

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CHUYÊN NGÀNH
HỌC KỲ 2, NĂM HỌC 2021-2022

**XÂY DỰNG MÔ HÌNH NHẬN DẠNG
KÝ SỔ VIẾT TAY SỬ DỤNG MẠNG
CNN (CONVOLUTIONAL NEURAL
NETWORK)**

Giáo viên hướng dẫn:
Họ tên: Ngô Thanh Huy

Sinh viên thực hiện:
Họ tên: Phan Minh Nhựt
MSSV: 110119038
Lớp: DA19TTA

Trà Vinh, tháng..... năm.....

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Trà Vinh, ngày tháng năm

Giáo viên hướng dẫn

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Trà Vinh, ngày tháng năm

Thành viên hội đồng

(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy, cô khoa Kỹ Thuật và Công Nghệ, bộ môn Công nghệ Thông Tin của Trường Đại học Trà Vinh đã tận tình dạy dỗ, truyền đạt cho em nhiều kiến thức, kinh nghiệm quý báu trong suốt quá trình học trong trường. Đặc biệt, em xin tỏ lòng biết ơn sâu sắc đến giảng viên, thầy Ngô Thanh Huy đã trực tiếp dìu dắt, giúp đỡ em tận tình, chu đáo trong suốt thời gian em hoàn thiện đồ án cơ sở ngành.

Xin chân thành cảm ơn các bạn trong lớp Công Nghệ Thông Tin A khoá 2019, trường Đại Học Trà Vinh đã giúp đỡ, động viên tôi rất nhiều trong quá trình thực hiện đề tài.

Em xin chân thành cảm ơn!

Trà Vinh, tháng 06 năm 2021

MỤC LỤC

MỤC LỤC.....	iv
DANH MỤC HÌNH ẢNH – BẢNG BIỂU	vi
TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH.....	1
MỞ ĐẦU	2
CHƯƠNG 1: TỔNG QUAN.....	4
1.1 Trước tiên cần tìm hiểu Convolutional Neural Network là gì?	4
1.2 Cấu trúc mạng CNN?	5
1.3 Ứng dụng của CNN?	6
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	9
2.1 Cở sở lý thuyết, lý luận.....	9
2.1.1 Lóp tích chập – Convolution layer	9
2.1.2 Ý tưởng xây dựng mạng CNN	12
2.1.3 Hàm kích hoạt	17
2.1.5 Mạng nơ-ron tích chập so với các kỹ thuật học máy khác.....	19
2.1.6 Mạng thần kinh nhân tạo so với AI cổ điển:	20
2.2 Giả thuyết khoa học	21
2.2.1 Huấn luyện mạng CNN.....	21
2.2.2 Phương pháp huấn luyện mạng CNN.	22
2.3 Phương pháp nghiên cứu.....	23
CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ	26
3.1 Mô tả công việc nghiên cứu	26
3.2 Kết quả nghiên cứu	26
3.3 Ưu điểm, nhược điểm của CNN.	29

CHƯƠNG 4: KẾT LUẬN	31
CHƯƠNG 5: HƯỚNG PHÁT TRIỂN.....	33
DANH MỤC TÀI LIỆU THAM KHẢO.....	34

DANH MỤC HÌNH ẢNH – BẢNG BIỂU

Hình 1: Mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).....	4
Hình 2: Cấu trúc mạng CNN.....	6
Hình 3: Xử lý đơn hình ảnh và xử lý đa hình ảnh.....	6
Hình 4: Ví dụ Classification.....	7
Hình 5: Ví dụ Localization.....	7
Hình 6: Ví dụ Detection	8
Hình 7: Ví dụ Segmentation.....	8
Hình 8: Ảnh minh họa.....	9
Hình 9: Ma trận bộ lọc 3 x 3	10
Hình 10: Feature Map	10
Hình 11: Hình ảnh khi áp dụng các Kernel khác nhau	11
Hình 12: Trường tiếp nhận cục bộ (local receptive field).....	12
Hình 13: Tạo ra neural ẩn đầu tiên trong lớp ẩn 1	13
Hình 14: Dịch filter qua bên phải một cột sẽ tạo được neural ẩn thứ 2.	13
Hình 15: Phân tách dữ liệu ảnh	14
Hình 16: Mô tả CNN2.....	15
Hình 17: Pooling-layer.....	16
Hình 18: Pooling-layer.....	16
Hình 19: Hàm kích hoạt	17
Hình 20: Các giá trị hàm kích hoạt	19
Hình 21: Mỗi lớp của mạng nơ-ron sẽ trích xuất các tính năng từ hình ảnh đầu vào	22

Hình 22: Một số ảnh đã được gán nhãn tương ứng.....	24
Hình 23 Giao diện form dùng để vẽ ký số	25
Hình 24 Lựa chọn batch_size là 128.....	27
Hình 25 Lựa chọn batch_size là 64.....	27
Hình 26 Khung nhận dạng chữ số bằng cách vẽ bằng chuột	28
Hình 27: Kết quả nhận dạng thành công.....	29
Hình 28: Nhận dạng thành công số 4	31
Hình 29: Nhận dạng thành công số 7	32

TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH

Mạng neural tích chập (Convolutional Neural Network – CNN) là 1 trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định đối tượng và nhận dạng khuôn mặt là một trong số những lĩnh vực mà CNN được sử dụng rộng rãi. 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. Vấn đề nghiên cứu ở đây là xây dựng mô hình nhận dạng ký số viết tay sử dụng mạng neural tích chập. Kết quả nghiên cứu cho thấy CNN nhận dạng chữ số với độ chính xác lên đến 98%, có thể nhận dạng thông qua hình ảnh được lưu trữ sẵn và vẽ trực tiếp trên khung bằng chuột.

MỞ ĐẦU

1. Lý do chọn đề tài

Có thể nói, với sự thỏa mãn về cả ba yếu tố: nguồn dữ liệu đủ lớn, phần cứng hỗ trợ mạnh và các thuật toán tiên tiến, trí tuệ nhân tạo (Artificial Intelligent - AI) đã tạo nên một phong trào công nghệ mới trong kỉ nguyên số hóa hiện tại. Trong đó, việc thu thập thông tin từ hệ thống dữ liệu hình ảnh khổng lồ trên toàn thế giới đang là một lĩnh vực được nhiều nhà khoa học trên toàn thế giới quan tâm và nghiên cứu [1]. Đây là cơ hội và cũng là một thách thức hàng đầu, việc ứng dụng trí tuệ nhân tạo nói chung hay kĩ thuật học sâu (Deep Learning - DL) nói riêng đang là một lĩnh vực đầy tính cạnh tranh, mục tiêu hướng đến là tăng tốc độ xử lý, khả năng trích xuất và thu thập thông tin từ nguồn dữ liệu nói trên cho các mục đích sử dụng khác nhau.

Nghiên cứu trình bày về mạng nơ-ron tích chập (Convolutional Neural Network – CNN), và khả năng ứng dụng của nó trên mô hình nhận dạng ký số viết tay. Trong phạm vi bài báo, tôi sẽ cố gắng làm rõ mô hình CNN và đánh giá các khối chức năng, tác động của các tham số đến kết quả nhận dạng ký số. Đối tượng cụ thể ở đây là tập cơ sở dữ liệu chữ số viết tay MNIST [2] (Modified National Institute of Standards and Technology). Cuối cùng, bài báo cũng đề cập đến tính khả dụng của việc xây dựng mô hình nhận dạng ký số viết tay, từ đó có thể nâng cao sự hiểu biết của mình về trí tuệ nhân tạo.

2. Mục đích nghiên cứu

Tìm hiểu về mạng nơ-ron tích chập (Convolutional Neural Network – CNN), tìm hiểu những thuật ngữ, cấu tạo của CNN cũng như cách hoạt động của đó. Cách nó “học” các thông số để đánh giá cho mô hình nhận dạng ký số viết tay.

3. Đối tượng nghiên cứu

- Mạng nơ-ron tích chập (Convolutional Neural Network – CNN),
- Mô hình nhận dạng ký số viết tay dùng mạng Convolutional Neural Network (CNN) và tập cơ sở dữ liệu chữ số viết tay MNIST.

4. Phạm vi nghiên cứu

- Tìm hiểu về lý thuyết của CNN trên các trang web.
- Xây dựng hệ thống nhận dạng ký số viết tay trên ngôn ngữ python với thư viện MNIST.

5. Phương pháp nghiên cứu

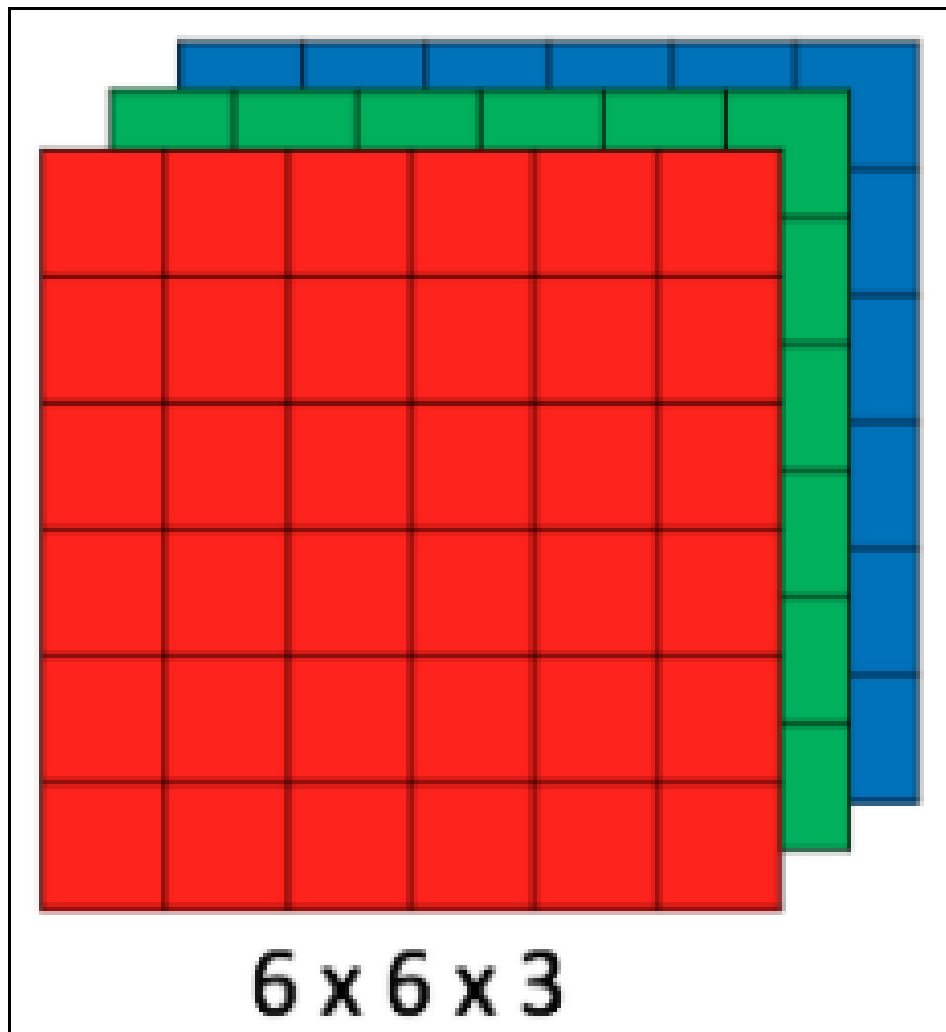
- Đề tài này sử dụng một vài phương pháp nghiên cứu nhưng chủ yếu là phương pháp phân tích và tổng kết kinh nghiệm. Cụ thể là từ những công trình nghiên cứu liên quan đến đề tài và sự hỗ trợ từ thư viện MNIST cùng Tensorflow, đề đề xuất một cách tiếp cận mới trong giải quyết vấn đề đặt ra.

CHƯƠNG 1: TỔNG QUAN

1.1 Trước tiên cần tìm hiểu Convolutional Neural Network là gì?

Convolutional Neural Network (CNN – 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 phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy $H \times W \times D$ (H: Chiều cao, W: Chiều rộng, D: Độ dày). Ví dụ:



Hình 1: Mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).

Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernels), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1.

1.2 Cấu trúc mạng CNN?

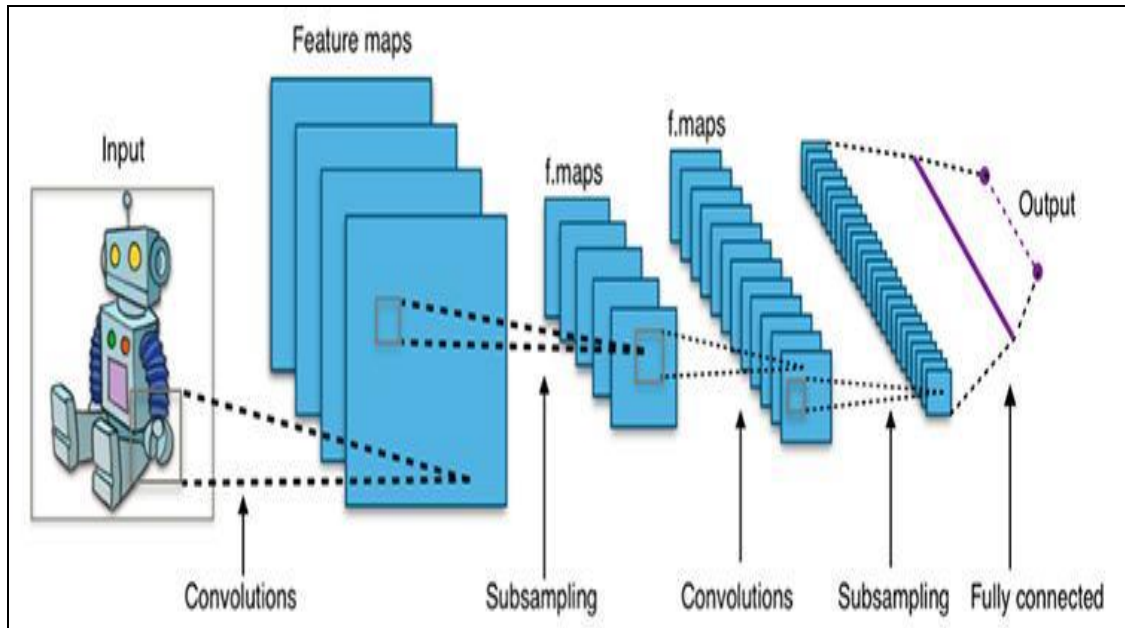
Mạng CNN là một tập hợp các lớp tích chập chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

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). Còn trong mô hình CNNs thì ngược lại.

Các lớp liên kết được với nhau thông qua cơ chế tích chập. Lớp tiếp theo là kết quả phép tính tích chập từ lớp trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neural ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neural trước đó. Mỗi 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, còn có một số lớp khác như pooling/subsampling lớp dùng để chốt lọc lại các thông tin hữu ích hơn. Trong quá trình huấn luyện mạng (training) CNN tự động học các giá trị qua các lớp filter.

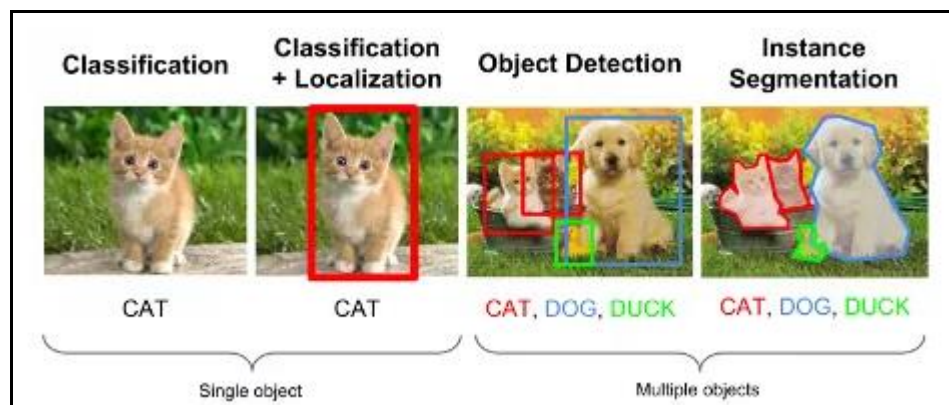
Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Lớp cuối cùng được dùng để phân lớp ảnh.



Hình 2: Cấu trúc mạng CNN

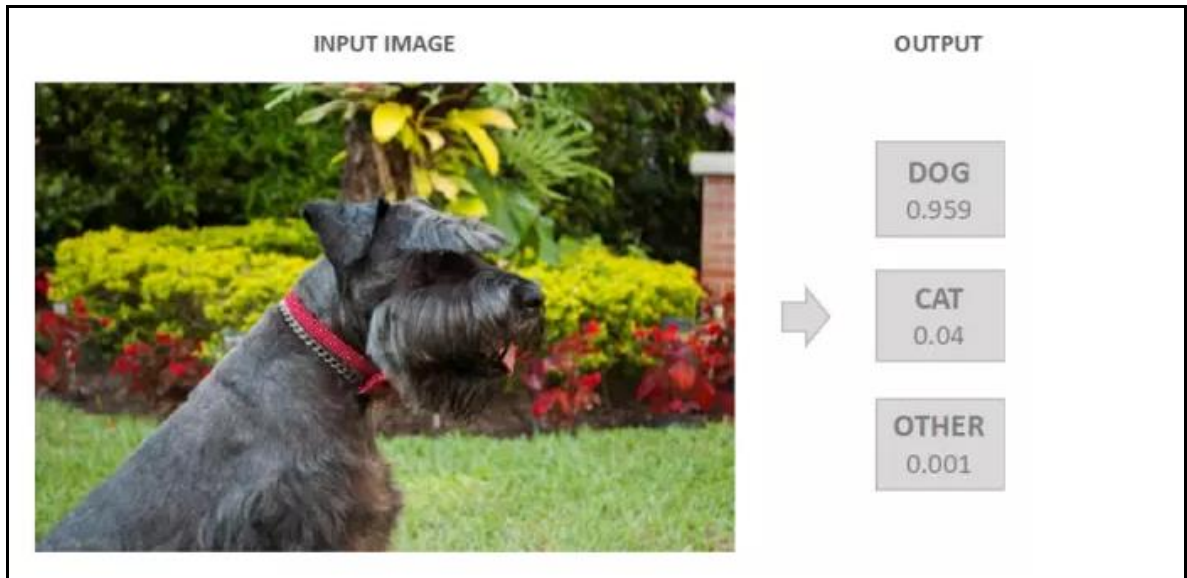
1.3 Ứng dụng của CNN?

Mặc dù CNN chủ yếu được sử dụng cho các vấn đề về computer vision, nhưng điều quan trọng là đề cập đến khả năng giải quyết các vấn đề học tập khác của họ, chủ yếu liên quan đến tích chuỗi dữ liệu. Ví dụ: CNN đã được biết là hoạt động tốt trên chuỗi văn bản, âm thanh và video, đôi khi kết hợp với các mạng khác qua cầu kiến trúc hoặc bằng cách chuyển đổi các chuỗi thành hình ảnh có thể được xử lý của CNN. Một số vấn đề dữ liệu cụ thể có thể được giải quyết bằng cách sử dụng CNN với chuỗi dữ liệu là các bản dịch văn bản bằng máy, xử lý ngôn ngữ tự nhiên và gắn thẻ khung video, trong số nhiều người khác.



Hình 3: Xử lý đơn hình ảnh và xử lý đa hình ảnh

- Classification: Đây là nhiệm vụ được biết đến nhiều nhất trong computer vision. Ý tưởng chính là phân loại nội dung chung của hình ảnh thành một tập hợp các danh mục, được gọi là nhãn. Ví dụ: phân loại có thể xác định xem một hình ảnh có phải là của một con chó, một con mèo hay bất kỳ động vật khác. Việc phân loại này được thực hiện bằng cách xuất ra xác suất của hình ảnh thuộc từng lớp, như được thấy trong hình ảnh sau:



Hình 4: Ví dụ Classification

- Localization: Mục đích chính của localization là tạo ra một hộp giới hạn mô tả vị trí của đối tượng trong hình ảnh. Đầu ra bao gồm một nhãn lớp và một hộp giới hạn. Tác vụ này có thể được sử dụng trong cảm biến để xác định xem một đối tượng ở bên trái hay bên phải của màn hình:



Hình 5: Ví dụ Localization

- Detection: Nhiệm vụ này bao gồm thực hiện localization trên tất cả các đối tượng trong ảnh. Các đầu ra bao gồm nhiều hộp giới hạn, cũng như nhãn lớp (một cho mỗi hộp). Nhiệm vụ này được sử dụng trong việc chế tạo ô tô tự lái, với mục tiêu là có thể xác định vị trí các biển báo giao thông, đường, ô tô khác, người đi bộ và bất kỳ đối tượng nào khác có thể phù hợp để đảm bảo trải nghiệm lái xe an toàn:



Hình 6: Ví dụ Detection

- Segmentation: Nhiệm vụ ở đây là xuất ra cả nhãn lớp và đường viền của mỗi đối tượng hiện diện trong hình ảnh. Điều này chủ yếu được sử dụng để đánh dấu các đối tượng quan trọng của hình ảnh cho phân tích sâu hơn. Ví dụ: tác vụ này có thể được sử dụng để phân định rõ ràng khu vực tương ứng với khối u trong hình ảnh phổi của bệnh nhân. Hình sau mô tả cách vật thể quan tâm được phác thảo và gán nhãn:



Hình 7: Ví dụ Segmentation

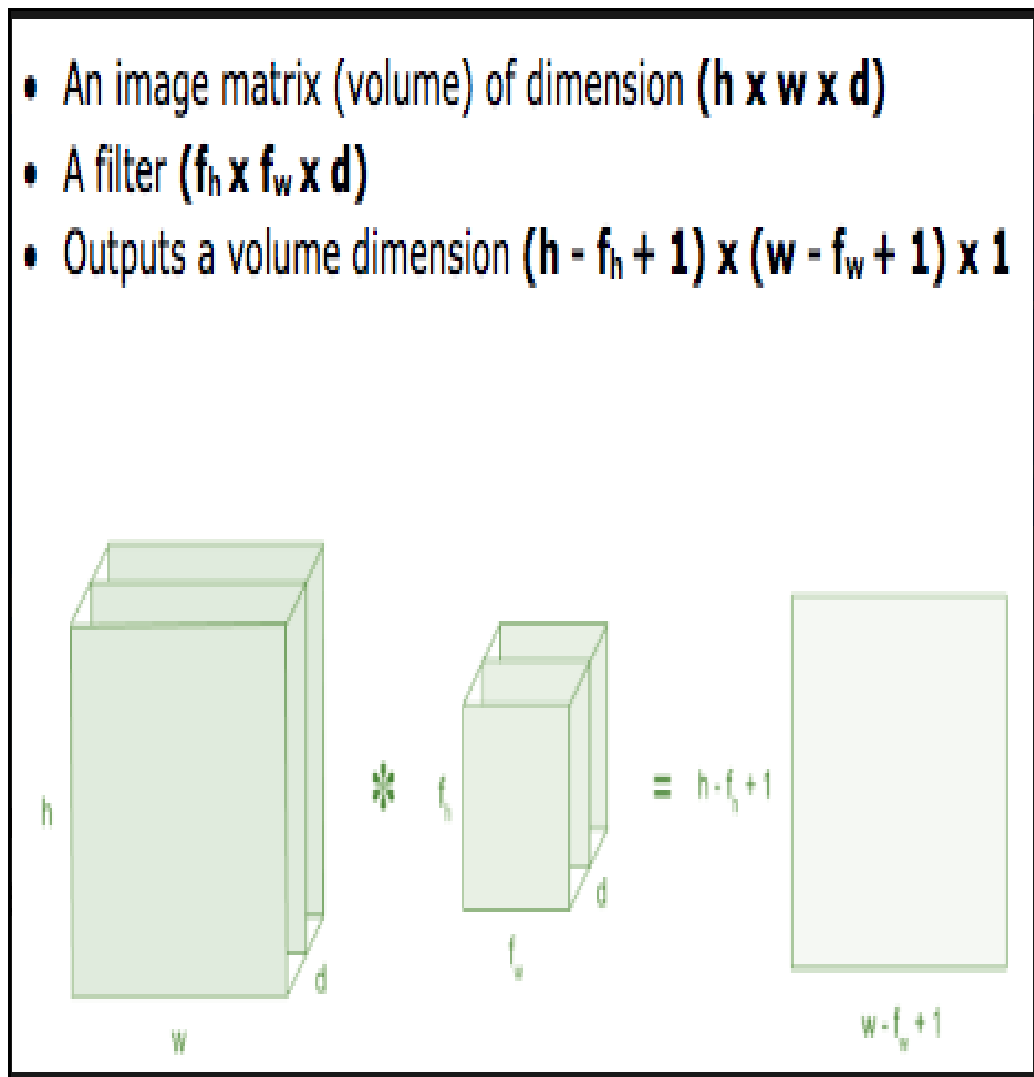
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Cở sở lý thuyết, lý luận

2.1.1 Lớp tích chập – Convolution layer

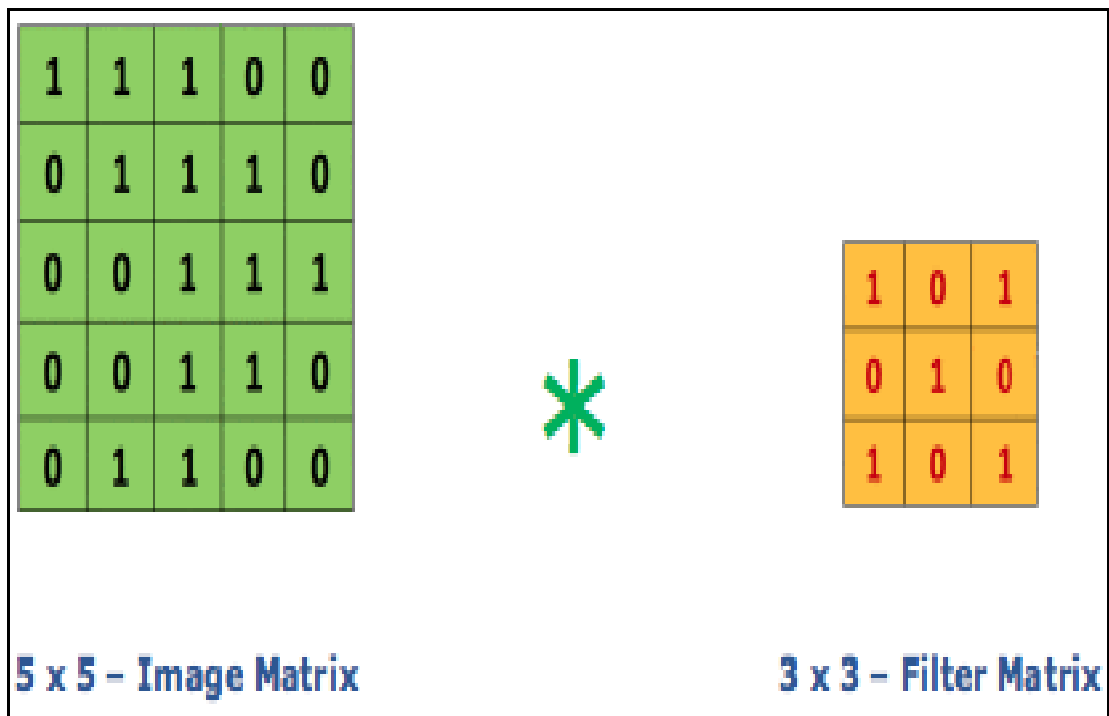
Tích chập là lớp đầu tiên để trích xuất các tính năng từ hình ảnh đầu vào. Tích chập duy trì mối quan hệ giữa các pixel bằng cách tìm hiểu các tính năng hình ảnh bằng cách sử dụng các ô vuông nhỏ của dữ liệu đầu vào. Nó là 1 phép toán có 2 đầu vào như ma trận hình ảnh và 1 bộ lọc hoặc hạt nhân.

- An image matrix (volume) of dimension $(h \times w \times d)$
- A filter $(f_h \times f_w \times d)$
- Outputs a volume dimension $(h - f_h + 1) \times (w - f_w + 1) \times 1$



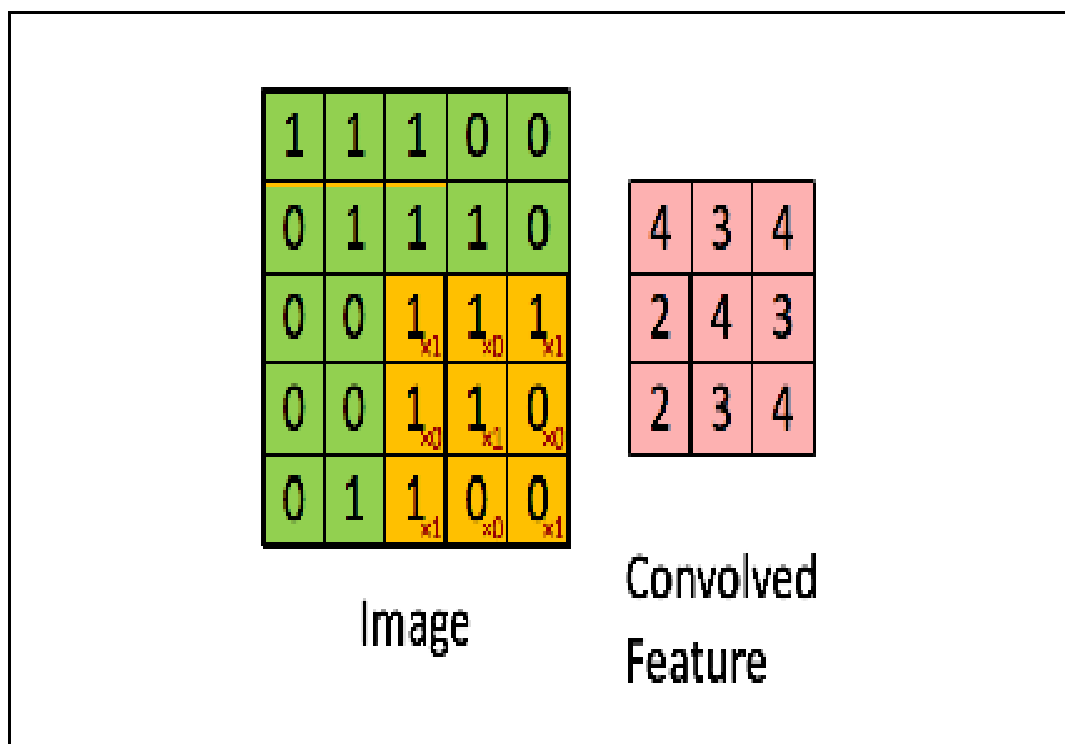
Hình 8: Ảnh minh họa

Xem xét 1 ma trận 5×5 có giá trị pixel là 0 và 1. Ma trận bộ lọc 3×3 như hình bên dưới.










Hình 9: Ma trận bộ lọc 3 x 3

Sau đó, lớp tích chập của ma trận hình ảnh 5 x 5 nhân với ma trận bộ lọc 3 x 3 gọi là 'Feature Map' như hình bên dưới.



Hình 10: Feature Map

Sự kết hợp của 1 hình ảnh với các bộ lọc khác nhau có thể thực hiện các hoạt động như phát hiện cạnh, làm mờ và làm sắc nét bằng cách áp dụng các bộ lọc. Ví dụ dưới đây cho thấy hình ảnh tích chập khác nhau sau khi áp dụng các Kernel khác nhau.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Hình 11: Hình ảnh khi áp dụng các Kernel khác nhau

2.1.2 Ý tưởng xây dựng mạng CNN

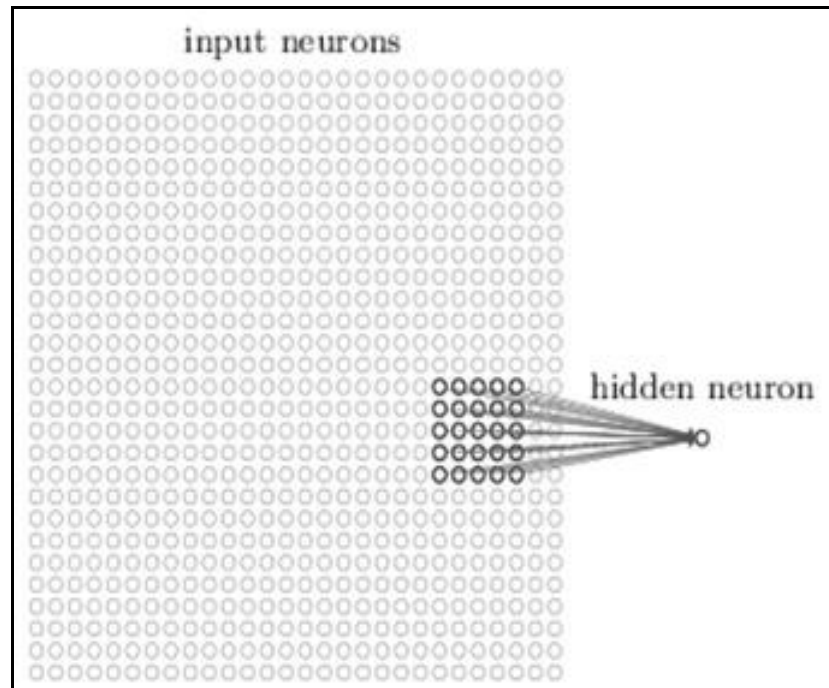
Mạng CNN sử dụng 3 ý tưởng cơ bản:

- 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):

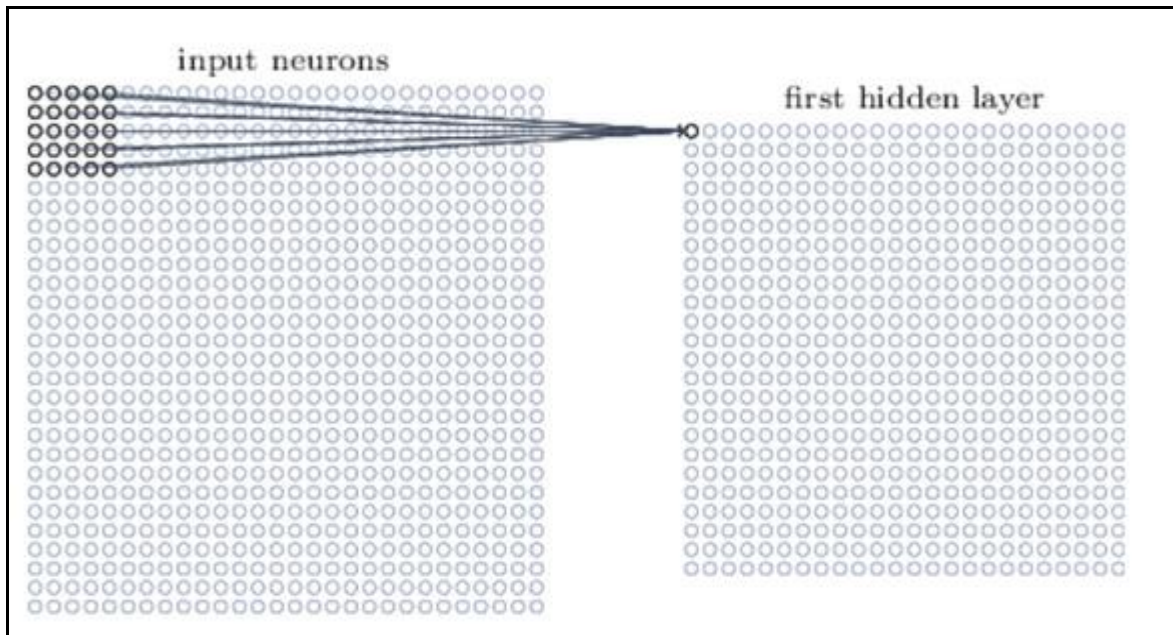
- Đầu vào của mạng CNN là một ảnh. Ví dụ như ảnh có kích thước 28×28 thì tương ứng đầu vào là một ma trận có 28×28 và giá trị mỗi điểm ảnh là một ô trong ma trận. Trong mô hình mạng CNN truyền thống thì chúng ta sẽ kết nối các neural đầu vào vào tầng ảnh.

- Tuy nhiên trong CNN chúng ta không làm như vậy mà chúng ta chỉ kết nối trong một vùng nhỏ của các neural 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 neural ẩn sẽ học một bias. Mỗi một vùng 5×5 đây gọi là một trường tiếp nhận cục bộ.

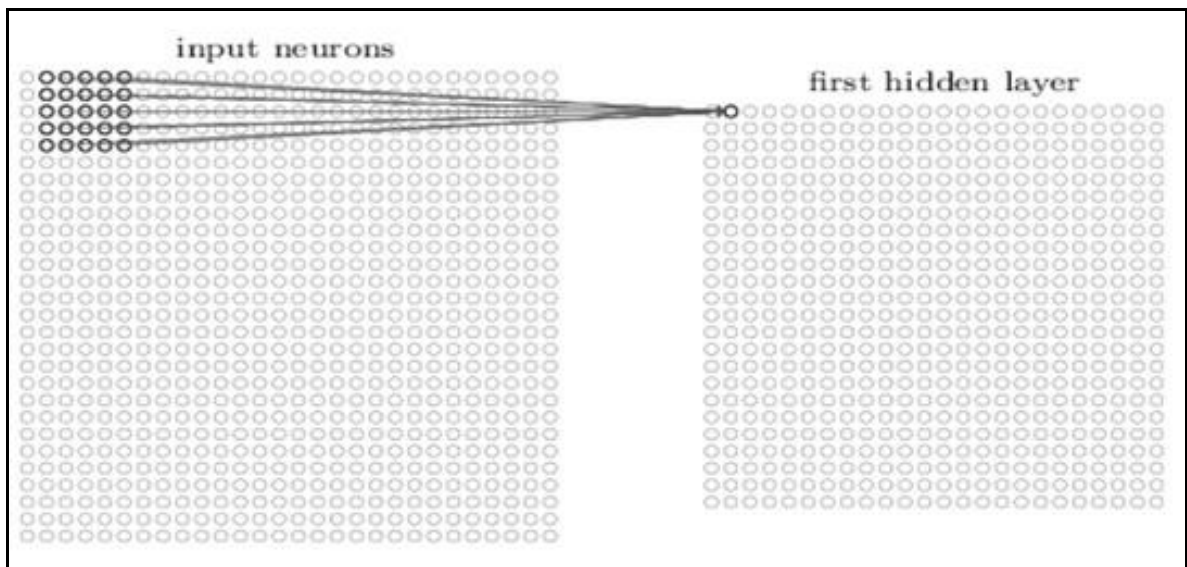


Hình 12: Trường tiếp nhận cục bộ (local receptive field)

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

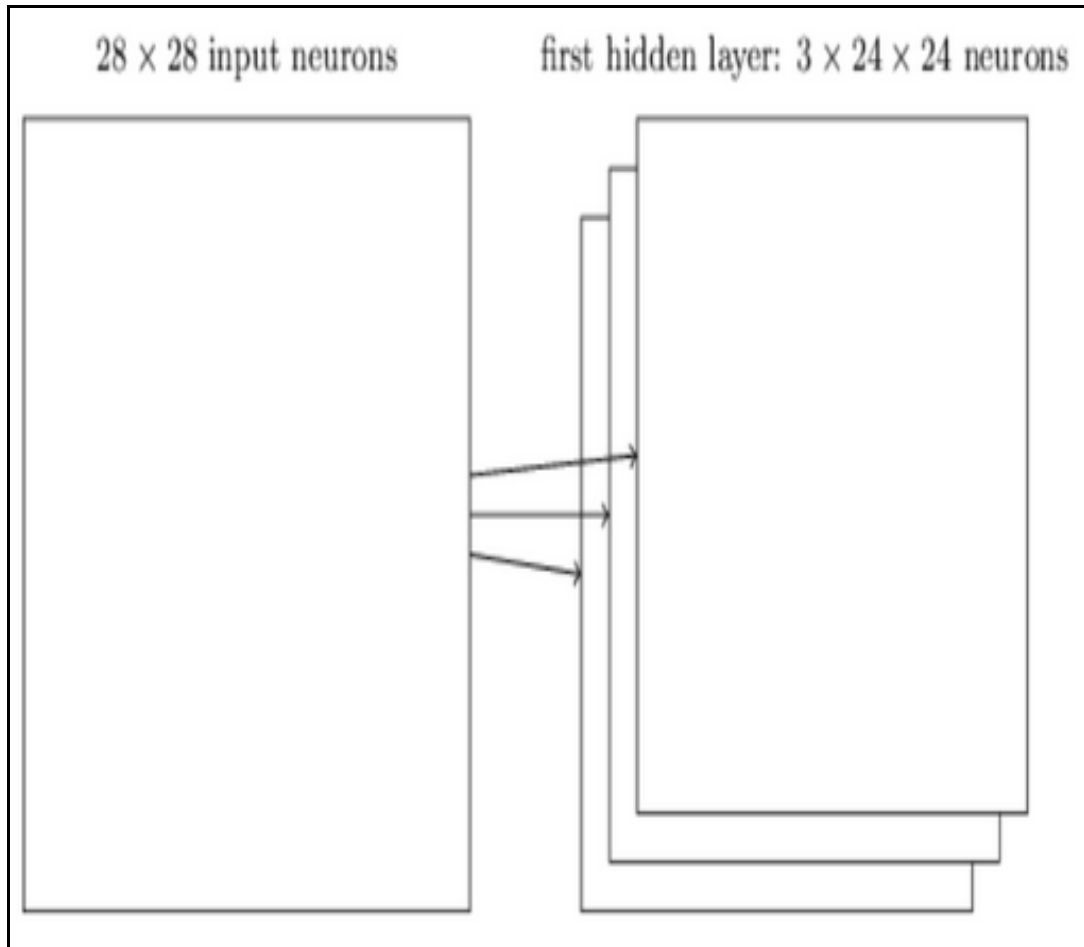


Hình 13: Tạo ra neural ẩn đầu tiên trong lớp ẩn 1



Hình 14: Dịch filter qua bên phải một cột sẽ tạo được neural 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 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 15: Phân tách dữ liệu ảnh

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 neuron 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?

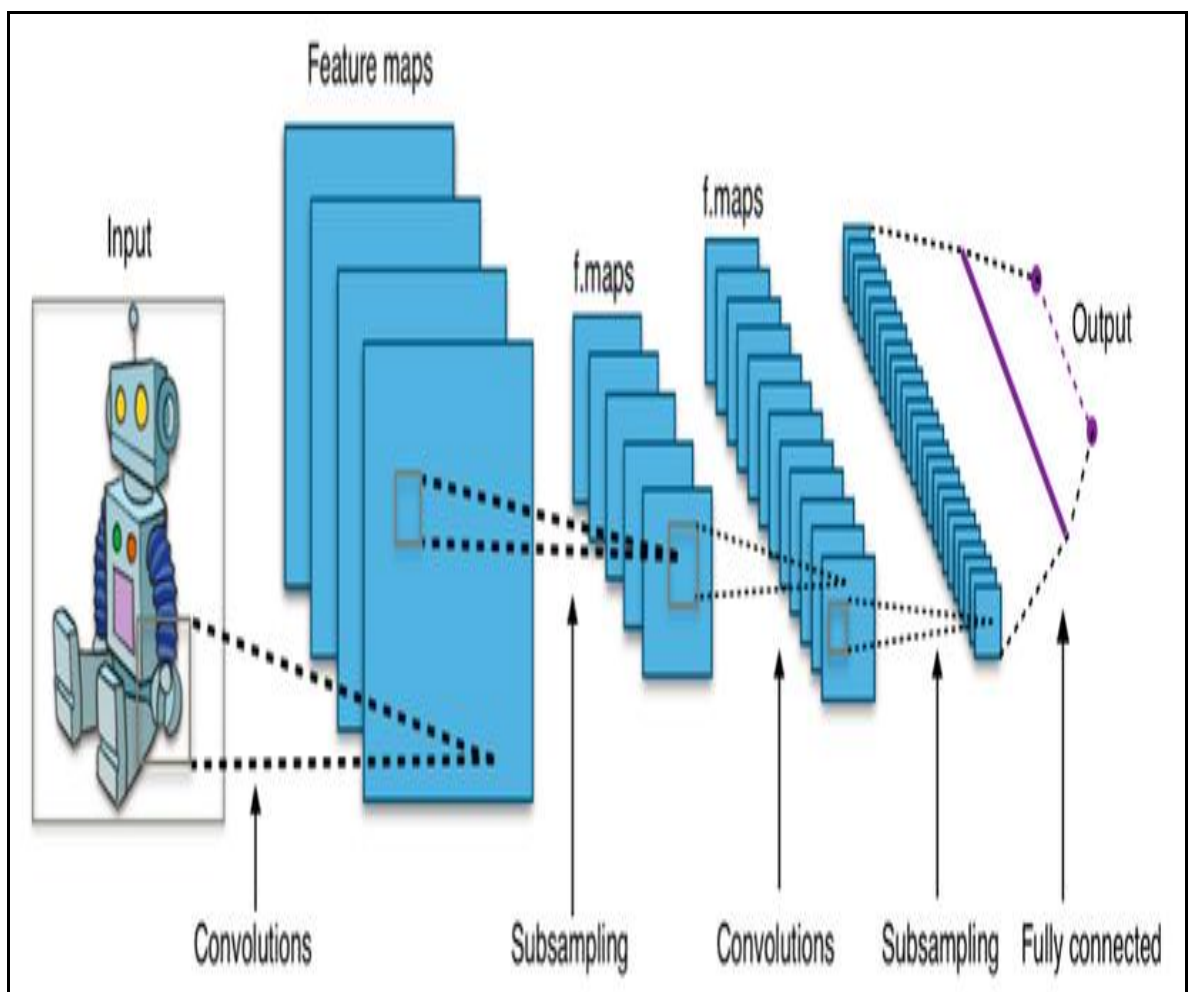
Chúng ta thấy mỗi fearture map cần $25 = 5 \times 5$ shared weight và 1 shared bias. Như vậy mỗi feature map cần $5 \times 5 + 1 = 26$ tham số. Như vậy nếu có 10 feature map thì có $10 \times 26 = 260$ tham số. Chúng ta xét lại nếu layer đầu tiên có kết nối đầy đủ nghĩa là chúng ta có $28 \times 28 = 784$ 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.

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.

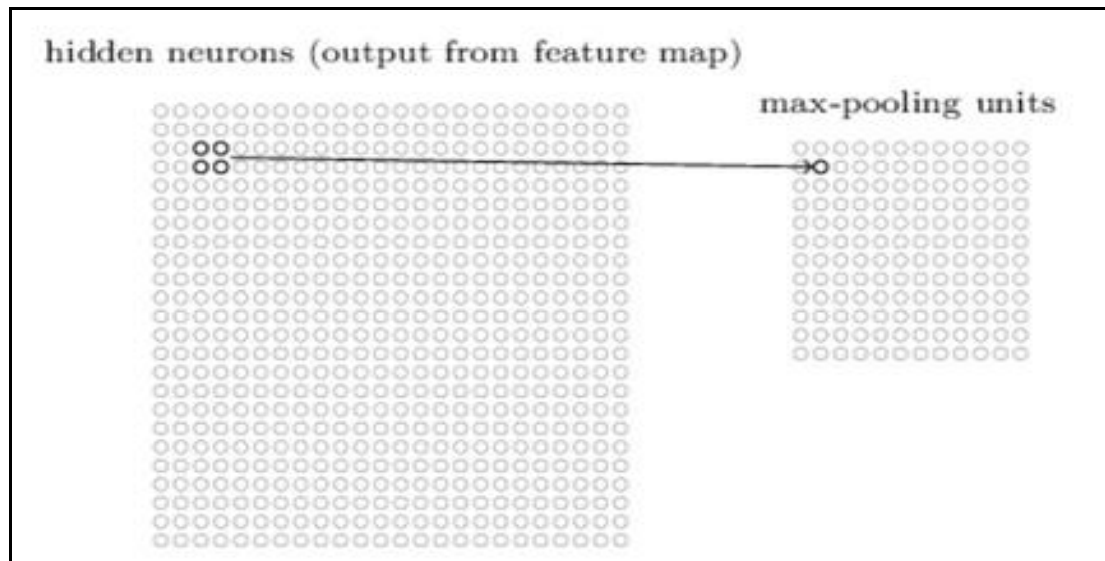
❖ Lớp tổng hợp (pooling layer):

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



Hình 16: Mô tả CNN2

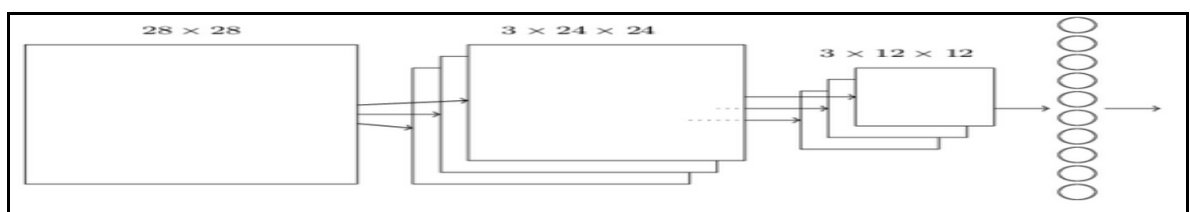
Thủ tục pooling phổ biến là max-pooling, thủ tục này chọn giá trị lớn nhất trong vùng đầu vào 2×2 .



Hình 17: Pooling-layer

Như vậy qua lớp Max Pooling thì số lượng neural 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 neural với số lượng tùy bài toán.



Hình 18: Pooling-layer

❖ **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

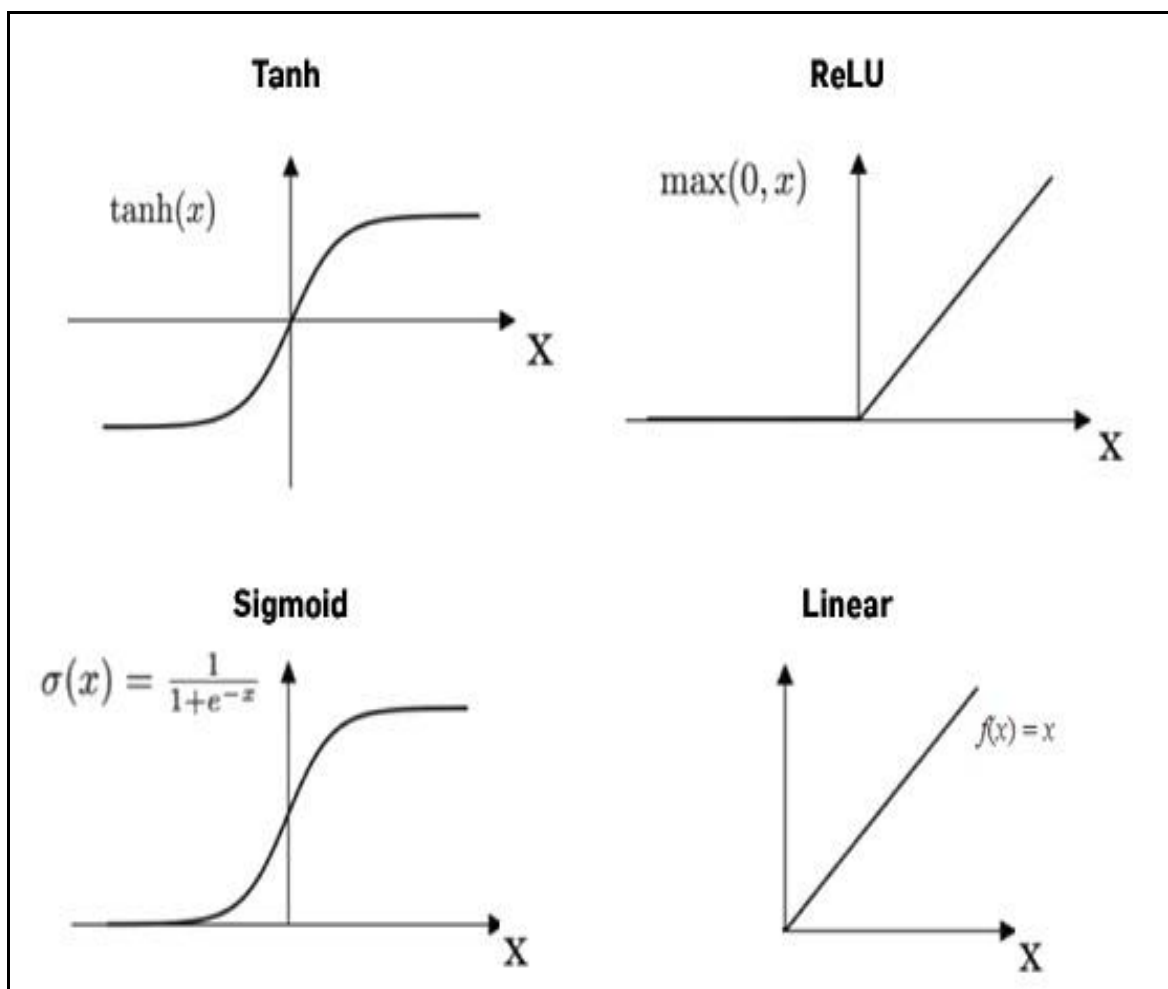
Filter size: thường filter theo size 5×5 hoặc 3×3

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

Cách cuối cùng là thực hiện nhiều lần việc train test để chọn ra được param tốt nhất.

2.1.3 Hàm kích hoạt

Hàm kích hoạt (activation function) luôn xuất hiện ở đầu ra của mỗi nơ-ron, nó sẽ đưa ra quyết định xem có đưa thông tin đã được tổng hợp sang nơ-ron khác hay không. Nhìn một cách tổng quát nó giống như một chiếc công tắc giúp bật tắt nơ-ron. Có rất nhiều loại hàm kích hoạt (activation function) và những nhà nghiên cứu vẫn đang tìm những hàm kích hoạt (activation function) mới hiệu quả hơn, 4 hàm kích hoạt (activation function) dưới đây là thông dụng hơn cả:



Hình 19: Hàm kích hoạt

Trong đó Sigmoid-Softmax và Linear được sử dụng ở các nơon đầu ra, tương ứng cho bài toán phân loại (classification) hay hồi quy (regression).

Các hàm kích hoạt (activation function) Sigmoid, Tanh và ReLU còn được gọi là các hàm kích hoạt phi tuyến tính (non-linear activation function) và được dùng trong kết nối giữa các nơon với nhau.

Nếu như hàm kích hoạt (activation function) Linear được sử dụng trong kết nối giữa các nơon, chúng ta sẽ vô hình chung quy tất cả các lớp nơon về một vì tổng hợp các hàm tuyến tính thì vẫn sẽ là một hàm tuyến tính.

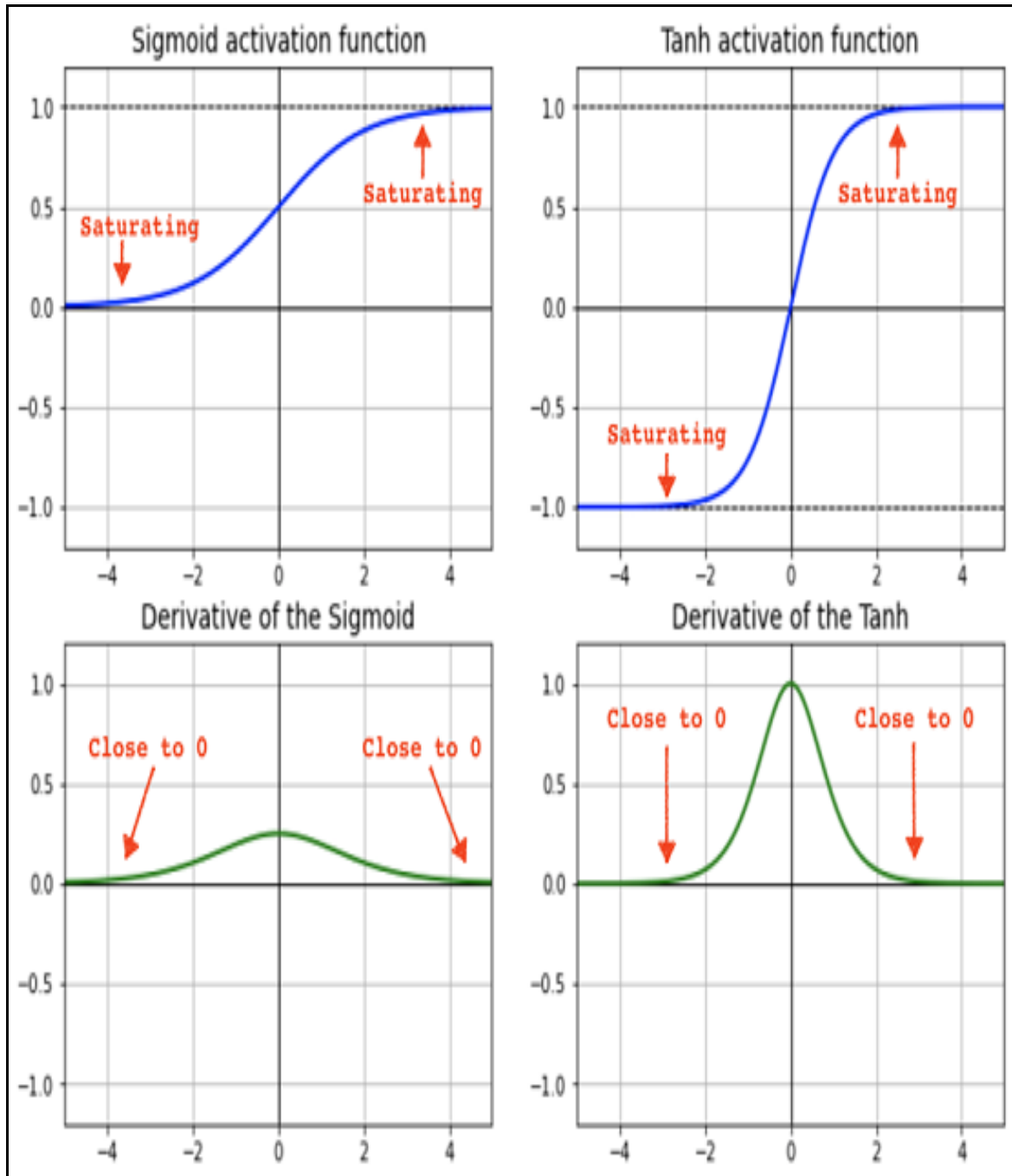
Đó là lý do mà các hàm kích hoạt phi tuyến tính (non-linear activation function) được sử dụng, từ đây chúng ta có thể kết hợp các lớp nơon một cách đúng nghĩa, mang đến khả năng xử lý những dữ liệu phức tạp với độ chính xác cao.

Trong số 3 hàm kích hoạt phi tuyến tính (non-linear activation function) trên, ReLU là hàm kích hoạt thường được sử dụng hơn cả, do hiện tượng tiêu biến độ dốc (vanishing gradient) ở hàm kích hoạt (activation function) Sigmoid và Tanh.

Hiện tượng tiêu biến độ dốc (vanishing gradient) xảy ra đối với các giá trị ở những điểm bão hòa của hàm số Tanh và Sigmoid, lúc này đạo hàm (derivative) trở nên rất nhỏ, hậu quả làm cho việc thay đổi các trọng số (weight) trong quá trình huấn luyện (training) gần như bằng 0, việc huấn luyện (training) của model gần như không có tác dụng.

Có một ví dụ rất hay về hiện tượng này, đó là như khi bạn nhấn ga một chiếc xe nhưng nó gần như không nhúc nhích về phía trước.

Với hàm kích hoạt (activation function) ReLU, các giá trị dương giờ đây được giữ nguyên giá trị, không còn những điểm bão hòa nữa, đạo hàm (derivative) luôn có độ lớn đủ tốt cho việc huấn luyện (training).



Hình 20: Các giá trị hàm kích hoạt

2.1.5 Mạng nơ-ron tích chập so với các kỹ thuật học máy khác

Mạng nơ-ron tích chập chỉ là một trong số các thuật toán để thực hiện học máy, một nhánh của trí tuệ nhân tạo phát triển hành vi dựa trên kinh nghiệm. Có nhiều kỹ thuật học máy khác có thể tìm thấy các mẫu trong dữ liệu và thực hiện các tác vụ như phân loại và dự đoán. Một số kỹ thuật này bao gồm mô hình hồi quy, máy vector hỗ trợ, phương pháp k-gần nhất và cây quyết định.

Tuy nhiên, khi nói đến việc xử lý dữ liệu lộn xộn và không có cấu trúc như hình ảnh, âm thanh và văn bản, mạng nơ-ron vượt trội hơn các kỹ thuật học máy khác.

Ví dụ: nếu bạn muốn thực hiện các nhiệm vụ phân loại hình ảnh bằng các thuật toán học máy cổ điển, bạn sẽ phải thực hiện nhiều “kỹ thuật tính năng” phức tạp, một quá trình phức tạp và gian khổ đòi hỏi nỗ lực của một số kỹ sư và chuyên gia miền. Mạng nơ-ron và thuật toán học sâu không yêu cầu kỹ thuật tính năng và tự động trích xuất các tính năng từ hình ảnh nếu được đào tạo tốt.

Tuy nhiên, điều này không có nghĩa là mạng nơ-ron là sự thay thế cho các kỹ thuật học máy khác. Các loại thuật toán khác yêu cầu ít tài nguyên tính toán hơn và ít phức tạp hơn, điều này khiến chúng thích hợp hơn khi bạn đang cố gắng giải quyết một vấn đề không yêu cầu mạng nơ-ron.

Các kỹ thuật học máy khác cũng có thể diễn giải được, có nghĩa là việc điều tra và sửa chữa các quyết định mà chúng đưa ra sẽ dễ dàng hơn. Điều này có thể làm cho chúng thích hợp hơn trong các trường hợp sử dụng mà khả năng diễn giải quan trọng hơn độ chính xác.

2.1.6 Mạng thần kinh nhân tạo so với AI cổ điển:

Các chương trình AI truyền thống, dựa trên quy tắc dựa trên các nguyên tắc của phần mềm cổ điển [3]. Các chương trình máy tính được thiết kế để chạy các hoạt động trên dữ liệu được lưu trữ trong các vị trí bộ nhớ và lưu kết quả trên một vị trí bộ nhớ khác. Logic của chương trình là tuần tự, xác định và dựa trên các quy tắc được xác định rõ ràng. Các hoạt động được điều hành bởi một hoặc nhiều bộ xử lý trung tâm.

Tuy nhiên, mạng nơ-ron không tuần tự, cũng không xác định. Ngoài ra, bất kể phần cứng bên dưới là gì, không có bộ xử lý trung tâm nào kiểm soát logic. Thay vào đó, logic được phân tán trên hàng nghìn tế bào thần kinh nhân tạo nhỏ hơn. CNN không chạy hướng dẫn; thay vào đó họ thực hiện các phép toán trên đầu vào của họ. Các hoạt động tập thể của họ phát triển hành vi của mô hình.

2.2 Giả thuyết khoa học

Để mạng nơ-ron nhân tạo có thể hoạt động cần phải huấn luyện cho nó “học”.

2.2.1 Huấn luyện mạng CNN.

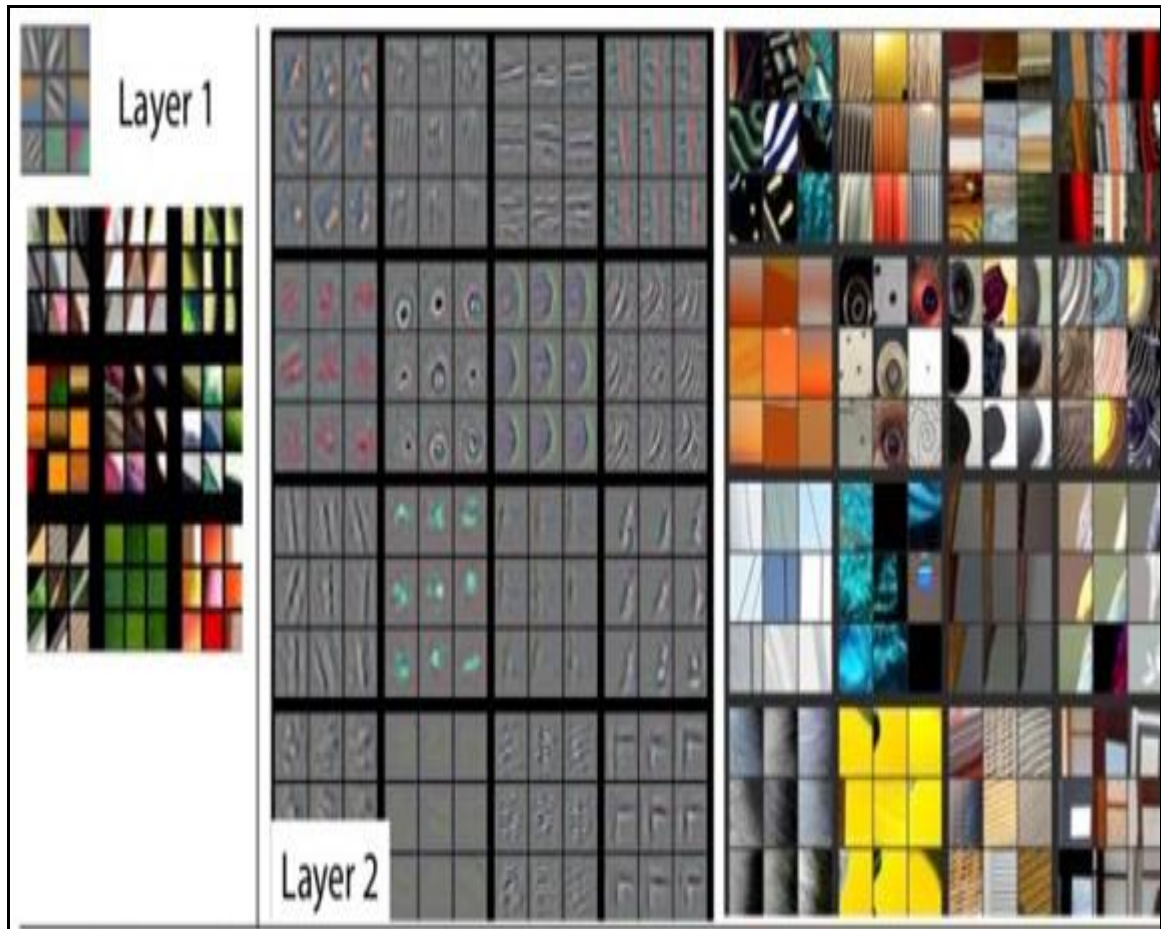
Mạng nơron tích chập bắt đầu bằng cách gán các giá trị ngẫu nhiên cho trọng số của các kết nối giữa các nơron. Chìa khóa để CNN thực hiện đúng và chính xác nhiệm vụ của mình là điều chỉnh các trọng số này về đúng số lượng. Nhưng việc tìm kiếm trọng lượng phù hợp không phải là điều dễ dàng, đặc biệt là khi bạn đang xử lý nhiều lớp và hàng nghìn tế bào thần kinh.

Việc hiệu chuẩn này được thực hiện bằng cách “huấn luyện” mạng với các ví dụ được chú thích.

Ví dụ: nếu muốn đào tạo bộ phân loại hình ảnh được đề cập ở trên, cần cung cấp cho nó nhiều ảnh, mỗi ảnh được gắn nhãn với lớp tương ứng (người, ô tô hoặc động vật). Khi cung cấp cho nó ngày càng nhiều ví dụ đào tạo, mạng nơ-ron tích chập dần dần điều chỉnh trọng số của nó để ánh xạ từng đầu vào thành các đầu ra chính xác.

Về cơ bản, những gì xảy ra trong quá trình đào tạo là mạng tự điều chỉnh để thu thập các mẫu cụ thể từ dữ liệu. Một lần nữa, trong trường hợp của mạng phân loại hình ảnh, khi đào tạo mô hình AI với các ví dụ chất lượng, mỗi lớp sẽ phát hiện một lớp tính năng cụ thể.

Ví dụ: lớp đầu tiên có thể phát hiện các cạnh ngang và dọc, các lớp tiếp theo có thể phát hiện các góc và hình tròn. Xa hơn nữa trong mạng, các lớp sâu hơn sẽ bắt đầu chọn ra các tính năng nâng cao hơn như khuôn mặt và vật thể.



Hình 21: Mỗi lớp của mạng nơ-ron sẽ trích xuất các tính năng từ hình ảnh đầu vào

Khi chạy một hình ảnh mới thông qua một mạng nơ-ron được đào tạo tốt, trọng lượng được điều chỉnh của các nơ-ron sẽ có thể trích xuất các đặc điểm phù hợp và xác định chính xác hình ảnh thuộc lớp đầu ra nào.

2.2.2 Phương pháp huấn luyện mạng CNN.

- *Phương pháp học:* đặc trưng cơ bản của mạng là có khả năng học, khả năng tái tạo các hình ảnh và dữ liệu khi đã học. Trong trạng thái học, thông tin được lan truyền theo hai chiều nhiều lần để học các trọng số. Có 3 kiểu học chính, mỗi kiểu học tương ứng với một nhiệm vụ học trừu tượng. Đó là học có giám sát (có mẫu), học không giám sát và học tăng cường. Thông thường, loại kiến trúc mạng nào cũng có thể dùng được cho các nhiệm vụ.

- *Học có giám sát:* Một thành phần không thể thiếu của phương pháp này là sự có mặt của một người thầy (ở bên ngoài hệ thống). Người thầy này có kiến thức

về môi trường thể hiện qua một tập hợp các cặp đầu vào - đầu ra đã được biết trước. Hệ thống học (ở đây là mạng neural tích chập) sẽ phải tìm cách thay đổi các tham số bên trong của mình (các trọng số và các ngưỡng) để tạo nên một ánh xạ có khả năng ánh xạ các đầu vào thành các đầu ra mong muốn. Sự thay đổi này được tiến hành nhờ việc so sánh giữa đầu ra thực sự và đầu ra mong muốn.

- *Học không giám sát*: Trong học không có giám sát, ta được cho trước một số dữ liệu x và hàm chi phí cần được cực tiểu hóa có thể là một hàm bất kỳ của dữ liệu x và đầu ra của mạng, f – hàm chi phí được quyết định bởi phát biểu của bài toán. Phần lớn các ứng dụng nằm trong vùng của các bài toán ước lượng như mô hình hóa thống kê, nén, lọc, phân cụm.

- *Học tăng cường*: Dữ liệu x thường không được tạo trước mà được tạo ra trong quá trình một agent tương tác với môi trường. Tại mỗi thời điểm t , agent thực hiện hành động yt và môi trường tạo một quan sát xt với một chi phí tức thời Ct , theo một quy trình động nào đó (thường là không được biết). Mục tiêu là một sách lược lựa chọn hành động để cực tiểu hóa một chi phí dài hạn nào đó, nghĩa là chi phí tích lũy mong đợi. Quy trình hoạt động của môi trường và chi phí dài hạn cho mỗi sách lược thường không được biết, nhưng có thể ước lượng được. Mạng neural tích chập thường được dùng trong học tăng cường như một phần của thuật toán toàn cục. Các bài toán thường được giải quyết bằng học tăng cường là các bài toán điều khiển, trò chơi và các nhiệm vụ quyết định tuần tự (sequential decision making) khác.

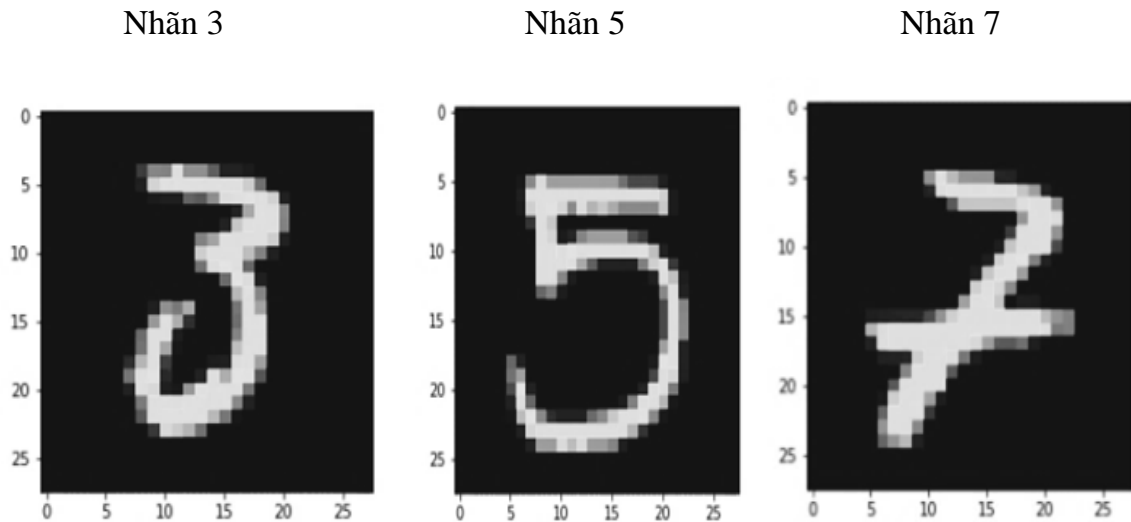
2.3 Phương pháp nghiên cứu

- ❖ **Xây dựng file training.py chứa mô hình nhận dạng ký số viết tay với CNN (file chứa mô hình dành cho việc huấn luyện).**

Ở đây, em sử dụng mạng CNN và để xử lý bài toán. Mục đích là để kiểm chứng khả năng nhận dạng của mạng nơ-ron tích chập, nhận dạng được các số từ 0 đến 9.

Cơ sở dữ liệu: Bài báo sử dụng tập dữ liệu MNIST để làm cơ sở dữ liệu đánh giá hệ thống. Bộ dữ liệu MNIST được chia thành hai phần: dữ liệu dành cho quá

trình huấn luyện và dữ liệu dành cho quá trình kiểm tra. Dữ liệu huấn luyện gồm 60000 ảnh, dữ liệu kiểm tra gồm 10000 ảnh. Tất cả đều là hình ảnh đen trắng các chữ số viết tay từ 0 tới 9 có kích thước 28 pixel x 28 pixel đã được gắn nhãn đúng. Một số hình ảnh ngẫu nhiên được trích xuất từ bộ dữ liệu MNIST:



Hình 22: Một số ảnh đã được gắn nhãn tương ứng.

Lựa chọn các thông số ban đầu: Ngõ vào và ngõ ra của mạng sẽ là các thông số cố định của một mô hình mạng neural tích chập. Ở bài toán nhận diện chữ số viết tay sử dụng bộ dữ liệu MNIST, mục đích của mạng sẽ nhận dạng được 10 chữ số khác nhau từ 0 đến 9, từ đó lớp ngõ ra cuối cùng cần 10 neural thể hiện 10 chữ số khác nhau. Với ảnh 2 chiều có kích thước 28 pixel x 28 pixel chuyển về dạng dữ liệu 1 chiều 784 pixel x 1 pixel cho mỗi hình ảnh. Ta xây dựng một mô hình mạng CNN với 784 neural đầu vào tương ứng với 784 pixel. Với 10 giá trị ngõ ra (tương ứng với các nhãn 0, 1, 2, ..., 9), sẽ có 10 neural ở lớp ngõ ra.

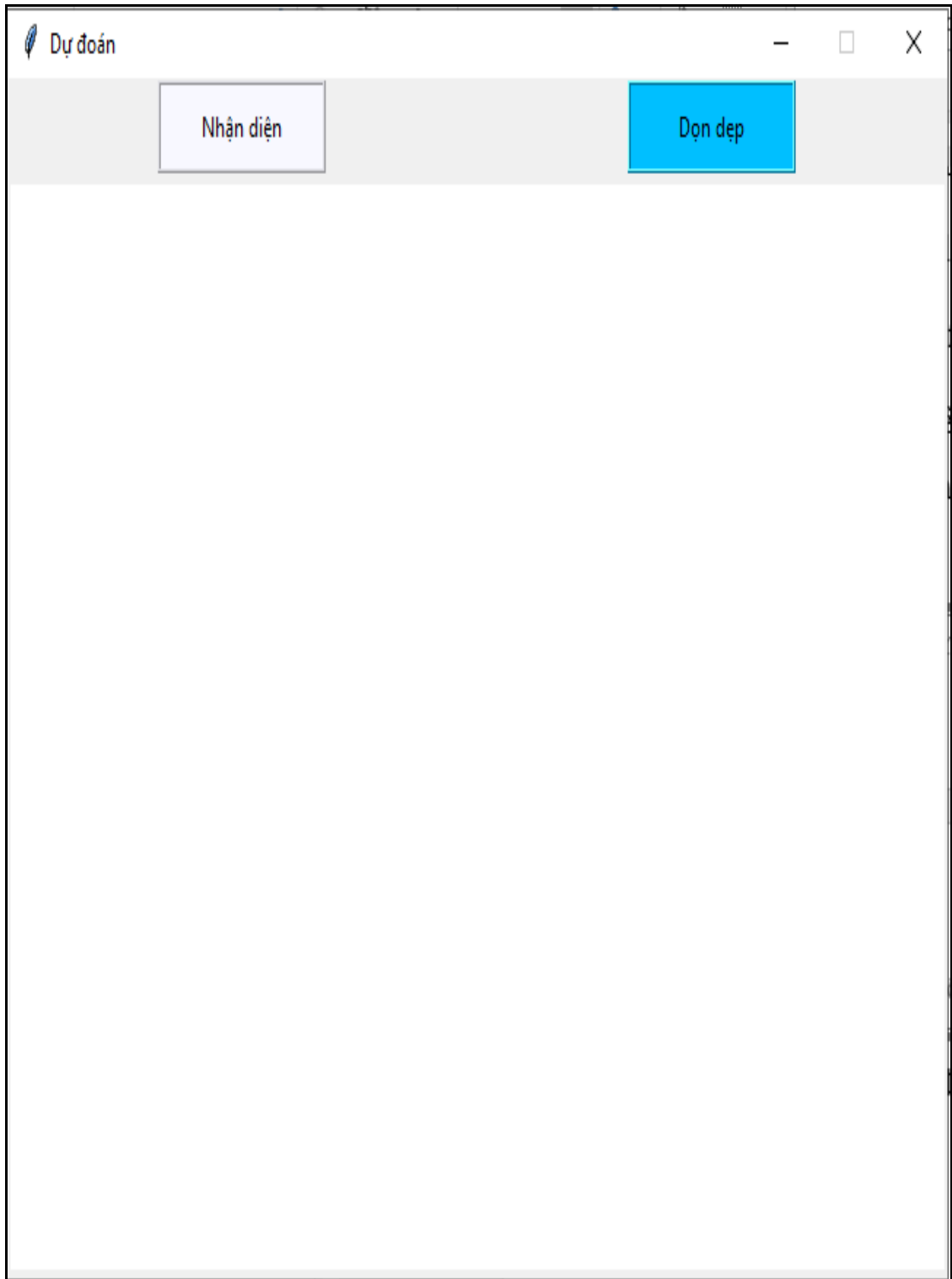
Các thông số khác như: số neural lớp ẩn, số lớp tích chập, mini-batch size sẽ được làm rõ ở phần đánh giá.

Các dữ liệu huấn luyện trong này sẽ được lưu dưới dạng tệp mnist.h5.

❖ **Xây dựng file form.py chứa code dùng cho việc vẽ ký số viết tay để**

Tạo một file đặt tên làm form.py chứa code cho việc vẽ các ký số, form.py sẽ lấy dữ liệu từ file mnist.h5 đã huấn luyện trong khi chạy file training.py.

Khi chạy sẽ tạo ra form để vẽ với chiều rộng là 640, chiều cao là 480



Hình 23 Giao diện form dùng để vẽ ký số

CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ

3.1 Mô tả công việc nghiên cứu

Mô hình thực nghiệm được lập trình với ngôn ngữ Python, framework được sử dụng là tensorflow. Khởi tạo với mô hình mạng CNN gồm 1 lớp: 1 lớp ngõ vào 32 neural sử dụng hàm kích hoạt relu, 3 lớp tích chập có 64 neural sử dụng hàm kích hoạt relu, lớp ngõ ra 10 neural sử dụng hàm kích hoạt là softmax (softmax có tác dụng chọn ra giá trị lớn nhất trong tập số).

Thử nghiệm với 50 chu kỳ học, lúc này kết quả huấn luyện gần như đạt bão hòa trong 40 chu kỳ liên tiếp gần (mỗi chu kỳ học tương đương một lần quét qua hết tất cả ảnh huấn luyện), sử dụng phương pháp học là “adam”, loss là “categorical_crossentropy”, metrics là “ accuracy”.

Kết quả thực nghiệm được tiến hành trên máy tính cá nhân có cấu hình: Intel(R) Core i3-4030U CPU@ 1.92GHz, RAM 8GB. Thời gian huấn luyện cho mô hình trên khoảng 15 phút, cho tỷ lệ nhận dạng đúng gần 98% ở trên dữ liệu kiểm tra.

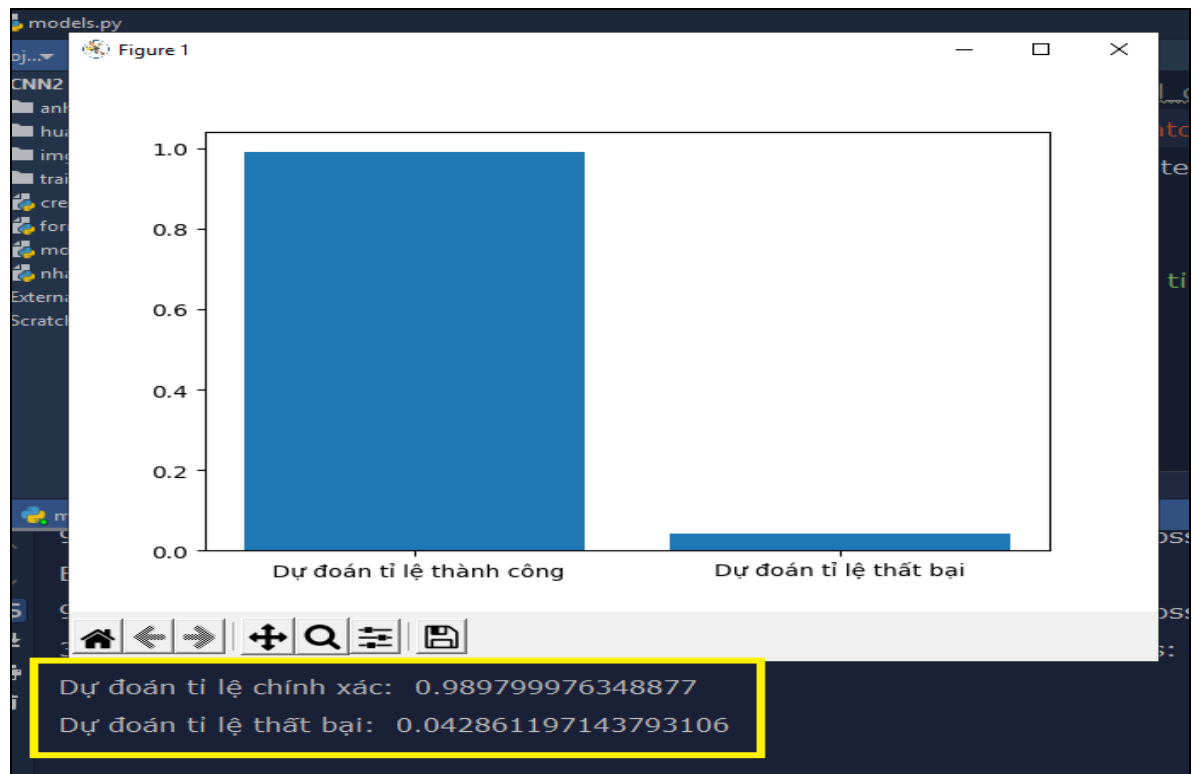
Bên cạnh mô hình khởi tạo ban đầu, các mô hình với các thông số khác nhau sẽ được khảo sát để làm rõ chức năng của từng thông số trong mạng.

3.2 Kết quả nghiên cứu

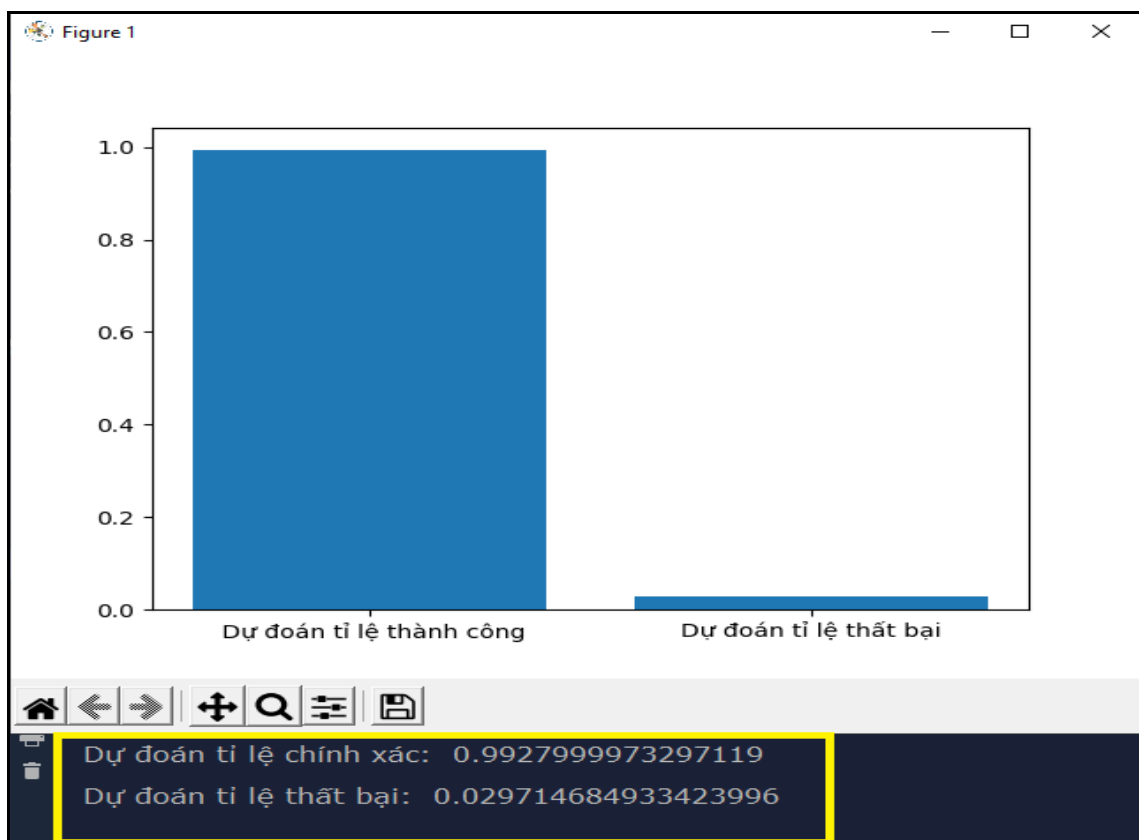
❖ Ảnh hưởng của mini-batch size

Mini-batch size được tạo ra bằng cách chia nhỏ tập dữ liệu huấn luyện và tập kiểm tra thành nhiều phần bằng nhau. Ví dụ tập huấn luyện MNIST có 60000 ảnh, chọn mini-batch size bằng 10 có nghĩa là chia tập huấn luyện thành 5000 tập con, mỗi tập con có 10 ảnh. Số lượng ảnh trong tập con quyết định có bao nhiêu ảnh dùng để huấn luyện cho một lần cập nhật weight, bias. Để hoàn thành một chu kỳ học, mạng CNN sẽ lần lượt chọn lần lượt các tập con để huấn luyện, sau mỗi tập con, các weight và bias sẽ được cập nhật cho đến khi hết số lượng các tập con.

Thực nghiệm với các mini-batch size khác nhau cho kết quả trên tập kiểm tra như hình 23:



Hình 24 Lựa chọn batch_size là 128



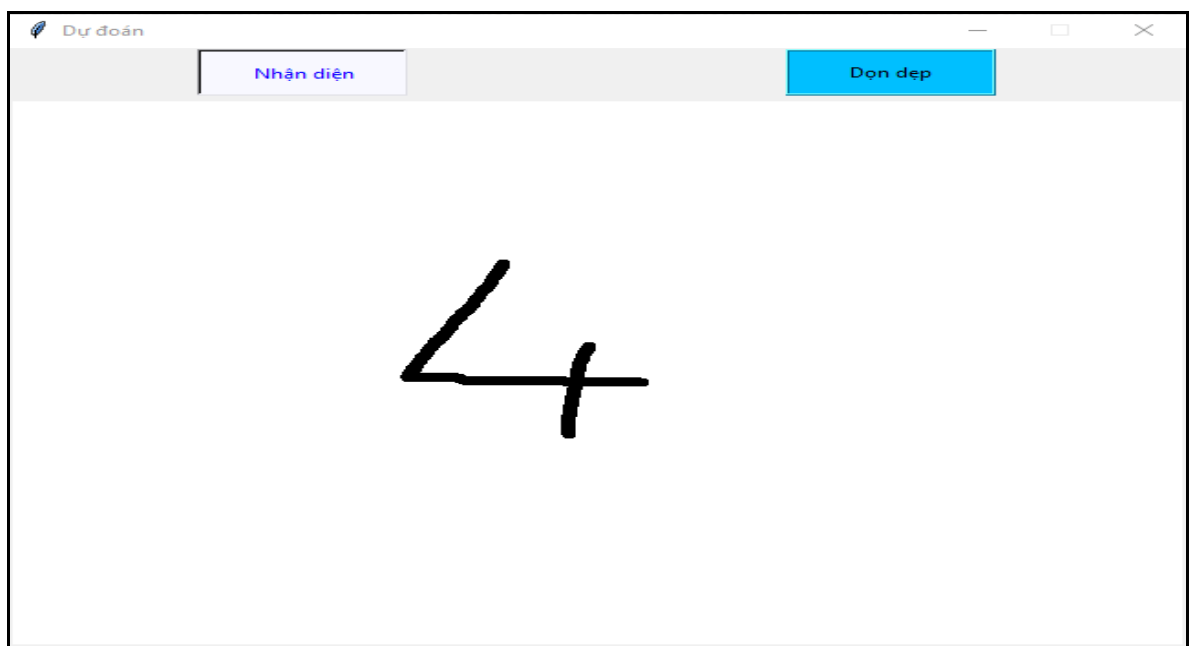
Hình 25 Lựa chọn batch_size là 64

Mini-batch size lớn sẽ mở rộng cơ sở dữ liệu cho quá trình học, tỉ lệ nhận dạng càng thấp nhưng thời gian huấn luyện ngắn hơn, song song với đó, việc chọn các mini-batch size lớn cũng yêu cầu bộ nhớ lớn hơn cho việc tính toán.

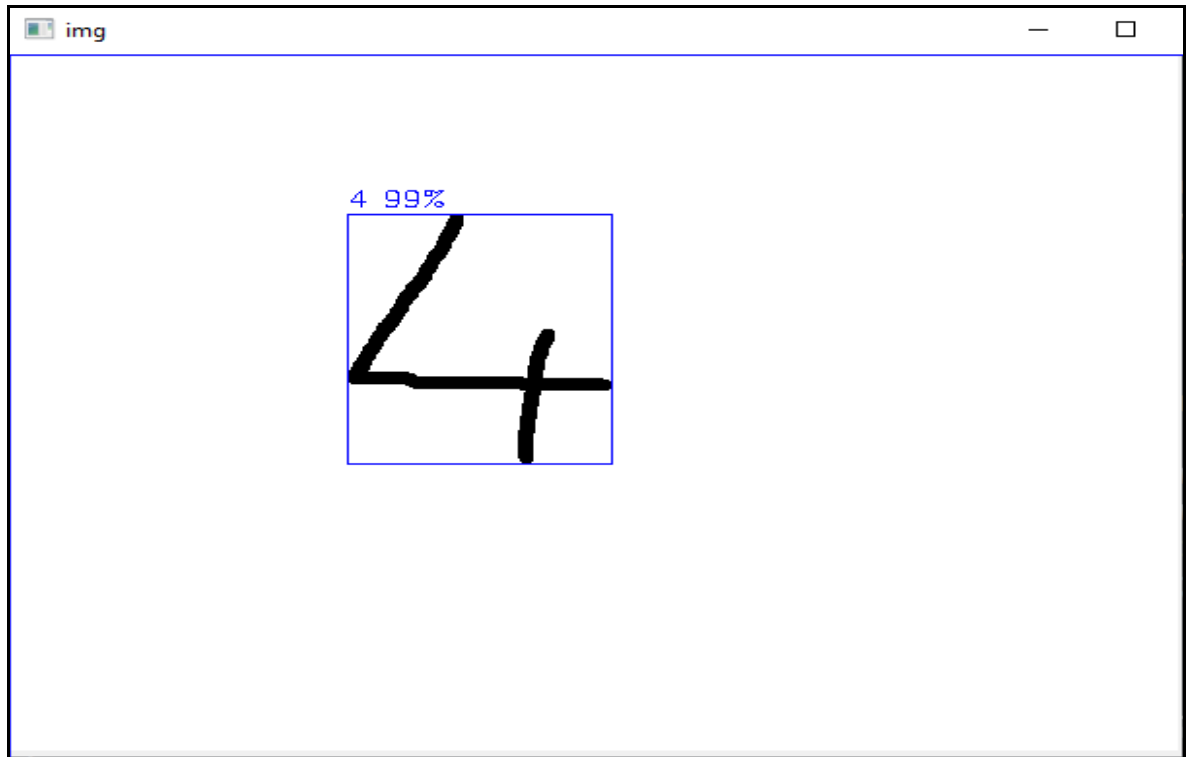
Việc chọn Mini-batch size là 64 tổng thời gian cho 50 lần huấn luyện khoảng đến hơn 20 phút, còn việc chọn Mini-batch size là 128 thì tổng thời gian cho 50 lần huấn luyện rơi vào khoảng 10 phút đến 20 phút. Vì dữ liệu được chia thành nhiều tập con hơn nên Mini-batch size là 64 sẽ duyệt qua tập dữ liệu nhiều lần hơn nên tỉ lệ nhận dạng cao hơn.

Qua đánh giá thực nghiệm, ta thấy được rằng với các mô hình nhỏ vị đơn giản, cập nhật các thông số nhiều lần qua các mini-batch size nhỏ cho kết quả tốt hơn các minibatch size lớn. Ngoài ra, ứng dụng với batch size nhỏ cũng phù hợp là bài toán nhận dạng thực tiễn, bộ não phải đưa ra quyết định sau khi học với một số lượng mẫu khá bé, do đó kết quả tối ưu chính là hướng đến chính là tăng tỷ lệ nhận dạng đúng với số lượng mẫu bé.

Bên dưới là một bài bức ảnh nhận dạng chữ số đã thực nghiệm dựa trên mô hình nhận dạng ký số viết tay sử dụng mạng CNN:



Hình 26 Khung nhận dạng chữ số bằng cách vẽ bằng chuột



Hình 27: Kết quả nhận dạng thành công

3.3 Ưu điểm, nhược điểm của CNN.

❖ **Ưu điểm:**

- Dựa vào nghiên cứu trên ta thấy CNN nhận dạng chữ số với độ chính xác lên đến hơn 97%.

- Khả năng làm việc với kiến thức chưa đầy đủ: Sau khi đào tạo, thông tin có thể tạo ra đầu ra ngay cả với dữ liệu không đủ. Việc mất hiệu suất ở đây phụ thuộc vào tầm quan trọng của việc thiếu dữ liệu.

- Có phân phối bộ nhớ: Đối với CNN là để có thể thích ứng, điều quan trọng là phải xác định các ví dụ và khuyến khích mạng theo đầu ra mong muốn bằng cách trình diễn các ví dụ này cho mạng. Sự kế thừa của mạng tỷ lệ thuận với các trường hợp đã chọn và nếu sự kiện không thể xuất hiện trên mạng theo tất cả các khía cạnh của nó, nó có thể tạo ra kết quả sai.

❖ **Nhược điểm:**

- Cần nhiều dữ liệu: Không giống như bộ não con người, có thể học cách làm

mọi thứ với rất ít ví dụ, mạng nơ-ron cần hàng nghìn và hàng triệu ví dụ.

- Khả năng tổng quát hóa kém: Một mạng nơ-ron sẽ thực hiện chính xác một nhiệm vụ mà nó đã được huấn luyện, nhưng rất kém ở bất kỳ nhiệm vụ nào khác, ngay cả khi nó tương tự như vấn đề ban đầu.

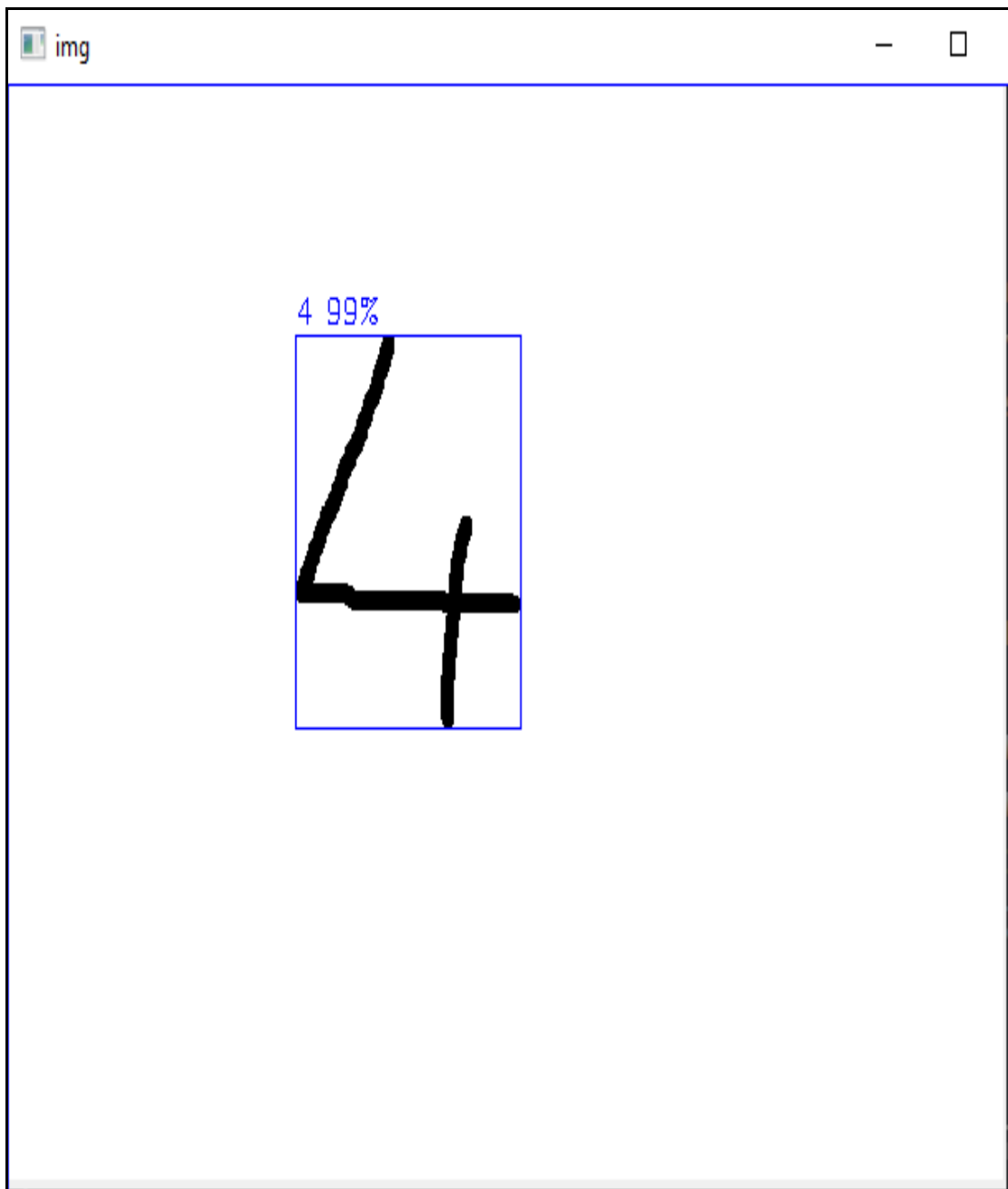
- Mạng nơ-ron không rõ ràng: Vì mạng nơ-ron thể hiện hành vi của chúng về trọng lượng và kích hoạt nơ-ron nên rất khó xác định logic đằng sau các quyết định của chúng.

- Khó khăn khi hiển thị sự cố với mạng: CNN có thể hoạt động với dữ liệu số. Các vấn đề phải được chuyển đổi thành các giá trị số trước khi được đưa vào CNN. Cơ chế trình bày được giải quyết ở đây sẽ tác động trực tiếp đến hiệu suất của mạng. Nó phụ thuộc vào khả năng của người dùng.

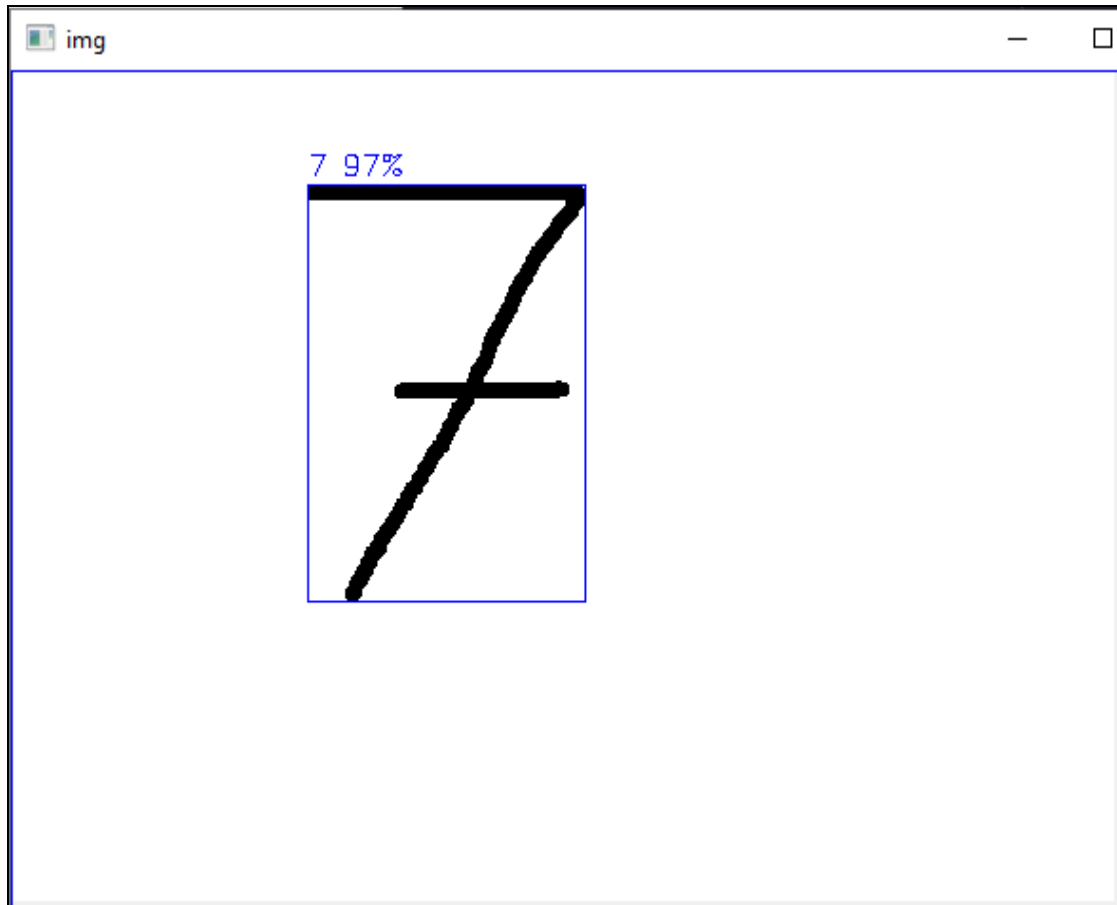
CHƯƠNG 4: KẾT LUẬN

Nghiên cứu trên đã trình bày những thuật ngữ cơ bản, cấu tạo mạng CNN (Convolutional Neural Network) cũng như cách mà mạng nơ-ron tích chập “học” các thông số để xây dựng mô hình nhận dạng ký số viết tay. Xây dựng thành công mô hình có thể nhận dạng ký số viết tay.

Một vài hình ảnh chứng minh:



Hình 28: Nhận dạng thành công số 4



Hình 29: Nhận dạng thành công số 7

Qua kết quả nghiên cứu, ta còn nhận thấy mạng CNN, để nâng cao độ chính xác của mô hình huấn luyện cần phải lựa chọn các thông số sao cho phù hợp. Ví dụ: như Mini-batch size cần phải lựa chọn thông số phù hợp, việc chọn Mini-batch size là 64 tổng thời gian cho 50 lần huấn luyện khoảng đến hơn 20 phút, còn việc chọn Mini-batch size là 128 thì tổng thời gian cho 50 lần huấn luyện rơi vào khoảng 10 phút đến 20 phút. Vì dữ liệu được chia thành nhiều tập con hơn nên Mini-batch size là 64 sẽ duyệt qua tập dữ liệu nhiều lần hơn nên tỉ lệ nhận dạng cao hơn.

CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

Đưa CNN (Convolutional Neural Network) trở thành hệ thống nơ-ron nhân tạo cấp độ con người, giảm số lần huấn luyện (Convolutional Neural Network) nhưng độ chính xác tăng lên.

Có một số nỗ lực để vượt qua giới hạn của mạng nơ-ron nhân tạo, chẳng hạn như một sáng kiến do DARPA[5] tài trợ nhằm tạo ra các mô hình AI có thể giải thích được . Những phát triển thú vị khác bao gồm phát triển các mô hình lai kết hợp mạng nơ-ron nhân tạo và AI dựa trên quy tắc để tạo ra các hệ thống AI có thể diễn giải được và yêu cầu ít dữ liệu đào tạo hơn.

Mặc dù chúng ta vẫn còn một chặng đường dài trước khi đạt được mục tiêu về AI cấp độ con người (nếu chúng ta sẽ đạt được nó), mạng thần kinh đã đưa chúng ta đến gần hơn nhiều.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Thông tin xu hướng công nghệ thông tin năm 2022 tham khảo tại “<https://br.atsit.in/vi/?p=123913>”, truy cập ngày 01/07/2022.
- [2] Trang chủ thư viện MNIST “ <http://yCNN.lecun.com/exdb/mnist/index.html>”, truy cập ngày 01/07/2022.
- [3] Dave Anderson, George McNeill (2006), Artificial Neural Networks Technology, Prepared for Rome Laboratory RL/C3C Griffiss AFB, NY 13441-5700, USA.
- [4] Khái niệm học chuyển tiếp, Website: <https://bdtechtalks.com/2019/06/10/what-is-transfer-learning/>, truy cập ngày 04/07/2022.
- [5]: Sáng kiến do DARPD tài trợ, <https://bdtechtalks.com/2019/01/10/darpa-xai-explainable-artificial-intelligence/> , truy cập ngày 05/07/2022