**ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**- - - 🙞 🕮** **🙜 - - -**

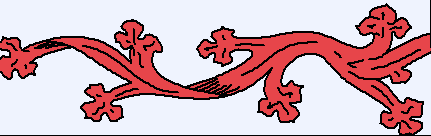
**Tiểu luận chuyên ngành**

**ỨNG DỤNG THUẬT TOÁN MÁY HỌC TRONG VIỆC XỬ LÝ ẢNH CHUẨN ĐOÁN BỆNH Y KHOA**

BOOKS

***Giảng viên hướng dẫn : TS. Vũ Quang Huy***

***Sinh viên thực hiện : Phạm Gia Bảo Đại 15110029***

SHAPE056 ***Nguyễn Nhật Hào 15110038***

***Năm học 2018–2019 220192019***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................

.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................  
.......................................................................................................................................

Giáo viên hướng dẫn

TS.Vũ Quang Huy

**MỤC LỤC**

[1. Giới thiệu về đề tài 6](#_Toc532542113)

[1.1 Lý do lựa chọn đề tài 6](#_Toc532542114)

[1.2 Ý nghĩa đề tài 7](#_Toc532542115)

[1.3 Kết quả mong đợi – Goal 8](#_Toc532542116)

[1.4 Mục tiêu 8](#_Toc532542117)

[2. Tổng quan về học máy 9](#_Toc532542118)

[2.1 Tổng quan 9](#_Toc532542119)

[2.2 Các ứng dụng của học máy 11](#_Toc532542120)

[3. Phương pháp thực hiện 12](#_Toc532542121)

[3.1 Tổng quan về CNN 12](#_Toc532542122)

[3.2 Đặc điểm của CNN 13](#_Toc532542123)

[3.3 Activation function 15](#_Toc532542124)

[3.4 Các thành phần cơ bản của mạng CNN 17](#_Toc532542125)

[3.5 Cách hoạt động của CNN 23](#_Toc532542126)

[3.6 Diagram cho bài toán 25](#_Toc532542127)

[4. Áp dụng vào bài toán nhận diện xương 26](#_Toc532542128)

[4.1 Planning 26](#_Toc532542129)

[4.2 Mô hình Convolution Neural Network cụ thể áp dụng vào bài toán 26](#_Toc532542130)

[4.3 Chuẩn bị dữ liệu 27](#_Toc532542131)

[4.4 Áp dụng CNN vào bài toán 28](#_Toc532542132)

[4.5 Xây dựng các layer 30](#_Toc532542133)

[4.6 Quá trình compile và xây dựng model 33](#_Toc532542134)

[5. Kết quả và đánh giá 34](#_Toc532542135)

[6. Kết luận 35](#_Toc532542136)

[TÀI LIỆU THAM KHẢO 35](#_Toc532542137)

**DANH MỤC HÌNH**

[Hình 1: Machine Learning 5](#_Toc532507048)

[Hình 2: Xương gãy (trái) và xương lành (phải) 6](#_Toc532507049)

[Hình 3: Hoàn cảnh y tế nước ta 7](#_Toc532507050)

[Hình 4: Phân loại Machine Learning 10](#_Toc532507051)

[Hình 5: Natural Language Processing 11](#_Toc532507052)

[Hình 6: Người máy chơi cờ 12](#_Toc532507053)

[Hình 7: Convolutional Neural Network 13](#_Toc532507054)

[Hình 8: Mạng Neural Network cơ bản 14](#_Toc532507055)

[Hình 9: Đồ thị hàm Sigmoid 16](#_Toc532507056)

[Hình 10: Đồ thị hàm ReLU 17](#_Toc532507057)

[Hình 11: Ma trận đầu vào (trái) và kernel (phải) 18](#_Toc532507058)

[Hình 12: Kernel quét qua từng phần tử của mảng input 18](#_Toc532507059)

[Hình 13: Kết quả 19](#_Toc532507060)

[Hình 14: Ví dụ thực tế, trước và sau 19](#_Toc532507061)

[Hình 15: Mô hình tổng quát của Conv layer 20](#_Toc532507062)

[Hình 16: Minh họa toán học 21](#_Toc532507063)

[Hình 17: Minh họa Max Pooling (1) 21](#_Toc532507064)

[Hình 18: Minh họa Max pooling (2) 22](#_Toc532507065)

[Hình 19: Sơ đồ Fully - Connected 23](#_Toc532507066)

[Hình 20: Sơ đồ hoạt động của CNN 24](#_Toc532507067)

[Hình 21: Sơ đồ hoạt động của CNN với ảnh thực tế 24](#_Toc532507068)

[Hình 22: Diagram 25](#_Toc532507069)

[Hình 23: Planning 26](#_Toc532507070)

[Hình 24: Mô hình CNN áp dụng vào bài toán 27](#_Toc532507071)

[Hình 25: Ảnh X-quang của xương 28](#_Toc532507072)

[Hình 26: Sơ đồ tổng quát 30](#_Toc532507073)

[Hình 27: Layer 1&2 31](#_Toc532507074)

[Hình 28: Layer 3 31](#_Toc532507075)

[Hình 29: Layer 4 31](#_Toc532507076)

[Hình 30: Layer 5 31](#_Toc532507077)

[Hình 31: Layer 6-10 32](#_Toc532507078)

[Hình 32: Layer 11 32](#_Toc532507079)

[Hình 33: Layer 12 32](#_Toc532507080)

[Hình 34: Layer 13-15 32](#_Toc532507081)

[Hình 35: Compile & Build Model 33](#_Toc532507082)

[Hình 36: Kết quả sau khi train 33](#_Toc532507083)

[Hình 37: Kết quả cuối cùng sau khi test 34](#_Toc532507084)

**DANH MỤC BẢNG**

[Bảng 1: Thống kê model 34](#_Toc532507085)

**Lời mở đầu**

Công nghệ Thông tin ngày nay không ngừng phát triển mạnh mẽ. Sự ra đời của công nghệ thông tin đã góp phần giúp cho xã hội và đời sống của con người vươn lên một tầm cao mới, rõ rệt, ảnh hưởng đặc biệt đến sự phát triển của nhân loại.

Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1 - động cơ hơi nước, 2 - năng lượng điện, 3 - công nghệ thông tin). Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, …, chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning.

Machine Learning là một tập con của AI. Theo định nghĩa của Wikipedia, Machine learning is the subfield of computer science that “gives computers the ability to learn without being explicitly programmed”. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.



Hình 1: Machine Learning

# Giới thiệu về đề tài

* Đồ án: Ứng dụng thuật toán máy học trong việc xử lý ảnh chuẩn đoán bệnh y khoa (Xương người).
* Mô tả:
* Đề tài: Nhận diện hình ảnh X-quang xương người, sau đó trả ra kết quả xương có bị gãy hay không.
* Kiến thức: áp dụng những kiến thức cơ bản của Neural Networks, kiến thức của một số thuật toán, activation function, cost functions để phục vụ cho bài toán chuẩn đoán bệnh y khoa.
* Sản phẩm: sản phẩm là chương trình được viết bằng ngôn ngữ lập trình Python. Sử dụng bộ data set MURA 1.1 để đào tạo và đánh giá.



Hình 2: Xương gãy (trái) và xương lành (phải)

1.1 Lý do lựa chọn đề tài

* Hoàn cảnh:
* **Y khoa:** Theo báo cáo của Bộ Y tế năm 2017, các bệnh viện lớn và nhỏ trên toàn quốc thường xuyên lâm vào tình trạng quá tải trong khâu chuẩn đoán bệnh. Trong khi các công việc này có sự tương đồng, đơn giản và lặp đi lặp lại hàng ngày. Việc này dẫn đến nhu cầu, các bệnh viện cần có một hệ thống công nghệ thay cho các bác sĩ trong việc phân loại và chuẩn đoán các loại bệnh với độ chính xác cao.
* **Công nghệ:** Machine Learning nói riêng và AI (Artificial Intelligence) nói chung trong tương lai đang phát triển rất nhanh chóng và hỗ trợ, phục vụ con người trong nhiều lĩnh vực như: kinh tế, công nghiệp, đời sống, giáo dục,… và đặt biệt là Y khoa.



Hình 3: Hoàn cảnh y tế nước ta

* Lý do tại sao nhóm chọn vấn đề về xương người:
* Vấn đề gãy xương rất phổ biến ở nước ta khi mà tình trạng tai nạn giao thông hằng năm đều rất cao.
* Đây là loại bệnh đơn giản nhưng lại tốn nhiều thời gian để chuẩn đoán và độ rủi ro thấp khi áp dụng công nghệ AI vào.
* Nếu hệ thống được cộng đồng chú ý và có độ chính xác cao, chúng ta có thể áp dụng nó vào những lĩnh vực khác trong Y khoa.

## Ý nghĩa đề tài

* Áp dụng Machine Learning & AI vào trong y khoa sẽ giúp rút ngắn thời gian khám và chuẩn bệnh của các bác sĩ. Điều này sẽ tiết kiệm công sức, thời gian và chi phí rất nhiều.
* Hỗ trợ các bác sĩ từ các quốc gia có nên y khoa chưa phát triển từ đó giúp giảm thiểu rủi ro tử vong của bệnh nhân vì chẩn đoán không chính xác .
* Giảm thiểu sai sót trong quá trình chuẩn đoán bệnh.

## Kết quả mong đợi – Goal

* Đáp ứng được yêu cầu đề ra của giáo viên và người hướng dẫn.
* Học hỏi thêm những kiến thức và công nghệ mới.
* Kết quả đầu ra có độ chính xác cao (>80%).
* Áp dụng vào các hệ thống lớn trong tương lai.

## Mục tiêu

# Đặt vấn đề

* Phân tích vấn đề
* Lựa chọn giải pháp và thuật toán.

1. Chuẩn bị dữ liệu

* Thu thập dữ liệu.
* Phân loại (dán nhãn) dữ liệu.
* Gray scale (xám hóa) và denoise (giảm nhiễu) cho dữ liệu.

1. Tìm hiểu thuật toán

* Tìm hiểu các định nghĩa, nguyên lý cơ bản về thuật toán đã chọn.
* Nghiên cứu sâu và chạy tay thuật toán.
* Chuyển dữ liệu đầu vào thành feature vectors.

1. Training model

* Viết code.
* Dựa vào các feature vectors và thuật toán đã có để tạo thành một model.

# Testing

* So sánh dữ liệu test với model đã training.
* Cập nhật lại model nếu kết quả không như mong muốn nhằm tăng độ chính xác của chương trình.

# Tổng quan về học máy

2.1 Tổng quan

Học máy (Machine Learning) là một ngành khoa học nghiên cứu các thuật toán cho phép máy tính có thể học được các khái niệm (concept).

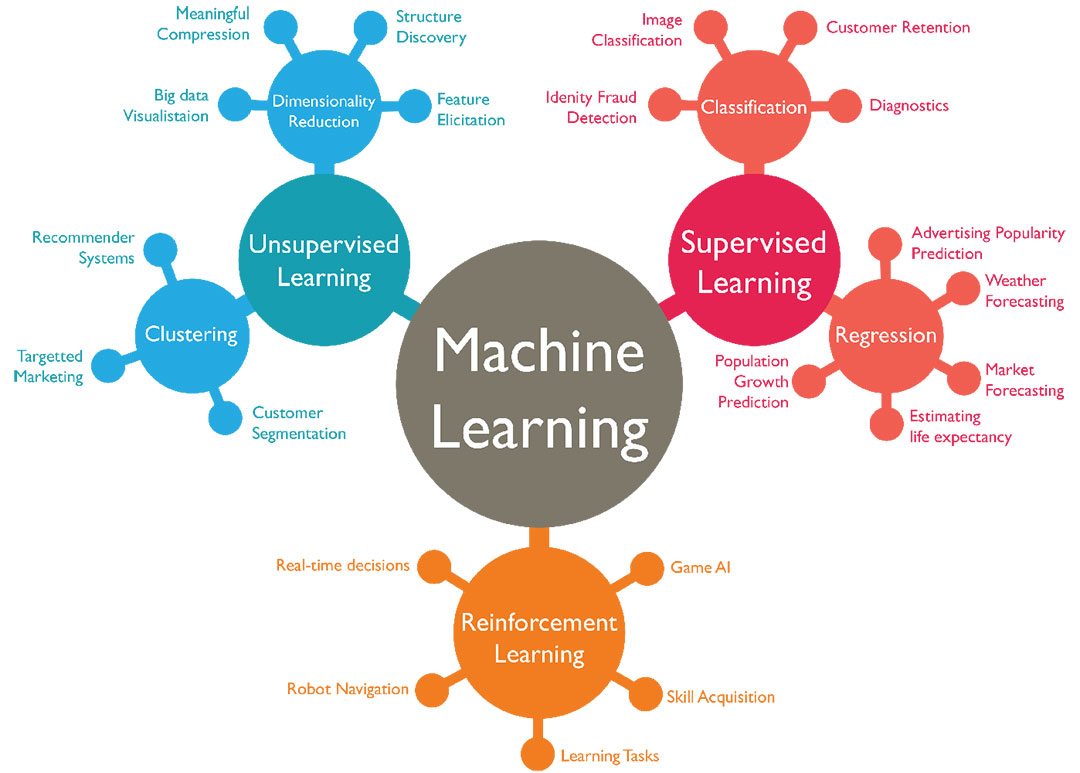
Phân loại: Có hai loại phương pháp học máy chính

* **Phương pháp quy nạp:** Máy học/phân biệt các khái niệm dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.
* **Phương pháp suy diễn:** Máy học/phân biệt các khái niệm dựa vào các luật. Phương pháp này cho phép tận dụng được các kiến thức chuyên ngành để hỗ trợ máy tính.

Hiện nay, các thuật toán đều cố gắng tận dụng được ưu điểm của hai phương pháp này.

Các ngành khoa học liên quan:

* Lý thuyết thống kê: các kết quả trong xác suất thống kê là tiền đề cho rất nhiều phương pháp học máy. Đặc biệt, lý thuyết thống kê cho phép ước lượng sai số của các phương pháp học máy.
* Các phương pháp tính: các thuật toán học máy thường sử dụng các tính toán số thực/số nguyên trên dữ liệu rất lớn. Trong đó, các bài toán như: tối ưu có/không ràng buộc, giải phương trình tuyến tính, v.v… được sử dụng rất phổ biến.
* Khoa học máy tính: là cơ sở để thiết kế các thuật toán, đồng thời đánh giá thời gian chạy, bộ nhớ của các thuật toán học máy.



Hình 4: Phân loại Machine Learning

Các nhóm giải thuật học máy:

* **Học có giám sát (Supervised Learning):** Máy tính được xem một số mẫu gồm đầu vào (input) và đầu ra (output) tương ứng trước. Sau khi học xong các mẫu này, máy tính quan sát một đầu vào mới và cho ra kết quả.
* **Học không giám sát (Unsupervised Learning):** Máy tính chỉ được xem các mẫu không có đầu ra, sau đó máy tính phải tự tìm cách phân loại các mẫu này và các mẫu mới.
* **Học nửa giám sát (Semi-Supervised Learning):** Một dạng lai giữa hai nhóm giải thuật trên.
* **Học tăng cường (Reinforcement Learning):** Máy tính đưa ra quyết định hành động (action) và nhận kết quả phản hồi (response/reward) từ môi trường (environment). Sau đó máy tính tìm cách chỉnh sửa cách ra quyết định hành động của mình.

## Các ứng dụng của học máy

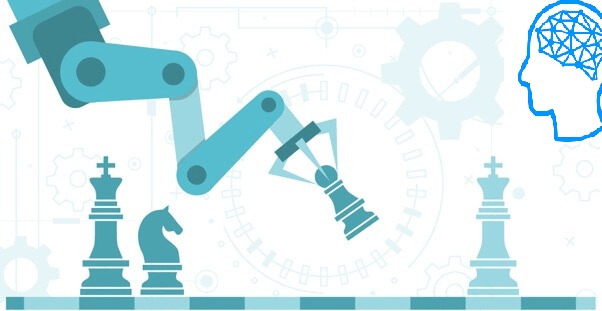
Học máy có ứng dụng rộng khắp trong các ngành khoa học/sản xuất, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ. Một số ứng dụng thường thấy:

* Xử lý ngôn ngữ tự nhiên (Natural Language Processing): xử lý văn bản, giao tiếp người – máy,…



Hình 5: Natural Language Processing

* Nhận dạng (Pattern Recognition): nhận dạng tiếng nói, chữ viết tay, vân tay, thị giác máy (Computer Vision).
* Tìm kiếm (Search Engine).
* Chẩn đoán trong y tế: phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động.
* Tin sinh học: phân loại chuỗi gene, quá trình hình thành gene/protein.
* Vật lý: phân tích ảnh thiên văn, tác động giữa các hạt.
* Phát hiện gian lận tài chính (financial fraud): gian lận thẻ tỉn dụng.
* Phân tích thị trường chứng khoán (stock market analysis).
* Chơi trò chơi: tự động chơi cờ, hành động của các nhân vật ảo.



Hình 6: Người máy chơi cờ

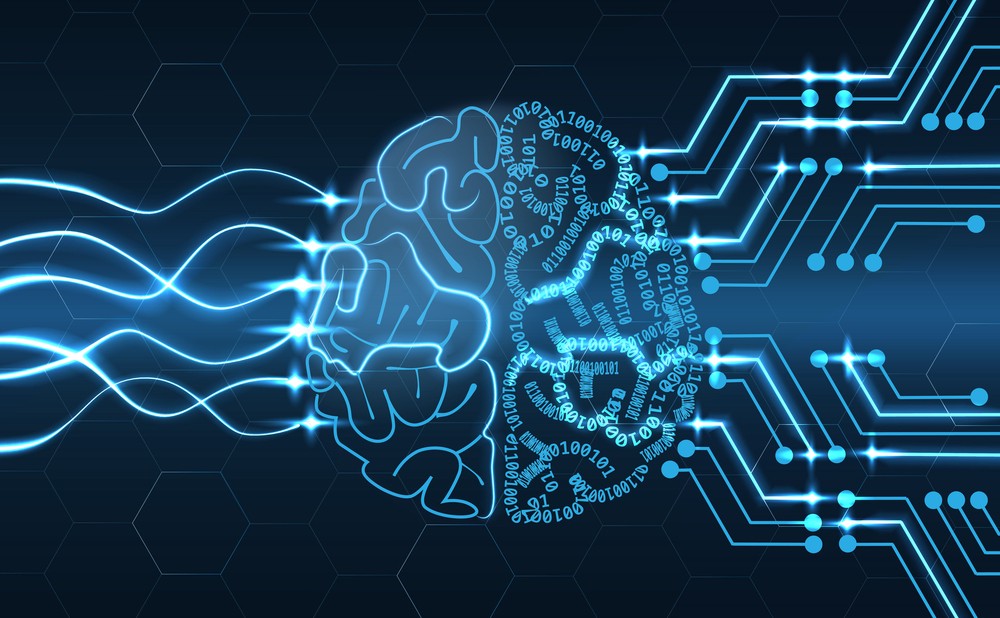
Rôbốt: là tổng hợp của rất nhiều ngành khoa học, trong đó học máy tạo nên hệ thần kinh/bộ não của người máy.

# Phương pháp thực hiện

Trong bài toán này, nhóm chúng em sẽ sử dụng thuật toán CNN (Convolutional Neural Network) để giải quyết vấn đề. 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.

## Tổng quan về CNN

**CNN (Convolutional Neural Network)** là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay như hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. Ví dụ như tự động nhận diện khi chúng ta đăng một ảnh lên Facebook, hay khi search một từ bất kì lên google search, ví dụ "Mèo" thì trong tab "Hình ảnh", google sẽ hiển thị rất nhiều ảnh có mèo trong đó. CNN chính là một dạnh Artificial Neural Network, một Multiplayer Perceptron nhưng mang thêm 1 vài cải tiến, đó là **Convolution** và **Pooling**.

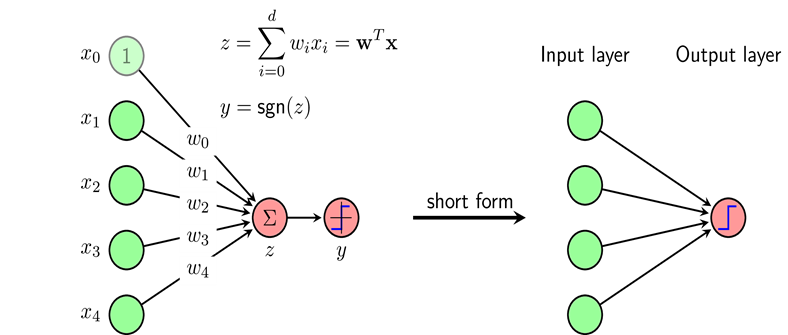


Hình 7: Convolutional Neural Network

3.2 Đặc điểm của CNN

* 1. Neural Network

Mạng nơron nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron.



Hình 8: Mạng Neural Network cơ bản

Các thành phần cơ bản của một nơron nhân tạo bao gồm:

* Tập các đầu vào: Là các tín hiệu vào (input signals) của nơron, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.
* Tập các liên kết: Mỗi liên kết đƣợc thể hiện bởi một trọng số liên kết – Synaptic weight. Trọng số liên kết giữa tín hiệu vào thứ j với nơron k thường được kí hiệu là w\_kj. Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.
* Bộ tổng (Summing function): Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.
* Ngưỡng (còn gọi là một độ lệch - bias): Ngưỡng này thường được đưa vào như một thành phần của hàm truyền.
* Hàm truyền (Transfer function): Hàm này được dùng để giới hạn phạm vi đầu ra của mỗi nơron. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng.
* Đầu ra: Là tín hiệu đầu ra của một nơron, với mỗi nơron sẽ có tối đa là một đầu ra.
  1. Ưu và nhược điểm của CNN

Trong thực tế, mỗi bức ảnh đầu vào của một bài toán có kích thước là D \* R \* C (D là chiều dài, R là chiều rộng, C là số màu) thì số lượng tham số cần phải xử lý là rất lớn. Giả sử một bức hình đơn giản sẽ có 28x28x3=2352 tham số, đây là một con số rất lớn nhưng trong thực tế thì kích thước của một bức ảnh còn lớn hơn rất nhiều (Đối với một bức ảnh X-ray thông thường thì chiều dài và chiều rộng ít nhất là 425 và 512). Và CNN sẽ giải quyết được vấn đề này, cái mà ANN không thể làm được. Vì sao? Vì CNN sẽ giúp số tham số đầu vào của ANN trở nên ít lại nhưng vẫn giữ nguyên các thuộc tính cần thiết cho bức hình.

Nhưng CNN gặp phải các khó khăn sau:

* Cần lượng dữ liệu rất lớn để có thể xây dựng một model chuẩn.
* Tiêu tốn nhiều tài nguyên để thực hiện thuật toán.
* Các bức ảnh đầu vào phải có cùng một kích thước (width, height, dimension,…).
* Khi các bức hình đầu vào có kích thước lớn hơn thì kéo theo số lượng layer tăng rất nhiều.

## Activation function

Activation function giới hạn phạm vi giá trị tới một giá trị hữu hạn cho phép. Một số hàm activation thường dùng : Sigmoid, Rectified linear unit (ReLU), Softmax.

1. Sigmoid function

Sigmoid function là một logistic function có phạm vi (0, 1). Do đó, nó đặc biệt được sử dụng cho các mô hình mà phải dự đoán xác suất. Vì vậy xác suất của bất cứ điều gì chỉ tồn tại giữa phạm vi 0 và 1.

Áp dụng tốt nhất có bài toán classification với chỉ hai class.

(1)

(2)



Hình 9: Đồ thị hàm Sigmoid

1. Softmax

Hàm softmax có output nằm trong khoảng từ 0 đến 1, giống như hàm Sigmoid. Nhưng nó chia mỗi output sao cho tổng số output bằng 1.

(3)

1. Rectified linear unit (ReLU)

ReLU (Rectified Linear Unit) được sử dụng rộng rãi gần đây vì tính đơn giản của nó. Nó có công thức toán học - rất đơn giản. Ưu điểm chính của nó là:

* ReLU được chứng minh giúp cho việc training các Deep Networks nhanh hơn rất nhiều. Sự tăng tốc này được cho là vì ReLU được tính toán gần như tức thời và gradient của nó cũng được tính cực nhanh với gradient bằng 1 nếu đầu vào lớn hơn 0, bằng 0 nếu đầu vào nhỏ hơn 0.
* Mặc dù hàm ReLU không có đạo hàm tại , trong thực nghiệm, người ta vẫn thường định nghĩa ReLU và khẳng định thêm rằng, xác suất để input của một unit bằng 0 là rất nhỏ.



Hình 10: Đồ thị hàm ReLU

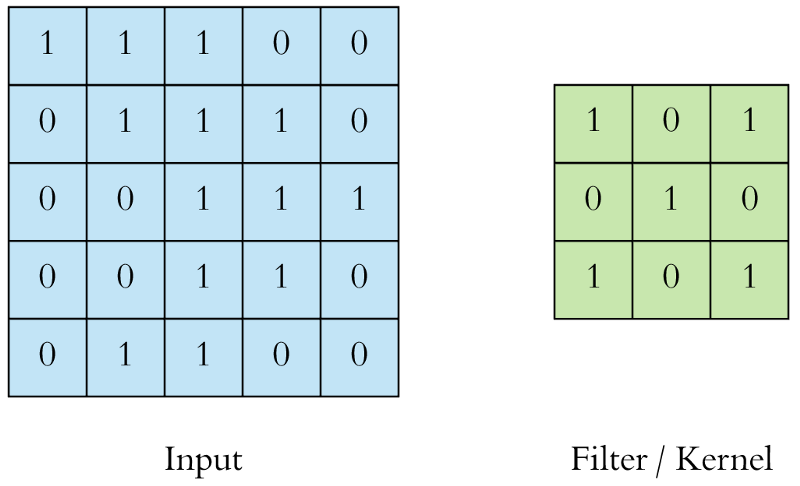
Với bất kỳ input có giá trị âm đưa vào cho hàm ReLU sẽ biến giá trị thành 0, do đó ảnh hưởng đến biểu đồ kết quả khi không ánh xạ các giá trị âm một cách thích hợp.

## Các thành phần cơ bản của mạng CNN

1. Convolution

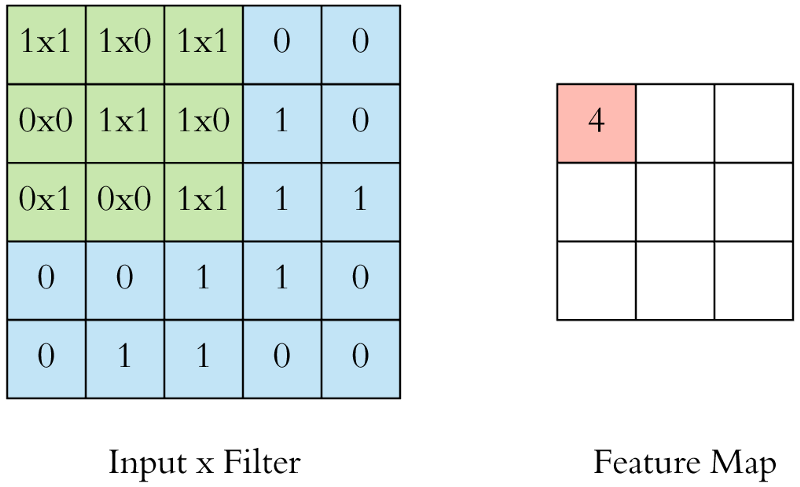
Convolution gồm 2 khái niệm khác là **Convolution Filter** và **Convolutional Layer.** Trong mạng Neural Network thông thường, từ input, ta cho qua các Hidden Layer rồi ra được output. Với CNN, Convolutional Layer cũng chính là Hidden Layer, khác ở chỗ, Convolutional Layer là một tập các feature map và mỗi feature map này là một bản scan của input ban đầu, nhưng được trích xuất ra các feature/đặc tính cụ thể. Scan như thế nào thì lại dựa vào Convolution Filter hay kernel. Đây là một ma trận sẽ quét qua ma trận dữ liệu đầu vào, từ trái qua phải, trên xuống dưới, và nhân tương ứng từng giá trị của ma trận đầu vào mà ma trận kernel rồi cộng tổng lại, đưa qua activation funciton (sigmoid, relu, elu, ... ), kết quả sẽ là một con số cụ thể, tập hợp các con số này lại là 1 ma trận nữa, chính là feature map.

Ví dụ: có một ma trận đầu vào input và một kernel.



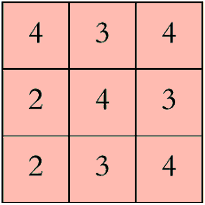
Hình 11: Ma trận đầu vào (trái) và kernel (phải)

Quét kernel qua từng phần tử của input. Và tính toán như trên: nhân tương ứng, rồi cộng tổng kết quả, đưa qua activation function, ta thu được một giá trị tại feature map:



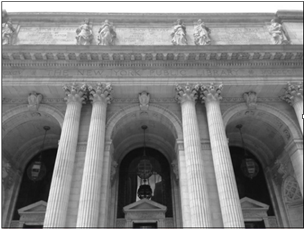
Hình 12: Kernel quét qua từng phần tử của mảng input

Thực hiện lần lượt cho đến hết, ta thu được Feature Map cuối cùng:



Hình 13: Kết quả

Chúng ta không cần phải quan tâm tới việc phải tìm bao nhiêu kernel hay lưu các kernel về để dùng dần. Đó là việc của CNN, nó sẽ tự động tìm các kernel, tự dò ra các feature.



Hình 14: Ví dụ thực tế, trước và sau

* **Chi tiết về Convolution Layer:** Là một layer đầu tiên để tách features từ một bức hình đầu vào. Tích chập bảo toàn mối quan hệ giữa các điểm ảnh bằng cách học features của bức hình sử dụng những hình vuông nhỏ của dữ liệu đầu vào. Nó là một thuật toán mà xem 2 dữ liệu đầu vào là ma trận hình ảnh và một kernel.

Một số khái niệm cơ bản: **Filter**, **Kernel** hay **Feature** **Detector** đều là cách gọi của ma trận lọc (như đã đề cập ở trên). Thông thường, ở các lớp đầu tiên của Conv Layer sẽ có kích thước là [5x5x3]:

- Convolved Feature, Activation Map hay Feature Map là đầu ra của ảnh khi cho bộ lọc chạy hết bức ảnh với phép tích vô hướng.

- Receptive field là vùng ảnh được chọn để tính tích chập, hay bằng đúng cái kích thước của bộ lọc.

- Depth là số lượng bộ lọc.

- Stride được hiểu là khoảng cách dịch chuyển của bộ lọc sau mỗi lần tính. Ví dụ khi stride=2. Tức sau khi tính xong tại 1 vùng ảnh, nó sẽ dịch sang phải 2 pixel. Tương tự với việc dịch xuống dưới.

- Zero-Padding là việc thêm các giá trị 0 ở xung quanh biên ảnh, để đảm bảo phép tích chập được thực hiện đủ trên toàn ảnh.

Cách tính kích thước đầu ra của ảnh mỗi layer:

Giả sử ảnh đầu ra là [ x x ]

Thì:

= (5)

= (6)

=

Trong đó:

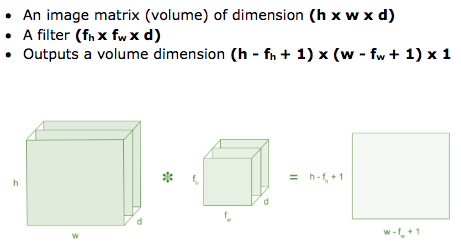
[ x x ]: Kích thước đầu vào

: Kích thước bộ lọc Kernel

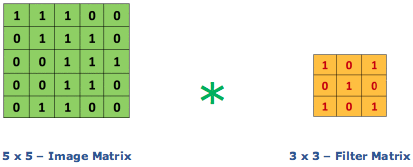
: giá trị Stride

: số lượng zero-padding thêm vào viền ảnh

: Số lượng bộ lọc (Depth)



Hình 15: Mô hình tổng quát của Conv layer

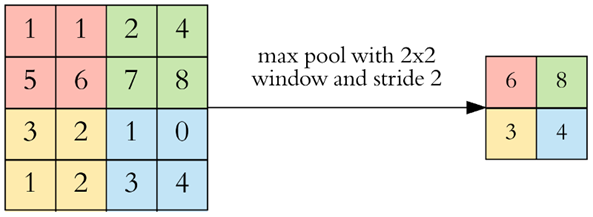


Hình 16: Minh họa toán học

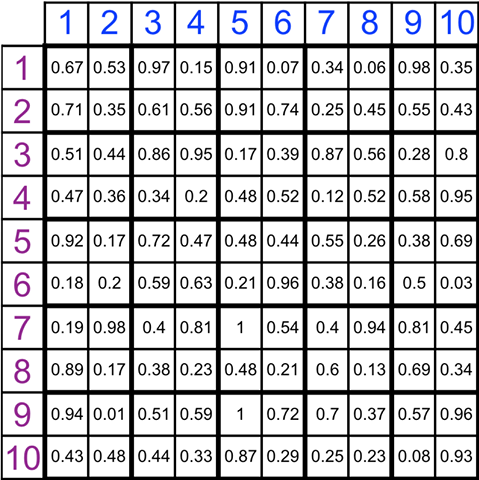
1. Max pooling

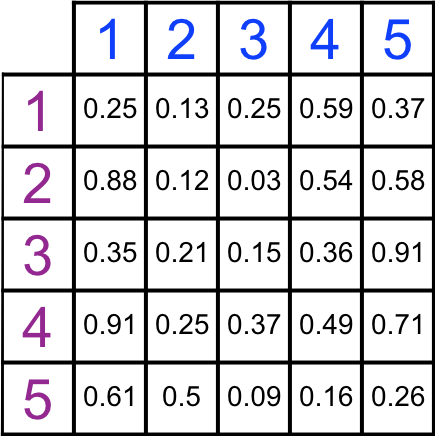
Pooling được sử dụng trong CNN để giảm số hyperparameter cần phải tính toán, từ đó giảm thời gian tính toán, tránh overfitting. Có 2 loại Pooling: Max pooling (dùng nhiều nhất) và Average pooling.

Pooling layer cung cấp tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling).

****Trong bài toán này, chúng ta sẽ sử dụng Max pooling: giữ lại chi tiết quan trọng hay hiểu ở trong bài toán này chính giữ lại pixel có giá trị lớn nhất. Nếu một đặc trưng được phát hiện ở một vùng nào đó bị bao phủ bởi bộ lọc, giá trị cao nhất trong vùng sẽ được giữ lại.

Hình 17: Minh họa Max Pooling (1)

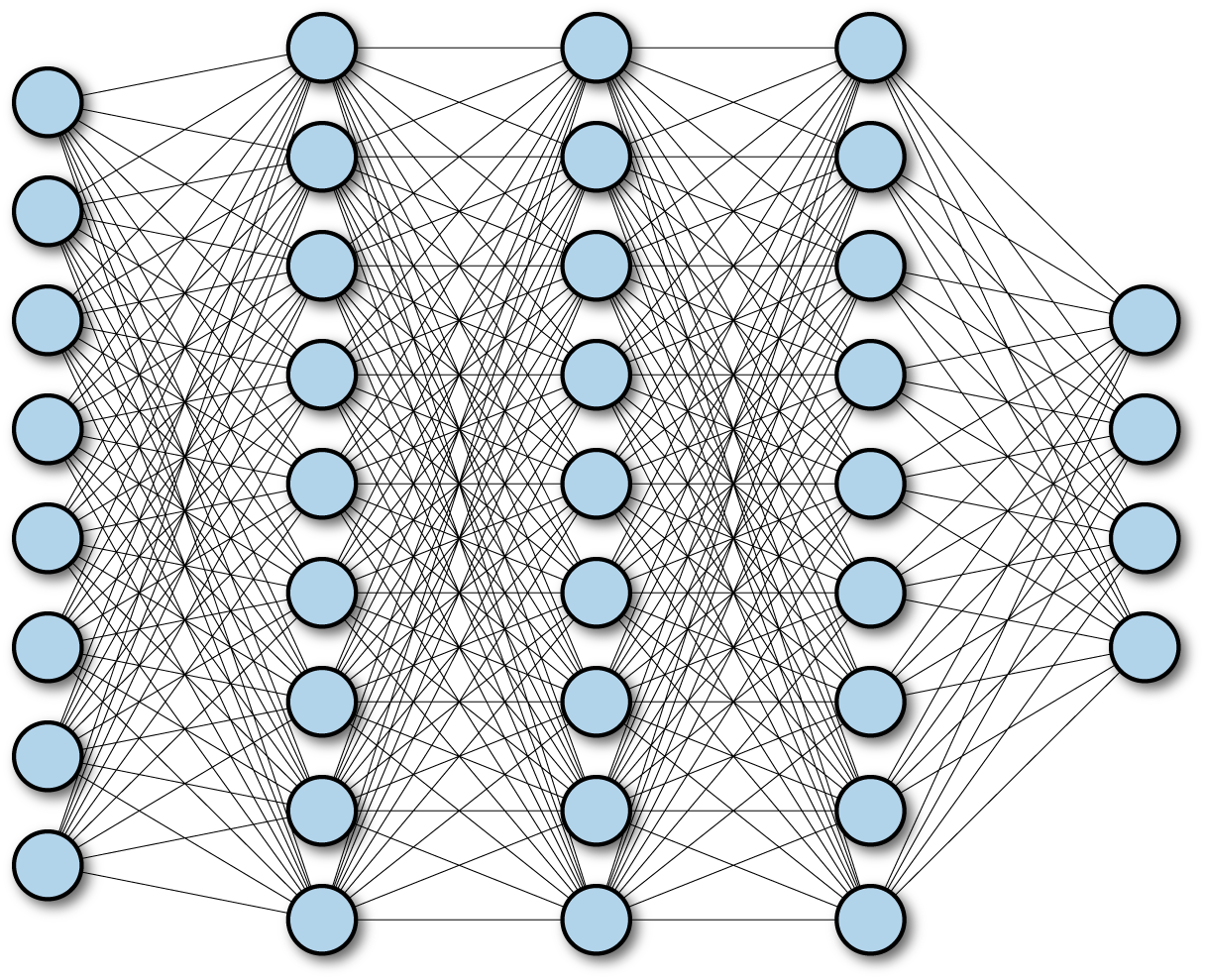




Hình 18: Minh họa Max pooling (2)

1. Fully – connected

Mục đích: kết nối mọi neuron trong một lớp với mọi neuron trong một lớp khác. Về nguyên tắc, nó giống như multi-layer perceptron neural network truyền thống. Thực hiên sau convolutional layers + pooling layers. Vì ouput của 2 layer này có kích thước 3D, còn output của fully connected layer là mảng 1D nên phải Flatten output của pooling layer cuối cùng. Flatten chỉ đơn giản là sắp xếp khối 3D của các số thành một vector 1D ( n hàng, 1 cột).



Hình 19: Sơ đồ Fully - Connected

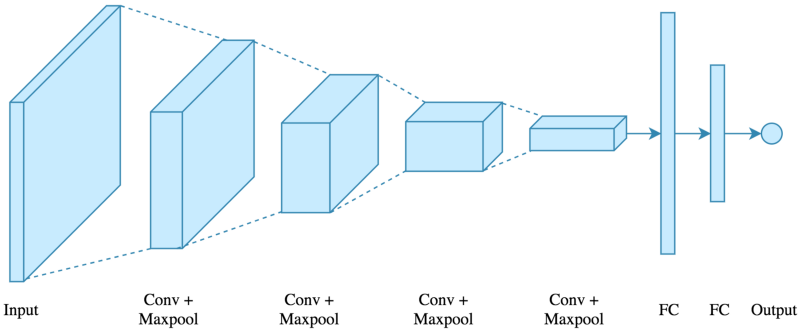
## Cách hoạt động của CNN

CNN có kiến trúc được hình thành từ các thành phần cơ bản bao gồm Convolution (CONV), Pooling (POOL), ReLU, Fully-connected (FC) về mặt xây dựng kiến trúc tổng quát CNN được mô tả như sau (dấu mũi tên thể hiện thứ tự sắp xếp các tầng từ trước đến sau):

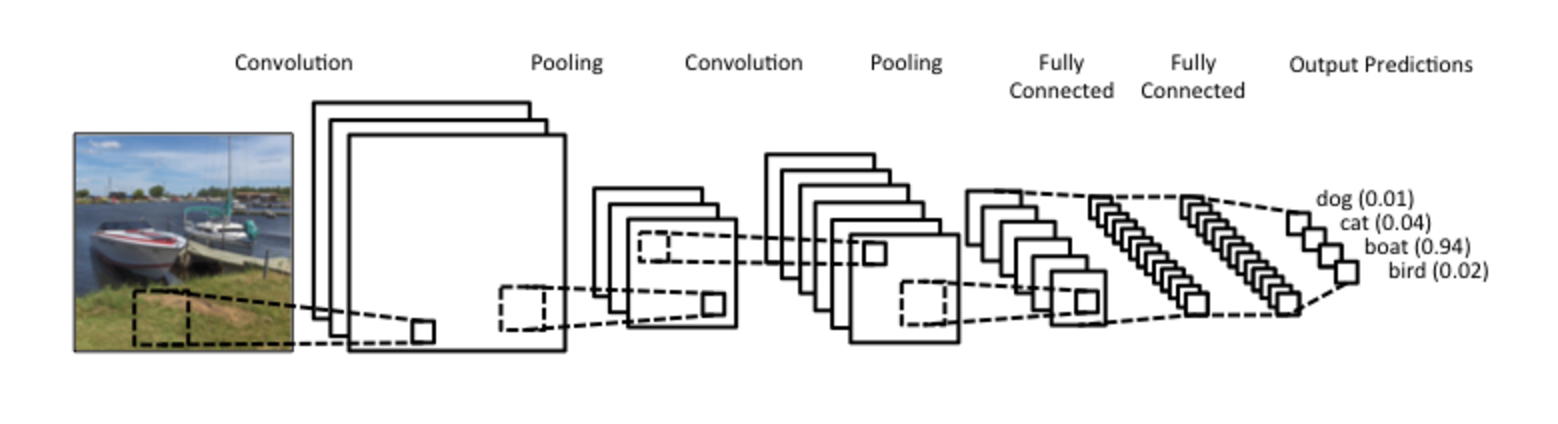
[[CONV 🡪 RELU]\*N 🡪 POOL?]\*M 🡪 [FC 🡪 RELU]\*K 🡪 FC

Trong đó:

* **[CONV -> RELU]\*N** tức là trong kiến trúc này sau tầng CONV là tầng RELU, trong CNN kiến trúc 2 tầng này có thể lặp N lần.
* **POOL?** là tầng Pooling cho người thiết kế quyết định có thể có hoặc không.
* **[[CONV -> RELU]\*N -> POOL?]\*M** trong kiến trúc CNN có thể lặp lại M lần kiểu sau tầng CONV là tầng RELU và kế tới là tầng Pooling.
* **[FC -> RELU]\*K** trong CNN có thể lặp K lần cấu trúc kiểu sau tầng FC là tầng RELU nhưng trước nó phải có tầng [CONV -> RELU].



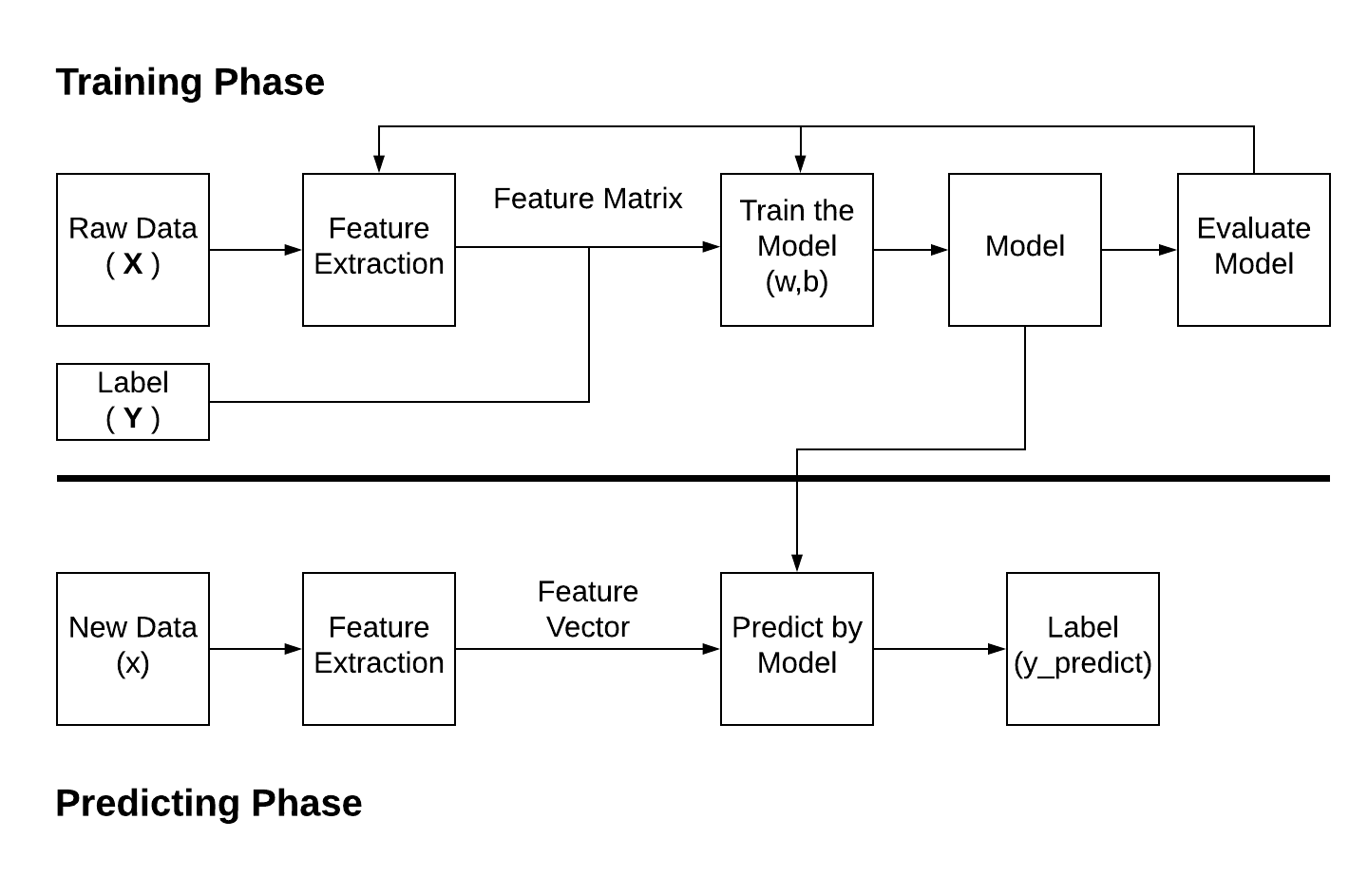
Hình 20: Sơ đồ hoạt động của CNN



Hình 21: Sơ đồ hoạt động của CNN với ảnh thực tế

Trong suốt quá trình huấn luyện, CNNs sẽ tự động học được các thông số.

## Diagram cho bài toán

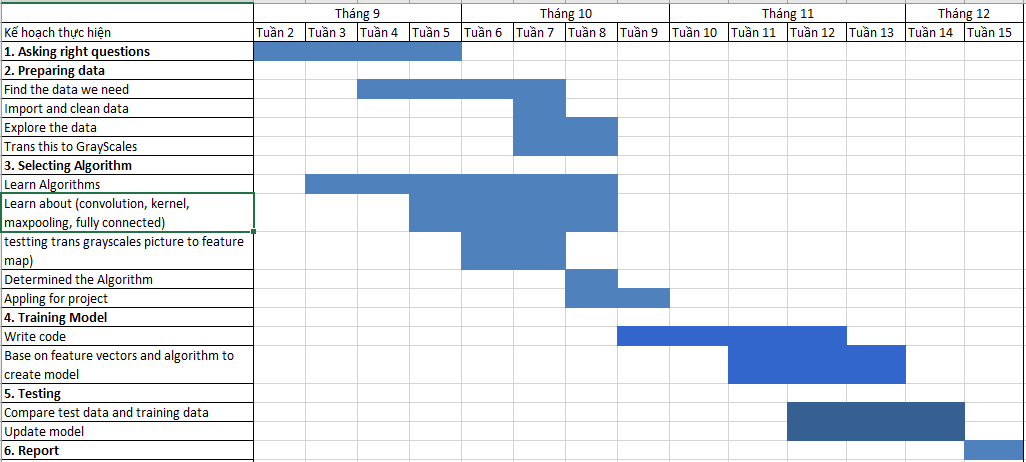


Hình 22: Diagram

* Training phase:
* Feature Extraction: mục đích là tạo ra một Feature Extractor biến dữ liệu thô ban đầu thành dữ liệu phù hợp với từng mục đích khác nhau.
* Raw data: là tất cả các thông tin ta biết về dữ liệu. Ví dụ: với ảnh thì là giá trị của từng pixel; với văn bản thì là từng từ, từng câu; với file âm thanh thì nó là một đoạn tín hiệu,… Dữ liệu thô này thường không ở dạng vector, không có số chiều như nhau. Thậm chí có thể có số chiều như nhau nhưng số chiều quá lớn, như một bức ảnh màu 1000 pixel x 1000 pixel thì số elements đã là 3×106 (3 vì ảnh màu thường có 3 channels: Red, Green, Blue). Đây là một con số quá lớn, không lợi cho lưu trữ và tính toán.
* Sau quá trình Train model thì chúng ta sẽ thu được một model với dữ liệu (X) được gắn nhãn (Y) tương ứng. Tiếp theo là quá trình đánh giá model, nếu vẫn chưa tốt thì quay lại bước Feature Extraction và lại tiếp tục tiến trình.
* Predicting phase:
* New data: là dữ liệu mới hoàn toàn được đưa vào để thực hiện quá trình dự đoán. Dữ liệu này cũng sẽ trải qua quá trình Feature Extraction ở trên để phân tích.
* Sau khi hoàn tất quá trình Feature Extraction, quá trình dự đoán sẽ được thực hiện với model được tạo ra ở Training phase. Cuối cùng dữ liệu đầu ra sẽ được gán nhãn tương ứng.

# Áp dụng vào bài toán nhận diện xương

## 4.1 Planning



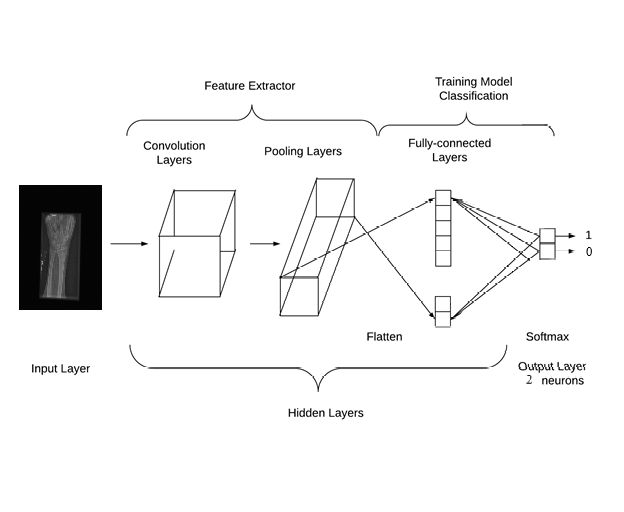
Hình 23: Planning

## 4.2 Mô hình Convolution Neural Network cụ thể áp dụng vào bài toán

Quá trình xử lý sẽ trải qua 2 giai đoạn chính: Feature Extractor và Training Model Classification.

* Feature Extractor: bao gồm Convolution Layers và Pooling Layers.
* Training Model Classification: chính là quá trình Fully – connected Layers.

Số neuron của Output Layer bằng số class phân loại. Vì kết quả nhắm đến của bài toán này là xác định xem xương có bị vấn đề hay không, do đó số neuron của Output Layer là 2.



Hình 24: Mô hình CNN áp dụng vào bài toán

## 4.3 Chuẩn bị dữ liệu

Về cơn bản, chúng ta sẽ chia tập dữ liệu thành 3 phần:

* Train: đây là tập dữ liệu dùng để training. Trong này sẽ chia được chia tiếp thành 2 tập là xương có vấn đề và không có vấn đề.
* Valid: đây là tập dữ liệu dùng để xác thực. Trong này sẽ chia được chia tiếp thành 2 tập là xương có vấn đề và không có vấn đề.
* Test: đây là tập dữ liệu dùng để test (bước cuối). Trong này sẽ chia được chia tiếp thành 2 tập là xương có vấn đề và không có vấn đề.

Dataset được nhóm lấy từ website này:

<https://stanfordmlgroup.github.io/competitions/mura/>

Dataset này gồm tổng cộng 10,400 hình X-quang về xương người ở các vị trí như cẳng tay, bàn tay, cẳng chân, bàn chân,… Ví dụ về một hình X-quang trong dataset:



Hình 25: Ảnh X-quang của xương

## 4.4 Áp dụng CNN vào bài toán

**Quá trình CNN sẽ trải qua 4 bước:**

* Bước 1: Sử dụng Convolution để tách các đặc trưng của ảnh, giảm các tham số trước khi train.
* Bước 2: Sử dụng Maxpooling để giảm các tham số thừa đồng thời sẽ giảm số lượng phép tính khi train.
* Bước 3: Flatten hình ảnh với mục đích dễ dàng tính toán.
* Bước 4: Fully Connected.
* Bước 5: So sánh label được tính toán và label được gắn sẵn để thay đổi model cho phù hợp.

Ở tầng Convolution và Maxpooling, hình ảnh sẽ trải qua các bước sau:

* Bước 1: Hình ảnh X-ray (input) sẽ được xám hóa (grayscale) để đồng nhất kênh màu (dimension).
* Bước 2: Hình ảnh sẽ được tích chập (convolution) hai lần với mỗi lần là 32 filter, ứng với mỗi filter có độ lớn là 3x3, output sẽ được relu nhằm giảm bớt các feature âm.
* Bước 3: Khối ma trận sẽ được BatchNormalization sẽ giúp làm giảm sự thay đổi của các lớp ẩn thông qua việc chuẩn hóa.
* Bước 4: Các hình ảnh sẽ được Maxpooling để giảm một nửa số tham số xử lý với tỷ lệ là 2x2. Sau đó dữ liệu sẽ được Dropout với tỷ lệ 0.25 giả định các phần tử nào không quan trọng để có thể bỏ qua nhằm tránh overfitting.
* Bước 5: Các hình ảnh sẽ được lặp lại bước 2, 3, 4 nhưng với số filter là 64.

🡺 Output ở 2 tầng này sẽ là một tập các ma trận 2 chiều nhỏ.

Ở tầng Flatten: Hình ảnh 2 chiều sẽ được đổ thành chiều dọc nhằm dễ tính toán.

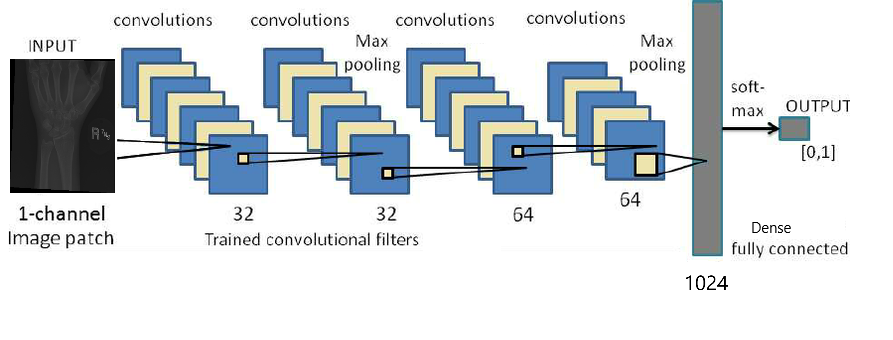
🡺Output của tầng này là các mãng 1 chiều.

Ở tầng Fully Connected:

* Bước 1: Dữ liệu lúc này trở thành các mảnh nhỏ và đưa vào với Dense layer có input là 1024 sau đó sẽ được đi qua 1 activation function là reLu.
* Bước 2: Dữ liệu sẽ được BatchNormalization và Dropout với tỷ lệ là 0.4.
* Bước 3: Dữ liệu sẽ được đi qua softmax với số output quy định là 2 nhằm xác định hình ảnh này là bị gãy xương hay không bị gãy xương.

🡺Output vừa tính sẽ được so sánh với label đã được gắn ban đầu nhằm thay đổi model cho phù hợp.

Dưới đây là sơ đồ tổng quát các bước thực hiện của CNN trong bài toán chuẩn đoán bệnh về xương:



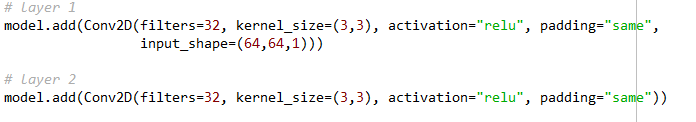
Hình 26: Sơ đồ tổng quát

## 4.5 Xây dựng các layer

Thông thường, một CNN gồm một stack các convolutional module, mỗi một module thực hiện việc extract feature. Mỗi module cũng bao gồm một convolutional layer theo sau là một pooling layer. Module convolutional cuối cùng bao gồm một hoặc nhiều các dense layer thực hiện việc phân lớp. Layer dense cuối cùng trong một CNN bao gồm một single node cho mỗi một lớp cụ thể trong model (tất cả các class khả thi mà model có thể dự đoán được). Với mỗi một node, ta sử dụng hàm kích hoạt softmax để generate ra giá trị predict (từ 0 - 1).

Trong bài toán này, nhóm sử dụng tổng cộng là 15 layer.

* Layer 1 & 2:
* Dùng convolution layer (*Conv2D*) dùng để lấy feature từ image, trong đó:
* *filters:* số filter của convolution layer
* *kernel\_size:* size Sliding window trượt trên image
* *relu:* max(0,x) dùng trong các layer cnn để giảm chi phí tính toán. Có tác dụng đưa các giá trị âm về thành 0. Để loại bỏ các giá trị âm không cần thiết mà có thể sẽ ảnh hưởng cho việc tính toán ở các layer sau đó.
* *padding="same":* có sử dụng padding (="valid": không dùng)
* *input\_shape:* chính là kích thước của dữ liệu đầu vào.
* Layer 1 đầu tiên là layer input nên có *input\_shape.*



Hình 27: Layer 1&2

* Layer 3: *BatchNormalization()* giúp chuẩn hóa (normalizing) và zero-centering (mean subtracting) các inputs (mean của inputs sẽ là 0).



Hình 28: Layer 3

* Layer 4:
* *MaxPooling2D:* lấy những đặc điểm nổi bật nhất và resize lại ảnh.
* *pool\_size:* size pooling, thường có size ma trận 2x2, đối với ảnh lớn thì 4x4.



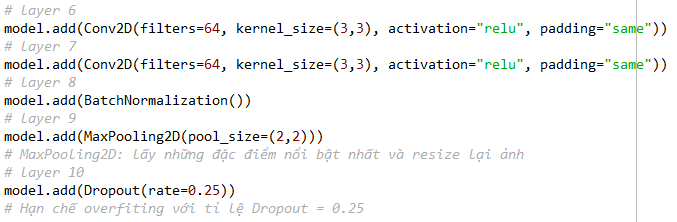
Hình 29: Layer 4

* Layer 5: Hạn chế overfiting với tỉ lệ *Dropout = 0.25*.



Hình 30: Layer 5

* Layer 6-10: Từ layer 6-10 chúng ta thực hiện lại quá trình giống từ 1-5, mục đích nhằm tăng số lượng data input qua đó tăng số lượng phép tính và giảm độ phức tạp trên mỗi phép tính.



Hình 31: Layer 6-10

* Layer 11: *Flatten()* dùng để lát phẳng layer để Fully Connection và chuyển thành ma trận cột.



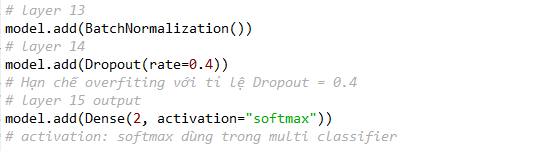
Hình 32: Layer 11

* Layer 12:
* *Dense* layer này sử dụng như một layer neural network bình thường, 1024: chiều output (với filters ảnh đầu vào là 32x32).
* Activation: dùng để chọn *activation="relu"*, relu max(0,x) dùng trong các layer CNN để giảm chi phí tính toán.



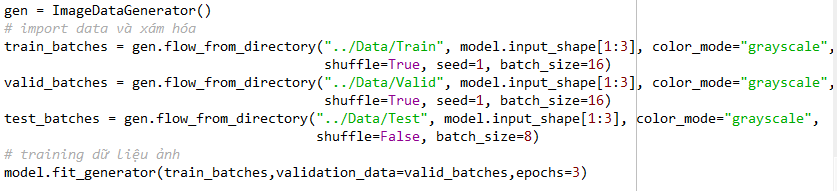
Hình 33: Layer 12

* Layer 13-15: *activation: softmax* dùng trong Multi classifier.



Hình 34: Layer 13-15

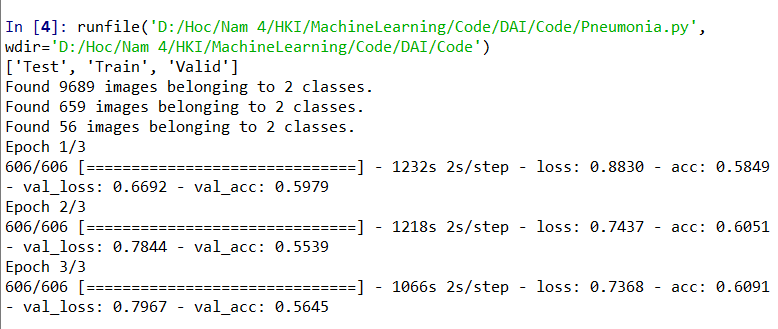
## 4.6 Quá trình compile và xây dựng model



Hình 35: Compile & Build Model

Trong mô hình Keras với API chức năng, cần gọi *fit\_generator()* để huấn luyện dữ liệu hình ảnh được tăng cường bằng cách sử dụng *ImageDataGenerator()*. Sau đó, chúng ta nhúng hình ảnh theo từng tập định sẵn là train, valid và test. Đồng thời ta cũng thực hiện tác vụ xám hóa. Ở đây, ta quy định *batch\_size=16*, với mục đích là sẽ có 16 bức hình được chương trình xử lý mỗi lần.

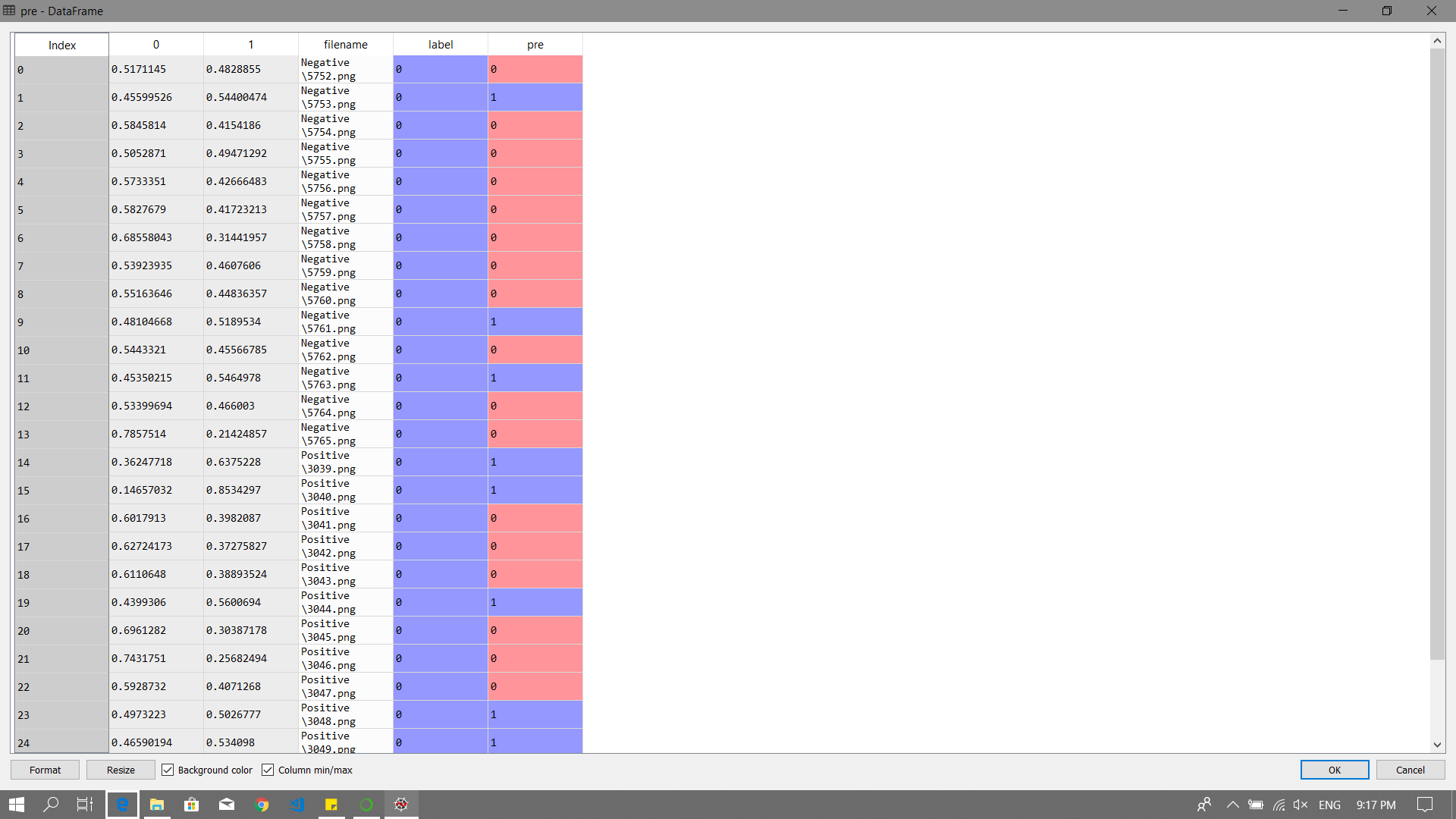
Chương trình sẽ thực thi 3 epoch lần lượt, mỗi epoch gồm 606 tiến trình. Lý do vì trong tập dữ liệu train có tổng cộng là 9689 bức hình và mỗi lần chương trình xử lý 16 bức. Quá trình chạy diễn ra như hình dưới:



Hình 36: Kết quả sau khi train

Sau khi có được model, chúng ta sẽ chạy file **Prediction.py** với model vừa được tạo. Nghĩa là chúng ta sẽ sử dụng số lượng hình ảnh trong tập dữ liệu Test (ở trong ví dụ này là 56) để đánh giá độ chính xác của model.

Kết quả thu được:



Hình 37: Kết quả cuối cùng sau khi test

# Kết quả và đánh giá

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Learning rate | Số hình ảnh train | Số hình ảnh test | Tỷ lệ hình ảnh | Độ chính xác |
| MODEL\_1 | 0,001 | 9760 | 29 | 336:1 | 77.6% |
| MODEL\_2 | 0,002 | 9689 | 56 | 173:1 | 78.5% |
| MODEL\_3 | 0,002 | 9689 | 28 | 346:1 | 67.9% |
| MODEL\_4 | 0,001 | 9707 | 38 | 255:1 | 60.5% |

Bảng 1: Thống kê model

* Qua 4 model nhóm thu được, có thể dễ dàng nhận ra rằng MODEL\_2 có vẻ như là tốt nhất vì có độ chính xác cao nhất (78.5%). Khi so sánh với các model còn lại, điều tạo ra sự khác biệt là ở **Learning rate**. Do đó, trong tương lai, nhóm sẽ cố gắng tìm được một Learning rate tối ưu nhất đối với bài toán này.

# Kết luận

Qua đồ án trên, nhóm đã tiếp cận với các khía cạnh của Machine Learning (Học máy), hiểu được tầm quan trọng của nó. Đồ án còn giúp ta phân biệt được sự giống và khác nhau của ANN (Artificial Neural Network) và CNN (Convolutional Neural Network). Biết được các tiến trình bên trong một thuật toán CNN cơ bản và làm thế nào để sử dụng chúng. Hiểu được vai trò của toán học trong các công nghệ tương lai. Ngoài ra, nhóm cũng hiểu được tiến trình xử lý hình ảnh của máy tính. Dù cho kết quả cuối cùng chưa thực sự tốt nhưng trong tương lai nhóm sẽ cố gắng hoàn thiện sản phẩm hơn. Tổng kết lại, ta có thể thấy trong suốt vài thập niên qua, con người luôn cố gắng vay mượn sức mạnh tính toán của máy tính (một trong những giới hạn của con người) để cải thiện chất lượng cuộc sống cũng như công việc của mình. Machine learning là một trong những công cụ giúp cho chúng ta có thể giải quyết những bài toán mà phương pháp lập trình truyền thống chỉ dựa trên tiến trình không thể nào đạt được. Bằng cách vay mượn một lượng lớn dữ liệu có sẵn, ta có thể “dạy” cho máy tính “học” để chúng có khả năng tự động thực hiện những tác vụ hữu ích cho con người và điển hình nhất là trong lĩnh vực y tế.

# TÀI LIỆU THAM KHẢO

[1] Machine Learning in Python – Edouard Duchesnay (Release 0.1)

[2] Introduction to Machine Learning (Ethem Alpaydin – The MIT Press Cambridge, Massachusetts London, England)

[3] Machine Learning in Python Essential Techniques For Predictive Analysis (Book)

[4] Bài chỉ dẫn Python – Python Software (Foundation Fred L. Drake, Jr., Biên tập viên)

[5] <https://machinelearningcoban.com>

[6] <https://viblo.asia/p/ung-dung-convolutional-neural-network-trong-bai-toan-phan-loai-anh-4dbZNg8ylYM>

[7] <https://www.tensorflow.org>

[8] <https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050>

[5] <https://www.kaggle.com>

[6] <https://stanfordmlgroup.github.io/competitions/mura/>

[7] <https://keras.io/>

[8] <https://stanfordmlgroup.github.io/competitions/mura/>

--- **HẾT ---**