

LỜI CAM ĐOAN

Tôi xin cam đoan: Luận văn thạc sỹ chuyên ngành Khoa học máy tính, tên đề tài “Nghiên cứu hỗ trợ chuẩn đoán bệnh lao dựa vào học máy” là công trình nghiên cứu, tìm hiểu và trình bày do tôi thực hiện dưới sự hướng dẫn khoa học của **PGS.TS. Đỗ Năng Toàn**, Viện Công nghệ Thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.

Kết quả tìm hiểu, nghiên cứu trong luận văn là hoàn toàn trung thực, không vi phạm bất cứ điều gì trong luật sở hữu trí tuệ và pháp luật Việt Nam. Nếu sai, tôi hoàn toàn chịu trách nhiệm trước pháp luật.

Tất cả các tài liệu, bài báo, khóa luận, công cụ phần mềm của các tác giả khác được sử dụng lại trong luận văn này đều được chỉ dẫn tường minh về tác giả và đều có trong danh mục tài liệu tham khảo.

Thái Nguyên, ngày 30 tháng 6 năm 2022.

Tác giả luận văn
Nguyễn Hữu Khánh

LỜI CẢM ƠN

Tác giả xin chân thành cảm ơn **PGS.TS. Đỗ Năng Toàn**, Viện Công nghệ Thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, là giáo viên hướng dẫn khoa học đã hướng dẫn tác giả hoàn thành luận văn này, xin được cảm ơn các thầy, cô giáo trường Đại học công nghệ thông tin và truyền thông nơi tác giả theo học và hoàn thành chương trình cao học đã nhiệt tình giảng dạy và giúp đỡ.

Xin cảm ơn Trung tâm Đào tạo Từ xa - Đại học Thái Nguyên nơi tác giả công tác đã tạo mọi điều kiện thuận lợi để tác giả có thời gian, tâm trí để hoàn thành nhiệm vụ nghiên cứu và chương trình học tập.

Và cuối cùng xin cảm ơn gia đình, bạn bè, đồng nghiệp đã động viên, giúp đỡ tác giả trong suốt thời gian học tập, nghiên cứu và hoàn thành luận văn này.

Xin chân thành cảm ơn.

Thái Nguyên, ngày 30 tháng 06 năm 2022

Tác giả luận văn
Nguyễn Hữu Khánh

Mục lục

Lời cam đoan.....	i
Lời cảm ơn	ii
Mở đầu.....	1
Chương 1. Khái quát về CNN và bài toán chuẩn đoán bệnh lao...	4
1.1. Khái quát về CNN.....	4
1.1.1. Giới thiệu.....	4
1.1.2. Kiến trúc mạng CNN.....	6
1.1.3. Ứng dụng CNN trong phân loại ảnh.....	11
1.1.4. Xây dựng mạng CNN cho phân loại ảnh.....	12
1.2. Bài toán chuẩn đoán bệnh lao.....	21
1.2.1. Mô tả bài toán.....	22
Chương 2. Một số mô hình hỗ trợ chuẩn đoán.....	24
2.1. VGG-16.....	24
2.1.1. VGG16 là gì.....	24
2.1.2. Kiến trúc VGG16	25
2.1.3. Trường hợp sử dụng VGG16.....	27
2.1.4. Áp dụng VGG16 vào bài toán của luận văn.....	28
2.1.5. Cài đặt mô hình.....	28
Chương 3. Chương trình thử nghiệm.....	30
Tài liệu tham khảo	30

MỞ ĐẦU

Theo báo cáo của Tổ chức Y tế thế giới (TCYTTG - WHO Report 2020 - Global Tuberculosis Control)[1], mặc dù đã đạt được một số thành tựu đáng kể trong công tác chống lao trong thời gian qua, bệnh lao vẫn đang tiếp tục là một trong các vấn đề sức khỏe cộng đồng chính trên toàn cầu. TCYTTG ước tính năm 2019 trên toàn cầu có khoảng 10 triệu người hiện mắc lao, một con số đã giảm rất chậm trong những năm gần đây; 8,2% trong số mắc lao có đồng nhiễm HIV. Bệnh lao là nguyên nhân gây tử vong đứng hàng thứ hai trong các bệnh nhiễm trùng với khoảng 1,2 triệu người tử vong do lao và khoảng 208.000 người chết do lao trong số những người nhiễm HIV. Số tử vong này làm cho lao là một trong các bệnh gây tử vong hàng đầu ở nữ giới. WHO đã công bố kết quả của mô hình đánh giá tác động ngắn hạn của đại dịch Covid-19 lên số ca tử vong do lao trong năm 2020. Kết quả cho thấy rằng tử vong do lao có thể tăng đáng kể trong năm 2020 và sẽ ảnh hưởng đến nhóm bệnh nhân lao dễ bị tổn thương nhất, tăng khoảng 200.000 – 400.000 ca tử vong, nếu như các dịch vụ chẩn đoán và phát hiện bệnh nhân trên toàn cầu bị ngưng trệ và giảm từ 25 – 50% trong khoảng 3 tháng. Con số tử vong sẽ tương ứng với mức tử vong toàn cầu do lao vào năm 2015, một bước lùi nghiêm trọng trong quá trình hướng tới mục tiêu của Hội nghị Cấp Cao Liên Hợp Quốc về Lao và Chiến lược thanh toán bệnh lao của WHO.

Về Việt nam, hiện chúng ta vẫn là nước có gánh nặng bệnh lao cao, đứng thứ 11 trong 30 nước có số người bệnh lao cao nhất trên toàn cầu, đồng thời đứng thứ 11 trong số 30 nước có gánh nặng bệnh lao kháng đa thuốc cao nhất thế giới[1]. Bệnh lao vẫn là một trong những bệnh truyền nhiễm phổ biến ở Việt Nam. Hàng năm, ước tính có 17.000 trường hợp tử vong do lao tại Việt Nam, cao hơn gấp hai lần so với con số tử vong do tai nạn giao thông. Mỗi năm ước tính có 180.000 người có bệnh lao hoạt động; 5.000 trường hợp trong số đó được xác định nhiễm lao kháng đa thuốc. Chẩn đoán bệnh lao không thật sự khó trong đa số các trường hợp. Điều đáng chú ý là làm sao chẩn đoán sớm và chẩn đoán đúng để khởi động điều trị sớm nhằm giảm các tổn thương cũng như biến chứng của lao gây ra. Để làm được điều trên, việc ứng dụng công nghệ thông tin vào quá trình chuẩn đoán là thực sự cần thiết, đặc biệt là áp dụng những tiến bộ của học máy để xây dựng lên hệ thống hỗ trợ chuẩn đoán bệnh lao.

Đối tượng nghiên cứu: Ảnh X-quang lồng ngực trong y tế thu nhận bởi các máy chiếu, chụp chuyên dụng.

Phạm vi nghiên cứu: Ảnh đa mức xám chụp phổi thẳng thường quy (tư thế sau - trước), chụp phổi nghiêng thường quy và chụp đỉnh phổi tư thế ưỡn ngực.

Những nội dung nghiên cứu chính: Dự kiến nội dung báo cáo của luận văn gồm: phần mở đầu, 3 chương chính, phần kết luận, tài liệu tham khảo, phụ lục. Bố cục được trình bày như sau:

Phần mở đầu: Nêu lý do chọn đề tài và hướng nghiên cứu chính

Chương 1: Khái quát về CNN và bài toán chuẩn đoán bệnh lao.

Chương 2: Một số mô hình hỗ trợ chuẩn đoán.

Chương 3: Chương trình thử nghiệm.

Mặc dù đã có cố gắng nỗ lực, song luận văn không tránh khỏi những thiếu sót do năng lực và thời gian hạn chế. Em chân thành mong muốn lắng nghe những đóng góp, góp ý của thầy, cô, bạn bè, đồng nghiệp để luận văn được cải thiện tốt hơn.

Em xin chân thành cảm ơn.

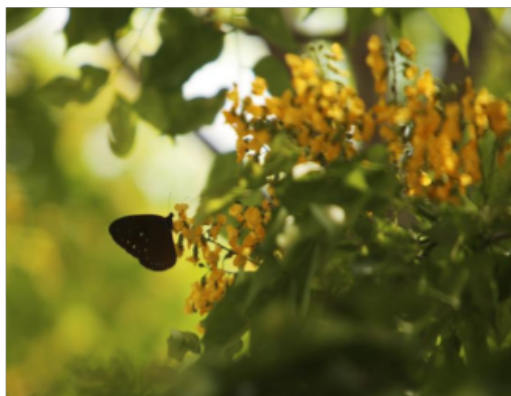
CHƯƠNG 1

Khái quát về CNN và bài toán chuẩn đoán bệnh lao.

1.1. Khái quát về CNN

1.1.1. Giới thiệu

Tương tự như việc trẻ em học cách nhận diện đối tượng, chúng ta cần cho thuật toán học rất nhiều hình ảnh trước khi nó có thể đưa ra phân loại cho hình ảnh đầu vào mà nó chưa từng thấy [2].



Chúng ta nhìn thấy



157	48	240	49	2	78	229	64	207	32
159	54	94	218	126	97	60	163	60	69
128	201	202	100	53	5	4	131	49	209
199	132	202	121	119	238	49	220	76	149
192	63	21	129	16	226	104	32	255	15
124	225	229	180	141	155	153	100	20	252
90	17	238	232	34	209	64	187	197	210
212	244	86	30	192	160	85	195	36	111
226	221	201	223	161	170	114	154	9	68
252	217	1	107	127	126	50	26	151	193

Máy tính thấy

Hình 1.1: Cách máy tính "nhìn" một hình ảnh.

Máy tính “nhìn” theo cách khác con người. Trong thế giới máy tính chỉ có những con số. Mỗi hình ảnh có thể được biểu diễn dưới dạng mảng 2 chiều những con số được gọi là các pixel.

Mặc dù máy tính nhìn nhận theo cách khác con người, chúng ta vẫn có thể dạy máy tính nhận diện các mẫu như con người. Điều quan trọng là chúng ta cần nghĩ về hình ảnh theo một cách khác đi.

Để dạy thuật toán nhận diện đối tượng trong hình ảnh, ta sử dụng một loại mạng ANN, đó là Convolutional Neural Networks (CNN). Tên của nó

được dựa trên phép tính quan trọng được sử dụng trong mạng- tích chập.

Convolutional Neural Networks (CNN) là một trong những mô hình deep learning phổ biến nhất và có ảnh hưởng nhiều nhất trong cộng đồng Computer Vision. CNN được dùng trong nhiều bài toán như nhận dạng ảnh, phân tích video, ảnh MRI, hoặc cho các bài của lĩnh vực xử lý ngôn ngữ tự nhiên, và hầu hết đều giải quyết tốt các bài toán này.

Mạng CNN lấy cảm hứng từ não người [5]. Nghiên cứu trong những thập niên 1950 và 1960 của D.H Hubel và T.N Wiesel trên não của động vật đã đề xuất một mô hình mới cho việc cách mà động vật nhìn nhận thế giới. Trong báo cáo, hai ông đã diễn tả 2 loại tế bào neural trong não và cách hoạt động khác nhau: tế bào đơn giản (simple cell – S cell) và tế bào phức tạp (complex cell – C cell).

Các tế bào đơn giản được kích hoạt khi nhận diện các hình dáng đơn giản như đường nằm trong một khu vực cố định và một góc cạnh của nó. Các tế bào phức tạp có vùng tiếp nhận lớn hơn và đầu ra của nó không nhạy cảm với những vị trí cố định trong vùng.

Trong thị giác, vùng tiếp nhận của một neural tương ứng với một vùng trên võng mạc nơi mà sẽ kích hoạt neural tương ứng.

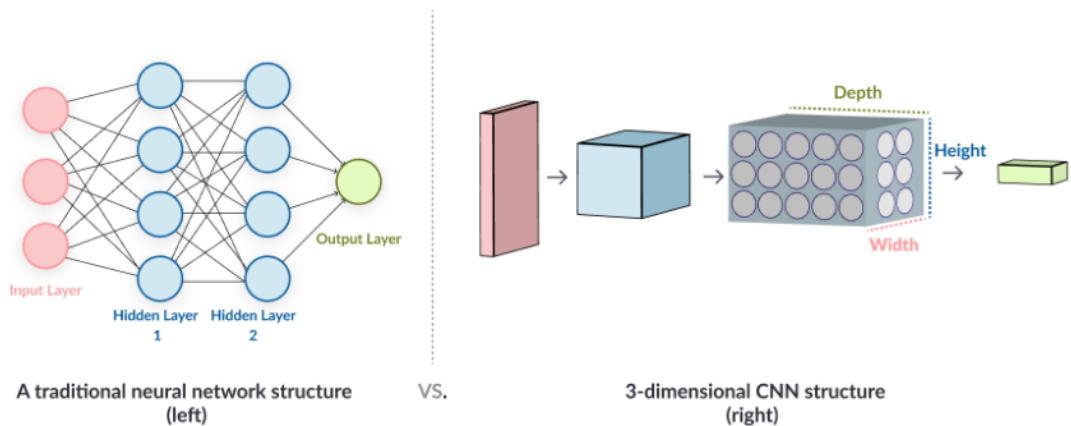
Năm 1980, Fukushima đề xuất mô hình mạng neural có cấp bậc gọi là neocognitron. Mô hình này dựa trên khái niệm về S cell và C cell. Mạng neocognitron có thể nhận diện mẫu dựa trên việc học hình dáng của đối tượng.

Sau đó vào năm 1998, mạng CNN được giới thiệu bởi Bengio, Le Cun, Bottou và Haffner. Mô hình đầu tiên của họ được gọi tên là LeNet-5. Mô hình này có thể nhận diện chữ số viết tay.

1.1.2. Kiến trúc mạng CNN

Mạng CNN có kiến trúc khác với Mạng Neural thông thường. Mạng ANN bình thường chuyển đổi đầu vào thông qua hàng loạt các tầng ẩn. Mỗi tầng là một tập các neural và các tầng được liên kết đầy đủ với các neural ở tầng trước đó. Và ở tầng cuối cùng sẽ là tầng kết quả đại diện cho dự đoán của mạng.

Đầu tiên, mạng CNN được chia thành 3 chiều: rộng, cao, và sâu. Kể đến, các neural trong mạng không liên kết hoàn toàn với toàn bộ neural lớp sau mà chỉ liên kết tới một vùng nhỏ. Cuối cùng, một tầng đầu ra được tối giản thành véc-tơ của giá trị xác suất (hình 1.2).



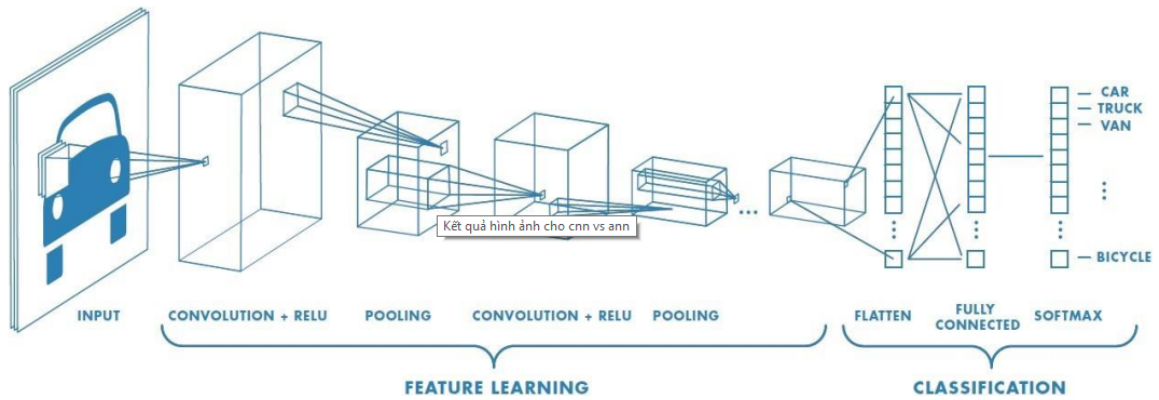
Hình 1.2: Mạng Neural thông thường (trái) và CNN (phải).

Mạng CNN gồm hai thành phần:

Phần tầng ẩn hay phần rút trích đặc trưng: trong phần này, mạng sẽ tiến hành tính toán hàng loạt phép tích chập và phép hợp nhất (pooling) để phát hiện các đặc trưng. Ví dụ: nếu ta có hình ảnh con ngựa vằn, thì trong phần này mạng sẽ nhận diện các sọc vằn, hai tai, và bốn chân của nó.

Phần phân lớp: tại phần này, một lớp với các liên kết đầy đủ sẽ

đóng vai trò như một bộ phân lớp các đặc trưng đã rút trích được trước đó. Tầng này sẽ đưa ra xác suất của một đối tượng trong hình 1.3.



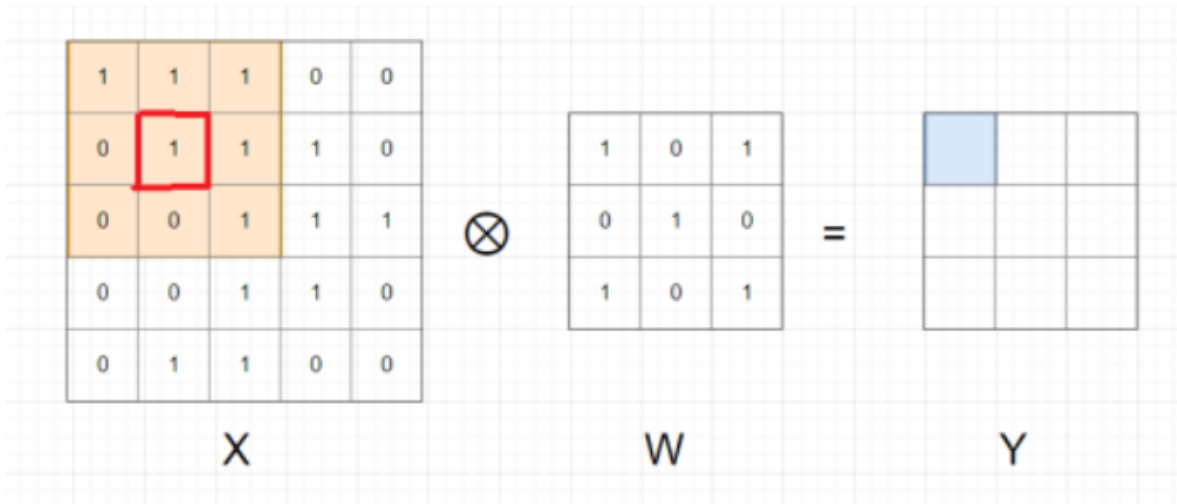
Hình 1.3: Kiến trúc mạng CNN.

1.1.2.1. Trích rút đặc trưng

1.1.2.1.1. Lớp tích chập Tích chập là một khối quan trọng trong CNN. Thuật ngữ tích chập được dựa trên một phép hợp nhất toán học của hai hàm tạo thành hàm thứ ba. Phép toán này kết hợp hai tập thông tin khác nhau.

Trong trường hợp CNN, tích chập được thực hiện trên giá trị đầu vào của dữ liệu và kernel/filter (thuật ngữ này được sử dụng khác nhau tùy tình huống) để tạo ra một bản đồ đặc trưng (feature map).

Ta thực hiện phép tích chập bằng cách trượt kernel/filter theo dữ liệu đầu vào. Tại mỗi vị trí, ta tiến hành phép nhân ma trận và tính tổng các giá trị để đưa vào bản đồ đặc trưng. Thao tác này đã được minh họa cụ thể trong hình 1.4



Hình 1.4: Minh họa phép tích chập.

Trong thực tế, tích chập được thực hiện hiện trên không gian 3 chiều. Vì mỗi hình ảnh được biểu diễn dưới dạng 3 chiều: rộng, cao, và sâu. Chiều sâu ở đây chính là giá trị màu sắc của hình (RGB).

Ta thực hiện phép tích chập trên đầu vào nhiều lần khác nhau. Mỗi lần sử dụng một kernel/filter khác nhau. Kết quả ta sẽ thu được những bản đồ đặc trưng khác nhau. Cuối cùng, ta kết hợp toàn bộ bản đồ đặc trưng này thành kết quả cuối cùng của tầng tích chập.

1.1.2.1.2. Lớp ReLU Tương tự như mạng neural thông thường, ta sử dụng một hàm kích hoạt (activate function) để có đầu ra dưới dạng phi tuyến. Trong trường hợp CNN, đầu ra của phép tích chập sẽ đi qua hàm kích hoạt nào đó ví dụ như hàm tinh chỉnh các đơn vị tuyến tính (Rectified linear units - ReLU).

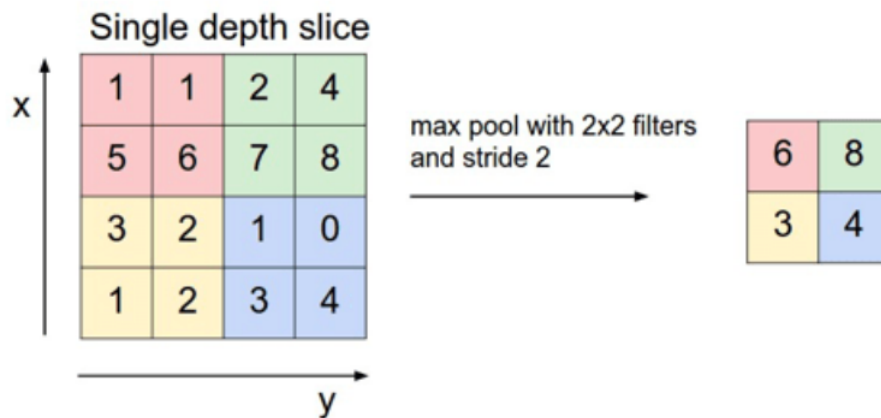
Trong quá trình trượt kernel/filter trên dữ liệu đầu vào, ta sẽ quy định một bước nhảy (stride) với mỗi lần di chuyển. Thông thường ta lựa chọn thường chọn bước nhảy là 1. Nếu kích thước bước nhảy tăng, kernel/filter sẽ có ít ô trùng lặp.

Bởi vì kích thước đầu ra luôn nhỏ hơn đầu vào nên ta cần một phép

xử lí đầu vào để đầu ra không bị co giãn. Đơn giản ta chỉ cần thêm một lề nhỏ vào đầu vào. Một lề (padding) với giá trị 0 sẽ được thêm vào xung quanh đầu vào trước khi thực hiện phép tích chập.

1.1.2.1.3. Lớp pooling Thông thường, sau mỗi tầng tích chập, ta sẽ cho kết quả đi qua một tầng hợp nhất (pooling layer). Mục đích của tầng này là để nhanh chóng giảm số chiều. Việc này giúp giảm thời gian học và hạn chế việc overfitting.

Một phép hợp nhất đơn giản thường được dùng đó là max pooling, phép này lấy giá trị lớn nhất của một vùng để đại diện cho vùng đó. Kích thước của vùng sẽ được xác định trước để giảm kích thước của bản đồ đặc trưng nhanh chóng nhưng vẫn giữ được thông tin cần thiết (hình 1.5).



Hình 1.5: Max pooling kích thước 2x2

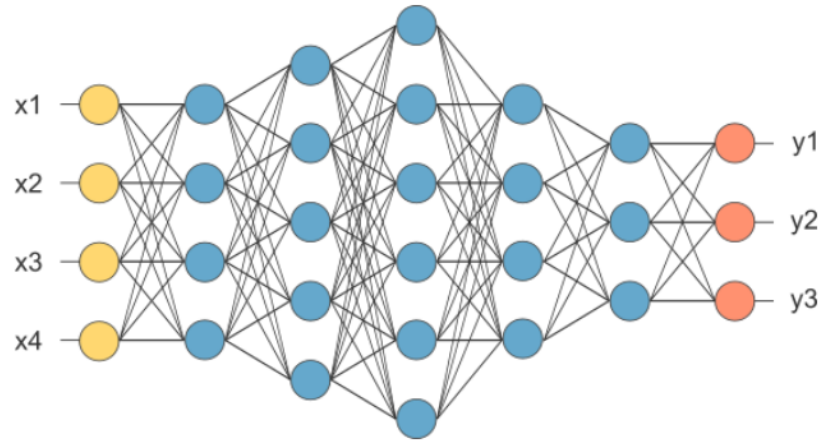
Như vậy, khi thiết kế phần rút trích đặc trưng của mạng CNN, ta cần chú ý đến 4 siêu tham số quan trọng là: Kích thước kernel/filter, Số lượng kernel/filter, Kích thước bước nhảy (stride), Kích thước lề (padding).

1.1.2.2. Phân lớp

Trong phần phân lớp, ta sử dụng một vài tầng với kết nối đầy đủ để xử lí kết quả của phần tích chập. Vì đầu vào của mạng liên kết đầy đủ là 1

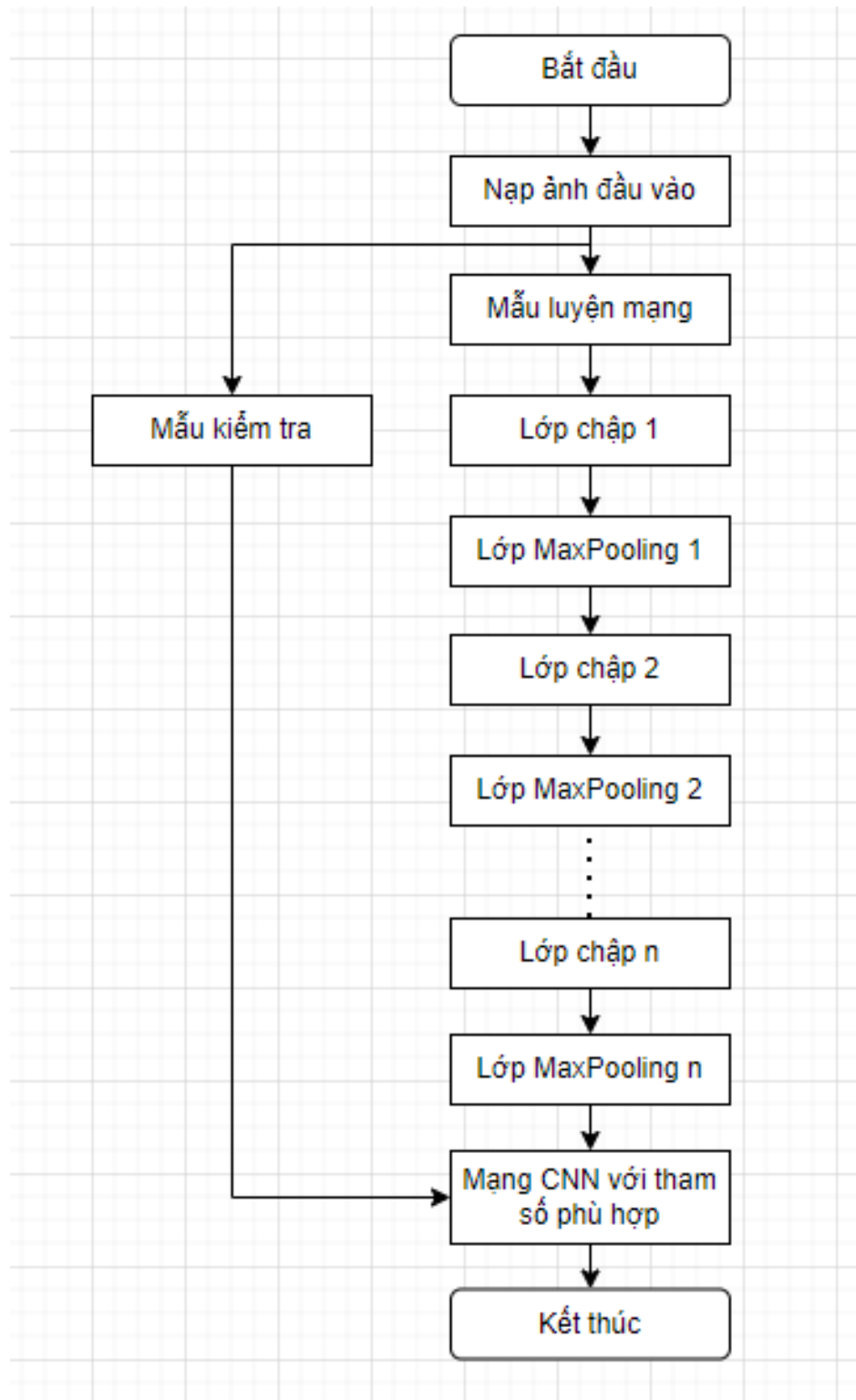
chiều, ta cần làm phẳng đầu vào trước khi phân lớp. Tầng cuối cùng trong mạng CNN là một tầng liên kết đầy đủ, phần này hoạt động tương tự như mạng neural thông thường.

Kết quả thu được cuối cùng cũng sẽ là một véc-tơ với các giá trị xác suất cho việc dự đoán như mạng neural thông thường.



Hình 1.6: Lớp kết nối đầy đủ

1.1.3. Ứng dụng CNN trong phân loại ảnh



Hình 1.7: Max pooling kích thước 2×2

Các bước để thực hiện phân loại hình ảnh dựa trên mạng CNN được mô tả trong Hình 1.7. Đầu tiên, kho dữ liệu ảnh đầu vào được nạp. Ảnh này được chia làm hai phần, một phần dành cho luyện mạng và một phần cho kiểm tra. Trước tiên, ta phải lựa chọn cấu trúc mạng CNN bao gồm số lượng lớp ẩn, các tham số trong mỗi lớp ẩn như kích thước trường tiếp nhận cục bộ, stride, padding. Ảnh luyện mạng sau đó được đưa vào lớp chập 1 để thực hiện tích chập trên ảnh và thực hiện hàm ReLU. Sau đó, kết quả được đưa đến quá trình thực hiện pooling với tham số pooling size phù hợp để giảm kích cỡ ảnh. Ảnh sẽ tiếp tục được đưa thêm qua các lớp tích chập nữa cho đến khi đạt được kết quả mong muốn. Kết quả này được dàn phẳng và đưa vào lớp kết nối đầy đủ. Cuối cùng là quá trình thực hiện các activation function và phân loại ảnh. Quá trình luyện mạng sẽ kết thúc sau khi tổng sai số nhỏ hơn một ngưỡng cho phép hoặc sau một số thế hệ cho trước (điều kiện hội tụ). Kết thúc của quá trình luyện mạng là cấu trúc mạng CNN với các tham số phù hợp. Để kiểm tra, các mẫu ảnh kiểm tra được đưa qua mạng CNN rồi thực hiện đánh giá sai số.

1.1.4. Xây dựng mạng CNN cho phân loại ảnh

Trước tiên, đối với mỗi điểm ảnh trong ảnh đầu vào, ta mã hóa cường độ của điểm ảnh là giá trị của neural tương ứng trong tầng đầu vào.

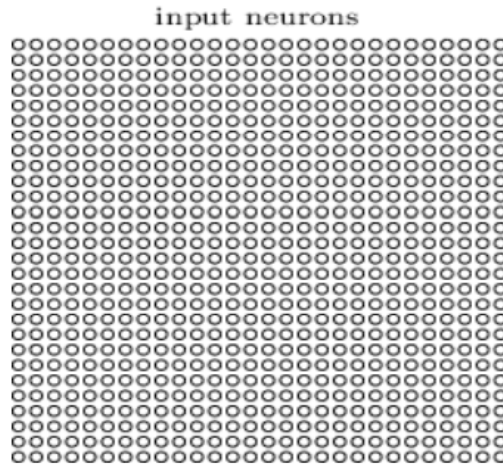
Ví dụ đối với bài toán nhận dạng chữ viết tay từ tập dữ liệu MNIST, mỗi bức ảnh kích thước 28x28 điểm ảnh. Do vậy, mạng có 784 (28x28) neural đầu vào (Hình 1.8). Sau đó ta huấn luyện trọng số (weight) và độ lệch (bias) để đầu ra của mạng như ta mong đợi là xác định chính xác ảnh các chữ số 0, 1, 2...8, 9.

Mạng tích chập sử dụng 3 ý tưởng cơ bản: các trường tiếp nhận cục bộ (local receptive field), trọng số chia sẻ (shared weights) và tổng hợp

(pooling). Chúng ta hãy xem xét lần lượt từng ý tưởng.

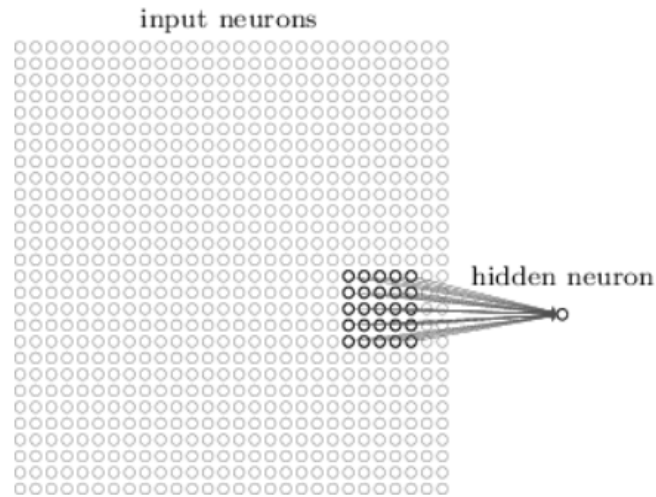
1.1.4.1. Trường tiếp nhận cục bộ (Local receptive fields)

Như thường lệ chúng ta sẽ kết nối các điểm ảnh đầu vào cho các neural ở tầng ẩn. Nhưng chúng ta sẽ không kết nối mỗi điểm ảnh đầu vào cho mỗi neural ẩn. Thay vào đó, chúng ta chỉ kết nối trong phạm vi nhỏ, các vùng cục bộ của bức ảnh.



Hình 1.8: Lớp input gồm 28x28 neural cho nhận dạng chữ từ tập dữ liệu MNIST

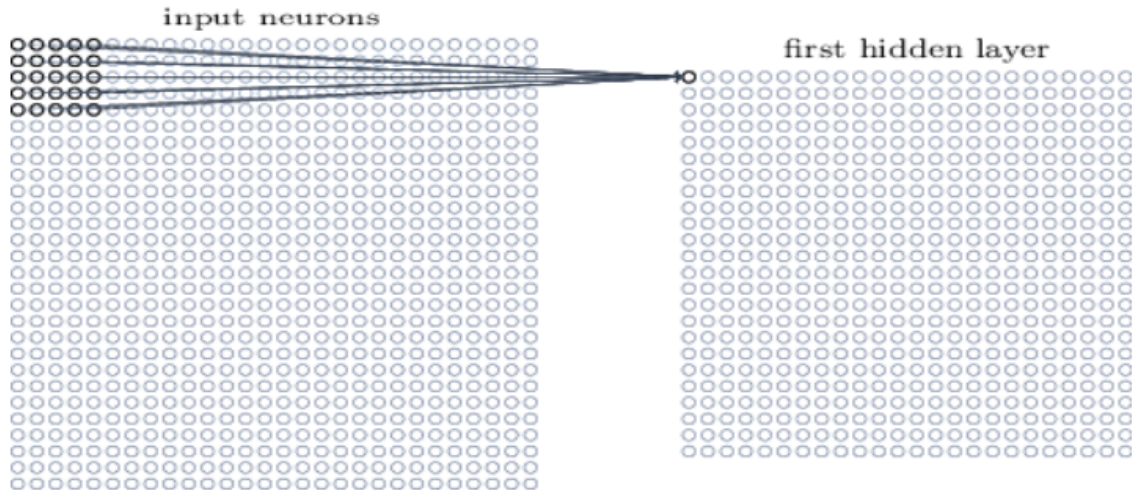
Để được chính xác hơn, mỗi neural trong lớp ẩn đầu tiên sẽ được kết nối với một vùng nhỏ của các neural đầu vào, ví dụ, một vùng 5×5 , tương ứng với 25 điểm ảnh đầu vào. Vì vậy, đối với một neural ẩn cụ thể, chúng ta có thể có các kết nối như Hình 1.9 sau:



Hình 1.9: Kết nối vùng 5x5 neural input với neural lớp ẩn

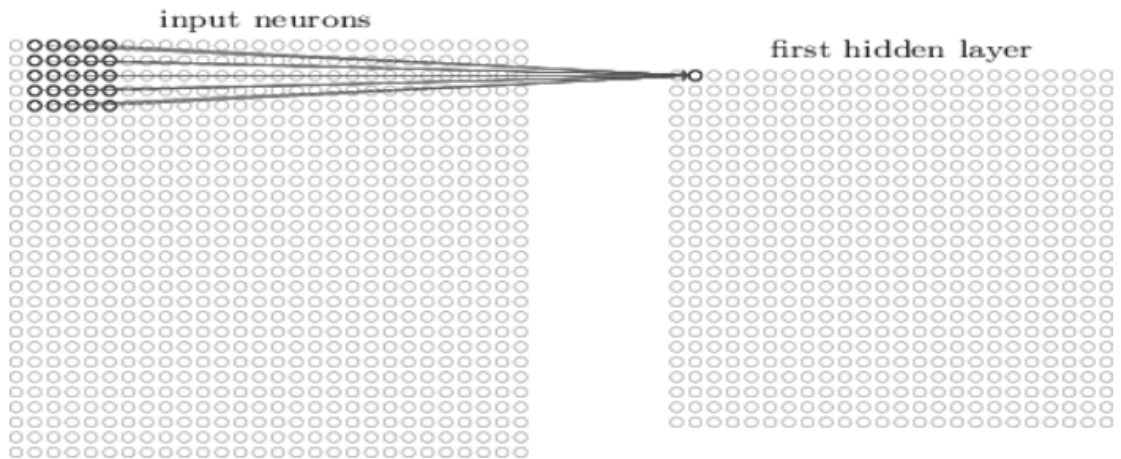
Vùng đó trong bức ảnh đầu vào được gọi là vùng tiếp nhận cục bộ cho neural ẩn. Đó là một cửa sổ nhỏ trên các điểm ảnh đầu vào. Mỗi kết nối sẽ học một trọng số và neural ẩn cũng sẽ học một độ lệch (overall bias). Ta có thể hiểu rằng, neural lớp ẩn cụ thể học để phân tích trường tiếp nhận cục bộ cụ thể của nó.

Sau đó chúng ta trượt trường tiếp nhận cục bộ trên toàn bộ bức ảnh. Đối với mỗi trường tiếp nhận cục bộ, có một neural ẩn khác trong tầng ẩn đầu tiên. Để minh họa điều này một cách cụ thể, chúng ta hãy bắt đầu với một trường tiếp nhận cục bộ ở góc trên bên trái (Hình 1.10):



Hình 1.10: Vị trí bắt đầu của trường tiếp nhận cục bộ

Sau đó, chúng ta trượt trường tiếp nhận cục bộ trên bởi một điểm ảnh bên phải (tức là bằng một neuron), để kết nối với một neural ẩn thứ hai (Hình 1.11):

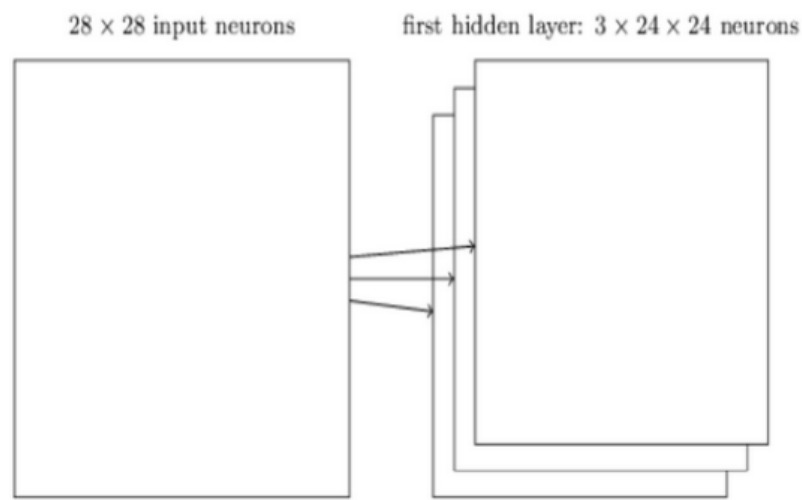


Hình 1.11: Vị trí thứ 2 của trường tiếp nhận cục bộ và neural lớp ẩn

Cứ như vậy, ta sẽ xây dựng các lớp ẩn đầu tiên. Lưu ý rằng nếu chúng ta có một ảnh đầu vào 28×28 và 5×5 trường tiếp nhận cục bộ thì ta sẽ có 24×24 neural trong lớp ẩn. Có được điều này là do chúng ta chỉ có thể di chuyển các trường tiếp nhận cục bộ ngang qua 23 neural (hoặc

xuống dưới 23 neuron), trước khi chạm với phía bên phải (hoặc dưới) của ảnh đầu vào.

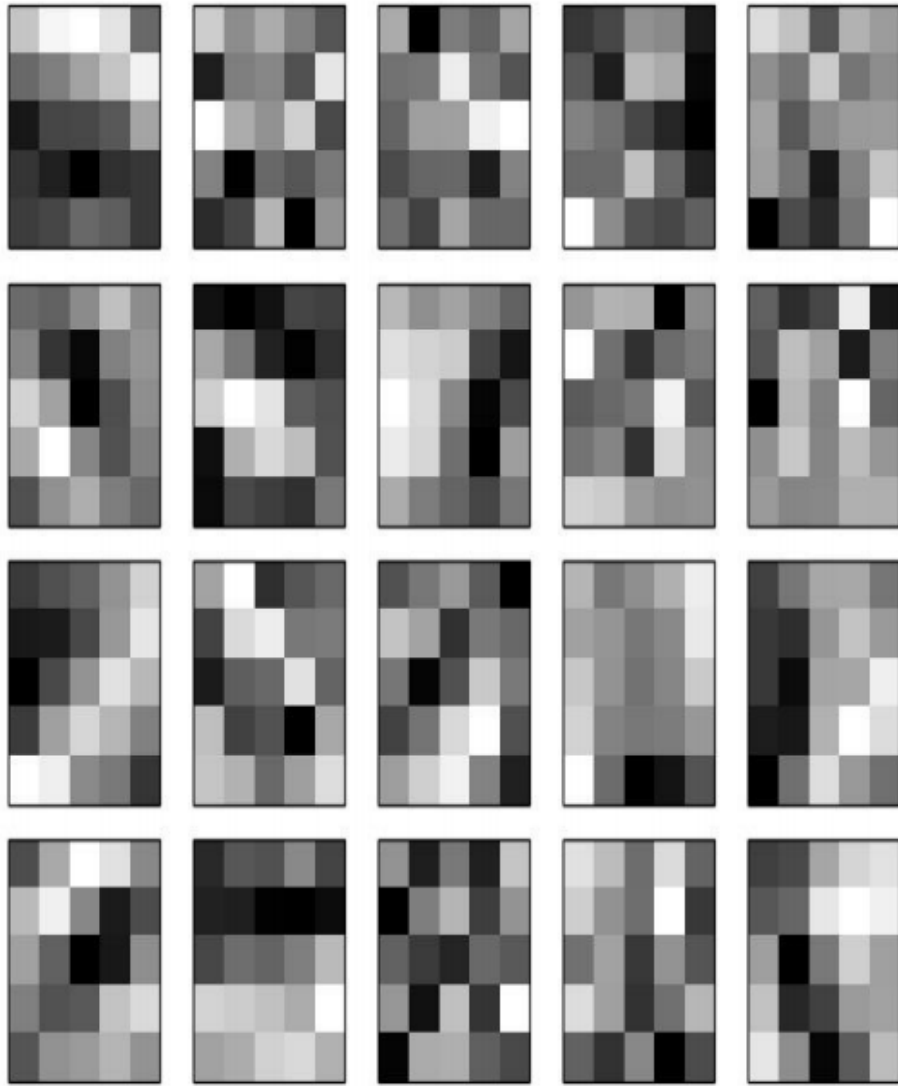
Với bài toán nhận dạng ảnh người ta thường gọi ma trận lớp ẩn đầu vào là feature map, trọng số xác định các đặc trưng là shared weight và độ lệch xác định một feature map là shared bias. Như vậy đơn giản nhất là qua các bước trên chúng ta chỉ có 1 feature map. Tuy nhiên trong nhận dạng ảnh chúng ta cần nhiều hơn một feature map.



Hình 1.12: Trường tiếp nhận cục bộ với ba bản đồ đặc trưng

Trong ví dụ ở Hình 1.12, có 3 bản đồ đặc trưng. Mỗi bản đồ đặc trưng được xác định bởi một tập 5×5 trọng số chia sẻ, và một độ lệch chia sẻ duy nhất. Kết quả là các mạng có thể phát hiện 3 loại đặc trưng khác nhau, với mỗi đặc trưng được phát hiện trên toàn bộ ảnh.

Trong thực tế mạng CNN có thể sử dụng nhiều bản đồ đặc trưng hơn. Một trong những mạng chập đầu tiên là LeNet-5, sử dụng 6 bản đồ đặc trưng, mỗi bản đồ được liên kết đến một trường tiếp nhận cục bộ 5×5 , để phát hiện các kí tự MNIST. Vì vậy, các ví dụ minh họa ở trên là thực sự khá gần LeNet-5. Trong một số nghiên cứu gần đây sử dụng lớp tích chập với 20 và 40 bản đồ đặc trưng.



Hình 1.13: Trường tiếp nhận cục bộ với 20 bản đồ đặc trưng

Trên đây (hình 1.13) là 20 ảnh tương ứng với 20 bản đồ đặc trưng khác nhau (hay còn gọi là bộ lọc, hay là nhân). Mỗi bản đồ được thể hiện là một hình khối kích thước 5×5 , tương ứng với 5×5 trọng số trong trường tiếp nhận cục bộ. Khối trắng có nghĩa là một trọng số nhỏ hơn, vì vậy các bản đồ đặc trưng đáp ứng ít hơn để tương ứng với điểm ảnh đầu vào. Khối sẫm màu hơn có nghĩa là trọng số lớn hơn, do đó, các bản đồ đặc trưng đáp ứng nhiều hơn với các điểm ảnh đầu vào tương ứng.

Có thể thấy rằng, trường tiếp nhận cục bộ 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.

1.1.4.2. Trọng số chia sẻ và độ lệch (Shared weights and biases)

Đầu tiên, các trọng số cho mỗi filter (kernel) phải giống nhau. Tất cả các neural 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. Ta cần tìm ra mối quan hệ giữa số lượng Feature map với số lượng tham số.

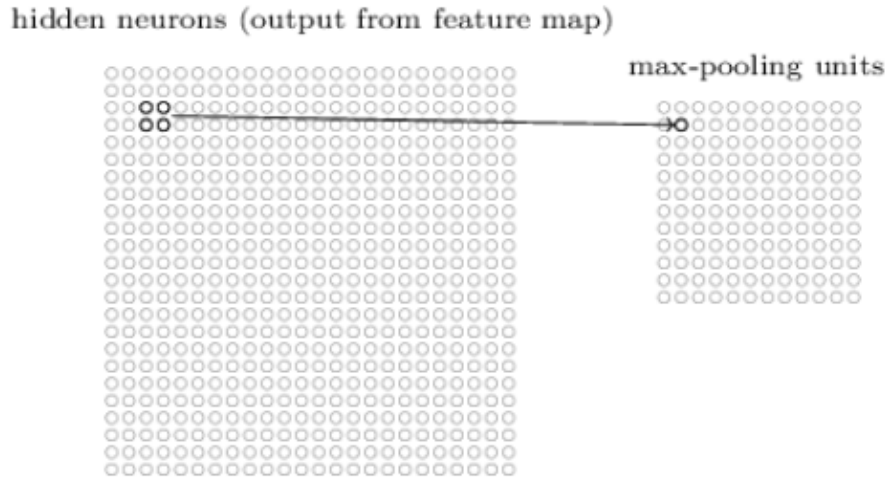
Chúng ta thấy mỗi fearture map cần $25 = 5 \times 5$ shared weight và 1 shared bias. Như vậy mỗi feature map cần $5 \times 5 + 1 = 26$ tham số. Như vậy nếu có 10 feature map thì có $10 \times 26 = 260$ tham số. Chúng ta xét lại nếu layer đầu tiên có kết nối đầy đủ nghĩa là chúng ta có $28 \times 28 = 784$ neural đầu vào như vậy ta chỉ có 30 neural ẩn. Như vậy ta cần $28 \times 28 \times 30$ shared weight và 30 shared bias. Tổng số tham số là $28 \times 28 \times 30 + 30$ tham số lớn hơn nhiều so với CNN. Ví dụ vừa rồi chỉ mô tả để thấy được sự ước lượng số lượng tham số chứ chúng ta không so sánh được trực tiếp vì 2 mô hình khác nhau. Nhưng điều chắc chắn là nếu mô hình có số lượng tham số ít hơn thì nó sẽ chạy nhanh hơn.

1.1.4.3. Lớp chứa hay lớp tổng hợp (Pooling layer)

Ngoài các lớp tích chập vừa mô tả, mạng neural tích chập cũng chứa các lớp pooling. Lớp pooling thường được sử dụng ngay sau lớp tích chập. Những gì các lớp pooling làm là đơn giản hóa các thông tin ở đầu ra từ các lớp tích chập.

Ví dụ, mỗi đơn vị trong lớp pooling có thể thu gọn một vùng 2×2 neural trong lớp trước. Một thủ tục pooling phổ biến là max-pooling.

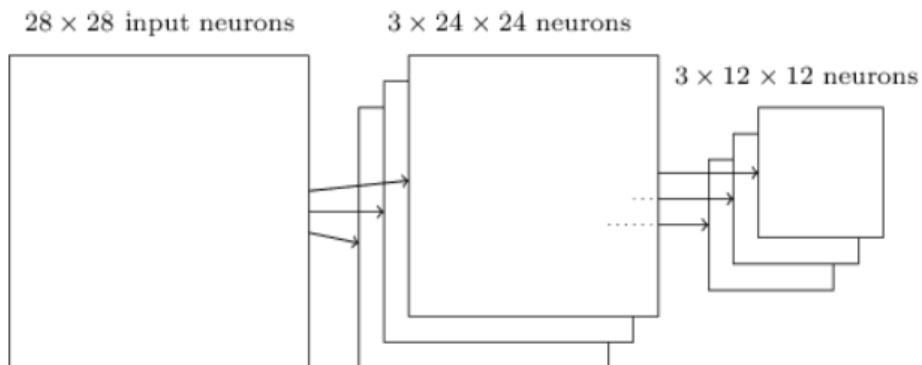
Trong maxpooling, một đơn vị pooling chỉ đơn giản là kết quả đầu ra kích hoạt giá trị lớn nhất trong vùng đầu vào 2×2 , như minh họa trong hình 1.14 sau:



Hình 1.14: Trường tiếp nhận cục bộ với 20 bản đồ đặc trưng

Lưu ý rằng bởi vì chúng ta có 24×24 neural đầu ra từ các lớp tích chập, sau khi pooling chúng ta có 12×12 neuron.

Như đã đề cập ở trên, lớp tích chập thường có nhiều hơn một bản đồ đặc trưng. Chúng ta áp dụng max-pooling cho mỗi bản đồ đặc trưng riêng biệt. Vì vậy, nếu có ba bản đồ đặc trưng, các lớp tích chập và max-pooling sẽ kết hợp như hình 1.15 sau:



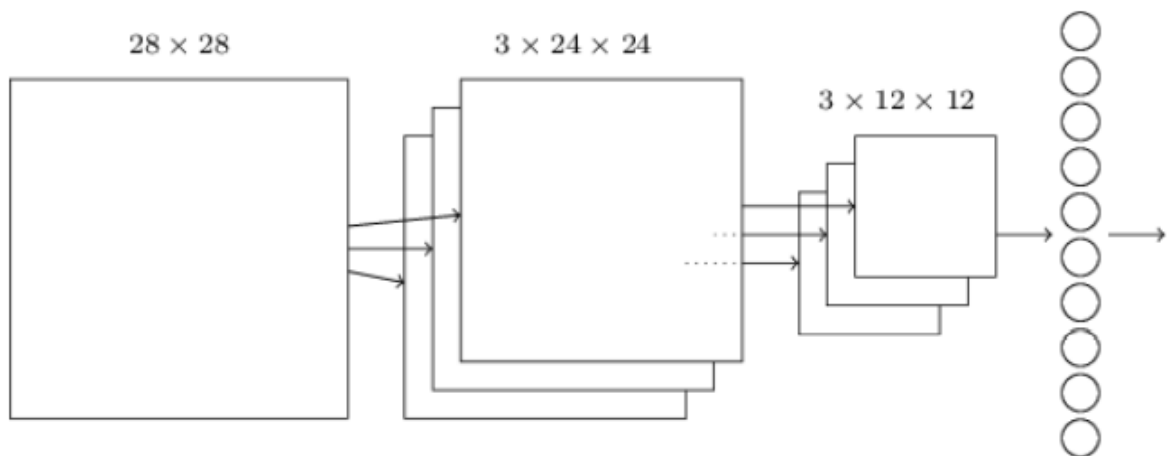
Hình 1.15: Trường tiếp nhận cục bộ với 20 bản đồ đặc trưng

Chúng ta có thể hiểu max-pooling như là một cách cho mạng để hỏi

xem một đặc trưng nhất được tìm thấy ở bất cứ đâu trong một khu vực của ảnh. Sau đó nó bỏ đi những thông tin định vị chính xác. Trục giác là một khi một đặc trưng đã được tìm thấy, vị trí chính xác của nó là không quan trọng như vị trí thô của nó so với các đặc trưng khác. Một lợi ích lớn là có rất nhiều tính năng gộp ít hơn (fewer pooled features), và vì vậy điều này sẽ giúp giảm số lượng các tham số cần thiết trong các lớp sau.

Max-pooling không phải là kỹ thuật duy nhất được sử dụng để pooling. Một phương pháp phổ biến khác được gọi là L2 pooling. Ở đây, thay vì lấy giá trị kích hoạt tối đa (maximum activation) của một vùng 2×2 neuron, chúng ta lấy căn bậc hai của tổng các bình phương của kích hoạt trong vùng 2×2 . Trong khi các chi tiết thì khác nhau, nhưng về trục giác thì tương tự như max-pooling: L2 pooling là một cách để cô đọng thông tin từ các lớp tích chập. Trong thực tế, cả hai kỹ thuật đã được sử dụng rộng rãi. Và đôi khi người ta sử dụng các loại pooling khác.

Như vậy, chúng ta có thể đặt tất cả những ý tưởng lại với nhau để tạo thành một mạng tích chập hoàn chỉnh. Nó tương tự như kiến trúc chúng ta phân tích ở trên, nhưng có thêm một lớp 10 neural đầu ra, tương ứng với 10 giá trị có thể cho các số MNIST ('0', '1', '2', v.v...) (hình 1.16):



Hình 1.16: Một kiến trúc mạng CNN cho nhận dạng chữ viết từ dữ liệu MNIST

Mạng bắt đầu với 28×28 neural đầu vào, được sử dụng để mã hóa các cường độ điểm ảnh cho ảnh MNIST. Sau đó là một lớp tích chập sử dụng 5×5 trường tiếp nhận cục bộ và 3 bản đồ đặc trưng. Kết quả là một lớp $3 \times 24 \times 24$ neural lớp ẩn. Bước tiếp theo là một lớp max-pooling, áp dụng cho 2×2 vùng qua 3 bản đồ đặc trưng (feature maps). Kết quả là một lớp $3 \times 12 \times 12$ neural đặc trưng ở tầng ẩn.

Lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối. Lớp này nối mọi neural từ lớp max-pooled tới mọi neural của tầng ra.

1.1.4.4. Cách chọn tham số cho CNN

Hiệu quả hoạt động của mạng CNN phụ thuộc rất nhiều vào việc lựa chọn các tham số sau:

- 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
- Pooling size: thường là 2×2 hoặc 4×4 cho ảnh đầu vào lớn

Trong thực tế, tùy vào ứng dụng cụ thể mà ta chọn các tham số khác nhau. Thông thường ta sẽ thực hiện nhiều lần việc train test để chọn ra được param tốt nhất (Phương pháp thử sai).

1.2. Bài toán chuẩn đoán bệnh lao

Để chuẩn đoán được bệnh lao cần dựa vào rất nhiều dữ liệu [3] như:

- **Lâm sàng**
 - **Toàn thân:** Sốt nhẹ về chiều, ra mồ hôi đêm, chán ăn, mệt mỏi, gầy sút cân.

- **Cơ năng:** Ho, khạc đờm, ho ra máu, đau ngực, khó thở.
- **Thực thể:** Nghe phổi có thể có tiếng bệnh lý (ran ẩm, ran nổ,...).

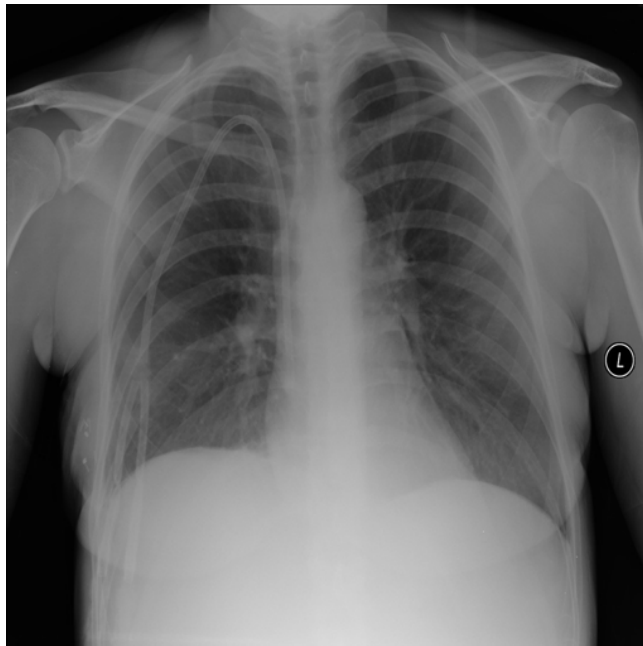
- **Cận lâm sàng**

- Nhuộm soi đờm trực tiếp tìm AFB.
- Xét nghiệm Xpert MTB/RIF (nếu có thể).
- Nuôi cấy tìm vi khuẩn lao.
- X-Quang phổi thường quy.

Tuy vậy, do mục tiêu, đối tượng nghiên cứu, phạm vi nghiên cứu, nên luận văn chỉ tập trung nghiên cứu hỗ trợ chuẩn đoán bệnh lao qua các hình ảnh x-quang phổi thường quy dựa vào học máy, học sâu.

1.2.1. Mô tả bài toán

Ta có tập dữ liệu với 3500 ảnh của người không có lao và 700 ảnh của người có lao đã được gán nhãn. Mỗi hình ảnh có kích thước 512 x 512 định dạng PNG như hình 1.17.



Hình 1.17: Ảnh chụp x-quang của người không có lao

Tập dữ liệu trên được một nhóm các nhà nghiên cứu từ Đại học Qatar, Doha, và Đại học Dhaka, Bangladesh cùng với các cộng tác viên của họ từ Malaysia phối hợp với các bác sĩ từ Hamad Medical Corporation và Bangladesh đã tạo ra một cơ sở dữ liệu về hình ảnh X-quang phổi cho người có lao cùng với hình ảnh người không có lao [4]. Dữ liệu trên được cung cấp miễn phí tại <https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset>.

Từ tập dữ liệu trên, ta tách riêng biệt 100 ảnh ngẫu nhiên từ cả 2 nhãn để làm dữ liệu đánh giá, kiểm thử độc lập. 4100 ảnh còn lại sẽ được dùng để huấn luyện các mô hình học sâu của luận văn. Sau khi hoàn thành quá trình huấn luyện, ta sẽ dùng dữ liệu đánh giá riêng biệt (100 ảnh đã tách riêng) để các mô hình phân loại, dự đoán về khả năng ảnh x-quang đó là của người có lao hay không.

CHƯƠNG 2

Một số mô hình hỗ trợ chuẩn đoán.

Từ mạng CNN cơ bản người ta có thể tạo ra rất nhiều mô hình khác nhau, từ những mạng neural cơ bản 1 đến 2 layer đến 100 layer. Khi thêm nhiều layer hơn thì theo lý thuyết độ chính xác phải cao hơn, nhưng thực tế lại không phải độ chính xác không tăng thậm chí là có lúc lại giảm. Về kernel ta có 3x3, 5x5 hay thậm chí 7x7, vậy thì dùng kernel nào tốt, càng nhỏ liệu có càng chính xác? Vì vậy, chương này ta sẽ tìm hiểu một số mô hình nổi tiếng của CNN, các cài đặt, cấu hình của chúng và áp dụng vào bài toán chuẩn đoán của luận văn.

2.1. VGG-16

2.1.1. VGG16 là gì

VGG là viết tắt của Visual Geometry Group; nó là một kiến trúc CNN sâu tiêu chuẩn với nhiều lớp. Kiến trúc VGG là cơ sở của các mô hình nhận dạng đối tượng mang tính đột phá. Được phát triển như một mạng nơ-ron sâu, VGGNet cũng vượt qua các đường cơ sở về nhiều tác vụ và bộ dữ liệu ngoài ImageNet. Hơn nữa, bây giờ nó vẫn là một trong những kiến trúc nhận dạng hình ảnh phổ biến nhất.

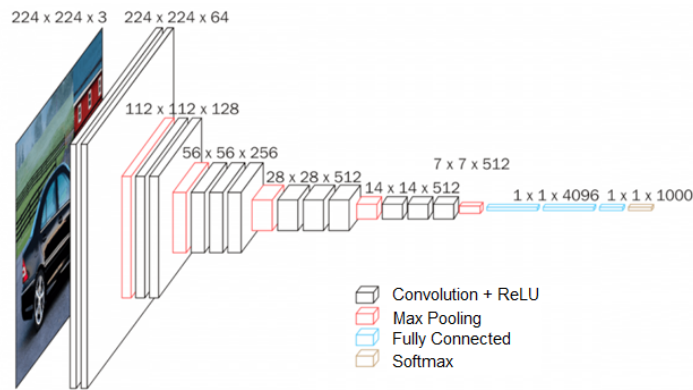
Karen Simonyan và Andrew Zisserman [6] đã đề xuất ý tưởng về mạng VGG vào năm 2013 và gửi mô hình thực tế dựa trên ý tưởng này trong ImageNet Challenge 2014. Họ gọi nó là VGG theo tên bộ phận của Visual Geometry Group tại Đại học Oxford nơi họ làm việc.

Mô hình VGG16, hoặc VGGNet, là một mạng nơ-ron phức hợp hỗ trợ 16 lớp. Mô hình VGG16 đạt được độ chính xác gần như 92,7% trong bài kiểm tra top 5 trong ImageNet (ImageNet là một tập dữ liệu bao gồm

hơn 14 triệu hình ảnh thuộc gần 1000 lớp), nó có thể phân loại hình ảnh thành 1000 loại đối tượng, bao gồm bàn phím, động vật, bút chì, chuột, v.v. Ngoài ra, mô hình có kích thước đầu vào hình ảnh là 224×224 . Nó thay thế các bộ lọc kích thước hạt nhân lớn bằng một số bộ lọc kích thước hạt nhân 3×3 lần lượt.

2.1.2. Kiến trúc VGG16

Trong tất cả các cấu hình, VGG16 được xác định là mô hình hoạt động tốt nhất trên tập dữ liệu ImageNet. Hãy xem lại kiến trúc thực tế của cấu hình này (Hình 2.1).



Hình 2.1: Kiến trúc VGG16.

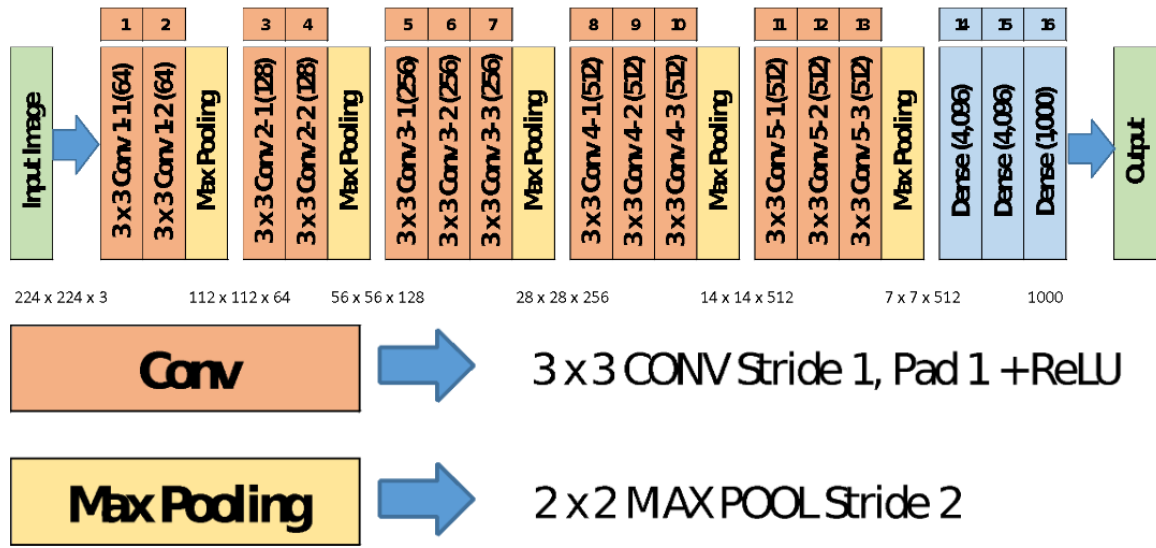
Đầu vào cho bất kỳ cấu hình mạng nào được coi là hình ảnh có kích thước cố định 224×224 với ba kênh R, G và B. Quá trình xử lý trước duy nhất được thực hiện là chuẩn hóa các giá trị RGB cho mỗi pixel. Điều này đạt được bằng cách trừ đi giá trị trung bình cho mỗi pixel.

Hình ảnh được chuyển qua ngăn xếp đầu tiên gồm 2 lớp tích chập có kích thước tiếp nhận rất nhỏ là 3×3 , tiếp theo là kích hoạt ReLU. Mỗi lớp trong số hai lớp này chứa 64 bộ lọc. Stride được cố định ở 1 pixel và padding là 1 pixel. Cấu hình này bảo toàn độ phân giải không gian và kích thước của bản đồ kích hoạt đầu ra giống với kích thước hình ảnh đầu vào.

Các bản đồ kích hoạt sau đó được chuyển qua tổng hợp tối đa không gian trên cửa sổ 2×2 pixel, với stride là 2 pixel. Điều này làm giảm một nửa kích thước của các lần kích hoạt. Do đó, kích thước của các kích hoạt ở cuối ngăn xếp đầu tiên là $112 \times 112 \times 64$.

Các kích hoạt sau đó chảy qua ngăn xếp thứ hai tương tự, nhưng với 128 bộ lọc so với 64 bộ lọc trong ngăn xếp thứ nhất. Do đó, kích thước sau ngăn xếp thứ hai trở thành $56 \times 56 \times 128$. Tiếp theo là ngăn xếp thứ ba với ba lớp chập và một lớp tổng hợp tối đa. Số lượng bộ lọc được áp dụng ở đây là 256, làm cho kích thước đầu ra của ngăn xếp là $28 \times 28 \times 256$. Tiếp theo là hai ngăn xếp gồm ba lớp chập, với mỗi ngăn chứa 512 bộ lọc. Đầu ra ở cuối cả hai ngăn xếp này sẽ là $7 \times 7 \times 512$.

Các chồng lớp chập trùng được theo sau bởi ba lớp được kết nối hoàn chỉnh với một lớp làm phẳng ở giữa. Hai lớp đầu tiên có 4.096 tế bào thần kinh mỗi lớp và lớp được kết nối đầy đủ cuối cùng đóng vai trò là lớp đầu ra và có 1.000 tế bào thần kinh tương ứng với 1.000 lớp có thể có cho tập dữ liệu ImageNet. Tiếp theo là lớp đầu ra là lớp kích hoạt Softmax được sử dụng để phân loại (Hình 2.2).



Hình 2.2: Kiến trúc VGG16.

2.1.3. Trường hợp sử dụng VGG16

VGG16 khiến các nhà khoa học và nghiên cứu dữ liệu trên toàn thế giới quan tâm mặc dù sự ra đời của nhiều mô hình tính điểm mới và tốt hơn kể từ thời điểm VGG ban đầu được đề xuất. Dưới đây là một số trường hợp sử dụng mà bạn có thể thấy VGG16 được sử dụng thực tế:

1. Nhận dạng hoặc phân loại hình ảnh - VGG16 có thể được sử dụng để chẩn đoán bệnh bằng hình ảnh y tế như X-quang hoặc MRI. Nó cũng có thể được sử dụng để nhận biết các biển báo đường phố từ một phương tiện đang di chuyển.
2. Phát hiện và bản địa hóa hình ảnh - VGG16 có thể hoạt động thực sự tốt trong các trường hợp sử dụng phát hiện hình ảnh. Trên thực tế, nó đã là người chiến thắng trong thử thách phát hiện ImageNet vào năm 2014 (nơi nó kết thúc với vị trí á quân đầu tiên cho thử thách phân loại)
3. Vectơ nhúng hình ảnh - Sau khi bật ra lớp đầu ra trên cùng, mô hình

có thể được sử dụng để đào tạo để tạo vectơ nhúng hình ảnh có thể được sử dụng cho một vấn đề như xác minh khuôn mặt bằng cách sử dụng VGG16 bên trong mạng Siamese.

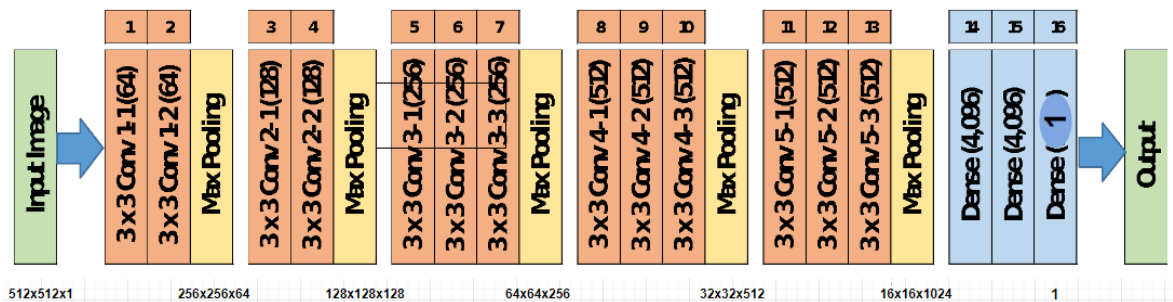
2.1.4. Áp dụng VGG16 vào bài toán của luận văn

Có hai điểm khác biệt của tập dữ liệu ImageNet và tập dữ liệu [4] được sử dụng trong bài toán của luận văn đó là:

1. Kích thước ảnh đầu vào là 512x512 pixel thay vì 224x224 pixel và ảnh đầu vào là ảnh đa mức xám thay vì ảnh RGB.
2. Chỉ có 2 lớp đầu ra: Có lao hay không có lao.

Điều này dẫn tới 1 chút thay đổi nhỏ trong cấu hình của mô hình VGG16 cho bài toán này (hình 2.3):

- Ảnh đầu vào 512x512x1
- Vì chỉ có 2 lớp đầu ra nên neural đầu ra là 1, hàm kích hoạt là sigmoid.



Hình 2.3: Kiến trúc VGG16 cho bài toán của luận văn.

2.1.5. Cài đặt mô hình

```
def make_vgg16_model(input_shape):
    inputs = keras.Input(shape=input_shape)
    x = data_augmentation(inputs)
    x = Rescaling(1.0 / 255)(x)
    #Block 1
    x = Conv2D(filters=start_step, kernel_size=(3, 3), padding='same', activation='relu',
               name='conv1_1')(x)
```

```
x = Conv2D(filters=start_step, kernel_size=(3, 3), padding='same', activation='relu',
           name='conv1_2')(x)
x = MaxPooling2D(pool_size=(2,2), strides=(2,2), name='max_pooling2d_1')(x)

#Block 2
x = Conv2D(filters=(start_step*2), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv2_1')(x)
x = Conv2D(filters=(start_step*2), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv2_2')(x)
x = MaxPooling2D(pool_size=(2,2), strides=(2,2), name='max_pooling2d_2')(x)

#Block (x)
x = Conv2D(filters=(start_step*4), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv3_1')(x)
x = Conv2D(filters=(start_step*4), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv3_2')(x)
x = Conv2D(filters=(start_step*4), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv3_3')(x)
x = MaxPooling2D(pool_size=(2,2), strides=(2,2), name='max_pooling2d_3')(x)

#Block (x)
x = Conv2D(filters=(start_step*8), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv4_1')(x)
x = Conv2D(filters=(start_step*8), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv4_2')(x)
x = Conv2D(filters=(start_step*8), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv4_3')(x)
x = MaxPooling2D(pool_size=(2,2), strides=(2,2), name='max_pooling2d_4')(x)

#Block (x)
x = Conv2D(filters=(start_step*8), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv5_1')(x)
x = Conv2D(filters=(start_step*8), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv5_2')(x)
x = Conv2D(filters=(start_step*8), kernel_size=(3, 3), padding='same', activation='relu',
           name='conv5_3')(x)
x = MaxPooling2D(pool_size=(2,2), strides=(2,2), name='max_pooling2d_5')(x)

#Flatten and FC
x = Flatten(name='flatten')(x)
x = Dense((start_step*8), activation='relu', name='fc_1')(x)
x = Dropout(0.5, name='dropout_1')(x)
x = Dense((start_step*4), activation='relu', name='fc_2')(x)
x = Dropout(0.5, name='dropout_2')(x)
outputs = Dense(1, activation='sigmoid', name='output')(x)
return keras.Model(inputs,outputs)
```

CHƯƠNG 3

Chương trình thử nghiệm.

Tài liệu tham khảo

- [1] WHO *Global tuberculosis report 2020*, báo cáo tại <https://www.who.int/publications/i/item/9789240013131>, 2020
- [2] Nguyễn Thanh Tuấn *Deep learning cơ bản (Chưa xuất bản)*, <https://drive.google.com/file/d/1lNjzISABdoc7SRq8tg-xkCRRZRABPCKi/view>, 2019
- [3] Bộ Y tế *Chuẩn đoán bệnh lao*, <https://healthvietnam.vn/thu-vien/tai-lieu-tieng-viet/ho-hap/chan-doan-benh-lao>, 2015
- [4] Tawsifur Rahman, Amith Khandakar, Muhammad A. Kadir, Khandaker R. Islam, Khandaker F. Islam, Zaid B. Mahbub, Mohamed Arselene Ayari, Muhammad E. H. Chowdhury *Reliable Tuberculosis Detection using Chest X-ray with Deep Learning, Segmentation and Visualization*, IEEE Access - Vol. 8, 2020
- [5] Grace W. Lindsay *Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future*, arXiv - Cornell University (PDF), 2001
- [6] Karen Simonyan, Andrew Zisserman *Very deep convolutional networks for large-scale image recognition*, ICLR 2015 (PDF), 2015