra tương ứng. Tức là, mô hình được cung cấp một tập dữ liệu XXX, nhưng không biết trước các nhãn YYY tương ứng, và nhiệm vụ của nó là tự khám phá cấu trúc ẩn bên trong dữ liệu.

Các thuật toán học không giám sát thường được ứng dụng trong các bài toán sau:

**Phân cụm (Clustering):** Là quá trình nhóm các đối tượng có đặc điểm tương đồng vào cùng một cụm. Ví dụ: trong lĩnh vực thương mại điện tử, khách hàng có thể được phân nhóm dựa trên hành vi mua sắm; trong tài chính, người dùng có thể được phân loại theo mức thu nhập hoặc mô hình tiêu dùng.



Hình 1.2. Ví dụ về phân cụm.

**Phát hiện luật kết hợp (Association Rule Mining):** Nhằm khai thác các mối liên hệ giữa các mục dữ liệu thường xuyên xuất hiện cùng nhau. Chẳng hạn, trong một siêu thị, từ dữ liệu lịch sử mua hàng có thể suy ra rằng khách hàng mua bánh mì thường mua kèm sữa, từ đó phục vụ cho hệ thống gợi ý sản phẩm (Recommendation System).

CHƯƠNG 2. TÌM HIỂU VỀ MÔ HÌNH

2.1. Tổng quan về CNN

2.1.1. CNN ( Convolutional Neural Network ) là gì

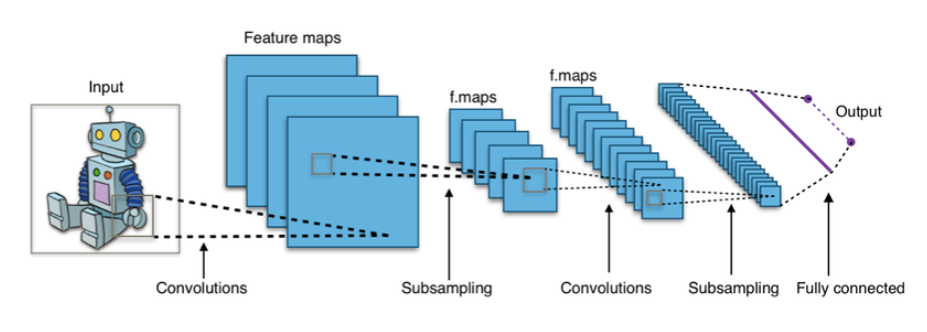
Mạng nơ-ron tích chập (Convolutional Neural Network – CNN) là một trong những kiến trúc nổi bật của học sâu (Deep Learning), được thiết kế đặc biệt để xử lý dữ liệu dạng lưới như ảnh. CNN đã trở thành nền tảng cốt lõi trong nhiều hệ thống trí tuệ nhân tạo hiện đại nhờ khả năng học đặc trưng trực tiếp từ dữ liệu đầu vào mà không cần trích xuất thủ công.

CNN được ứng dụng rộng rãi trong các bài toán thị giác máy tính như nhận diện đối tượng, phân loại hình ảnh, phát hiện vật thể, và thậm chí trong lĩnh vực y tế như phân tích ảnh X-quang. Cấu trúc của CNN gồm nhiều lớp tích chập (convolutional layers), lớp phi tuyến (activation), lớp pooling, và lớp fully connected – cho phép mô hình học từ các đặc trưng cục bộ đến khái quát hơn.

Trong đề tài này, CNN đóng vai trò như một kiến trúc nền tảng để so sánh hiệu suất với các mô hình học sâu khác trong nhiệm vụ phát hiện viêm phổi từ ảnh X-ray ngực

2.1.2. Cấu trúc mạng CNN.

Mạng nơ-ron tích chập (CNN) bao gồm nhiều lớp tích chập (convolutional layers) xếp chồng lên nhau, trong đó mỗi lớp sử dụng các hàm kích hoạt phi tuyến tính như ReLU hoặc Tanh nhằm tăng khả năng biểu diễn phi tuyến của mô hình. Qua từng lớp, các đặc trưng ngày càng trừu tượng hơn được trích xuất và truyền đến các lớp tiếp theo.



Hình 2.1. Cấu trúc mạng CNN

Không giống với mạng truyền thẳng thông thường, nơi mọi nút đầu vào kết nối đến mọi nút ở lớp sau (fully connected layers), CNN sử dụng cơ chế kết nối cục bộ – tức là mỗi nơ-ron chỉ liên kết với một vùng nhỏ trên đầu vào. Điều này được thực hiện thông qua các bộ lọc (filter) hay kernel được “trượt” trên ảnh đầu vào để tạo ra các bản đồ đặc trưng (feature maps).

Mỗi bộ lọc học một đặc trưng riêng và các lớp khác nhau sẽ trích xuất các mức độ thông tin khác nhau, từ cạnh, góc cạnh, đến hình dạng và cấu trúc phức tạp hơn. Trong quá trình huấn luyện, CNN học tự động các giá trị tối ưu cho các bộ lọc thông qua lan truyền ngược (backpropagation).

Ngoài các lớp tích chập, mạng CNN còn có các lớp giảm kích thước như pooling hoặc subsampling để làm nổi bật đặc trưng chính và giảm nhiễu. Pooling layer giúp mạng học được tính bất biến (invariance) đối với phép dịch chuyển, co giãn, xoay, từ đó tăng khả năng nhận diện đối tượng dù có thay đổi về vị trí hoặc kích thước trong ảnh.

Một đặc điểm quan trọng của CNN là tính **kết hợp (compositionality)** – nghĩa là các đặc trưng đơn giản ở lớp dưới có thể kết hợp thành đặc trưng phức tạp hơn ở các lớp trên. Đây là cơ sở cho khả năng nhận diện mạnh mẽ của CNN, tương tự như cách con người nhận biết vật thể từ chi tiết đến tổng thể.

Ba nguyên lý then chốt tạo nên thành công của CNN bao gồm:

**Trường tiếp nhận cục bộ (local receptive field)**

**Trọng số chia sẻ (shared weights)**

**Lớp tổng hợp (pooling)**

2.1.3. Cấu trúc mạng CNN.

Phép tích chập (convolution) là một phép toán cốt lõi trong mạng nơ-ron tích chập (CNN), được sử dụng để trích xuất đặc trưng từ ảnh đầu vào. Phép toán này thường được ký hiệu là Y=X⊗W, trong đó:

X: Ma trận biểu diễn giá trị điểm ảnh (input image)

W: Ma trận kernel (hay filter), thường có kích thước k×kk \times kk×k với kkk là số lẻ (ví dụ: 3, 5, 7,...)

⊗: Ký hiệu phép tích chập (thực hiện nhân từng phần tử và cộng tổng)

Trong quá trình tính toán, kernel được “trượt” qua từng vùng nhỏ (local region) của ảnh đầu vào. Tại mỗi bước, kernel nhân từng phần tử tương ứng với vùng ảnh và tổng kết lại thành một giá trị duy nhất, giá trị này trở thành phần tử tại vị trí tương ứng trong ảnh đầu ra.

Ví dụ: tính giá trị

A grid with numbers and symbols

Description automatically generated

Hình 2.2. Ví dụ về phép tính Convolution

Mỗi bộ lọc (kernel) khác nhau sẽ làm nổi bật các đặc trưng khác nhau của ảnh, chẳng hạn như:

Làm mờ (blur)

Làm nét (sharpen)

Phát hiện cạnh (edge detection)

Chính sự đa dạng của các kernel giúp mạng CNN có khả năng học được nhiều loại đặc trưng khác nhau trong quá trình huấn luyện, từ đó hỗ trợ mạnh mẽ cho các tác vụ như nhận dạng, phân loại hoặc phát hiện đối tượng trong ảnh.

2.1.4. Padding

Padding là một kỹ thuật được sử dụng để thay đổi kích thước của đầu vào bằng cách thêm các giá trị (thường là giá trị 0) vào xung quanh biên của ảnh hoặc đầu vào của mạng neural. Mục tiêu chính của việc sử dụng padding là để duy trì kích thước của đầu vào qua các lớp xử lý mà không làm giảm kích thước của đầu ra.

A grid with numbers and symbols

Description automatically generated

Hình 2.3. Padding

2.1.5. Stride

Stride định nghĩa bước nhảy của cửa sổ (window) hoặc bộ lọc (filter) khi di chuyển trên đầu vào (input) để thực hiện phép tính convolution hoặc pooling.

Khi áp dụng stride trong phép tính convolution hoặc pooling, cửa sổ hoặc bộ lọc không di chuyển mỗi pixel một cách liên tục, mà sẽ nhảy qua một số pixel cố định được xác định bởi stride. Giá trị stride được xác định trước và thường được chọn sao cho đảm bảo kích thước của đầu ra (output) là phù hợp với yêu cầu của mô hình.

Ví dụ: ta có ma trận X (hình bên dưới) với padding = 1, stride = 2.

**A grid with numbers and symbols

Description automatically generated**

Hình 2.4. Stride

Hiểu đơn giản là bắt đầu từ vị trí x11 sau đó nhảy k bước theo chiều dọc và ngang cho đến hết ma trận X .

Kích thước của ma trận Y là 3\*3 đã giảm đi đáng kể so với ma trận X.

Công thức tổng quát cho phép tính convolution của ma trận X kích thước m\*n với kernel kích thước ,stride = s, padding = p ra ma trận Y kích thước .

Trong đó:

m và n là kích thước của đầu vào.

k là kích thước của bộ lọc.

p là padding.

s là stride.

Stride thường dùng để giảm kích thước của ma trận sau phép tính convolution.

2.1.6. Pooling layer

Pooling layer thường được dùng giữa các convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Việc giảm kích thước dữ liệu giúp giảm các phép tính toán trong model.

Bên cạnh đó, với phép pooling kích thước ảnh giảm, do đó lớp convolution học được các vùng có kích thước lớn hơn. Ví dụ như ảnh kích thước 224\*224 qua pooling về 112\*112 thì vùng 3\*3 ở ảnh 112\*112 tương ứng với vùng 6\*6 ở ảnh ban đầu. Vì vậy qua các pooling thì kích thước ảnh nhỏ đi và convolutional layer sẽ học được các thuộc tính lớn hơn.

Gọi pooling size kích thước K\*K. Input của pooling layer có kích thước H\*W\*D, ta tách ra làm D ma trận kích thước H\*W. Với mỗi ma trận, trên vùng kích thước K\*K trên ma trận ta tìm maximum hoặc average của dữ liệu rồi viết vào ma trận kết quả. Quy tắc về stride và padding áp dụng như phép tính convolution trên ảnh.

Ví dụ: max pooling layer với size= (3x3), stride=1, padding=0.

**A close-up of a grid

Description automatically generated**

Hình 2.5. Max Pooling

Nhưng hầu hết khi dùng pooling layer thì sẽ dùng size=(2,2), stride=2, padding=0. Khi đó output width và height của dữ liệu giảm đi một nửa, depth thì được giữ nguyên.

**A diagram of a pool and pool

Description automatically generated**

Hình 2.6. Giảm kích thước ảnh khi dùng pooling layer

Có 2 loại pooling layer phổ biến là:

Max pooling

Average pooling

**A diagram of a pool

Description automatically generated**

Hình 2.7. Max Pooling và Average Pooling

**2.2 Tổng quan về mô hình VGG**

VGG (Visual Geometry Group) là một kiến trúc mạng nơ-ron tích chập sâu (Deep Convolutional Neural Network) được phát triển bởi nhóm nghiên cứu tại Đại học Oxford.

Mô hình VGG được biết đến với cấu trúc đơn giản nhưng hiệu quả, trong đó các lớp tích chập 3x3 được xếp chồng để trích xuất đặc trưng từ ảnh. Hai phiên bản phổ biến của mô hình là VGG16 và VGG19, tương ứng với 16 và 19 lớp có trọng số huấn luyện được.

VGG đóng vai trò quan trọng trong sự phát triển của các hệ thống nhận dạng hình ảnh hiện đại, và vẫn được sử dụng rộng rãi trong nhiều bài toán thị giác máy tính.

**2.2.1 Mô hình VGG16**

VGG16 là một biến thể của mô hình VGG với tổng cộng 16 lớp có trọng số, bao gồm 13 lớp tích chập và 3 lớp kết nối đầy đủ.

Mô hình được đề xuất bởi Simonyan và Zisserman trong bài báo công bố năm 2014. VGG16 đạt độ chính xác top-5 lên đến 92.7% trên bộ dữ liệu ImageNet gồm hơn 14 triệu ảnh thuộc 1000 lớp khác nhau.

Khác với các mô hình trước như AlexNet, VGG16 sử dụng bộ lọc 3x3 nhỏ với bước trượt 1 pixel để xây dựng các tầng trích xuất đặc trưng sâu và chính xác hơn. Cách tiếp cận này giúp mô hình học được các đặc trưng không gian phức tạp với độ sâu lớn hơn mà không tăng quá nhiều tham số.

**2.2.2 Kiến trúc VGG16**

VGG16 có đầu vào là ảnh với kích thước cố định (ví dụ: 224x224 pixel). Ảnh đầu vào sẽ lần lượt đi qua các khối tích chập (convolution blocks), mỗi khối bao gồm 2 hoặc 3 lớp convolution sử dụng bộ lọc 3x3, tiếp theo là một lớp gộp (max pooling) để giảm kích thước không gian.

Các lớp convolution sử dụng hàm kích hoạt ReLU để tăng tính phi tuyến cho mạng. Sau các khối tích chập là ba lớp fully connected (kết nối đầy đủ), trong đó lớp cuối cùng sử dụng hàm Softmax để đưa ra xác suất dự đoán cho từng lớp đầu ra.

Toàn bộ mạng được huấn luyện bằng lan truyền ngược (backpropagation) để tối ưu hàm mất mát giữa nhãn thực tế và đầu ra mô hình.

**2.2.3 Sự phức tạp và thách thức**

Một trong những đặc điểm của VGG16 là độ sâu lớn với số lượng tham số lên tới hàng trăm triệu, làm cho mô hình rất nặng và tốn thời gian huấn luyện.

Dung lượng của mô hình vào khoảng 533MB, gây ra khó khăn khi triển khai trong môi trường tài nguyên hạn chế.

Tuy nhiên, nhờ cấu trúc đồng nhất và đơn giản (sử dụng bộ lọc 3x3 xuyên suốt), VGG16 vẫn là một lựa chọn phổ biến trong nghiên cứu và giảng dạy, đồng thời được dùng như mạng nền cho các mô hình nâng cao khác.

**2.2.4 Hiệu suất của mô hình**

VGG16 đạt hiệu suất cao trên nhiều bài toán phân loại ảnh. Trong cuộc thi ILSVRC 2014, mô hình này đạt top-5 error rate khoảng 7.0%, vượt trội so với nhiều mô hình cùng thời điểm.

Mặc dù không chiến thắng trong cuộc thi, VGG16 đã trở thành một tiêu chuẩn tham khảo trong lĩnh vực thị giác máy tính nhờ khả năng tổng quát tốt và cấu trúc dễ triển khai.

**2.2.5 Ứng dụng của VGG16**

VGG16 được ứng dụng rộng rãi trong nhiều lĩnh vực như:

- Phân loại hình ảnh: Làm nền tảng trong các hệ thống nhận dạng hình ảnh tự động.

- Phát hiện đối tượng: Sử dụng để xác định và phân loại đối tượng trong ảnh.

- Nhận diện khuôn mặt: Ứng dụng trong hệ thống an ninh, giám sát, và các nền tảng mạng xã hội.

**2.3 Mô hình ResNet50**

**2.3.1 Tổng quan về ResNet50**

ResNet50 (Residual Network) là một mô hình mạng nơ-ron sâu được giới thiệu bởi nhóm nghiên cứu Microsoft trong cuộc thi ImageNet năm 2015. Kiến trúc này nổi bật với ý tưởng "residual learning" (học phần dư), cho phép xây dựng các mạng cực kỳ sâu mà vẫn tránh được vấn đề suy giảm độ chính xác khi tăng số lớp.

ResNet50 gồm 50 lớp có trọng số học được, trong đó điểm khác biệt cốt lõi là việc sử dụng các "skip connection" để bỏ qua một số lớp nhất định, giúp lan truyền gradient hiệu quả hơn trong quá trình huấn luyện.

**2.3.2 Kiến trúc ResNet50**

Kiến trúc của ResNet50 bắt đầu bằng một lớp tích chập với kernel 7x7 và stride 2, tiếp theo là một lớp pooling. Sau đó là 4 khối chính (stages), mỗi khối gồm nhiều 'residual blocks'.

Mỗi residual block bao gồm ba lớp tích chập (1x1, 3x3, 1x1) với batch normalization và hàm kích hoạt ReLU, cùng với một đường kết nối tắt (identity shortcut) giữa đầu vào và đầu ra của block.

Tổng cộng ResNet50 có khoảng 23 triệu tham số, tuy nhiên nhờ thiết kế hiệu quả nên khả năng huấn luyện và khả năng tổng quát của mô hình vẫn rất cao.

**2.3.3 Ưu và nhược điểm của ResNet50**

Một trong những ưu điểm lớn nhất của ResNet50 là khả năng huấn luyện các mạng rất sâu mà không gặp phải vấn đề vanishing gradient như ở các mạng truyền thống.

Nhờ có skip connections, ResNet50 dễ hội tụ hơn và tránh hiện tượng quá khớp (overfitting) tốt hơn so với các kiến trúc trước đó như VGG.

Tuy nhiên, nhược điểm của ResNet50 là cấu trúc phức tạp hơn, yêu cầu thời gian huấn luyện lâu hơn và chi phí tính toán cao hơn so với các mô hình nhỏ gọn như MobileNet hoặc SqueezeNet.

CHƯƠNG 3. BÀI TOÁN CHUẨN ĐOÁN MỨC ĐỘ   
UNG THƯ PHỔI

3.1. Mô tả bài toán

Viêm phổi là một trong những bệnh lý đường hô hấp phổ biến và nguy hiểm, đặc biệt ở trẻ nhỏ, người cao tuổi và những bệnh nhân có hệ miễn dịch suy yếu. Nếu không được chẩn đoán và điều trị kịp thời, viêm phổi có thể dẫn đến nhiều biến chứng nghiêm trọng, thậm chí tử vong. Trong bối cảnh các cơ sở y tế ngày càng quá tải, việc ứng dụng công nghệ trí tuệ nhân tạo vào hỗ trợ chẩn đoán viêm phổi là một hướng tiếp cận có tiềm năng lớn nhằm giảm tải cho đội ngũ y tế và nâng cao chất lượng chẩn đoán.

Đề tài này tập trung vào bài toán phát hiện viêm phổi dựa trên ảnh X-quang ngực (chest X-ray) bằng cách sử dụng các mô hình học sâu. Cụ thể, ba kiến trúc mạng nơ-ron được lựa chọn để huấn luyện và đánh giá gồm: CNN truyền thống tự xây, mô hình VGG16 và mô hình ResNet50.

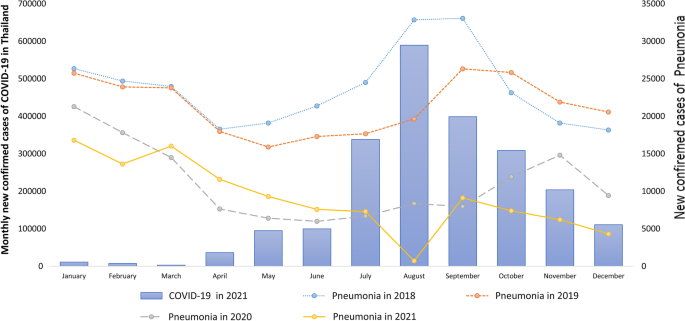
Mục tiêu chính là xây dựng hệ thống có khả năng tự động phân loại ảnh X-ray thành hai lớp: **bình thường** và **viêm phổi**, dựa trên các đặc trưng trích xuất từ ảnh đầu vào. Việc xây dựng mô hình được thực hiện theo các bước sau:

Thu thập và xử lý tập dữ liệu ảnh X-ray có nhãn (bình thường và viêm phổi).

Huấn luyện ba mô hình học sâu với cấu trúc khác nhau để thực hiện phân loại ảnh.

So sánh hiệu suất của các mô hình dựa trên độ chính xác, độ nhạy, độ đặc hiệu và tốc độ dự đoán.

Kết quả cuối cùng không chỉ giúp xác định mô hình tối ưu cho bài toán, mà còn góp phần xây dựng một ứng dụng hỗ trợ chẩn đoán viêm phổi thông minh, trực quan, phục vụ thực tiễn trong ngành y tế.



Hình 3.1. Tỷ lệ mắc viêm phổi các năm

3.2. Thu thập cơ sở dữ liệu

Dữ liệu huấn luyện được thu thập trên internet từ nhiều nguồn có chọn lọc

Bình thường



Bị viêm phổi :



3.3. Tiền xử lý dữ liệu

Trước khi tiến hành huấn luyện mô hình học sâu, dữ liệu đầu vào cần được tiền xử lý nhằm đảm bảo tính nhất quán, tối ưu hóa hiệu quả huấn luyện và giảm thiểu sai lệch trong quá trình học của mô hình.

Trong đề tài này, tập dữ liệu sử dụng là ảnh X-quang ngực (Chest X-ray) với hai lớp nhãn: **bình thường (Normal)** và **viêm phổi (Pneumonia)**. Các bước tiền xử lý được thực hiện như sau:

**Chuẩn hóa kích thước ảnh:** Tất cả các ảnh được chuyển đổi về cùng một kích thước cố định (150×150 pixel) nhằm đảm bảo đầu vào đồng nhất cho các mô hình CNN, VGG16 và ResNet50.

**Chuẩn hóa giá trị pixel:** Giá trị điểm ảnh ban đầu nằm trong khoảng từ 0 đến 255 được chia cho 255 để đưa về khoảng [0, 1], giúp mô hình hội tụ nhanh hơn và tránh hiện tượng gradient lớn gây bất ổn.

**Chuyển đổi nhãn:** Nhãn đầu ra được mã hóa dưới dạng nhị phân (0 cho bình thường, 1 cho viêm phổi) để phù hợp với bài toán phân loại nhị phân sử dụng hàm kích hoạt sigmoid ở đầu ra.

**Tăng cường dữ liệu (Data Augmentation):** Để tránh hiện tượng overfitting do thiếu dữ liệu đa dạng, các kỹ thuật biến đổi ảnh như xoay, phóng to, lật ngang và dịch chuyển được áp dụng. Điều này giúp mô hình học được các đặc trưng bền vững hơn với các biến đổi hình học.

**Tách tập dữ liệu:** Tập dữ liệu được chia thành ba phần: tập huấn luyện (training set), tập kiểm tra (test set) và tập xác thực (validation set) theo tỉ lệ phù hợp nhằm đánh giá khách quan hiệu suất của mô hình.

Quá trình tiền xử lý dữ liệu là bước nền tảng quan trọng, góp phần nâng cao độ chính xác và độ tổng quát hóa của các mô hình học sâu được sử dụng trong bài toán.

3.4. Huấn luyện mô hình

**3.4.1. Cấu hình và chuẩn bị đầu vào**

Trước khi huấn luyện mô hình, dữ liệu ảnh X-ray được tiền xử lý và đưa về kích thước chuẩn 150×150 pixel. Các tham số chung cho quá trình huấn luyện bao gồm:

**Batch size**: 32

**Số epoch tối đa**: 15

**Hàm mất mát**: Binary Crossentropy

**Hàm tối ưu**: Adam

**Đánh giá hiệu suất**: Accuracy

**Chiến lược dừng sớm (EarlyStopping)**: dừng huấn luyện khi mô hình không cải thiện sau 3 epoch liên tiếp

**3.4.2. Tăng cường dữ liệu (Data Augmentation)**

Để nâng cao khả năng tổng quát của mô hình và giảm hiện tượng overfitting, dữ liệu huấn luyện được tăng cường bằng các phép biến đổi hình học như:

Xoay ảnh ngẫu nhiên

Lật ngang (horizontal flip)

Phóng to/thu nhỏ (zoom)

Dịch chuyển chiều ngang/dọc (shift) Tăng cường dữ liệu được áp dụng thông qua thư viện ImageDataGenerator của Keras.

**3.4.3. Huấn luyện mô hình CNN tự xây**

Mô hình CNN được thiết kế từ đầu với các lớp tích chập (Conv2D), lớp chuẩn hóa (BatchNormalization), lớp gộp (MaxPooling2D) và các lớp kết nối đầy đủ (Dense).

Gồm 3 khối Conv2D + MaxPooling2D

Dùng hàm kích hoạt ReLU và lớp Dropout để giảm overfitting

Đầu ra là một lớp Dense (1 neuron) với sigmoid

Mô hình được lưu dưới tên cnn\_pneumonia\_model.h5

**3.3.4. Huấn luyện mô hình VGG16**

Mô hình VGG16 được sử dụng dưới dạng mô hình tiền huấn luyện với weights="imagenet" và include\_top=False để bỏ phần phân loại ban đầu.

Đóng băng toàn bộ phần thân (13 lớp tích chập gốc)

Thêm các lớp Dense vào cuối để phân loại nhị phân

Chỉ fine-tune phần đầu ra

Mô hình được lưu dưới tên vgg16\_pneumonia\_model.h5

**3.4.5. Huấn luyện mô hình ResNet50**

Tương tự như VGG16, mô hình ResNet50 được sử dụng với trọng số đã huấn luyện trên ImageNet.

Các residual block được giữ nguyên

Thêm các lớp phân loại tùy chỉnh phía sau

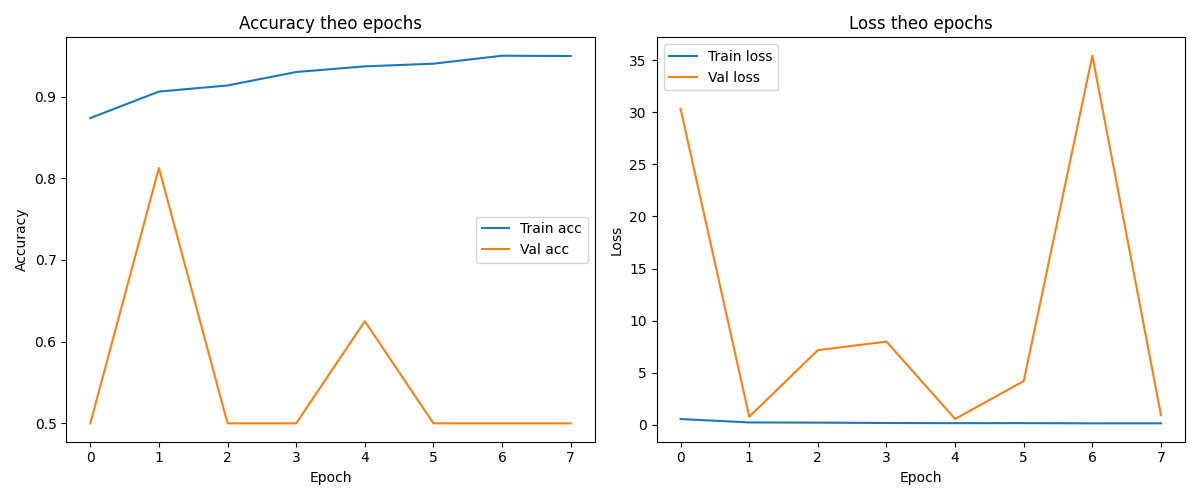
Lớp cuối là sigmoid cho bài toán nhị phân

Mô hình được lưu dưới tên resnet50\_pneumonia\_model.keras

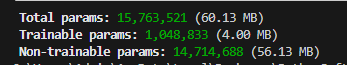
3.5. Kết quả huấn luyện mô hình

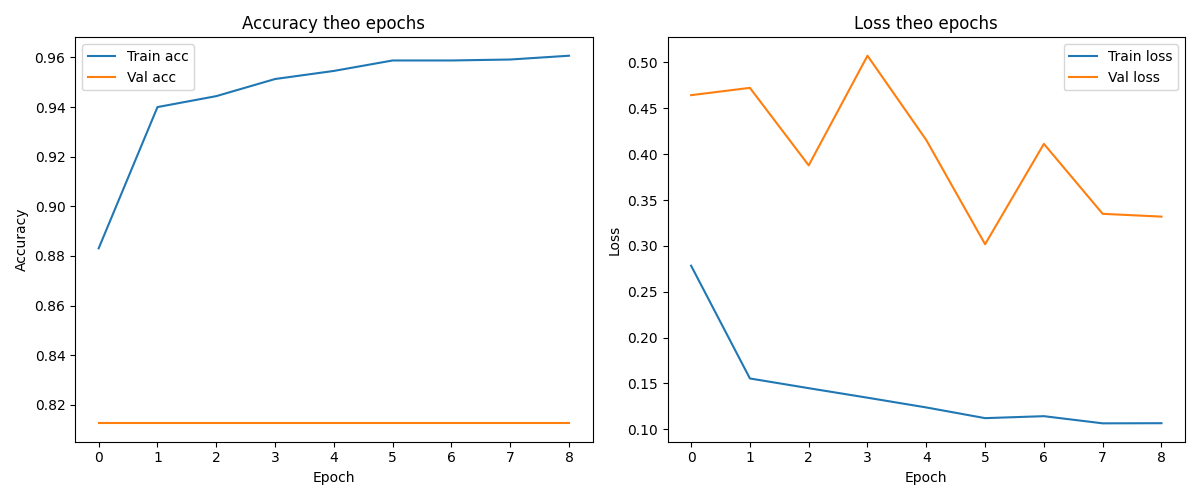
3.5.1. Mô hình CNN





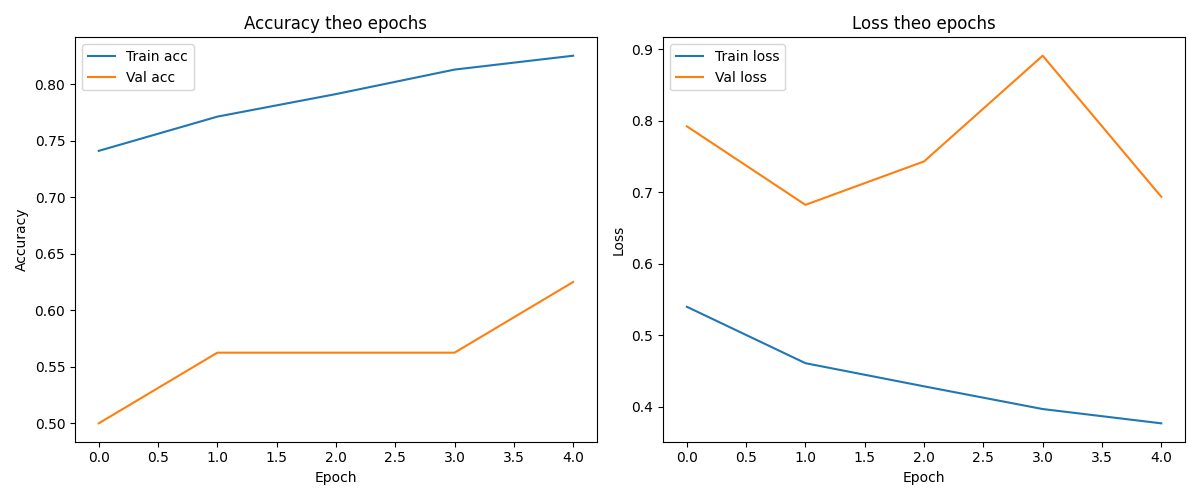
3.5.1. Mô hình VGG16





3.5.3 Mô hình ResNet50





**3.6 So sánh mô hình CNN, VGG16 và ResNet50**

Bảng dưới đây thể hiện sự so sánh giữa ba mô hình học sâu được sử dụng trong đề tài: CNN tự xây, VGG16 và ResNet50. Mỗi mô hình có những điểm mạnh và hạn chế riêng, phụ thuộc vào cấu trúc, độ sâu và khả năng tổng quát hóa.

|  |  |  |  |
| --- | --- | --- | --- |
| Mô hình | Ưu điểm chính | Nhược điểm chính | Hiệu quả thực tế |
| CNN tự xây | - Thiết kế đơn giản, dễ tùy chỉnh - Tốc độ huấn luyện nhanh, ít tài nguyên | - Hiệu suất hạn chế nếu không có dữ liệu lớn - Không tận dụng pre-trained model | Độ chính xác ổn định nhưng chưa tối ưu |
| VGG16 | - Mạnh trong trích xuất đặc trưng ảnh - Cấu trúc đồng nhất, dễ dùng | - Rất nhiều tham số (~138 triệu) - Tốn bộ nhớ, thời gian huấn luyện lâu | Độ chính xác cao, dễ triển khai |
| ResNet50 | - Khắc phục mất gradient - Tổng quát hóa tốt - Dễ hội tụ | - Cấu trúc phức tạp - Cần phần cứng mạnh hơn | Độ chính xác cao, phù hợp thực tế |

Nhìn chung, mô hình CNN tự xây dựng phù hợp với người mới bắt đầu, dễ triển khai nhưng cần nhiều tối ưu để đạt hiệu quả cao. VGG16 và ResNet50 đều là các mô hình mạnh, trong đó ResNet50 thể hiện độ chính xác vượt trội nhưng yêu cầu nhiều tài nguyên tính toán hơn.

**3.7 Tìm hiểu về Streamlit**

**Streamlit** là một thư viện mã nguồn mở bằng Python được thiết kế nhằm hỗ trợ việc xây dựng các ứng dụng web tương tác cho lĩnh vực khoa học dữ liệu và học máy một cách nhanh chóng, dễ dàng mà không cần kiến thức chuyên sâu về lập trình giao diện.

Streamlit cho phép các nhà nghiên cứu và lập trình viên triển khai trực tiếp các mô hình học máy và hiển thị kết quả thông qua giao diện người dùng thân thiện. Chỉ với vài dòng lệnh Python, người dùng có thể:

Tải lên dữ liệu và ảnh

Trực quan hóa biểu đồ, số liệu thống kê

Giao tiếp trực tiếp với mô hình học máy (như dự đoán, phân loại, nhận diện hình ảnh)

Tạo các thành phần giao diện như nút bấm, chọn mô hình, hiển thị kết quả...

Trong khuôn khổ đề tài, **Streamlit được sử dụng để xây dựng một ứng dụng web đơn giản**, giúp người dùng:

Tải ảnh X-ray lên hệ thống

Chọn mô hình muốn kiểm tra (CNN, VGG16 hoặc ResNet50)

Thực hiện dự đoán viêm phổi

Hiển thị kết quả bằng biểu đồ, bảng thống kê

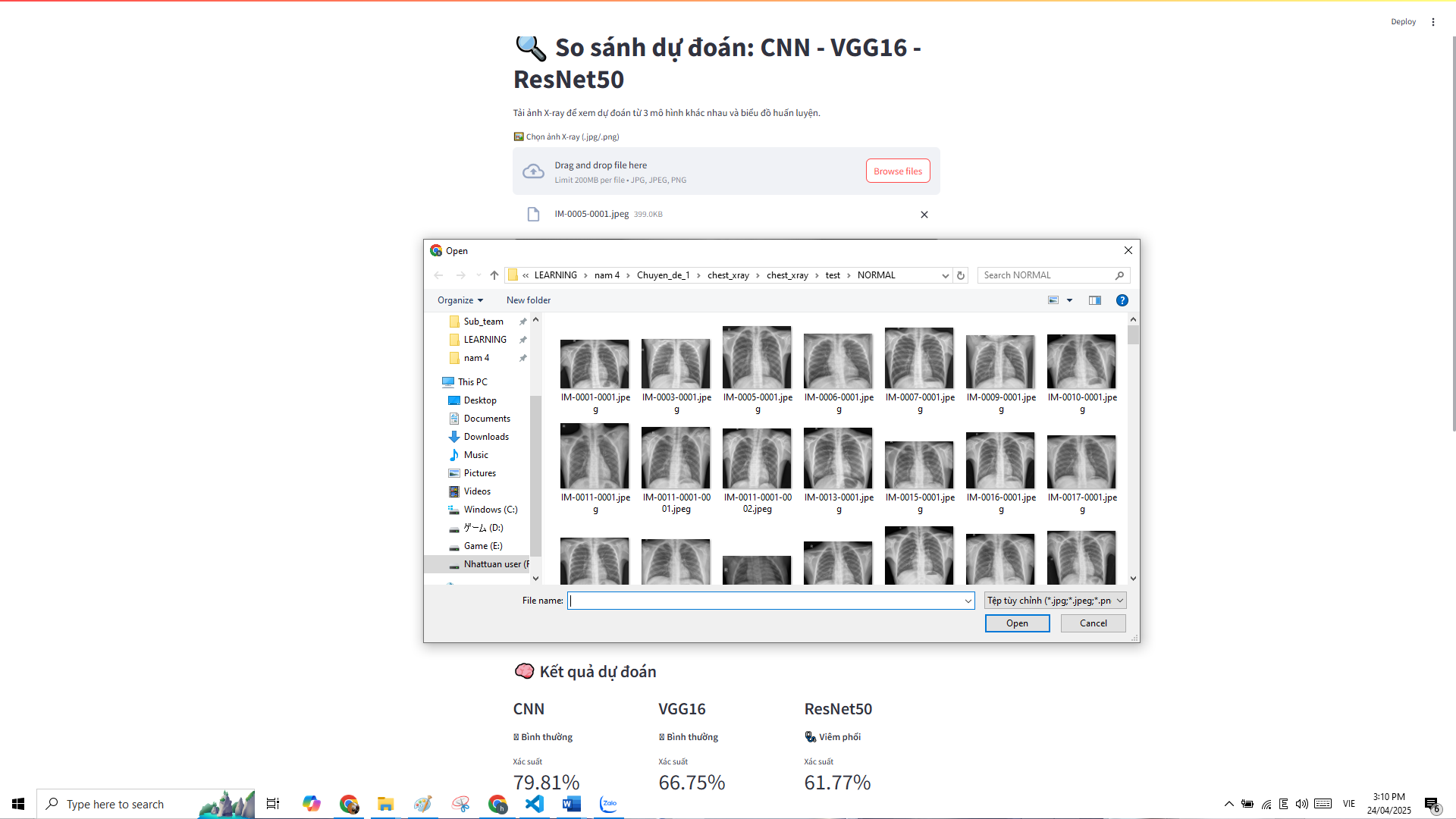
Tải về file báo cáo dưới dạng CSV hoặc PDF

Nhờ tính linh hoạt và tốc độ triển khai nhanh, Streamlit đã trở thành một lựa chọn lý tưởng cho các dự án trình bày kết quả học máy, đặc biệt trong bối cảnh cần tích hợp các mô hình AI vào hệ thống thực tế mà không cần phát triển phần mềm web từ đầu.

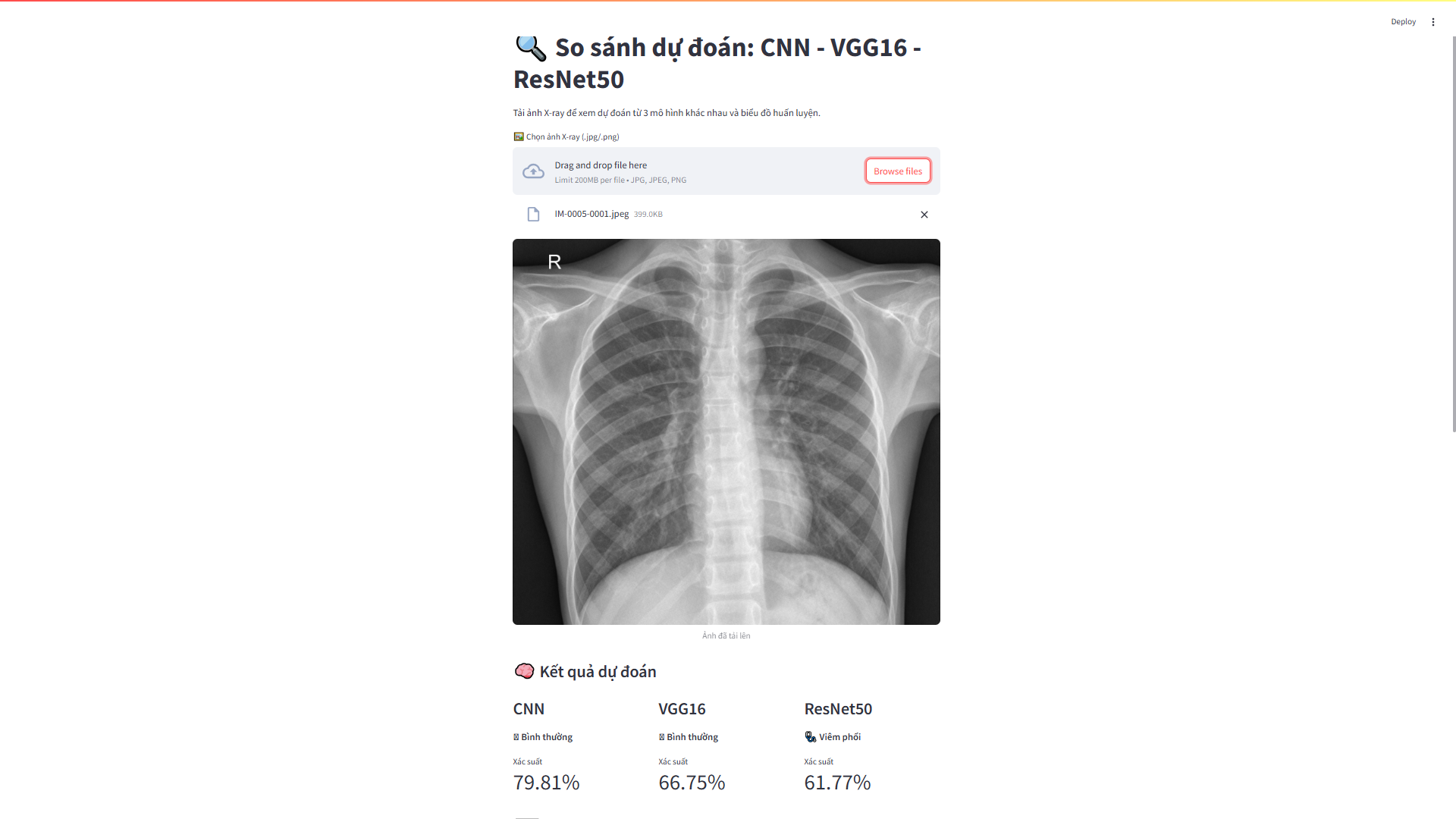
**3.8 Giao diện mô hình**

Cả 3 mô hình đều sử dụng chung một mã nguồn là streamlit để có thể đánh giá 1 cách toàn diện.

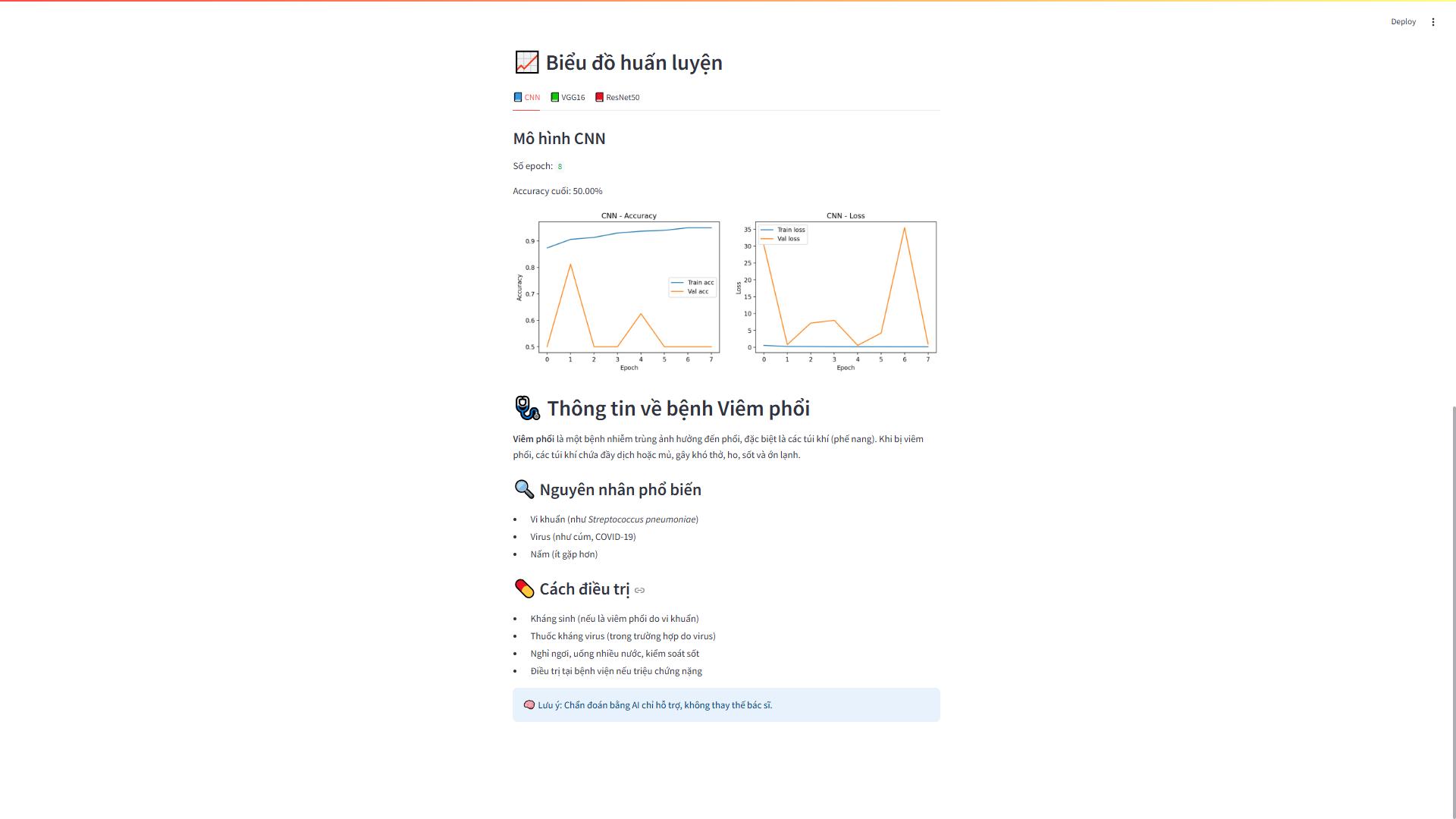
Giao diện chọn ảnh :



Giao diện đánh giá :



Giao diện so sánh :



**Đánh giá giao diện :**

Trong đề tài này, giao diện người dùng được xây dựng bằng thư viện **Streamlit**, một công cụ mã nguồn mở thân thiện với Python, chuyên dùng để tạo các ứng dụng web phục vụ trình bày mô hình học máy. Ứng dụng được thiết kế đơn giản, dễ sử dụng và hướng đến tính trực quan cho người dùng không chuyên.

Giao diện gồm các thành phần chính như:

**Khu vực tải ảnh X-ray**: Cho phép người dùng chọn ảnh từ thiết bị cá nhân.

**Lựa chọn mô hình**: Người dùng có thể chọn một hoặc nhiều mô hình để thực hiện dự đoán (CNN, VGG16, ResNet50).

**Nút kiểm tra**: Khi nhấn nút, hệ thống thực hiện phân loại ảnh và hiển thị kết quả.

**Hiển thị kết quả**: Gồm tên mô hình, dự đoán (viêm phổi / bình thường) và xác suất dự đoán. Các kết quả được trình bày dạng bảng và biểu đồ.

**Xuất báo cáo**: Cho phép người dùng tải về kết quả dưới dạng **file .CSV** và **file .PDF** phục vụ lưu trữ hoặc báo cáo y tế.

KẾT LUẬN

Sau thời gian nghiên cứu đề tài, dưới sự chỉ dẫn giúp đỡ nhiệt tình của giáo viên hướng dẫn đã giúp em có thêm nhiều hiểu biết và kiến thức về học máy, khai phá dữ liệu và đặc biệt là cách sử dụng công cụ Orange Data Mining. Một số kết quả đạt được bao gồm:

1. Tìm hiểu được tổng quan về Học máy (Machine learning).

2. Tìm hiểu được công cụ Orange Data Mining.

3. Ứng dụng được công cụ Orange Data Mining trong bài toán dự đoán mức độ ung thư phổi.

Bài báo cáo của em đã hoàn thành, tuy nhiên còn nhiều nội dung nâng cao cần nhiều thời gian để nghiên cứu trong tương lai. Em rất mong được sự đóng góp ý kiến của các thầy cô và các bạn để cuốn báo cáo này được hoàn thiện.

TÀI LIỆU THAM KHẢO

1. Đỗ Phúc, Giáo trình khai phá dữ liệu (Data mining), NXB Đại học Quốc gia TP. Hồ Chí Minh, 2017.
2. Nguyễn Hà Nam, Nguyễn Trí Thành, Hà Quang Thụy, Giáo trình khai phá dữ liệu, NXB Đại học Quốc gia Hà Nội, 2013.
3. Orange Data Mining, Online: https://orangedatamining.com/download/ #windows, Truy cập ngày 14/02/2023.
4. Data mining sử dụng Orange, Online: https://websitehcm.com/data-mining-su-dung-orange/, Truy cập ngày 20/02/2023.
5. Data mining là gì? Tổng hợp các thông tin mới nhất về data mining, Online: https://blog.cole.vn/data-mining-la-gi/, Truy cập ngày 20/02/2023.