TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

CHẨN ĐOÁN BỆNH VIÊM PHỔI TỪ ẢNH X-QUANG SỬ DỤNG CNN

GVHD: Từ Lãng Phiêu
NHÓM 11: Phan Trung Kiên - 3120410263
Lâm Chí Minh - 3120410321
Lữ Ngọc Hộp - 3120410187

TP. HỒ CHÍ MINH, THÁNG 5/2024

Mục lục

Lời Nói Đầu Tiên

Trong thời đại hiện đại, sự tiến bộ nhanh chóng của công nghệ đã mở ra những cánh cửa mới cho lĩnh vực y tế. Việc áp dụng các phương pháp và công nghệ mới không chỉ giúp nâng cao chất lượng dịch vụ y tế mà còn mở ra tiềm năng mới trong việc chẩn đoán và điều trị bệnh tật. Trong bối cảnh này, nghiên cứu và ứng dụng các mô hình học sâu, đặc biệt là trong lĩnh vực phân đoạn hình ảnh y tế, trở nên ngày càng phổ biến và quan trọng.

Dự án Phân loại X-quang Ngực sử dụng bộ dữ liệu được quản lý cẩn thận bao gồm 5.863 hình ảnh X-quang chất lượng cao ở định dạng JPEG. Tập dữ liệu được tổ chức thành ba thư mục, đó là đào tạo, kiểm tra và val, mỗi thư mục chứa các thư mục con cho hai loại hình ảnh riêng biệt: Viêm phổi và Bình thường. Mục tiêu chính của dự án này là phát triển mô hình phân loại chính xác và hiệu quả có khả năng xác định các trường hợp viêm phổi ở bệnh nhân dựa trên hình ảnh X-quang ngực. Viêm phổi là một bệnh nhiễm trùng đường hô hấp phổ biến và có khả năng đe dọa tính mạng. Việc phát hiện sớm có vai trò quan trọng để can thiệp kịp thời và điều trị hiệu quả.

Bằng cách tận dụng các Kỹ thuật học máy tiên tiến, bao gồm Mạng lưới thần kinh sâu và Thuật toán phân tích hình ảnh, dự án nhằm mục đích đào tạo một mô hình có thể phân loại chính xác hình ảnh X-quang ngực thành hai loại: Viêm phổi và Bình thường. Mô hình sẽ được đào tạo trên tập hợp con "đào tạo" của tập dữ liệu và được đánh giá trên các tập hợp con kiểm tra và giá trị để đảm bảo hiệu suất mạnh mẽ.

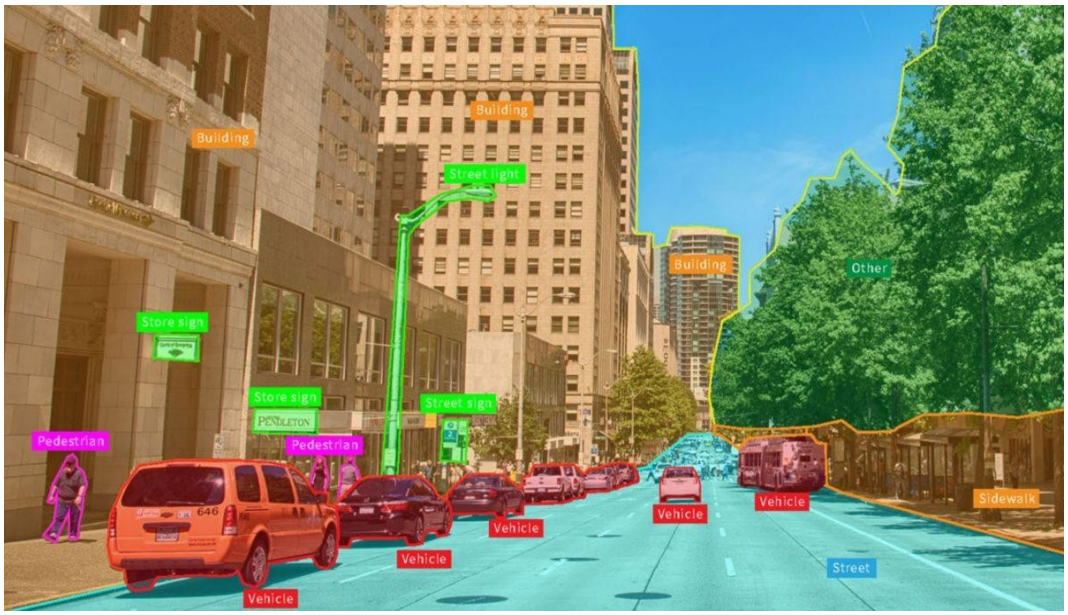
Kết quả của nghiên cứu này có ý nghĩa quan trọng đối với việc chăm sóc sức khỏe trẻ em. Một hệ thống phát hiện viêm phổi tự động và chính xác trên hình ảnh X-quang ngực có thể hỗ trợ các chuyên gia chăm sóc sức khỏe chẩn đoán và điều trị kịp thời các trường hợp viêm phổi. Điều này có khả năng dẫn đến cải thiện kết quả của bệnh nhân, giảm thời gian nằm viện và phân bổ nguồn lực y tế tốt hơn.

Hơn nữa, dự án còn đóng góp vào lĩnh vực hình ảnh y tế và chẩn đoán có sự hỗ trợ của máy tính. Những hiểu biết sâu sắc thu được từ nghiên cứu này có thể được áp dụng cho các phương thức và tình trạng hình ảnh y tế khác, dẫn đến những tiến bộ trong việc phát hiện và chẩn đoán bệnh tự động.

I. Thị giác máy tính (Computer Vision)

1.1 Thị giác máy tính (Computer Vision) là gì?

Thị giác máy tính (Computer Vision) là một trong những lĩnh vực hot nhất của khoa học máy tính và nghiên cứu trí tuệ nhân tạo. Dù chúng vẫn chưa thể cạnh tranh với sức mạnh thị giác của mắt người, đã có rất nhiều ứng dụng hữu ích được tạo ra khai thác tiềm năng của chúng.



Hình 1.1: Ứng dụng của thị giác máy tính

1.2 Ứng dụng của thị giác máy tính

Tích hợp công nghệ thị giác máy tính vào đa dạng lĩnh vực đã mở ra một loạt các ứng dụng tiềm năng, đóng góp tích cực vào sự tiến bộ trong nhiều ngành công nghiệp và phạm vi ứng dụng. Dưới đây là một số ví dụ phổ biến và đa dạng về những lĩnh vực mà thị giác máy tính có thể được áp dụng:

- **Nhận dạng vật thể và đối tượng:** Thị giác máy tính không chỉ giúp

trong việc nhận dạng khuôn mặt hoặc biển báo giao thông mà còn có thể được sử dụng để phát hiện và phân loại các đối tượng khác như xe hơi, động vật, hoặc sản phẩm trong cửa hàng.

- **Tự lái và xe tự hành:** Trong lĩnh vực xe tự lái, thị giác máy tính đóng một vai trò quan trọng trong việc nhận diện và dự đoán hành vi của các vật thể xung quanh, từ các phương tiện giao thông đến người đi bộ, giúp tăng cường an toàn và hiệu suất của hệ thống tự lái.

- **Y tế và chăm sóc sức khỏe:** Công nghệ thị giác máy tính đóng vai tíò quan trọng trong việc phân tích và chẩn đoán hình ảnh y khoa, từ phát hiện bất thường trong hình ảnh siêu âm đến phân loại các khối u trong hình ảnh MRI, giúp cải thiện quá trình chẩn đoán và điều trị các bệnh lý.

- **Giám sát và an ninh:** Hệ thống thị giác máy tính được triển khai rộng rãi trong việc giám sát an ninh và quản lý an toàn, từ việc giám sát người và phương tiện ở các khu vực công cộng đến giám sát biên giới và cơ sở hạ tầng quan trọng.

- **Quản lý sản xuất và logistics:** Trong sản xuất và quản lý hàng hóa, thị giác máy tính hỗ trợ trong việc kiểm tra chất lượng sản phẩm, theo dõi quy trình sản xuất, và tối ưu hóa quá trình logistics, từ việc phân loại sản phẩm đến quản lý kho hàng.

- **Tương tác người-máy thông minh:** Thị giác máy tính cũng được áp dụng trong việc phát triển giao diện người-máy thông minh, như nhận dạng cử chỉ và ngôn ngữ cơ thể của con người, tạo điều kiện thuận lợi cho việc tương tác người-máy một cách tự nhiên và hiệu quả.

Những ứng dụng này chỉ là một phần nhỏ của tiềm năng mà thị giác máy tính mang lại, và không ngừng mở ra các cơ hội mới trong tương lai.

1.3 Những hạn chế của thị giác máy tính

Mặc dù thị giác máy tính đã có những tiến bộ đáng kể và được sử dụng rộng rãi trong nhiều lĩnh vực, nhưng vẫn tồn tại một số hạn chế mà các nhà nghiên cứu và nhà phát triển phải đối mặt và cố gắng vượt qua:

- **Độ chính xác không hoàn hảo:** Mặc dù các mô hình thị giác máy tính ngày càng được cải thiện, nhưng độ chính xác vẫn chưa đạt tới mức hoàn hảo. Điều này đặc biệt đáng chú ý trong các tình huống phức tạp hoặc có nhiều biến thể, nơi các mô hình có thể gặp khó khăn trong việc phân loại đối tượng.

- **Yếu tố đa dạng và tinh vi của môi trường:** Thị giác máy tính có thể gặp khó khăn khi phải hoạt động trong các môi trường có sự đa dạng lớn hoặc có nhiều yếu tố ngoại lai, như ánh sáng yếu, độ ẩm cao, hoặc phong cảnh phức tạp.

- **Đòi hỏi lượng dữ liệu lớn:** Để huấn luyện một mô hình thị giác máy tính hiệu quả, thường cần một lượng lớn dữ liệu được gắn nhãn. Điều này có thể là một thách thức đối với các ngành công nghiệp hoặc ứng dụng nhỏ hoặc không có tài nguyên lớn.

- **Khả năng diễn giải kém:** Một số mô hình thị giác máy tính có thể được coi là "hộp đen" vì chúng không thể giải thích lý do tại sao một quyết định cụ thể đã được đưa ra. Điều này có thể gây khó khăn cho việc tin cậy và chấp nhận của người sử dụng.

- **Bất ổn trong các điều kiện biến đổi:** Thị giác máy tính có thể gặp khó khăn trong việc đảm bảo tính ổn định và khả năng chuyển giao giữa các điều kiện biến đổi, ví dụ như khi môi trường ánh sáng thay đổi hoặc khi có sự thay đổi về

góc chụp.

- **Tính đa dạng của đối tượng và ngữ cảnh:** Phân loại đối tượng trong các tình huống đa dạng và phức tạp vẫn là một thách thức. Các đối tượng có thể có sự biến đổi lớn về hình dáng, kích thước, và màu sắc, và có thể xuất hiện trong nhiều ngữ cảnh khác nhau.

Tuy nhiên, với sự tiến bộ liên tục trong nghiên cứu và công nghệ, nhiều nỗ lực đang được hướng tới việc giảm bớt các hạn chế này và tăng cường khả năng của thị giác máy tính trong các ứng dụng thực tiễn.

II. Dataset Chest X-Ray Images (Pneumonia)

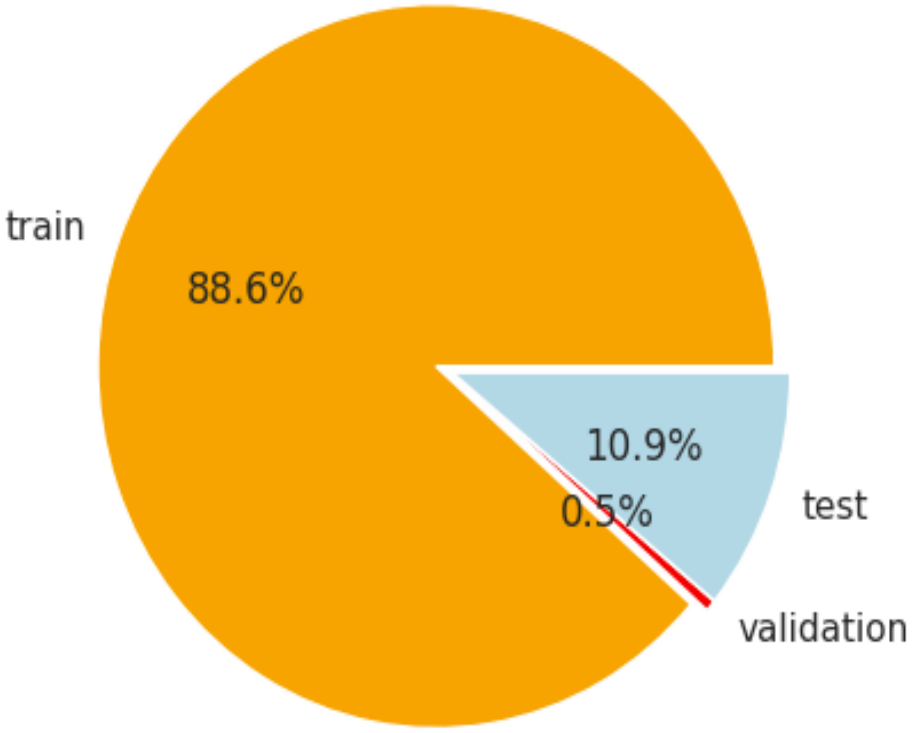
Hình S6. Ví dụ minh họa về X-quang ngực ở bệnh nhân viêm phổi

X-quang ngực bình thường (hình bên trái) cho thấy phổi rõ ràng và không có bất kỳ vùng mờ bất thường nào trên hình ảnh. Viêm phổi do vi khuẩn (giữa) thường biểu hiện đông đặc thùy khu trú, trong trường hợp này là ở thùy trên bên phải (mũi tên trắng), trong khi viêm phổi do vi rút (phải) biểu hiện với kiểu "kẽ" lan tỏa hơn ở cả hai phổi.

Nội dung Tập dữ liệu được tổ chức thành 3 thư mục (train, test, val) và chứa các thư mục con cho từng danh mục hình ảnh (Viêm phổi/Bình thường).

Có 5.863 hình ảnh X-Ray (JPEG) và 2 loại (Viêm phổi/Bình thường). Hình ảnh X-quang ngực (trước-sau) được chọn từ đoàn hệ hồi cứu của bệnh nhân nhi từ một đến năm tuổi từ Trung tâm Y tế Phụ nữ và Trẻ em Quảng Châu, Quảng Châu. Tất cả hình ảnh X-quang ngực được thực hiện như một phần của chăm sóc lâm sàng thông thường cho bệnh nhân.

Để phân tích hình ảnh X-quang ngực, tất cả các phim X-quang ngực ban đầu được sàng lọc để kiểm soát chất lượng bằng cách loại bỏ tất cả các bản quét chất lượng thấp hoặc không thể đọc được. Các chẩn đoán cho hình ảnh sau đó được hai bác sĩ chuyên môn phân loại trước khi được phép đào tạo hệ thống AI. Để giải quyết bất kỳ lỗi chấm điểm nào, bộ đánh giá cũng đã được chuyên gia thứ ba kiểm tra.



III. CNN (Convolutional Neural Network)

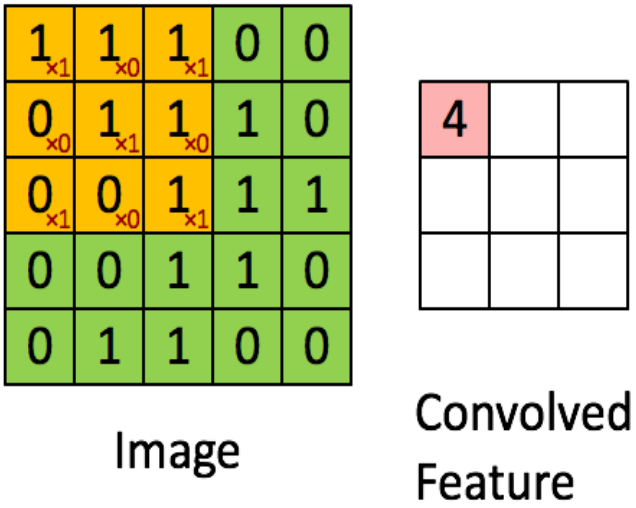
3.1 Giới thiệu về CNN

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

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

3.2 Tìm hiểu về Convolutional

Là một cửa sổ trượt (Sliding Windows) trên một ma trận như mô tả hình dưới:



Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature.

Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước 5×5 và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột.

Convolution hay tích chập là nhân từng phần tử trong ma trận 3. Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có kích thước nhỏ như trong ví dụ trên là 3×3.

Convolution hay tích chập là nhân từng phần tử bên trong ma trận 3×3 với ma trận bên trái. Kết quả được một ma trận gọi là Convoled feature được sinh ra từ việc nhận ma trận Filter với ma trận ảnh 5×5 bên trái.

3.3 Cấu trúc mạng CNN

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

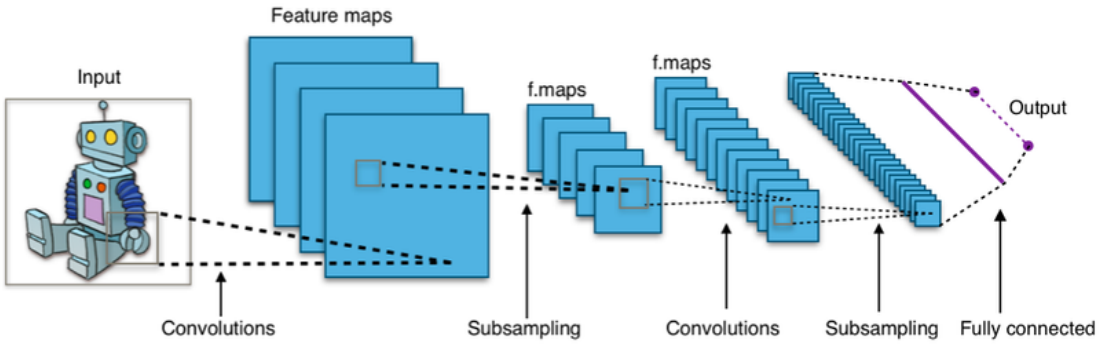
Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu

ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



Trong mô hình CNN có 2 khía cạnh cần quan tâm là tính bất biến (Location Invariance) và tính kết hợp (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

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

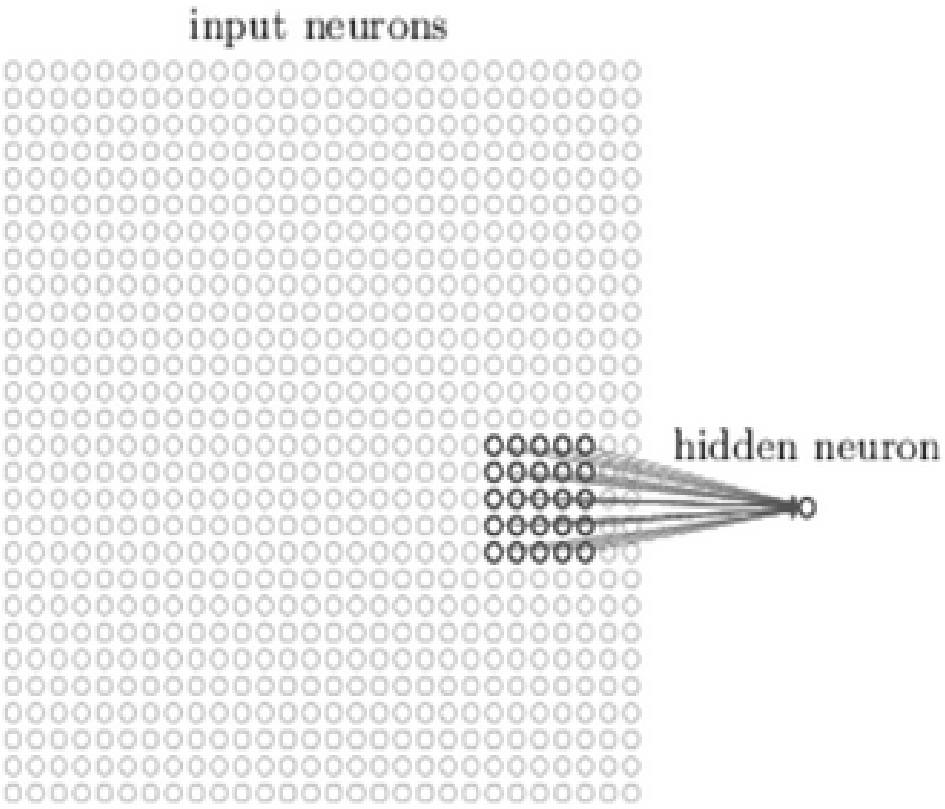
Mạng CNN sử dụng 3 ý tưởng cơ bản:

- Các trường tiếp nhận cục bộ (local receptive field)
- trọng số chia sẻ (shared weights)
- tổng hợp (pooling)

Trường tiếp nhận cục bộ (local receptive field)

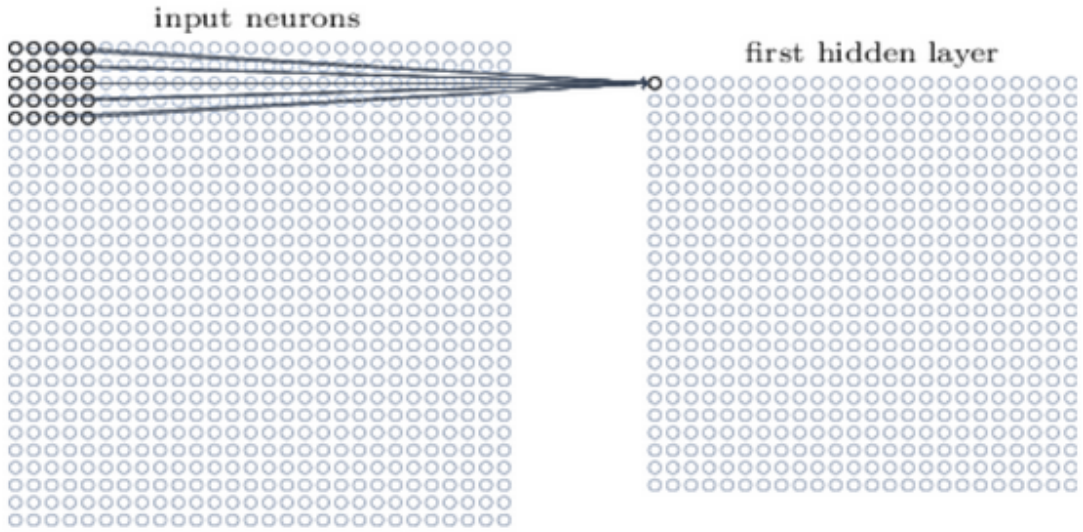
Đầu vào của mạng CNN là một ảnh. Ví dụ như ảnh có kích thước 28×28 thì tương ứng đầu vào là một ma trận có 28×28 và giá trị mỗi điểm ảnh là một ô trong ma trận. Trong mô hình mạng ANN truyền thống thì chúng ta sẽ kết nối các neuron đầu vào vào tầng ảnh.

Tuy nhiên trong CNN chúng ta không làm như vậy mà chúng ta chỉ kết nối trong một vùng nhỏ của các neuron đầu vào như một filter có kích thước 5×5 tương ứng (28- 5 + 1) 24 điểm ảnh đầu vào. Mỗi một kết nối sẽ học một trọng số và mỗi neuron ẩn sẽ học một bias. Mỗi một vùng 5×5 đấy gọi là một trường tiếp nhận cục bộ.

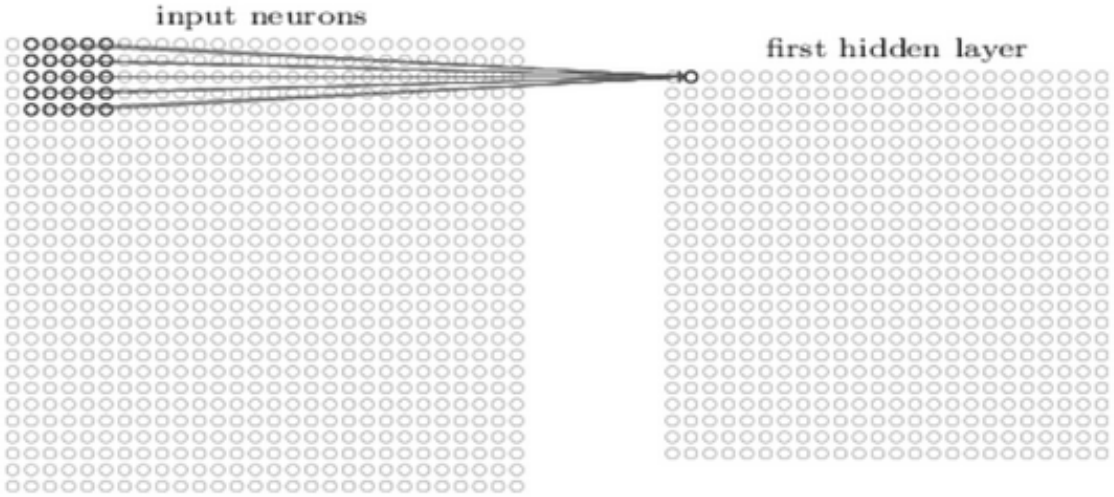


Một cách tổng quan, ta có thể tóm tắt các bước tạo ra 1 hidden layer bằng các cách sau:

- **Tạo ra neuron ẩn đầu tiên trong lớp ẩn 1**

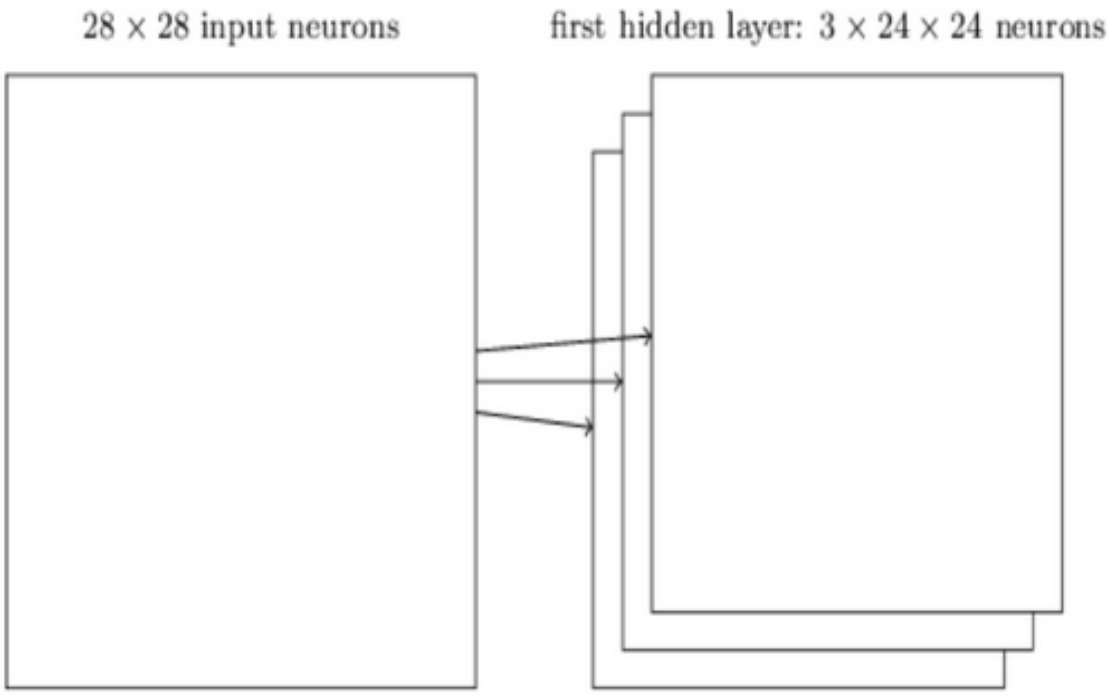


- **Dịch filter qua bên phải một cột sẽ tạo được neuron ẩn thứ 2**



Với bài toán nhận dạng ảnh người ta thường gọi ma trận lớp đầu vào là feature map, trọng số xác định các đặc trương là shared weight và độ lệch xác định một

feature map là shared bias. Như vậy đơn giản nhất là qua các bước trên chúng ta chỉ có 1 feature map. Tuy nhiên trong nhận dạng ảnh chúng ta cần nhiều hơn một feature map.



Như vậy, local receptive field thích hợp cho việc phân tách dữ liệu ảnh, giúp chọn ra những vùng ảnh có giá trị nhất cho việc đánh giá phân lớp.

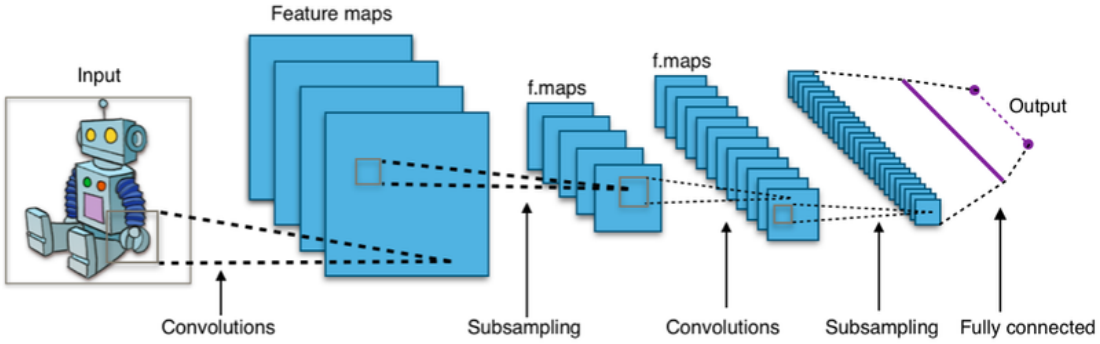
Trọng số chia sẻ (shared weight and bias))

Đầu tiên, các trọng số cho mỗi filter (kernel) phải giống nhau. Tất cả các nơ-ron trong lớp ẩn đầu sẽ phát hiện chính xác feature tương tự chỉ ở các vị trí khác nhau trong hình ảnh đầu vào. Chúng ta gọi việc map từ input layer sang hidden layer là một feature map. Vậy mối quan hệ giữa số lượng Feature map với số lượng tham số là gì?

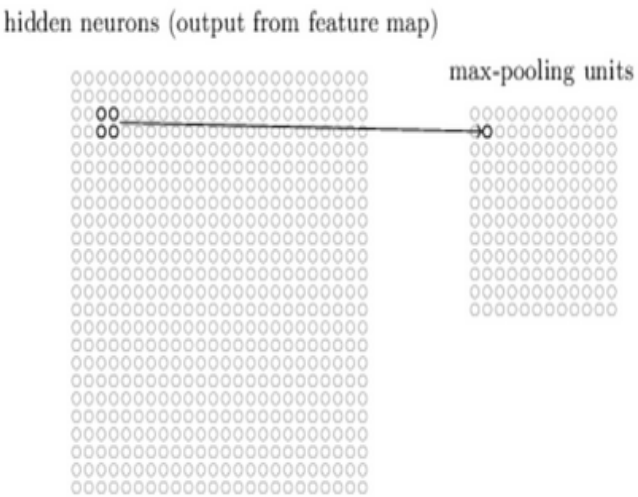
Tóm lại, một convolutional layer bao gồm các feature map khác nhau. Mỗi một feature map giúp detect một vài feature trong bức ảnh. Lợi ích lớn nhất của trọng số chia sẻ là giảm tối đa số lượng tham số trong mạng CNN.

Lớp tổng hợp (pooling layer)

Lớp pooling thường được sử dụng ngay sau lớp convulational để đơn giản hóa thông tin đầu ra để giảm bớt số lượng neuron.

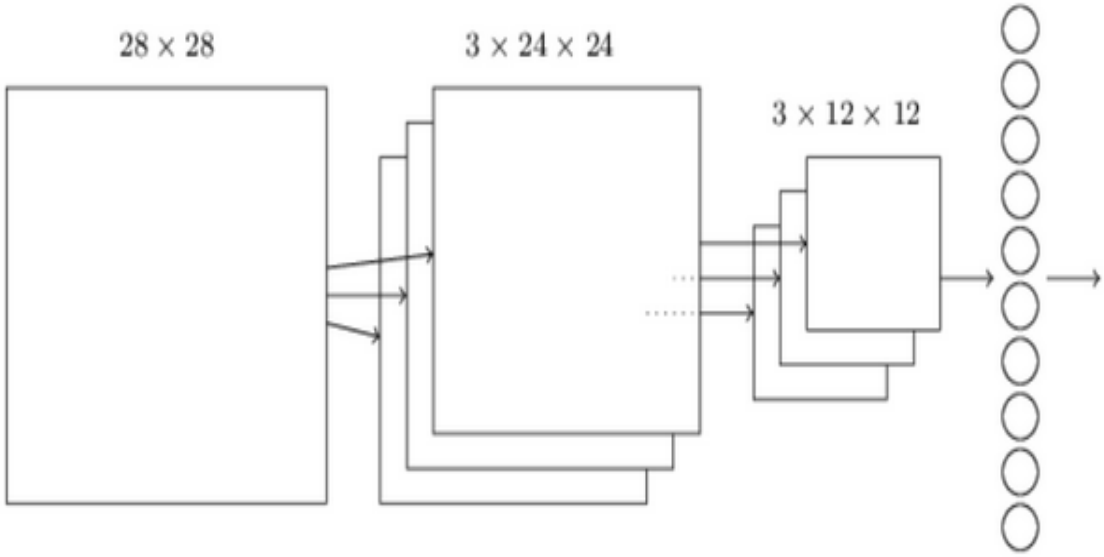


Thủ tục pooling phổ biến là max-pooling, thủ tục này chọn giá trị lớn nhất trong vùng đầu vào 2×2.



Như vậy qua lớp Max Pooling thì số lượng neuron giảm đi phân nửa. Trong một mạng CNN có nhiều Feature Map nên mỗi Feature Map chúng ta sẽ cho mỗi Max Pooling khác nhau. Chúng ta có thể thấy rằng Max Pooling là cách hỏi xem trong các đặc trưng này thì đặc trưng nào là đặc trưng nhất. Ngoài Max Pooling còn có L2 Pooling.

Cuối cùng ta đặt tất cả các lớp lại với nhau thành một CNN với đầu ra gồm các neuron với số lượng tùy bài toán.



2 lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối (fully connected layer) . Lớp này nối mọi nơron từ lớp max pooled tới mọi nơron của tầng ra.

Cách chọn tham số cho CNN)

- Số các convolution layer: càng nhiều các convolution layer thì performance càng được cải thiện. Sau khoảng 3 hoặc 4 layer, các tác động được giảm một cách đáng kể.

- Filter size: thường filter theo size 5×5 hoặc 3×3.
- Cách cuối cùng là thực hiện nhiều lần việc train test để chọn ra được param tốt nhất.

IV. Mô hình

4.1 Kiến trúc mô hình:

Sử dụng mô hình Convolutional Neural Network (CNN) dựa trên VGG16 với các lớp bổ sung để thực hiện fine-tuning.

4.2 VGG16

VGG16 là một mô hình CNN nổi tiếng, được phát triển bởi nhóm nghiên cứu Visual Geometry Group (VGG) tại Đại học Oxford. Mô hình này đã đạt kết quả tốt trong cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) năm 2014. VGG16 có các đặc điểm chính sau:

- **Kiến trúc:** Gồm 16 lớp chính, trong đó có 13 lớp Convolutional và 3 lớp Fully Connected (Dense).

- **Độ sâu:** Rất sâu, với nhiều lớp giúp trích xuất các đặc trưng phức tạp từ ảnh.

- **Trọng số đã học trước:** -VGG16 được huấn luyện trên bộ dữ liệu ImageNet với hàng triệu ảnh, giúp mô hình học được nhiều đặc trưng chung của ảnh.

4.3 Fine-Tuning

Fine-tuning là kỹ thuật điều chỉnh lại mô hình đã được huấn luyện trước để phù hợp với bài toán cụ thể. Thay vì huấn luyện một mô hình mới từ đầu (có thể mất nhiều thời gian và tài nguyên), chúng ta sử dụng một mô hình đã được huấn luyện trước (như VGG16) và huấn luyện lại các lớp cuối cùng của nó trên tập dữ liệu mới. Điều này giúp mô hình học được các đặc trưng cụ thể của bài toán mới mà vẫn tận dụng được các đặc trưng chung đã học từ trước.

4.4 Các lớp của mô hình

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 8, 8, 512)	14,714,688
conv2d (Conv2D)	(None, 6, 6, 64)	294,976
max_pooling2d (MaxPooling2D)	(None, 3, 3, 64)	0
flatten (Flatten)	(None, 576)	0
dropout (Dropout)	(None, 576)	0
dense (Dense)	(None, 128)	73,856
batch_normalization (BatchNormalization)	(None, 128)	512
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Mô hình sử dụng bao gồm các lớp sau:

- **Lớp đầu vào:** Kích thước: (256, 256, 3), tức là ảnh màu có kích thước 256x256 pixels với 3 kênh màu (RGB).
- **Mô hình cơ bản VGG16:**

Sử dụng phần mô hình VGG16 đã được huấn luyện trước trên ImageNet, nhưng không bao gồm các lớp cuối cùng (lớp phân loại).

Các lớp Convolutional của VGG16 sẽ trích xuất các đặc trưng từ ảnh như cạnh, góc, và các hình dạng cơ bản.

Mặc dù VGG16 bao gồm nhiều lớp, khi sử dụng mô hình đã được huấn luyện trước, nó được tích hợp như một lớp duy nhất trong kiến trúc mô hình lớn hơn.

- **Fine-tuning các lớp cuối của VGG16:** Mở khóa (unfreeze) các lớp cuối cùng của VGG16 để huấn luyện lại chúng trên tập dữ liệu mới. Điều này giúp mô hình học được các đặc trưng cụ thể hơn về viêm phổi từ ảnh X-quang.
- **Lớp Convolutional bổ sung:**

Thêm một lớp Convolutional với 64 filter kích thước 3x3 để trích xuất thêm các đặc trưng chi tiết từ ảnh.

Lớp MaxPooling giảm kích thước của các đặc trưng trích xuất, giúp giảm số lượng tham số và tránh overfitting.

- **Lớp Flatten:** Chuyển đổi đầu ra từ các lớp Convolutional thành một vector để đưa vào các lớp Fully Connected tiếp theo.
- **Lớp Dropout:** Tắt ngẫu nhiên một số đơn vị (neurons) trong quá trình huấn luyện để giảm overfitting, tức là tránh việc mô hình học quá chi tiết từ dữ liệu huấn luyện mà không tổng quát hóa được cho dữ liệu mới.
- **Lớp Dense với L2 Regularization:** Lớp Fully Connected (Dense) với 128 neurons và hàm kích hoạt ReLU. L2 Regularization giúp giảm overfitting bằng cách thêm một khoản phạt vào hàm loss cho các trọng số lớn.
- **Lớp Batch Normalization:** Giúp ổn định và tăng tốc quá trình huấn luyện bằng cách chuẩn hóa đầu ra của lớp trước đó.
- **Lớp Dense cuối cùng:** Lớp Fully Connected với 2 neurons và hàm kích hoạt sigmoid. Lớp này thực hiện phân loại cuối cùng cho hai lớp: viêm phổi và không

viêm phổi.

4.5 Lý do chọn mô hình:

VGG16 là một trong những mô hình CNN đã được huấn luyện trước trên tập dữ liệu ImageNet, có khả năng trích xuất đặc trưng tốt từ các ảnh y tế. Fine-tuning giúp tận dụng các đặc trưng đã học từ trước và điều chỉnh thêm cho phù hợp với bài toán cụ thể.

V. Quá trình huấn luyện

5.1 Tiền xử lý dữ liệu:

```
# Chuẩn bị dữ liệu với Data Augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.05,
    zoom_range=0.05,
    horizontal_flip=True,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1
)
```

Các ảnh được chuẩn hóa về kích thước (256x256) và giá trị pixel được chuẩn hóa về khoảng [0, 1]. Các kỹ thuật tăng cường dữ liệu (Data Augmentation) được áp dụng bao gồm:

- Lật ngang ảnh (horizontal flip)
- Thay đổi độ zoom (zoom range)
- Thay đổi độ cắt (shear range)
- Thay đổi vị trí ảnh (width_shift và height_shift)
- Xoay ảnh (rotation range)

5.2 Các tham số huấn luyện

- **Learning rate:** 1e-6

Learning rate là một trong những tham số quan trọng nhất trong quá trình huấn luyện mô hình. Nó quyết định tốc độ mà mô hình sẽ cập nhật các trọng số của nó. Nếu learning rate quá cao, mô hình có thể học quá nhanh và bỏ qua các đặc trưng quan trọng, dẫn đến không ổn định và kém hiệu quả. Nếu learning rate quá thấp, quá trình huấn luyện sẽ rất chậm và có thể bị mắc kẹt ở các điểm cực trị cục bộ. Chúng tôi chọn giá trị 1e-6 để đảm bảo mô hình học một cách ổn định và chính xác từ dữ liệu.

- **Batch size:** 64

Batch size là số lượng mẫu dữ liệu được đưa vào mô hình trong một lần cập nhật trọng số. Batch size lớn giúp quá trình huấn luyện ổn định hơn, vì các cập nhật trọng số được tính toán trên một tập dữ liệu lớn hơn, nhưng cũng đòi hỏi nhiều bộ nhớ hơn. Chúng tôi chọn giá trị 64 để cân bằng giữa độ ổn định và tài nguyên tính toán.

- **Số epoch:** 50

Epoch là số lần mà toàn bộ tập dữ liệu huấn luyện được đưa qua mô hình. Chúng tôi chọn 50 epoch để đảm bảo rằng mô hình có đủ thời gian để học các đặc trưng từ dữ liệu, nhưng không quá nhiều để tránh overfitting.

5.3 Kỹ thuật huấn luyện

- **Pretrained Model** (Mô hình đã huấn luyện trước)

```
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(img_width, img_height, 3))
for layer in base_model.layers[:-6]:
    layer.trainable = False
```

- Sử dụng mô hình VGG16 đã được huấn luyện trên tập dữ liệu ImageNet và loại bỏ phần top (phần fully connected layers).
- Chỉ các lớp cuối cùng của VGG16 được fine-tune, các lớp trước đó được "freeze" để giữ nguyên trọng số, giúp tận dụng kiến thức từ mô hình đã huấn luyện trước và giảm thời gian huấn luyện.

- **Regularization** (Điều chỉnh)

```
CNN.add(Dropout(0.5))
CNN.add(Dense(128, activation='relu', kernel_regularizer=l2(0.01)))
CNN.add(BatchNormalization())
CNN.add(Dropout(0.5))
```

- Dropout: Tắt ngẫu nhiên một phần của các đơn vị trong mạng nơ-ron trong quá trình huấn luyện để ngăn chặn overfitting. Trong đoạn mã này, Dropout được thiết lập với tỉ lệ 0.5, tức là 50% số đơn vị sẽ bị tắt ngẫu nhiên trong mỗi lần cập nhật.

- L2 Regularization: Áp dụng điều chuẩn L2 trên các trọng số của lớp Dense để ngăn chặn overfitting bằng cách thêm một hình phạt vào hàm mất mát.
- Batch Normalization: Giúp tăng tốc độ huấn luyện và ổn định mô hình bằng cách chuẩn hóa đầu vào của mỗi lớp sao cho có phân phối chuẩn.

- **Callbacks** (Gọi lại)

```
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-7)
```

- Early Stopping: Dừng huấn luyện khi không thấy sự cải thiện trên tập validation sau một số epoch nhất định (ở đây là 5 epoch). Điều này giúp tránh overfitting.
- ReduceLROnPlateau: Giảm learning rate khi hàm mất mát trên tập validation không cải thiện sau một số epoch nhất định (ở đây là 3 epoch). Điều này giúp mô hình tiếp tục học với learning rate nhỏ hơn khi gặp plateau (không còn cải thiện).

- **Optimizer with Gradient Clipping** (Tối ưu hóa với cắt gradient)

```
model.compile(optimizer=Adam(learning_rate=1e-6, clipnorm=1.0), loss='binary_crossentropy', metrics=['accuracy'])
```

- Adam Optimizer: Một thuật toán tối ưu hóa kết hợp giữa RMSProp và Stochastic Gradient Descent, giúp tăng tốc độ hội tụ.
- Gradient Clipping: Giới hạn norm của gradient (ở đây là 1.0) để tránh vấn đề gradient bùng nổ, giúp mô hình học ổn định hơn.

Những kỹ thuật này kết hợp lại giúp mô hình học tốt hơn, ổn định hơn và tránh overfitting, cải thiện hiệu suất và độ chính xác của mô hình trên dữ liệu kiểm thử.

VI. Kết quả sau khi học

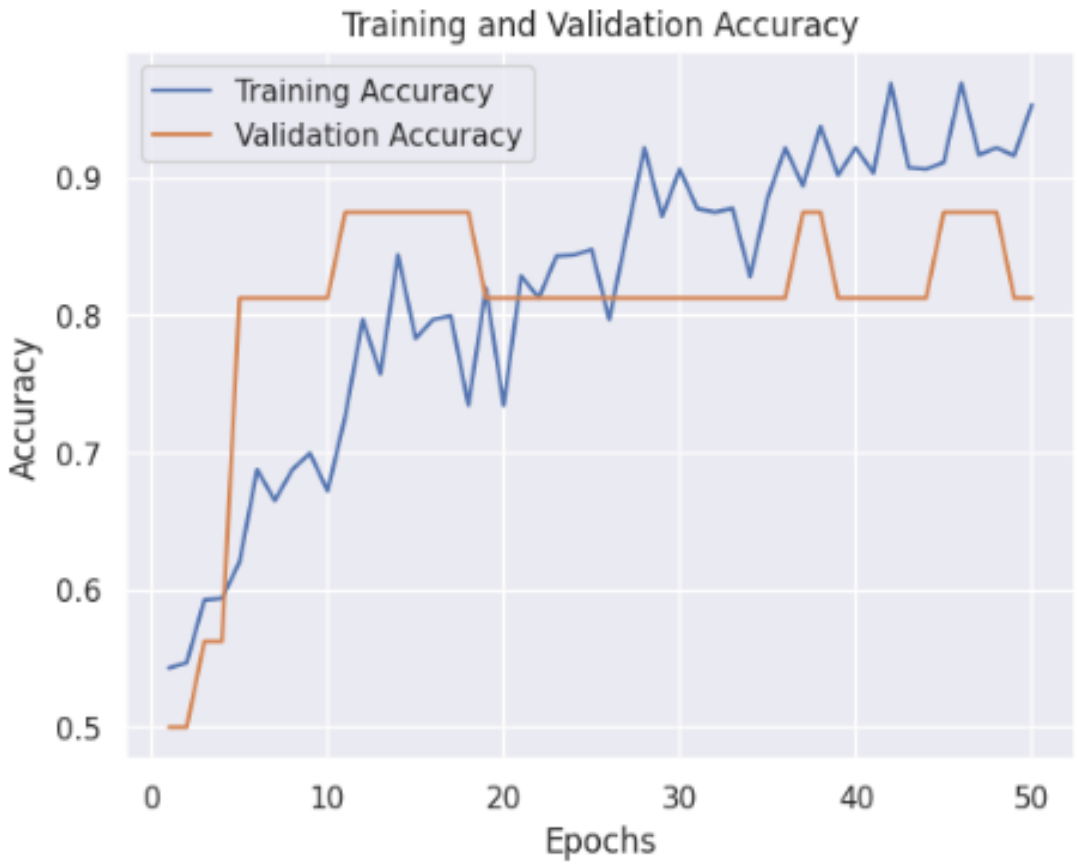
6.1 Training và Validation Loss

- Xu hướng giảm dần: Cả loss trên tập huấn luyện và tập validation đều giảm dần theo thời gian, điều này cho thấy mô hình đang học và cải thiện.

- Sự hội tụ: Loss trên tập validation hội tụ và ổn định quanh một giá trị, cho thấy mô hình đã tìm được một mức độ mất mát chấp nhận được.

- Chênh lệch loss: Loss trên tập huấn luyện thấp hơn so với tập validation, điều này là bình thường, nhưng sự chênh lệch nhỏ này cho thấy mô hình đang hoạt động khá tốt mà không bị overfitting nghiêm trọng.

6.2 Training and Validation Accuracy



- Độ chính xác trên tập huấn luyện: Độ chính xác trên tập huấn luyện tăng dần và dao động xung quanh mức cao (gần 0.9), cho thấy mô hình đang học và tối ưu hóa tốt trên tập huấn luyện.

- Độ chính xác trên tập validation: Độ chính xác trên tập validation dao động ít hơn và ổn định quanh mức 0.8. Điều này cho thấy mô hình có thể generalize tương đối tốt cho dữ liệu chưa từng thấy trước đó.

- Sự ổn định: Độ chính xác trên tập validation không tăng sau một số epoch nhất định và có xu hướng phẳng, điều này có thể là dấu hiệu của việc mô hình đã đạt đến ngưỡng tối ưu trên tập dữ liệu này.

6.3 Đánh giá tổng quan

- Không có overfitting nghiêm trọng: Mặc dù có một số chênh lệch giữa loss và accuracy trên tập huấn luyện và tập validation, nhưng điều này không quá lớn, cho thấy mô hình không bị overfitting nghiêm trọng.

- Dừng sớm hợp lý: Việc sử dụng early stopping giúp mô hình không tiếp tục huấn luyện khi không còn cải thiện, từ đó tránh overfitting.

- Học tốt trên tập huấn luyện: Mô hình học tốt trên tập huấn luyện với độ chính xác cao và giảm loss đáng kể.

- Generalization hợp lý: Mô hình generalize khá tốt trên tập validation, với độ chính xác ổn định ở mức 0.8, cho thấy mô hình không chỉ học thuộc lòng dữ liệu huấn luyện mà còn áp dụng được kiến thức học được cho dữ liệu mới.

VII. Đánh giá

[10]:

```python
# Đánh giá mô hình trên tập dữ liệu kiểm tra
test.batch_size = 16
test_loss = []
test_accuracy = []
for i in range(len(test)):
    batch = test[i]
    loss, accuracy = model.evaluate(batch[0], batch[1], verbose=0)
    test_loss.append(loss)
    test_accuracy.append(accuracy)

# Tính toán loss và accuracy trung bình
test_loss = np.mean(test_loss)
test_accuracy = np.mean(test_accuracy)

print(f'Test Loss: {test_loss}')
print(f'Test Accuracy: {test_accuracy}')
```

```
Test Loss: 2.1470929506497507
Test Accuracy: 0.8830128205128205
```

7.1 Độ chính xác (Accuracy)

Độ chính xác của mô hình đạt 88.3%, điều này cho thấy mô hình có khả năng phân loại đúng khoảng 88.3% số lượng ảnh trong tập dữ liệu kiểm tra. Đây là một kết quả khá tốt, đặc biệt khi xét đến tính chất phức tạp của việc phân loại ảnh y tế.

7.2 Mất mát (Loss)

Giá trị mất mát (loss) của mô hình là 2.147. Giá trị này tương đối cao, cho thấy mô hình có thể chưa tối ưu hoàn toàn và vẫn còn một số dự đoán sai lệch.