PHÁT HIỆN CÁC BỆNH VỀ PHỔI DỰA TRÊN HÌNH ẢNH Y TẾ SỬ DỤNG MÔ HÌNH HỌC SÂU

Khoa Toán - Cơ - Tin học

Trường Đại học Khoa học Tự nhiên - ĐHQGHN

Sinh viên thực hiện:
Hà Vũ Minh Đức
Bùi Xuân Hòa
Ngô Sĩ Hiếu
Nguyễn Quang Hướng

Giáo viên hướng dẫn:
TS. Nguyễn Hải Vinh

Ngày 16 tháng 11 năm 2024

Mục lục

Danh sách hình vẽ

Danh sách bảng

Giới thiệu

Trong thời đại công nghiệp phát triển như vũ bão, các vấn đề ô nhiễm không khí trở nên gia tăng và ảnh hưởng trực tiếp đến sức khỏe của con người, đặc biệt là phổi. Việc phát hiện sớm và chính xác các bệnh về phổi từ hình ảnh y tế đóng vai trò quan trọng trong việc cải thiện quá trình chẩn đoán và điều trị. Bài báo cáo này đề xuất một phương pháp dựa trên mô hình học sâu để tự động phát hiện một số bệnh lý phổi. Mô hình sử dụng thuật toán phát hiện You Only Look Once (YOLO) và mạng nơ-ron tích chập (CNN) để phân tích hình ảnh, kết hợp với các kỹ thuật xử lý và tăng cường dữ liệu nhằm cải thiện độ chính xác của hệ thống. Bộ dữ liệu bao gồm một số dị thường xuất hiện khá phổ biến trên phổi như xuất huyết phổi, tràn dịch màng phổi, tràn khí màng phổi và đặc biệt là ung thư. Thực nghiệm cho thấy mô hình đạt hiệu suất cao với độ chính xác. Kết quả nghiên cứu chứng minh tiềm năng của học sâu trong việc hỗ trợ các chuyên gia y tế trong quá trình chẩn đoán và điều trị các bệnh về u phổi. Bài báo cáo gồm những nội dung chính sau:

- Cơ sở lý thuyết liên quan: các bệnh lý về phổi(định nghĩa, cách nhận dạng), các mô hình học sâu sử dụng(YOLO, CNN)
- Xây dựng mô hình để phát hiện và chẩn đoán khối bệnh
- Kết quả thực nghiệm mô hình và đánh giá độ chính xác

Chương 1

Một số lý thuyết cơ sở liên quan

1.1 Một số khái niệm về sinh học được sử dụng

Trước khi tìm hiểu kỹ hơn về bài báo cáo, ta cần biết một số định nghĩa sau:

1. **Phổi** là cơ quan trao đổi khí của nhiều loài động vật thuộc lớp bò sát, chim, thú. Phổi cùng với đường dẫn khí, cơ hô hấp tạo nên hệ hô hấp của người. [1]

2. **Chụp cắt lớp(Chụp CT)** là một kỹ thuật mà trong đó một chùm tia X hẹp được chiếu vào bệnh nhân và quay nhanh xung quanh cơ thể. Tín hiệu thu được từ chùm tia X này sẽ được xử lý bởi máy tính để tạo ra các hình ảnh cắt ngang, gọi là ảnh cắt lớp (tomographic images). Những hình ảnh này cung cấp thông tin chi tiết hơn so với ảnh chụp X-quang thông thường. Khi máy tính thu thập được một loạt ảnh cắt lớp liên tiếp, chúng có thể được "xếp chồng"kỹ thuật số để tạo thành một hình ảnh ba chiều (3D) của cơ thể bệnh nhân. Điều này giúp dễ dàng xác định các cấu trúc cơ bản cũng như các khối u hoặc bất thường có thể có.[2]

3. **Xuất huyết(Hemorrhage)** là tình trạng chảy máu hoặc mất máu bất thường ra khỏi hệ tuần hoàn. Xuất huyết có thể xảy ra bên trong cơ thể, khi máu rò rỉ vào các khoang hoặc mô, hoặc bên ngoài, khi tổn thương làm da bị rách. Mức độ nghiêm trọng của xuất huyết có thể khác nhau, từ chảy máu nhẹ đến tình huống đe dọa tính mạng, tùy thuộc vào nguyên nhân và vị trí. **Xuất huyết phổi** là tình trạng máu chảy vào nhu mô phổi, phế nang, hoặc đường thở.

4. **Ung thư phổi(Lung cancer)** là một loại ung thư bắt đầu khi các tế bào bất thường phát triển không kiểm soát trong phổi. Các triệu chứng của ung thư phổi bao gồm ho không dứt, đau ngực và khó thở. Ung thư phổi là một vấn đề y tế công cộng nghiêm trọng, gây ra số lượng tử vong đáng kể trên toàn cầu. Theo ước tính của GLOBOCAN 2020, do Cơ quan Nghiên cứu Ung thư Quốc tế (IARC) thực hiện, ung thư phổi tiếp tục là nguyên nhân hàng đầu gây tử vong do ung thư, với ước tính khoảng 1,8 triệu ca tử vong, chiếm 18% tổng số ca tử vong do ung thư trong năm 2020. [3]

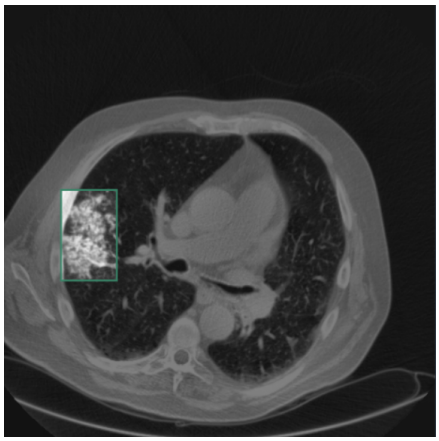
5. **Tràn dịch màng phổi(Pleural effusion)** là tình trạng tích tụ quá mức của dịch trong khoang màng phổi. Bệnh nhân thường có triệu chứng như khó thở khi gắng sức, ho khan và đau ngực. [**Karkhanis2012**]

6. **Tràn khí màng phổi(Pneumothorax)** hay còn được viết tắt là PNX, xảy ra khi không khí bị rò rỉ vào không gian giữa phổi và thành ngực. Không khí này đẩy lên phía ngoài của phổi và khiến phổi bị xẹp. Pneumothorax có thể gây xẹp hoàn toàn phổi hoặc chỉ một phần phổi bị xẹp.[4]

1.2 Đặc điểm trên ảnh chụp CT của các dị thường phổi đã giới thiệu

Bốn dị thường phổi nói trên, bao gồm xuất huyết, ung thư, tràn dịch màng phổi và tràn khí màng phổi có thể được phát hiện và phân tích thông qua các đặc điểm trên hình ảnh cụ thể. Mỗi loại dị thường có những dấu hiệu riêng biệt, từ sự thay đổi về mật độ đến cấu trúc và hình dạng, cho phép bác sĩ chẩn đoán chính xác và định hướng điều trị phù hợp.

- **Xuất huyết**: các dấu hiệu trên CT thường bao gồm các tổn thương lan tỏa hình mảng ở vùng quanh cửa phổi hai bên, phản ánh sự viêm nhiễm hoặc xuất huyết ở khu vực trung tâm của phổi. Ngoài ra, các nốt kính mờ dạng trung tâm tiểu thùy (ground-glass centrilobular nodules) cũng có thể được quan sát, cho thấy một quá trình lan tỏa trong nhu mô phổi. Trong một số trường hợp, có thể thấy khuyết tật trong lòng các đường thở chính, thường do cục máu đông gây tắc nghẽn dòng khí. [5]



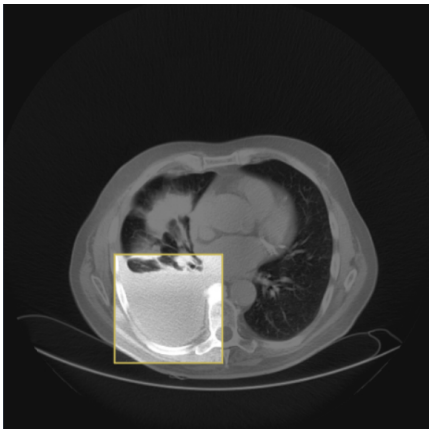
Hình 1.1: Phổi có xuất huyết

- **Ung thư phổi**: các nốt hoặc khối có bờ không đều, kèm theo tăng sinh mạch máu xung quanh. Khối lớn với kích thước thay đổi, xâm lấn các mô lân cận như màng phổi, mạch máu lớn hoặc xương sườn, cũng là dấu hiệu quan trọng. Bên cạnh đó, ung thư phổi có thể gây phì đại hạch bạch huyết vùng trung thất hoặc quanh rốn phổi.

Trong trường hợp lan rộng, các tổn thương thường xuất hiện dưới dạng nốt phổi liên quan đến khí quản hoặc phế quản, làm tăng nghi ngờ về sự di căn.

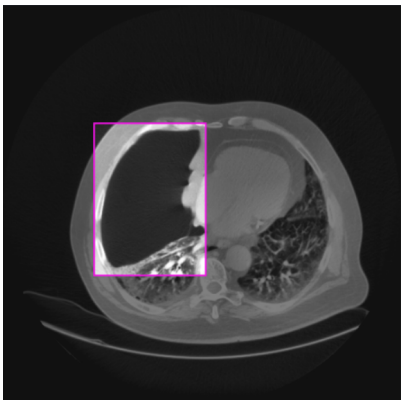
Hình 1.2: Ung thư phổi

- **Tràn dịch màng phổi**: Trên ảnh CT, tràn dịch màng phổi thường xuất hiện dưới dạng một mờ hình liềm ở phần thấp nhất của lồng ngực, nơi dịch tụ lại. Nếu bệnh nhân nằm ngửa, dịch sẽ tích tụ ở liên sườn phía sau. Tràn dịch màng phổi có thể có hình dạng lưỡi liềm, với các đường biên mượt mà và độ xóa mờ đồng nhất. CT còn có thể phát hiện sự dày lên của màng phổi, các nốt màng phổi, hoặc sự khu trú của dịch, đặc biệt trong trường hợp tràn dịch tiết. [6]



Hình 1.3: Tràn dịch màng phổi

- **Tràn khí màng phổi**: Trên ảnh, tràn khí màng phổi thường được phát hiện dễ dàng khi xem ở cửa sổ phổi. Đặc trưng của nó là sự hiện diện của một khoảng không khí giữa màng phổi tạng và màng phổi thành, được biểu thị bằng sự thiếu vắng các cấu trúc mạch máu trong không gian đó. Phần khí thường được biểu hiện ở phía góc trên của ảnh CT, chèn ép khung xương sườn và các tế bào phế nang của phổi.



Hình 1.4: Tràn khí màng phổi

1.3 Giới thiệu bài toán chẩn đoán các bệnh về phổi thông qua ảnh X-quang

Chụp cắt lớp là một trong những kỹ thuật y tế phổ biến để phát hiện các bệnh về phổi hiện nay, tuy nhiên, thời gian trả kết quả cho người bệnh thường mất một khoảng thời gian tương đối. Vì vậy xây dựng một mô hình có khả năng phát hiện và chẩn đoán các bệnh đó là cần thiết để tiết kiệm thời gian và vật lực cho các bệnh viện mà vẫn đảm bảo được độ chính xác cao. Mục tiêu của bài toán là từ hình ảnh CT của người khám, xác định được căn bệnh về phổi mà người đó mắc và xác định vị trí của bệnh trong phổi.

Để đạt được mục tiêu này, các kỹ thuật học sâu cần phải được sử dụng. Các kỹ thuật này có khả năng phát hiện các dấu hiệu bất thường trên ảnh cắt lớp. Bên cạnh đó, việc sử dụng tập dữ liệu hợp lý sẽ giúp mô hình học được đặc điểm của từng bệnh từ đó phân loại.

Bài báo cáo này sẽ trình bày quy trình xây dựng mô hình từ việc tiền xử lý dữ liệu đến lựa chọn mô hình đảm bảo hiệu quả cao trong thực nghiệm. Bên cạnh đó, việc đánh giá và tối ưu hóa hiệu suất của mô hình cũng sẽ được đề cập.

Chương 2

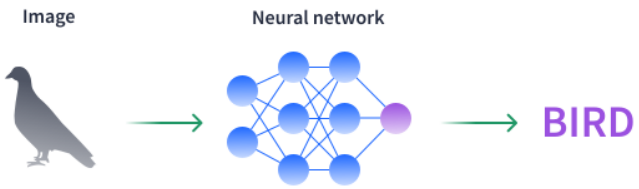
Xây dựng mô hình học sâu để dự đoán các bệnh về phổi từ ảnh X-quang

2.1 Học sâu

Các thông tin dưới đây được lấy trên trang thông tin ”dataquest.io” [7] :

Học sâu (Deep Learning) là một phương pháp học máy cho phép máy tự động học các biểu diễn đặc trưng "cấp cao"của dữ liệu. Nhờ vậy, các mô hình học sâu đạt được kết quả tốt nhất trong các nhiệm vụ xử lý ảnh hoặc xử lý ngôn ngữ tự nhiên.

Thuật toán học sâu sử dụng mạng nơ ron nhân tạo, một hệ thống tính toán có khả năng học các đặc trưng cấp cao từ dữ liệu bằng cách tăng độ sâu (tức là số tầng) trong mạng. Mạng nơ-ron được lấy cảm hứng một phần từ mạng nơ-ron thần kinh sinh học, nơi các tế bào trong hầu hết các bộ não (bao gồm cả bộ não của chúng ta) kết nối và hoạt động cùng nhau. Mỗi tế bào trong một mạng nơ-ron được gọi là một nơ-ron.



Hình 2.1: Mô hình học sâu

Một mạng nơ ron nhân tạo gồm 4 phần:

1) **Lớp đầu vào(Input Layer)**: Đây là nơi các quan sát huấn luyện được đưa vào thông qua các biến độc lập.

2) **Các lớp ẩn(Hidden Layers)**: Đây là các lớp trung gian giữa lớp đầu vào và lớp đầu ra, nơi mạng nơ-ron học về các mối quan hệ và tương tác của các biến được đưa vào lớp đầu vào.

3) **Lớp đầu ra(Output layer)**: Đây là lớp nơi kết quả cuối cùng được trích xuất sau tất cả các quá trình xử lý diễn ra trong các lớp ẩn.

4) **Nút(Node)**: Một nút, còn được gọi là một nơ-ron, trong mạng nơ-ron là một đơn vị tính toán nhận vào một hoặc nhiều giá trị đầu vào và tạo ra một giá trị đầu ra.

Học sâu hoạt động bằng cách lấy dữ liệu đầu vào và đưa nó vào một mạng lưới các nơ-ron nhân tạo. Mỗi nơ-ron nhận đầu vào từ lớp nơ-ron trước đó và sử dụng thông tin này để nhận diện các mẫu trong dữ liệu. Các nơ-ron sau đó gán trọng số cho dữ liệu đầu vào và đưa ra dự đoán về đầu ra. Đầu ra có thể là một lớp hoặc nhãn.

Ta đến với activation function của học sâu. Hãy tưởng tượng một mạng nơ-ron không có hàm kích hoạt. Trong trường hợp đó, mỗi nơ-ron sẽ chỉ thực hiện phép biến đổi tuyến tính trên các đầu vào bằng cách sử dụng trọng số và độ lệch. Mặc dù phép biến đổi tuyến tính làm cho mạng nơ-ron đơn giản hơn, nhưng mạng này sẽ kém mạnh hơn và sẽ không thể học được các mẫu phức tạp từ dữ liệu.

Một mạng nơ-ron không có hàm kích hoạt trong học sâu về cơ bản chỉ là một mô hình hồi quy tuyến tính. Do đó, chúng ta sử dụng phép biến đổi phi tuyến tính đối với các đầu vào của nơ-ron và tính phi tuyến tính này trong mạng được đưa vào bằng một hàm kích hoạt. Dưới đây là một số hàm kích hoạt phổ biến

Đầu tiên hàm kích hoạt sẽ là bộ phân loại dựa trên ngưỡng, tức là liệu nơ-ron có được kích hoạt hay không dựa trên giá trị từ phép biến đổi tuyến tính. **Hàm bước nhị phân (Binary Step Function)**: nếu đầu vào của hàm kích hoạt lớn hơn ngưỡng, thì nơ-ron được kích hoạt, nếu không thì nó bị vô hiệu hóa, tức là đầu ra của nó không được xem xét cho lớp ẩn tiếp theo:

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2.1}$$

Hàm bước nhị phân có thể được sử dụng như một hàm kích hoạt trong khi tạo một bộ phân loại nhị phân. Hàm này sẽ không hữu ích khi có nhiều lớp trong biến mục tiêu. Đó là một trong những hạn chế của hàm bước nhị phân. Hơn nữa, gradient của hàm bước bằng 0 gây cản trở cho quá trình truyền ngược. Nghĩa là, nếu bạn tính đạo hàm của f(x) theo x, kết quả sẽ bằng 0. Điều này là do không có thành phần nào của x trong hàm bước nhị phân. Thay vì hàm nhị phân, chúng ta có thể sử dụng **hàm tuyến tính (Linear Function)**. Định nghĩa hàm như sau:

$$f(x) = ax \tag{2.2}$$

trong đó:

Độ kích hoạt tỷ lệ thuận với đầu vào.

Biến 'a' trong trường hợp này có thể là bất kỳ giá trị hằng số nào.

Mặc dù gradient ở đây không trở thành số không, nhưng nó là một hằng số không phụ thuộc vào giá trị đầu vào x chút nào. Điều này ngụ ý rằng trọng số và độ lệch sẽ được cập nhật trong quá trình truyền ngược nhưng hệ số cập nhật sẽ giống nhau.

Hàm kích hoạt tiếp theo trong học sâu mà chúng ta sẽ xem xét là **hàm kích hoạt Sigmoid (Sigmoid Activation Function)**. Đây là một trong những hàm kích hoạt phi tuyến tính được sử dụng rộng rãi nhất. Sigmoid biến đổi các giá trị giữa phạm vi 0 và 1. Sau đây là biểu thức toán học cho sigmoid:

$$\text{Sigmoid(x)} = \sigma(x) = \frac{1}{1+e^{-x}} \tag{2.3}$$

Sigmoid là một hàm phi tuyến tính vậy nên đầu ra cũng phi tuyến tính. Ngoài ra, hàm sigmoid không đối xứng quanh số không. Vì vậy, đầu ra của tất cả các neuron sẽ có cùng dấu. Điều này có thể được giải quyết bằng cách mở rộng hàm sigmoid, đây chính xác là những gì xảy ra trong **hàm Tanh**.

Hàm Tanh (Tanh) rất giống với hàm sigmoid. Điểm khác biệt duy nhất là nó đối xứng quanh gốc tọa độ. Phạm vi giá trị trong trường hợp này là từ -1 đến 1. Do đó, các đầu vào cho các lớp tiếp theo sẽ không phải lúc nào cũng cùng dấu. Hàm tanh được định nghĩa là:

$$Tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.4}$$

Tanh thường được ưa chuộng hơn hàm sigmoid vì nó có tâm bằng không và độ dốc không bị giới hạn di chuyển theo một hướng nhất định.

Hàm kích hoạt ReLU (ReLU Activation Function) là một hàm kích hoạt phi tuyến tính khác đã trở nên phổ biến trong lĩnh vực học sâu. ReLU là viết tắt của Rectified Linear Unit. Ưu điểm chính của việc sử dụng hàm ReLU so với các hàm kích hoạt khác là nó không kích hoạt tất cả các neuron cùng một lúc. Điều này có nghĩa là các tế bào thần kinh sẽ chỉ bị vô hiệu hóa nếu đầu ra của phép biến đổi tuyến tính nhỏ hơn 0. Biểu thức

$$ReLU(x) = \max(0, x) \tag{2.5}$$

Đối với các giá trị đầu vào âm, kết quả là số không, nghĩa là neuron không được kích hoạt. Vì chỉ có một số neuron nhất định được kích hoạt, nên hàm ReLU hiệu quả hơn nhiều về mặt tính toán khi so sánh với hàm sigmoid và tanh. Nhưng trong quá trình truyền ngược, trọng số và độ lệch cho một số nơ-ron không được cập nhật bởi vì giá trị gradient bằng 0. Điều này có thể tạo ra các nơ-ron chết không bao giờ được kích hoạt. Điều này được xử lý bằng **'Leaky' ReLU function**.

$$f(x) = \begin{cases} 0.01x, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{2.6}$$

Hàm Leaky ReLU không gì khác ngoài phiên bản cải tiến của hàm ReLU. Thay vì định nghĩa hàm Relu là 0 cho các giá trị âm của x, nó được định nghĩa là một thành phần tuyến tính cực nhỏ của x. Bằng cách thực hiện sửa đổi nhỏ này, gradient của phía bên trái của đồ thị sẽ trở thành một giá trị khác không. Do đó, sẽ không còn gặp phải các tế bào thần kinh chết trong vùng đó nữa.

Exponential Linear Unit hay viết tắt là **ELU** cũng là một biến thể của Rectiufied Linear Unit (ReLU) sửa đổi độ dốc của phần âm của hàm. Không giống như leaky relu và các hàm ReLU tham số, thay vì đường thẳng, ELU sử dụng đường cong logarit để xác định các giá trị âm. Nó được định nghĩa là:

$$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases} \tag{2.7}$$

Tiếp đến là **Swish**, Swish là một hàm kích hoạt ít được biết đến hơn được các nhà nghiên cứu tại Google phát hiện. Swish có hiệu suất tính toán tương đương với ReLU và cho thấy hiệu suất tốt hơn ReLU trên các mô hình sâu hơn. Các giá trị của swish nằm trong khoảng từ âm vô cực đến vô cực. Hàm được định nghĩa là:

$$\text{Swish}(x) = x \cdot \text{Sigmoid}(x) = x \cdot \frac{1}{1+e^{-x}} \tag{2.8}$$

Một sự thật độc đáo về hàm này là 'swich function is not monotonic'. Điều này có nghĩa là giá trị của hàm có thể giảm ngay cả khi các giá trị đầu vào đang tăng.

Hàm Softmax thường được mô tả như một sự kết hợp của nhiều sigmoid. Sigmoid trả về các giá trị từ 0 đến 1, có thể được coi là xác suất của một điểm dữ liệu thuộc về một lớp cụ thể. Do đó, sigmoid được sử dụng rộng rãi cho các vấn đề phân loại nhị phân.

Hàm softmax có thể được sử dụng cho các bài toán phân loại đa lớp. Hàm này trả về xác suất cho một điểm dữ liệu thuộc về từng lớp riêng lẻ.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.9}$$

Trong khi xây dựng mạng cho một vấn đề đa lớp, lớp đầu ra sẽ có số lượng nơ-ron bằng số lớp trong mục tiêu. Ví dụ, nếu bạn có ba lớp, sẽ có ba nơ-ron trong lớp đầu ra.

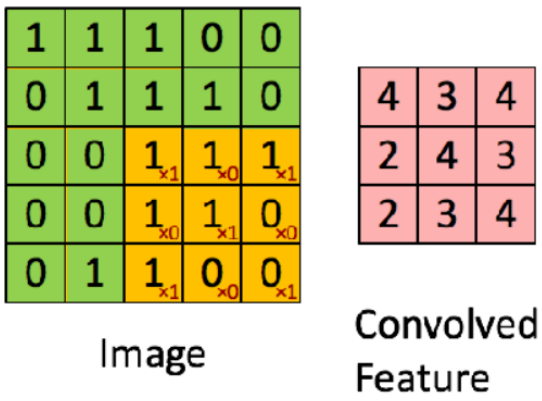
2.2 Mạng Nơ-ron tích chập(CNN)

Mạng Nơ-ron tích chập (Convolutional Neural Network) hay còn được gọi là CNN, là một trong những mô hình Deep Learning cực kỳ tiên tiến cho phép xây dựng những hệ thống có độ chính xác cao và thông minh. Nhờ khả năng đó, CNN có rất nhiều ứng dụng, đặc biệt là những bài toán cần nhận dạng vật thể (object) để phát hiện và phân loại các đối tượng trong hình ảnh. CNN vô cùng quan trọng để tạo nên những hệ thống nhận diện

thông minh với độ chính xác cao trong thời đại công nghệ ngày nay. Mô hình ảnh đầu vào sẽ chuyển nó qua một loạt các lớp tích chập với các bộ lọc, sau đó đến lớp Pooling, rồi tiếp theo là các lớp được kết nối đầy đủ (FC — fully connected layers) và cuối cùng áp dụng hàm softmax để phân loại một đối tượng dựa trên giá trị xác suất trong khoản từ 0 đến 1.

2.2.1 Lớp tích chập

Lớp tích chập (Convolution) là lớp đầu tiên thực hiện việc trích xuất đặc trưng từ hình ảnh. Lớp này có các tham số bao gồm một tập hợp các bộ lọc có thể học được. Các bộ lọc này thường có kích thước nhỏ, thường là 3x3 hoặc 5x5 ở hai chiều đầu tiên, và độ sâu tương đương với độ sâu của đầu vào. Khi các bộ lọc này di chuyển dọc và ngang trên hình ảnh, chúng tạo ra một bản đồ đặc trưng (Feature Map) chứa các đặc điểm được trích xuất từ hình ảnh đầu vào.

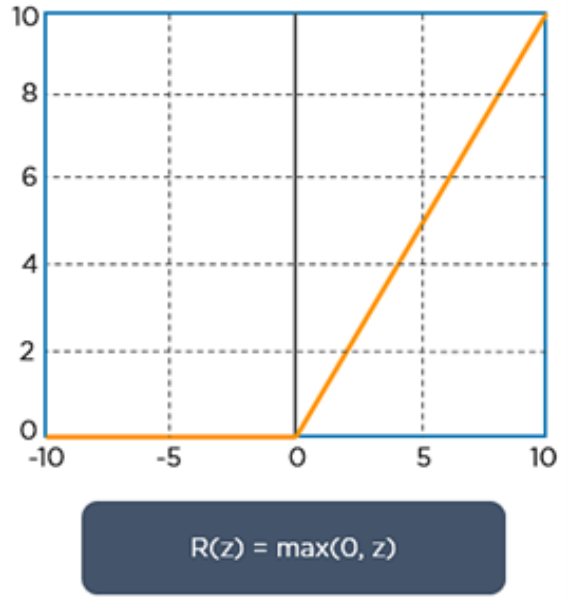


Hình 2.2: Phép toán tích chập

2.2.2 Lớp ReLU - Rectified Linear Unit

Sau khi trích xuất được các bản đồ đặc điểm, bước tiếp theo là chuyển chúng đến một lớp ReLU (Rectified Linear Unit). Lớp này thực hiện một phép biến đổi phi tuyến đơn giản nhưng vô cùng hiệu quả, bằng cách áp dụng một hàm kích hoạt theo từng phần tử, trong đó mọi giá trị âm đều được thay thế bằng 0, trong khi các giá trị dương vẫn được giữ nguyên. Điều này giúp đưa tính phi tuyến vào mạng, giúp mô hình có khả năng học các mối quan hệ phức tạp hơn từ dữ liệu.

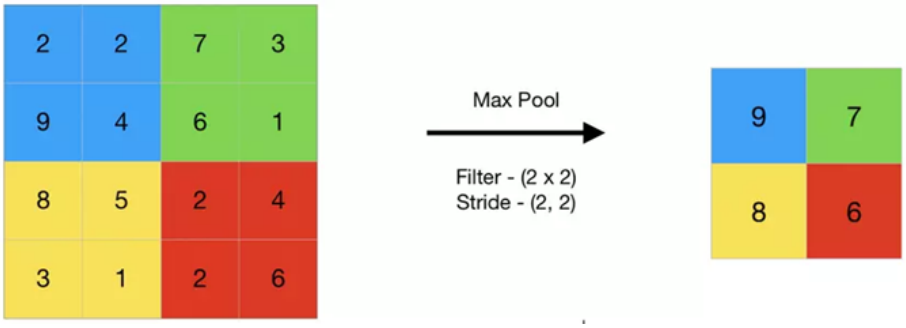
Ngoài ra, việc chỉnh lưu các giá trị âm cũng góp phần giảm thiểu nguy cơ gradient bị triệt tiêu trong quá trình huấn luyện, một vấn đề thường gặp ở các mô hình sử dụng hàm kích hoạt tuyến tính hoặc sigmoid. Đầu ra của lớp ReLU là một bản đồ tính năng đã được chỉnh lưu, cung cấp thông tin quan trọng để tiếp tục các bước xử lý và trích xuất đặc điểm sâu hơn trong mạng.



Hình 2.3: Đồ thị hàm ReLU

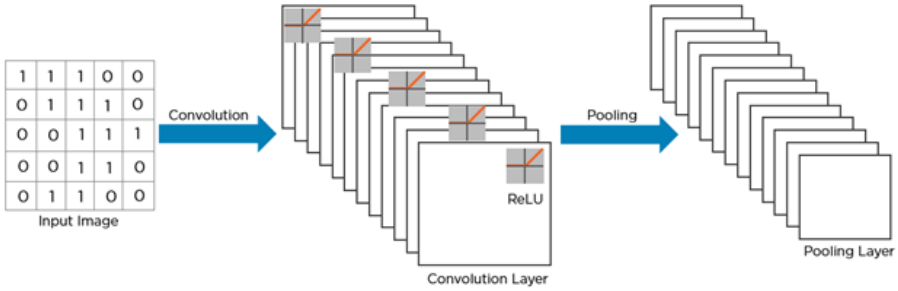
2.2.3 Pooling layer – Lớp gộp

Pooling layer thường được dùng giữa các convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm việc tính toán trong model, giảm độ phức tạp của mô hình và tránh overfitting.Các pooling có thể có nhiều loại khác nhau: **Max Pooling**, **Average Pooling**, **Sum Pooling**. Phổ biến là Max Pooling và Average Pooling. Trong đó Max pooling lấy phần tử lớn nhất từ ma trận đối tượng, hoặc lấy tổng trung bình. Tổng tất cả các phần tử trong map gọi là Sum Pooling



Hình 2.4: Pooling layer

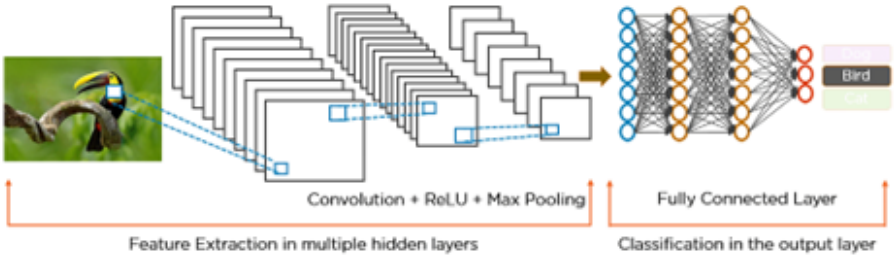
Cấu trúc của mạng nơ-ron tích chập trông như thế này cho đến nay:



Hình 2.5: Cấu trúc mạng nơ-ron tích chập

2.2.4 Lớp Fully Connected

Sau khi ảnh được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của ảnh thì tensor của output của layer cuối cùng sẽ được là phẳng thành vector và đưa vào một lớp được kết nối như một mạng nơ-ron. Với FC layer được kết hợp với các tính năng lại với nhau để tạo ra một mô hình. Cuối cùng sử dụng softmax hoặc sigmoid để phân loại đầu ra.



Hình 2.6: Fully Connected Layer

2.2.5 Cấu trúc của mạng CNN

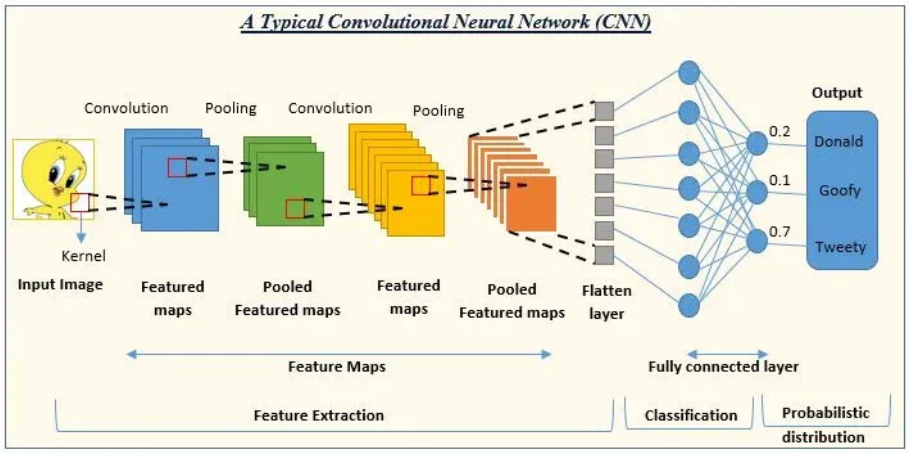
Mạng CNN là một trong những tập hợp của lớp Convolution được chồng lên nhau. Mạng CNN còn sử dụng các hàm nonlinear activation (như ReLU và tanh) nhằm kích hoạt trọng số trong node. Khi đã thông qua hàm, lớp này sẽ thu được trọng số trong các node và tạo ra nhiều thông tin trừu tượng hơn cho các lớp kế cận.

Đặc điểm mô hình CNN có 2 khía cạnh cần phải đặc biệt lưu ý là tính bất biến và tính kết hợp, do đó độ chính xác hoàn toàn có thể bị ảnh hưởng nếu có cùng một đối tượng được chiếu theo nhiều phương diện khác biệt. Với các loại chuyển dịch, co giãn và quay, người ta sẽ sử dụng pooli layer và làm bất biến những tính chất này. Từ đó, CNN sẽ cho ra kết quả có độ chính xác ứng với từng loại mô hình.

Pooling layer giúp tạo nên tính bất biến đối với phép dịch chuyển, phép co giãn và phép quay. Trong khi đó, tính kết hợp cục bộ sẽ thể hiện các cấp độ biểu diễn, thông tin từ mức độ thấp đến cao, cùng độ trừu tượng thông qua convolution từ các filter. Dựa trên cơ chế convolution, một mô hình sẽ liên kết được các layer với nhau.

Với cơ chế này, layer tiếp theo sẽ là kết quả được tạo ra từ convolution thuộc layer kế trước. Điều này đảm bảo bạn có được kết nối cục bộ hiệu quả nhất. Mỗi nơ-ron sinh ra ở

lớp tiếp theo từ kết quả filter sẽ áp đặt lên vùng ảnh cục bộ của nơ-ron tương ứng trước đó. Cũng có một số layer khác như pooling/subsampling layer được dùng để chắt lọc lại các thông tin hữu ích hơn.



Hình 2.7: Cấu trúc mạng CNN

2.2.6 Convolutional Neural Network Training

Đào tạo Mạng nơ-ron tích chập (CNN) bao gồm hướng dẫn mô hình nhận dạng các mẫu trong dữ liệu thông qua quy trình học từng bước. Điều này thường được thực hiện bằng cách sử dụng học có giám sát, trong đó CNN được cung cấp một loạt hình ảnh có nhãn chính xác và dần dần học cách liên kết hình ảnh với nhãn phù hợp. Sau đây là cách quy trình hoạt động:

- Chuẩn bị dữ liệu - Data Preparation

Trước tiên, hình ảnh cần được chuẩn bị trước khi quá trình đào tạo có thể bắt đầu. Điều này có nghĩa là đảm bảo tất cả hình ảnh đều đồng nhất về định dạng và kích thước. Bằng cách xử lý trước dữ liệu theo cách này, bạn đảm bảo rằng CNN nhận được đầu vào nhất quán, điều này rất quan trọng cho quá trình học của nó.

- Chức năng mất mát - Loss Function

Khi hình ảnh đã sẵn sàng, bước tiếp theo là tìm hiểu xem CNN đang hoạt động tốt như thế nào. Đây là lúc hàm mất mát phát huy tác dụng. Hãy coi nó như một bảng điểm đo lường sự khác biệt giữa những gì mô hình dự đoán và nhãn thực tế của hình ảnh. Sự khác biệt càng nhỏ thì mô hình hoạt động càng tốt, do đó mục tiêu là giảm khoảng cách này càng nhiều càng tốt.

Loss Function có 3 loại chính là: Mean Squared Error Loss Function, Cross-Entropy Loss Function, Mean Absolute Percentage Error.

Mean Squared Error (MSE) loss function là tổng bình phương các hiệu số giữa các

mục trong vectơ dự đoán y và vectơ thực tế y-hat.

$$\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y}_i)^2$$

trong đó:

y_i: các giá trị trong vector dự đoán $\vec{y}$.

$\hat{y}_i$: các giá trị trong nhãn thực $\hat{\vec{y}}$.

Hàm MSE là hàm mất mát hoàn hảo nếu đang giải quyết vấn đề hồi quy. Nghĩa là, nếu muốn mạng nơ-ron của dự đoán giá trị vô hướng liên tục.

Hồi quy chỉ là một trong hai lĩnh vực mà mạng truyền thẳng được ưa chuộng. Lĩnh vực còn lại là phân loại. Trong các nhiệm vụ phân loại, chúng ta xử lý các dự đoán về xác suất, nghĩa là đầu ra của mạng nơ-ron phải nằm trong phạm vi từ 0 đến 1. Một hàm mất mát có thể đo lỗi giữa xác suất dự đoán và nhãn biểu diễn lớp thực tế được gọi là **Cross-Entropy Loss Function** và với vectơ dự đoán y và vectơ thực tế y_hat, có thể tính toán Cross-Entropy Loss Function giữa hai vectơ đó như sau:

$$\mathscr{L}(\theta) = -\sum_{i=0}^{N}\hat{y}_i \cdot \log(y_i)$$

trong đó:

y_i: các giá trị trong vector dự đoán $\vec{y}$.

$\hat{y}_i$: các giá trị trong nhãn thực $\hat{\vec{y}}$.

Cuối cùng, chúng ta đến với hàm mất mát **Mean Absolute Percentage Error (MAPE)**. Hàm mất mát này không được chú ý nhiều trong học sâu. Phần lớn, chúng ta sử dụng nó để đo hiệu suất của mạng nơ-ron trong các tác vụ dự báo nhu cầu.

$$\text{MAPE} = \frac{100\%}{N}\sum_{i=0}^{N}\frac{|y_i - \hat{y}_i|}{\hat{y}_i}$$

- Trình tối ưu hóa - Optimizer

Bây giờ chúng ta đã biết CNN hoạt động tốt (hoặc kém) như thế nào, đã đến lúc cải thiện nó. Bộ tối ưu hóa giống như một huấn luyện viên điều chỉnh trọng số của mạng để giúp nó hoạt động tốt hơn. Nó điều chỉnh các tham số của mô hình để giảm thiểu hàm mất mát, cuối cùng dẫn đến các dự đoán chính xác hơn theo thời gian.

- Truyền ngược - Backpropagation

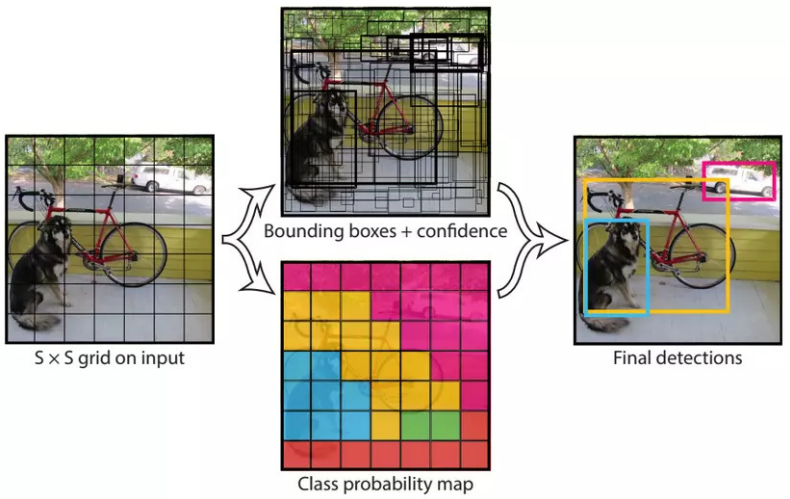
Backpropagation là phép thuật ẩn sau hậu trường khiến mọi thứ hoạt động. Đó là quá trình tìm ra mức độ mỗi trọng số trong mạng góp phần gây ra lỗi và sau đó điều chỉnh các trọng số đó cho phù hợp. Bộ tối ưu hóa sử dụng thông tin này để thực hiện các bản cập nhật thông minh hơn, giúp mô hình trở nên tốt hơn sau mỗi vòng đào tạo.

2.3 You Only Look Once(YOLO)

2.3.1 You Only Look Once cơ bản

You Only Look Once là một trong những thuật toán có giám sát của học sâu có mục tiêu phát hiện vật thể trong hình ảnh hoặc video. Mô hình này được sử dụng để phát hiện đối tượng và chỉ cần một mạng nơ-ron duy nhất cho toàn bộ hình ảnh. Đúng như tên gọi, YOLO có nghĩa là chỉ cần quét một lần để nhận diện và gán nhãn các đối tượng trong hình ảnh.

Mục tiêu của YOLO là xác định vị trí của vật thể và dự đoán dựa trên xác suất xem vật thể đó là gì. Đầu vào của tập dữ liệu mà YOLO huấn luyện thường có 2 phần: ảnh(hoặc video) và nhãn. Tệp nhãn sẽ lưu các chỉ số của vật thể tìm được: khung vật thể(bounding box) và tên của nó.



Hình 2.8: Cách thức hoạt động của YOLO

Đầu tiên, YOLO chuẩn hóa hình ảnh hoặc khung hình của video đầu vào từ hình chữ nhật thành hình vuông bằng phương pháp thêm viền(padding) để đưa ảnh về dạng ma trận vuông. Khi ảnh có kích thước không phải vuông, padding sẽ thêm các pixel vào các cạnh của ảnh (thường là màu đen hoặc giá trị 0) để làm đầy các vùng còn thiếu, giữ nguyên tỷ lệ khung hình gốc của ảnh mà không làm biến dạng nội dung.

Sau đó, ảnh được chia thành một lưới với kích thước S×S, với mỗi ô lưới sẽ chịu trách nhiệm phát hiện các đối tượng có tâm nằm trong ô đó. Với ảnh đầu vào có kích thước cố định, YOLO thường sử dụng các lưới như 13x13,19x19,52x52,... tùy thuộc vào độ phân giải của ảnh và phiên bản YOLO đang sử dụng. Mỗi ô được gán nhiều các anchor boxes, là các hộp hình chữ nhật có kích thước cố định, đại diện cho các hình dạng phổ biến của các đối tượng trong dữ liệu huấn luyện. Mỗi cell sẽ dự đoán một tập hợp các giá trị để điều chỉnh anchor boxes, biến chúng thành bounding boxes phù hợp với đối tượng trong ảnh.

Đối với mỗi bounding box, YOLO dự đoán vị trí trung tâm (x, y), chiều rộng w , chiều cao h của khung. Các giá trị (x, y, w, h) đều được chuẩn hóa so với kích thước của ô lưới để giúp mạng dễ dàng học và tăng cường khả năng tổng quát hóa cho nhiều kích thước ảnh khác nhau.

Mỗi bounding box sẽ kèm theo một giá trị xác suất, ký hiệu là p_{obj}, biểu diễn xác suất có đối tượng bên trong bounding box đó. Ngoài ra, YOLO cũng dự đoán xác suất đối tượng thuộc vào các lớp khác nhau(ví dụ như người, xe, chó, mèo), ký hiệu là c_i. Nếu một mạng YOLO được huấn luyện để phân loại 80 lớp đối tượng, đầu ra của mạng sẽ chứa 80 giá trị xác suất cho mỗi bounding box. Confidence score là một giá trị số biểu diễn mức độ tự tin của mô hình YOLO về việc một bounding box (hộp chứa) có chứa một đối tượng và xác định đúng lớp của đối tượng đó. Nó được tính bằng công thức:

$$\text{confidence score} = p_{obj} \times max(c_i) \tag{2.10}$$

Ví dụ: Với một ảnh đầu vào, ta có một ô lưới ảnh nằm trong ảnh đó phát hiện được bounding box với các thông tin sau:

- Xác suất bounding box chứa đối tượng $p_{obj} = 0.8$
- Mạng dự đoán xác suất của đối tượng thuộc vào các lớp khác nhau:
  - Lớp "Người"(Human) có xác suất là 0.7
  - Lớp "Xe"(Car) có xác suất là 0.2
  - Lớp "Chó"(Dog) có xác suất là 0.1

Như vậy, ta có thể thấy $max(c_i) = c_{human} = 0.7$ nên confidence score $= 0.8 \times 0.7 = 0.72$

Sau khi tính toán toàn điểm tin cậy của tất cả các khung, YOLO sẽ sử dụng một kỹ thuật gọi là Non-Max Suppression (NMS) để lọc ra các bounding boxes không cần thiết, chỉ giữ lại các bounding boxes có độ tin cậy cao nhất và tránh phát hiện trùng lặp. Thuật toán sau đó sẽ loại bỏ những vùng có điểm thấp hơn ngưỡng xác định(thường là 0.5) giúp giảm số lượng bounding boxes không cần thiết và chỉ giữ lại các khung hình có độ tin cậy cao. Sau đó, YOLO tiến hành NMS bằng cách chọn bounding box có confidence score cao nhất và đánh dấu nó là bounding box đáng tin cậy nhất cho đối tượng được phát hiện, ta có thể gọi là khung tham chiếu. Tiếp theo, YOLO tính toán chỉ số Intersection over Union (IoU) giữa bounding box này và các bounding boxes còn lại để đo độ chồng lấp giữa chúng bằng công thức:

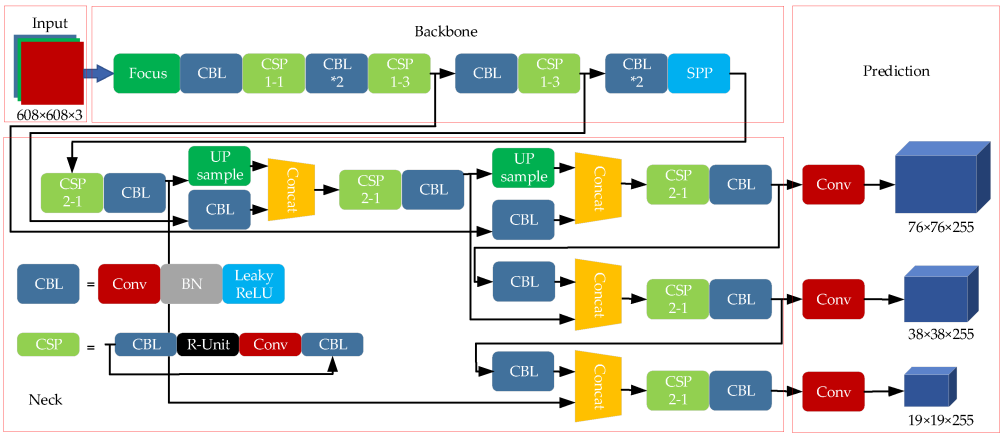
$$\text{IoU(A,B)} = \frac{\text{Diện tích phần giao của A và B}}{\text{Diện tích phần hợp của A và B}} \tag{2.11}$$

Nếu như giá trị IoU lớn hơn ngưỡng cho phép, bounding box đang xét sẽ bị loại bỏ vì nó được coi là trùng lặp với khung tham chiếu. Từ đó, thuật toán sẽ cho ra được đầu ra bao

gồm các khung vật thể không trùng lặp nhau và nhãn lớp của các đối tượng được phát hiện.

2.3.2 You Only Look Once phiên bản 5

YOLOv5 (You Only Look Once version 5) là phiên bản thứ 5 của YOLO, dựa trên các phiên bản YOLO trước, YOLOv5 tối ưu hơn về tốc độ và tính hiệu quả, nhờ các cải tiến về kiến trúc và kỹ thuật huấn luyện. YOLOv5 được phát triển bằng Python và PyTorch. Điều này giúp YOLOv5 dễ sử dụng và tích hợp hơn, cũng như dễ dàng thử nghiệm và tối ưu. Do phát triển bằng Python, nó dễ tích hợp với các thư viện phổ biến khác và dễ sử dụng cho các vấn đề học sâu.



Hình 2.9: Kiến trúc của YOLOv5 [8]

Kiến trúc của YOLOv5 gồm ba phần: Backbone, Neck và Head. Backbone là phần chịu trách nhiệm trích xuất các đặc trưng từ ảnh đầu vào. Trong hình **2.9**, các lớp convolutional (CONV) và các block đặc biệt như C3 giúp mô hình học được các đặc trưng khác nhau ở nhiều cấp độ. Phần SPP (Spatial Pyramid Pooling) giúp mô hình hiểu rõ hơn về các đặc trưng không gian ở nhiều tỷ lệ khác nhau, từ đó cải thiện khả năng nhận diện. Mỗi khối (C3) có các tham số như N là số lượng block trong khối, K là kích thước kernel, và S là bước nhảy (stride), giúp mô hình xử lý thông tin hiệu quả hơn.

Neck có nhiệm vụ kết hợp các đặc trưng từ các lớp khác nhau của backbone và chuẩn bị chúng cho phần head. Upsample được sử dụng để tăng kích thước của đặc trưng cho các lớp tiếp theo, hỗ trợ quá trình tổng hợp thông tin từ các cấp độ khác nhau. Các khối C3 trong phần này tiếp tục xử lý và tạo ra các đặc trưng, đặc biệt là đối với các đối tượng có kích thước khác nhau, giúp mô hình nhận diện cả các đối tượng nhỏ và lớn. Các lớp convolutional tiếp theo giúp lọc và tinh chỉnh các đặc trưng, chuẩn bị chúng cho việc dự đoán trong phần head.

Hàm mất mát của YOLOv5 gồm 3 thành phần chính: hàm mất mát khung bao(L_{box}),

mất mát phân loại(L_{cls}) và mất mát tin cậy (L_{obj}); được tính bằng công thức:

$$L = \lambda_{box}L_{box} + \lambda_{cls}L_{cls} + \lambda_{obj}L_{obj} \tag{2.12}$$

với λ_{box}, λ_{cls}, λ_{obj} là các hệ số để điều chỉnh tầm quan trọng của từng thành phần trong quá trình huấn luyện.

Hàm mất mát khung bao đo lường độ sai khác giữa khung bao dự đoán (x, y, h, w) so với khung bao thực tế $(x_{gt}, y_{gt}, h_{gt}, w_{gt})$ bằng cách sử dụng chỉ số Intersection over Union (IoU) đã nói ở trên, hoặc các biến thể của IoU như Generalized IoU (GIoU):

$$IoU(B, B_{gt}) = \frac{\text{Diện tích phần giao của } B \text{ và } B_{gt}}{\text{Diện tích phần hợp của } B \text{ và } B_{gt}} \tag{2.13}$$

$$GIoU(B, B_{gt}) = IOU(B, B_{gt}) - \frac{|C - (B \cup B_{gt})|}{|C|} \tag{2.14}$$

trong đó C là khung bao nhỏ nhất bao phủ cả B và B_{gt}

GIOU khắc phục hạn chế của IoU khi hai khung bao không giao nhau. Nó thêm vào phần bù bằng cách trừ đi tỷ lệ phần diện tích bên ngoài khung bao dự đoán nhưng nằm trong khung bao bao quanh nhỏ nhất. Điều này giúp mô hình học cách dịch chuyển khung bao để nó tiến gần hơn đến khung bao thực tế ngay cả khi chúng chưa có giao nhau.

Hàm mất mát phân loại giúp mô hình dự đoán chính xác lớp của đối tượng trong mỗi ô lưới. Với YOLOv5, thành phần này được tính qua Binary Cross-Entropy Loss, là một hàm mất mát đo lường sự khác biệt giữa xác suất dự đoán và nhãn thực tế:

$$L_{cls} = -\sum_{c} y_c log(\hat{p}_c) + (1 - y_c) log(1 - \hat{p}_c) \tag{2.15}$$

trong đó:

y_c là nhãn thực tế (1 nếu đối tượng thuộc lớp đó, 0 nếu không).

$\hat{p}_c$ là xác suất dự đoán của mô hình cho lớp.

Hàm mất mát tin cậy giúp mô hình quyết định liệu một ô lưới có chứa đối tượng không. Thành phần mất mát này được tính toán bằng Binary Cross-Entropy Loss dựa trên xác suất tin cậy p_{obj},nghĩa là xác suất có đối tượng ở vị trí đó:

$$L_{obj} = -(y_{obj} log(\hat{p}_{obj}) + (1 - y_{obj}) log(1 - \hat{p}_{obj})) \tag{2.16}$$

trong đó:

y_{obj} là nhãn thực tế cho biết ô có chứa đối tượng hay không (1 hoặc 0).

$\hat{p}_{obj}$ là xác suất tin cậy của mô hình.

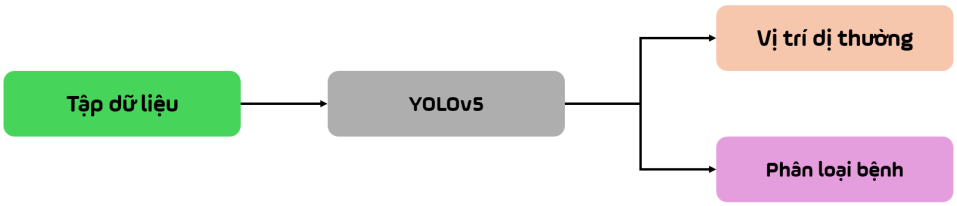
Tùy vào mục đích sử dụng mô hình YOLOv5, ta có thể điều chỉnh 3 hệ số λ_{box}, λ_{cls}, λ_{obj}. Ví dụ:

- Nếu mô hình có xu hướng dự đoán sai vị trí đối tượng (ví dụ: sai khung bao), có thể tăng λ_{box} để mô hình chú trọng hơn vào việc cải thiện vị trí của đối tượng.
- Nếu mô hình dự đoán sai lớp đối tượng, có thể tăng λ_{cls} để giúp phân loại chính xác hơn.
- Nếu mô hình gặp khó khăn trong việc phát hiện các đối tượng nhỏ (vì mô hình không nhận ra ô có đối tượng), tăng λ_{obj}

2.4 Áp dụng các mô hình trên cho bài toán dự đoán và phân loại bệnh phổi

Để có được đánh giá khách quan và tổng quát, bài báo cáo sẽ giới thiệu 3 mô hình được sử dụng đối với bộ dữ liệu trên:

- **Mô hình 1**: Sử dụng mô hình YOLO cho việc phát hiện và phân loại bệnh:

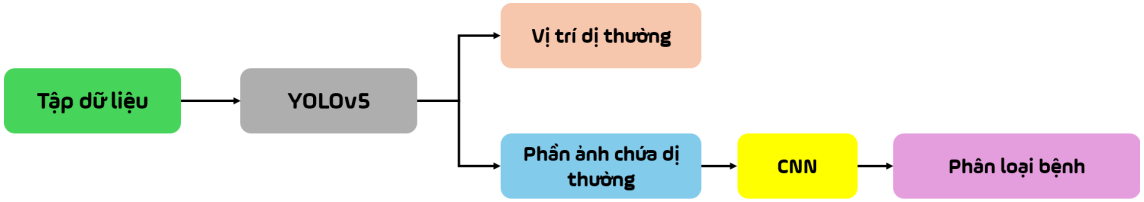


Hình 2.10: Mô hình sử dụng YOLOv5

- **Mô hình 2**: Sử dụng CNN trong việc phân loại bệnh:

Hình 2.11: Mô hình CNN

- **Mô hình 3**: Sử dụng kết hợp YOLO và CNN:



Hình 2.12: Mô hình sử dụng YOLOv5+CNN

Chương 3

Kết quả thực nghiệm

3.1 Thực nghiệm

Ba mô hình sử dụng bộ chung dữ liệu đầu vào gồm 2 thành phần chính: ảnh và nhãn tương ứng. Ảnh của tập dữ liệu là hình ảnh chụp cắt lớp của các bệnh nhân. Tập nhãn của từng ảnh sẽ gồm n dòng, ứng với n dị thường đã được đánh dấu trước của ảnh đó. Mỗi dòng là nhãn đã được chuẩn hóa sẵn theo mô hình YOLOv5 đã nói ở trên, gồm có 5 thành phần dạng số: (a, x, y, w, h) với:

- a là nhãn của dị thường $(a \in \{0; 1; 2; 3\})$

  trong đó:

  0 là nhãn ứng với tràn dịch màng phổi

  1 là nhãn ứng với xuất huyết phổi

  2 là nhãn ứng với tràn khí màng phổi

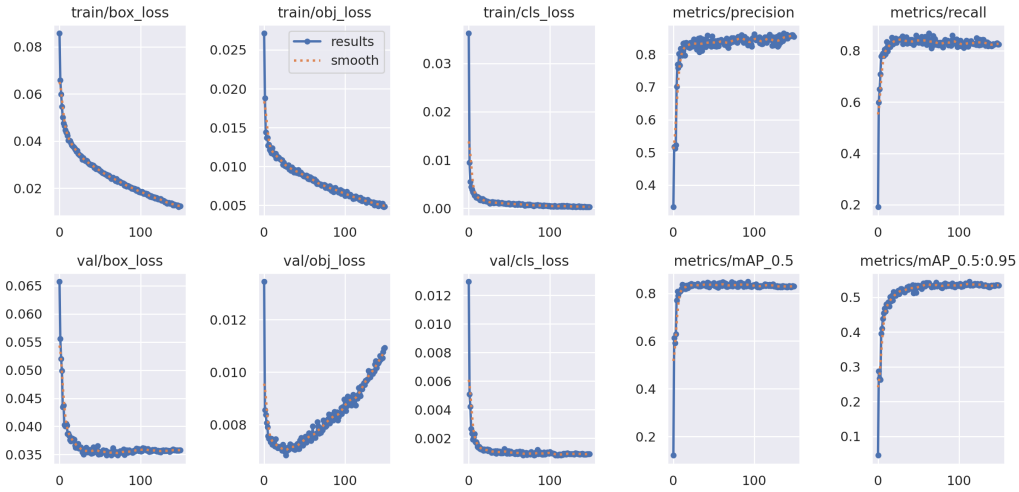
  3 là nhãn ứng với ung thư

- x và y là tọa độ 2 chiều của tâm khung dị thường
- w là chiều rộng của khung
- h là chiều cao của khung

Do hạn chế về mặt tài nguyên, bộ dữ liệu sử dụng chỉ gồm 2826 ảnh và nhãn tương ứng, trong đó có 4 lớp phân loại chính, tượng trưng cho 4 dị thường đã được trình bày. Bộ dữ liệu này được chia thành 2 phần: 2265 ảnh và nhãn tương ứng (chiếm 80% bộ dữ liệu) được sử dụng để huấn luyện, còn lại được sử dụng để kiểm tra, đánh giá độ chính xác của mô hình.

3.2 Kết quả thực nghiệm với mô hình YOLOv5

Mô hình được huấn luyện 150 epochs, trong tổng thời gian 2 giờ 54 phút, tương đương 1,16 phút 1 epoch cho ra kết quả sau:



Hình 3.1: Biểu đồ kết quả huấn luyện của YOLOv5

Các biểu đồ thể hiện giá trị tổn thất theo thời gian, đồng thời cũng là các giá trị chính xác, độ nhớ,.. với trục x đại diện cho số epoch.Đầu tiên là tổn thất huấn luyện: Dòng biểu đồ đầu tiên cho thấy tổn thất huấn luyện của mô hình, bao gồm tổn thất hộp (box loss), tổn thất đối tượng (object loss) và tổn thất phân loại (classification loss). Như mong đợi, tất cả các tổn thất này đều giảm theo thời gian, cho thấy rằng mô hình đang học cách dự đoán các hộp giới hạn và lớp của các đối tượng tốt hơn.

Tổn thất kiểm tra: Dòng biểu đồ thứ hai cho thấy tổn thất kiểm tra của mô hình, đo lường hiệu suất của mô hình trên một tập dữ liệu riêng biệt. Tổn thất kiểm tra thường cao hơn tổn thất huấn luyện, điều này là bình thường vì mô hình chưa thấy các điểm dữ liệu này trước đó. Tuy nhiên, tổn thất kiểm tra cũng có xu hướng giảm theo thời gian, cho thấy rằng mô hình đang tổng quát tốt với dữ liệu chưa thấy.

Chỉ số đo lường: Hai cột biểu đồ cuối cùng cho thấy các chỉ số huấn luyện và kiểm tra, bao gồm độ chính xác (precision), độ nhớ (recall) và độ chính xác trung bình (mAP) tại các ngưỡng IoU khác nhau. Những chỉ số này được sử dụng để đánh giá hiệu suất của mô hình trong các tác vụ phát hiện đối tượng. Tất cả các chỉ số dường như đều tăng theo thời gian, cho thấy mô hình đang cải thiện trong việc dự đoán các đối tượng một cách chính xác.

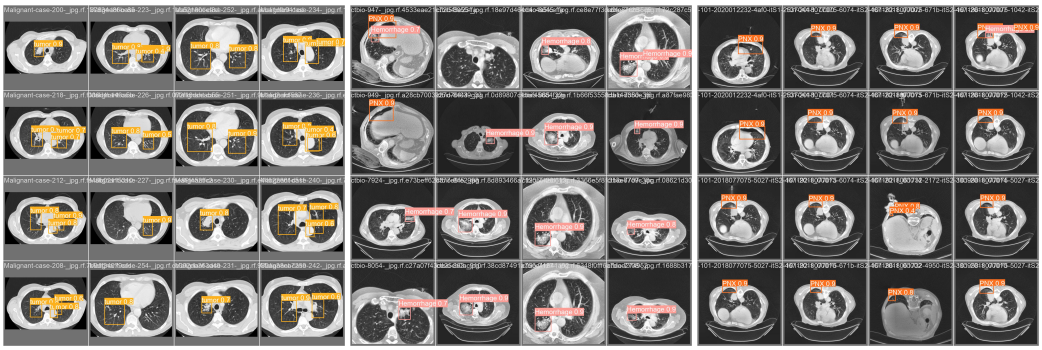
	precision	**recall**	**mAP50**	**mAP50-95**
Class 0	0.994	1	0.995	0.776
Class 1	0.919	0.903	0.952	0.543
Class 2	0.926	0.943	0.968	0.727
Class 3	0.535	0.474	0.51	0.148
all	0.844	0.83	0.84	0.548

Bảng 3.1: Báo cáo phân loại trên tập kiểm tra của YOLOv5

Kết quả đánh giá hiệu suất của YOLOv5 cho thấy mô hình đạt được kết quả khả quan trong việc phát hiện và phân loại các bệnh lý trên ảnh chụp CT phổi. Cụ thể:

Độ chính xác phát hiện: mAP50: YOLOv5 đạt được độ chính xác trung bình (mAP50) là 0.844 cho tất cả các lớp bệnh lý, thể hiện khả năng phát hiện đối tượng tương đối tốt. mAP50-95: Mô hình đạt được 0.54 mAP50-95, điều này cho thấy YOLOv5 vẫn chưa hiệu quả trong việc phát hiện các đối tượng với độ tin cậy cao (IOU> 0.5), điều này có thể cải thiện nếu được train nhiều lần hơn và xử lý ảnh tốt hơn cho YOLO.

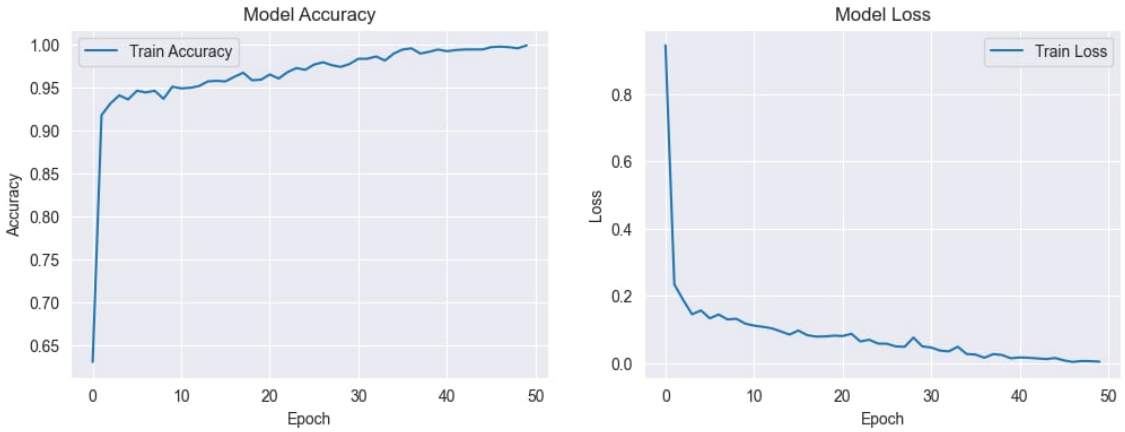
Khả năng phân loại: Precision: YOLOv5 có độ chính xác (Precision) cao đối với các lớp bệnh lý như Effusion (0.994) và Hemorrhage (0.919), cho thấy mô hình đã xác định chính xác các đối tượng thuộc lớp này. Tuy nhiên với Tumor độ chính xác chỉ đạt 0.535 , điều này có thể do các hình ảnh về ung thư chưa thực sự rõ ràng hoặc phân bố không đồng nhất. Recall: Mô hình đạt được Recall thấp ở lớp bệnh lý như tumor (0.474), chứng tỏ YOLOv5 chưa phân loại tốt các đối tượng thuộc lớp này. Các lớp còn lại đều có chỉ số Recall rất tốt (0.9 - 1) cho thấy khả năng phát hiện mạnh mẽ của YOLO. Dưới đây là ví dụ cụ thể khi YOLOv5 thực hiện phát hiện và khoanh vùng các lớp đã được học:



Hình 3.2: Mô hình YOLOv5 phát hiện và dự đoán bounding box trên ảnh

3.3 Kết quả thực nghiệm với mô hình CNN

Mô hình được huấn luyện 100 epochs, trong tổng thời gian 1118.63 giây (≈ 18 phút 39 giây), tương đương với khoảng 11 giây cho mỗi epoch cho ra kết quả sau:



Hình 3.3: Biểu đồ độ tăng trưởng độ chính xác và mất mát khi huấn luyện của CNN

Xu hướng của hai biểu đồ "Model Accuracy"và "Model Loss"trong quá trình huấn luyện mô hình cho thấy sự giảm mạnh của giá trị hàm mất mát và tăng mạnh độ chính xác trong những epoch đầu tiên. Điều này cho thấy mô hình đang học và tối ưu hóa rất nhanh ngay từ đầu, khi chưa có kiến thức gì về dữ liệu. Sau giai đoạn này, loss dần ổn định và không giảm nhiều nữa, điều này chỉ ra rằng mô hình đã học được các đặc trưng cơ bản của dữ liệu và gần đạt đến mức tối ưu. Tuy nhiên, tại mấy epoch đầu loss giảm và accuracy quá nhanh và ổn định sớm, có thể là dấu hiệu của overfitting, tức là mô hình quá khớp với dữ liệu huấn luyện và khó tổng quát khi áp dụng trên dữ liệu mới.

	precision	**recall**	**f1-score**	**support**
Class 0	0.74	0.83	0.78	63
Class 1	0.84	0.91	0.87	179
Class 2	0.78	0.84	0.81	133
Class 3	1.00	0.65	0.79	92
accuracy			0.83	467
macro avg	0.84	0.81	0.81	467
weighted avg	0.84	0.83	0.83	467

Bảng 3.2: Báo cáo phân loại trên tập kiểm tra của CNN

Dựa vào bảng trên, lớp 0 (Effusion) đạt độ chính xác 0.74 và độ nhạy 0.83, cho thấy mô hình có khả năng phát hiện 83% các trường hợp thực tế của lớp này, mặc dù một số dự đoán có thể không chính xác. Lớp 1 (Hemorrhage) có precision 0.84 và recall 0.91, cho thấy mô hình rất hiệu quả trong việc phát hiện các trường hợp ác tính, với 91% các trường hợp thực sự được nhận diện chính xác. Lớp 2 (PNX) có precision 0.78 và recall 0.84, cho thấy khả năng phát hiện tốt. Lớp 3 (Tumor) đạt precision hoàn hảo 1.00, nhưng recall chỉ 0.65, cho thấy mô hình có thể gặp khó khăn trong việc phát hiện tất cả các trường hợp thuộc lớp này, dẫn đến một số trường hợp bị bỏ sót.

Tổng thể, mô hình đạt độ chính xác 0.83 trên toàn bộ tập dữ liệu 467 mẫu, với điểm F1 trung bình là 0.81, cho thấy sự cân bằng giữa độ chính xác và độ nhạy. Các chỉ số trung bình theo trọng số và không trọng số đều đạt khoảng 0.84 và 0.81, chứng tỏ mô hình có khả năng phân loại tốt và đáng tin cậy trong việc phát hiện các lớp bệnh lý phổi qua ảnh CT scan.Tuy nhiên, vẫn cần cải thiện khả năng nhận diện, nhằm giảm thiểu các trường hợp bị bỏ sót.

3.4 Kết quả thực nghiệm với mô hình YOLOv5 kết hợp CNN

Từ những kết quả thực nghiệm ở trên, cùng với sự phù hợp trong việc kết hợp hai mô hình là CNN và YOLOv5, việc xây dựng một mô hình kết hợp hai mô hình ấy là cần thiết. Đi đến bước đầu tiên, mô hình phát hiện đối tượng YOLOv5, sẽ được sử dụng để xác định và đánh dấu các đối tượng trong ảnh bằng cách vẽ bounding box xung quanh chúng. Sau khi phát hiện, các đối tượng này sẽ được cắt ra từ ảnh gốc để tạo thành các ảnh nhỏ hơn, chứa riêng từng đối tượng. Tiếp theo, các ảnh đã cắt này sẽ được chuẩn bị cho mô hình phân loại mạng nơ-ron tích chập (CNN), thông qua các bước như thay đổi kích thước và chuẩn hóa dữ liệu. Sau đó, mô hình CNN sau đó sẽ phân loại các ảnh đã được xử lý, cho ra các nhãn tương ứng với từng đối tượng, cho ra kết quả cuối cùng.

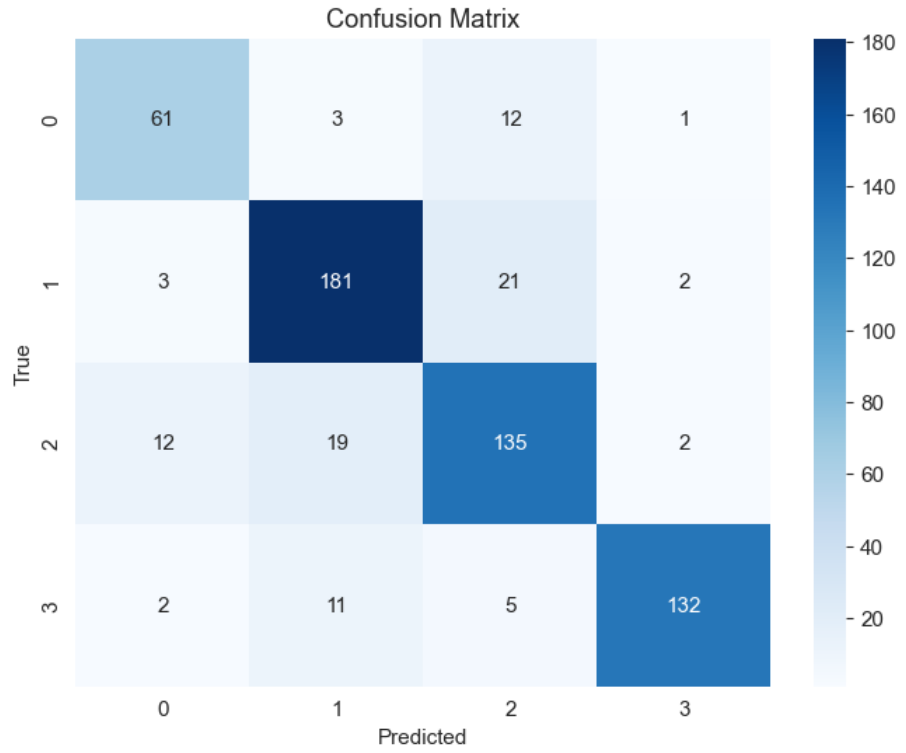
Mô hình được sử dụng lại kết quả huấn luyện của YOLOv5 và sử dụng lại mô hình CNN để xử lí phân loại bounding box, với thời gian kiểm tra là 503,61 giây ($\approx$ 8 phút 23 giây) cho 561 ảnh , cho ra kết quả như sau:

	precision	**recall**	**f1-score**	**support**
Class 0	0.79	0.81	0.79	77
Class 1	0.86	0.88	0.87	1207
Class 2	0.81	0.86	0.83	168
Class 3	0.95	0.90	0.92	150
accuracy			0.85	602
macro avg	0.84	0.84	0.84	602
weighted avg	0.85	0.85	0.85	602

Bảng 3.3: Báo cáo phân loại trên tập kiểm tra của mô hình CNN kết hợp YOLOv5

Trong báo cáo phân loại, các chỉ số như độ chính xác (precision), độ nhạy (recall), và điểm F1 (F1-score) đóng vai trò quan trọng trong việc đánh giá hiệu suất của mô hình. Độ chính xác của lớp 0 là 0.78, cho thấy 78% trong số các trường hợp mà mô hình dự đoán là chính xác, trong khi lớp 1 đạt precision 0.85, thể hiện khả năng phát hiện các trường hợp ác tính hiệu quả, điều này rất quan trọng trong y tế. Lớp 3 có precision cao nhất (0.96), cho thấy độ chính xác vượt trội trong việc phát hiện lớp này. Về độ nhạy, lớp 0 có recall

0.79, cho thấy mô hình đã phát hiện được 79% các trường hợp thực sự thuộc lớp này, còn lớp 1 đạt recall 0.87, cho thấy khả năng phát hiện cao các trường hợp ác tính. Lớp 3 có recall 0.88, chứng tỏ hiệu suất tốt trong việc nhận diện. Cuối cùng, điểm F1, chỉ số tổng hợp giữa precision và recall, cho thấy lớp 0 có F1-score 0.79, lớp 1 đạt 0.86, và lớp 3 có F1-score cao nhất, cho thấy mô hình đặc biệt mạnh trong việc phát hiện lớp này. Những chỉ số này không chỉ phản ánh hiệu suất của mô hình mà còn chỉ ra các lĩnh vực cần cải thiện để nâng cao khả năng phát hiện và phân loại trong tương lai.



Hình 3.4: Ma trận nhầm lẫn mô hình CNN kết hợp YOLOv5

Để thấy được tiềm năng và sự phù hợp của mô hình, ta có sự so sánh cả ba mô hình với nhau nhằm thấy được những phần tối ưu và chưa tối ưu riêng của chúng, từ đó đưa ra hướng phát triển phù hợp nhất. Dưới đây là bảng so sánh các chỉ số đánh giá thành phần sau khi training của các mô hình đã nêu ở phần trước và mô hình hiện tại:

Chỉ số	CNN	YOLOv5	CNN + YOLOv5
Độ chính xác (Accuracy)	82.87%	84%	85%
mAP50	Không áp dụng	84%	85%
mAP50-95	Không áp dụng	54%	55%
Precision (Lớp 0)	0.74	0.78	0.79
Precision (Lớp 1)	0.84	0.85	0.86
Precision (Lớp 2)	0.78	0.80	0.81
Precision (Lớp 3)	1.00	0.96	0.95
Recall (Lớp 0)	0.83	0.79	0.81
Recall (Lớp 1)	0.91	0.87	0.88
Recall (Lớp 2)	0.84	0.85	0.86
Recall (Lớp 3)	0.65	0.88	0.90
F1-Score (Lớp 0)	0.78	0.78	0.79
F1-Score (Lớp 1)	0.87	0.86	0.87
F1-Score (Lớp 2)	0.81	0.82	0.83
F1-Score (Lớp 3)	0.79	0.97	0.92

Hình 3.5: So sánh chỉ số đánh giá của ba mô hình phát hiện và phân loại

Nhìn vào bảng kết quả, ta thấy mô hình YOLOv5 kết hợp với CNN cho kết quả tổng thể tốt hơn cả hai mô hình riêng biệt. Điểm số mAP50 và mAP50-95 cao hơn khi so với YOLOv5 riêng lẻ, các chỉ số đánh giá thành phần cũng cao hơn với mô hình CNN. Ngoài ra, mô hình YOLOv5 kết hợp CNN cũng cho điểm số Precision, Recall và F1-Score cao hơn ở các lớp khác nhau, đặc biệt là ở lớp 3. Điều này cho thấy mô hình kết hợp này hiệu quả hơn trong việc xác định chính xác và phân loại các đối tượng trong các lớp khác nhau. Sự kết hợp giữa YOLOv5 và CNN cho thấy những lợi ích vượt trội trong việc phát hiện và phân loại các bệnh lý phổi. Các chỉ số cho thấy rằng mô hình này không chỉ cải thiện độ chính xác mà còn nâng cao khả năng phát hiện và phân loại, đặc biệt là ở các lớp khó phân loại.

Kết luận

Bài báo cáo đã trình bày quy trình xây dựng mô hình học sâu nhằm phát hiện các bệnh lý về phổi từ hình ảnh y tế, cụ thể là ảnh chụp cắt lớp CT. Phương pháp sử dụng kết hợp mô hình YOLOv5 và CNN đã cho thấy hiệu quả cao trong việc xử lý và phân tích dữ liệu hình ảnh y tế. Kết quả thực nghiệm chứng minh rằng hệ thống có khả năng phát hiện chính xác các dấu hiệu bất thường như xuất huyết phổi, tràn dịch màng phổi, tràn khí màng phổi và ung thư với độ nhạy và độ chính xác cao. Các nội dung đã thực hiện được bao gồm:

- Tìm hiểu cơ bản về cơ quan phổi của người và một số bệnh liên quan: các khái niệm, thuật ngữ thường được sử dụng
- Tìm hiểu về chụp cắt lớp CT và cách sử dụng ảnh để phát hiện bệnh lý về phổi
- Tìm hiểu về học sâu, các mô hình CNN, YOLO
- Nghiên cứu tìm hiểu bài toán nhận diện bệnh dựa trên ảnh CT sử dụng mô hình học sâu
- Triển khai mô hình với mẫu dữ liệu nhỏ, đánh giá độ chính xác.

Những kết quả đầu tiên này là nền móng cơ bản nhất để bắt đầu đi sâu vào học máy cũng như tin sinh học. Việc xây dựng mô hình bước đầu có kết quả khá tốt với độ chính xác trên 80 % . Tuy nhiên, báo cáo cũng gặp một số hạn chế, như phụ thuộc vào chất lượng tập dữ liệu, hạn chế trong số mẫu dữ liệu và khả năng giải thích kết quả của các mô hình học sâu. Trong tương lai, việc mở rộng tập dữ liệu đa dạng hơn, cải thiện tính minh bạch của mô hình là những hướng phát triển hứa hẹn, giúp cải thiện độ chính xác và khả năng ứng dụng thực tế của hệ thống.

Tài liệu tham khảo

[1] Phạm Văn Lập (Tổng chủ biên) **andothers**. *Sinh học 11*. Kết nối tri thức với cuộc sống, 2023, **page** 57.

[2] National Institute of Biomedical Imaging and Bioengineering. *Computed Tomography (CT)*. Accessed: 22-10-2024. 2024. URL: https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct.

[3] World Health Organization. *Lung Cancer*. Accessed: 10-11-2024. 2024. URL: https://www.who.int/news-room/fact-sheets/detail/lung-cancer.

[4] Mayo Clinic. *Pneumothorax*. Accessed: 22-10-2024. 2024. URL: https://www.mayoclinic.org/diseases-conditions/pneumothorax/symptoms-causes/syc-20350367#:~:text=A%20pneumothorax%20occurs%20when%20air,a%20portion%20of%20the%20lung..

[5] CTisus. *Pulmonary Hemorrhage*. Accessed: 22-10-2024. 2024. URL: https://www.ctisus.com/learning/pearls/chest/pulmonary-hemorrhage.

[6] Radiology Key. *Pleural Effusion*. Accessed: 22-10-2024. 2024. URL: https://radiologykey.com/pleural-effusion-2/.

[7] Dataquest. *An Introduction to Deep Learning*. Accessed on 28-10-2024. 2023. URL: https://www.dataquest.io/blog/tutorial-introduction-to-deep-learning/.

[8] Bùi Thanh Lâm **andothers**. "Nghiên cứu ứng dụng thuật toán học sâu kết hợp cảm biến Kinect trong phân loại vật thể". **in***Research for the Application of Deep Learning Combined Kinect Sensor in Object Classification*: (2023). DOI: 10.57001/huih5804.2023.142.