

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

LÊ THỊ THU HẰNG

NGHIÊN CỨU VỀ MẠNG NEURAL  
TÍCH CHẬP VÀ ỨNG DỤNG CHO BÀI  
TOÁN NHẬN DẠNG BIẾN SỐ XE

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

HÀ NỘI, 2016

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

LÊ THỊ THU HẰNG

**NGHIÊN CỨU VỀ MẠNG NEURAL  
TÍCH CHẬP VÀ ỨNG DỤNG CHO BÀI  
TOÁN NHẬN DẠNG BIỂN SỐ XE**

*Ngành* : Công nghệ thông tin  
*Chuyên ngành* : Kỹ thuật phần mềm  
*Mã số* : 60480103

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

*Người hướng dẫn khoa học:* TS. Nguyễn Văn Vinh

HÀ NỘI, 2016

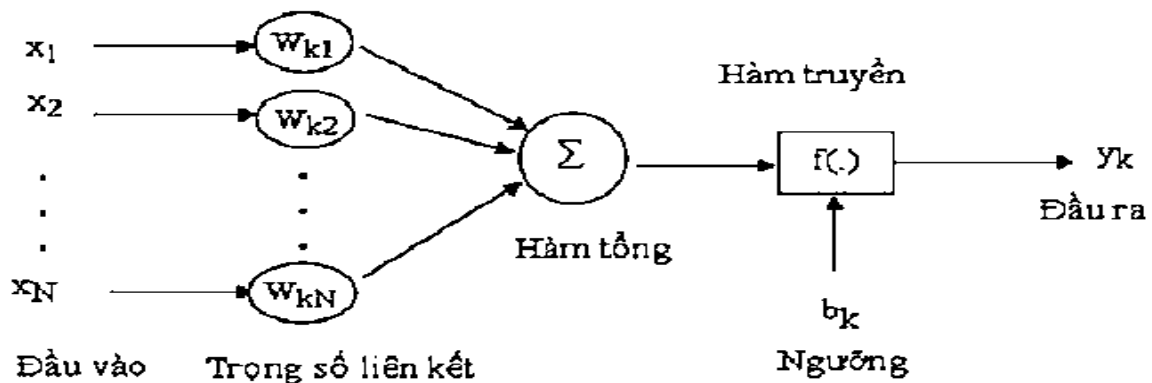
# CHƯƠNG 1: MẠNG NEURAL VÀ MẠNG NEURAL TÍCH CHẬP

## 1. Mạng neural và mạng neural lan truyền ngược

### 1.1. Giới thiệu về mạng Neural

**Định nghĩa:** 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. Cấu trúc neural nhân tạo:

Hình 1.1. Cấu tạo một Neural



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 được thể hiện bởi một trọng số liên kết – Synaptic weight.
- 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.

Xét về mặt toán học, cấu trúc của một nơron  $k$ , được mô tả bằng cặp biểu thức sau:

$$u_k = \sum_{j=1}^p w_{kj} x_j$$

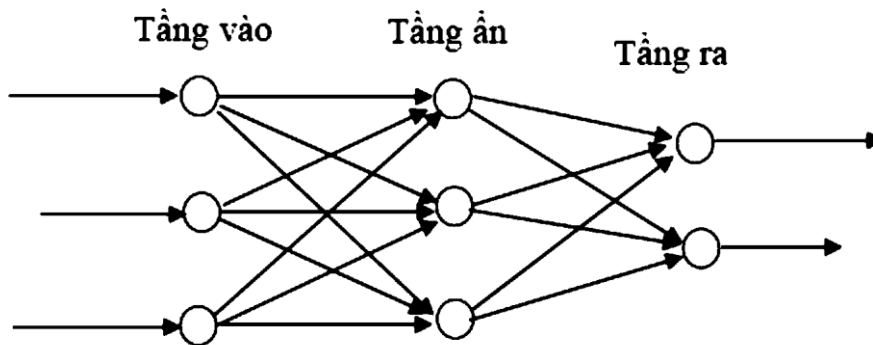
và  $y_k = f(u_k - b_k)$

Trong đó:  $x_1, x_2, \dots, x_p$ : là các tín hiệu vào;  $(w_{k1}, w_{k2}, \dots, w_{kp})$  là các trọng số liên kết của nơron thứ  $k$ ;  $u_k$  là hàm tổng;  $b_k$  là một ngưỡng;  $f$  là hàm truyền và  $y_k$  là tín hiệu đầu ra của nơron.

### 1.2. Một số kiểu mạng Neural

- ◆ Tự kết hợp (*autoassociative*)
- ◆ Kết hợp khác kiểu (*heteroassociative*)
- ◆ Kiến trúc truyền thẳng (*feedforward architecture*)

Hình 1.4. Mạng truyền thẳng

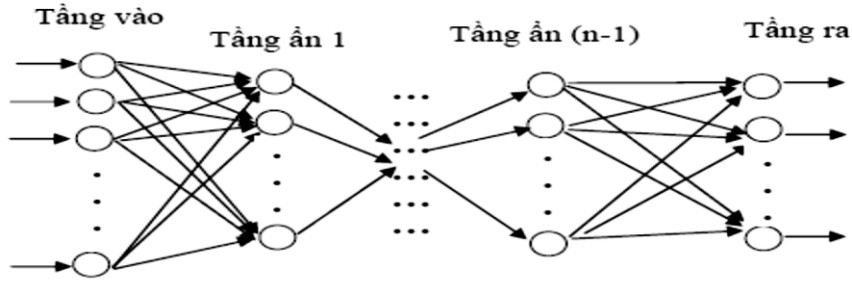


- ◆ Kiến trúc phản hồi (*Feedback architecture*): Hình 1.5. Mạng phản hồi

### 1.3. Mạng Neural lan truyền ngược MLP

#### a. Kiến trúc mạng MLP

Hình 1.6. Mạng neural lan truyền ngược MLP



Kiến trúc của một mạng MLP tổng quát có thể mô tả như sau:

◆ Đầu vào là các vector  $(x_1, x_2, \dots, x_p)$  trong không gian  $p$  chiều, đầu ra là các vector  $(y_1, y_2, \dots, y_q)$  trong không gian  $q$  chiều. Mỗi nơron thuộc tầng sau liên kết với tất cả các nơron thuộc tầng liền trước nó. Đầu ra của nơron tầng trước là đầu vào của nơron thuộc tầng liền sau nó.

b. Huấn luyện mạng MLP:

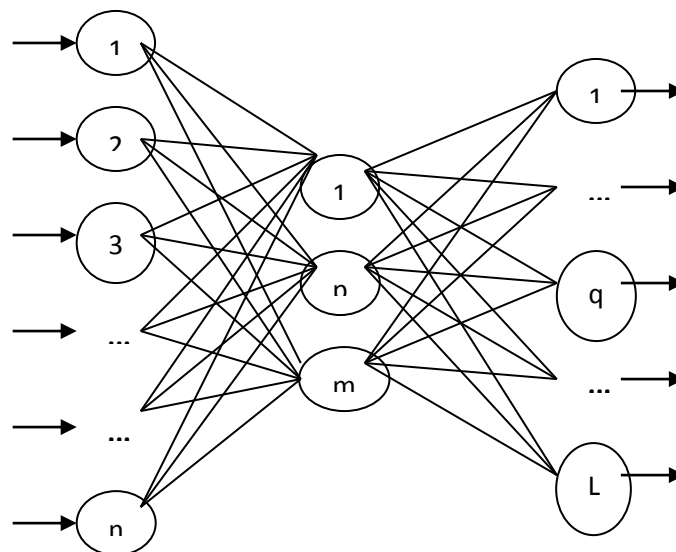
+ Các phương pháp học:

- ◆ Học có giám sát
- ◆ Học không giám sát
- ◆ Học tăng cường

c. Học có giám sát trong các mạng nơron

d. Mạng lan truyền ngược <Back Propagation Network>

Mô hình mạng: Mạng neural lan truyền ngược có mô hình như sau



Mạng có 3 lớp:

- Lớp vào (Input Layer) – số node vào là số thuộc tính của đối tượng cần phân lớp.
- Lớp ra (Output Layer) – Số node ra là số đặc điểm cần hướng tới của đối tượng (giá trị ra cần hướng đến – học có giám sát).
- Lớp ẩn (Hidden Layer) – Số node ẩn thường là không xác định trước, nó thường là do kinh nghiệm của người thiết kế mạng, nếu số node ẩn quá nhiều mạng sẽ công kênh, quá trình học sẽ chậm, còn nếu số node ẩn quá ít làm mạng học không chính xác.

e) Một số vấn đề lưu ý trong xây dựng mạng MLP.

Xác định kích thước mẫu:

Hai yếu tố quan trọng ảnh hưởng đến kích thước mẫu:

- Dạng hàm đích: khi hàm đích càng phức tạp thì kích thước mẫu cần tăng.
- Nhiễu: khi dữ liệu bị nhiễu (thông tin sai hoặc thiếu thông tin) kích thước mẫu cần tăng.

## **2. Mạng neural tích chập**

### **2.1. Định nghĩa mạng neural tích chập**

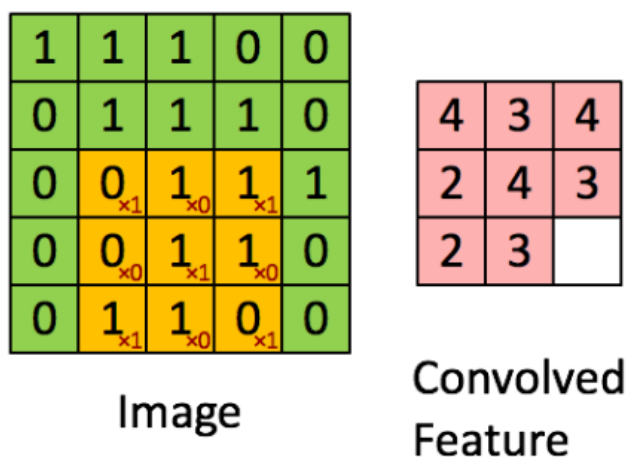
Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Trong luận văn này, chúng ta sẽ trình bày về Convolution (tích chập) cũng như ý tưởng của mô hình CNNs trong phân lớp chữ viết áp dụng trong bài toán nhận dạng biển số xe (Image Classification).

### **2.2. Convolution (tích chập)**

Tích chập được sử dụng đầu tiên trong xử lý tín hiệu số (Signal processing). Nhờ vào nguyên lý biến đổi thông tin, các nhà khoa học đã áp dụng kỹ thuật này vào xử lý ảnh và video số.

Để dễ hình dung, ta có thể xem tích chập như một cửa sổ trượt (sliding window) áp đặt lên một ma trận. Bạn có thể theo dõi cơ chế của tích chập qua hình minh họa bên dưới.

Hình 1.8. Minh họa tích chập

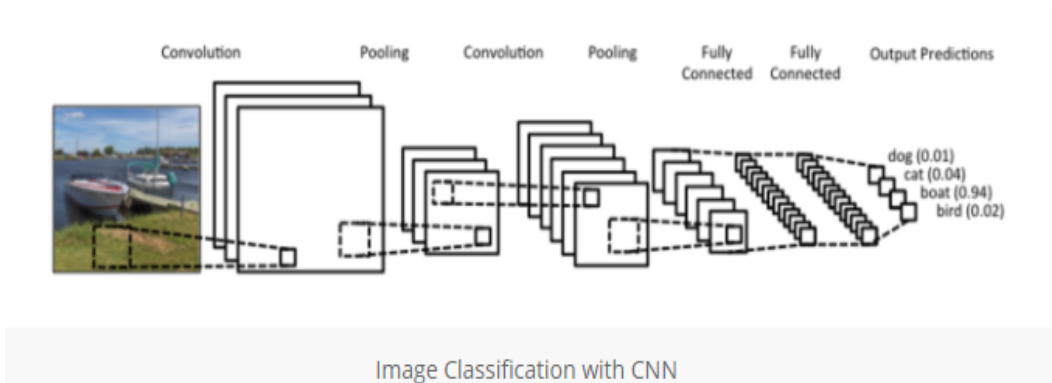


Ma trận bên trái là một bức ảnh đen trắng. Mỗi giá trị của ma trận tương đương với một điểm ảnh (pixel), 0 là màu đen, 1 là màu trắng (nếu là ảnh grayscale thì giá trị biến thiên từ 0 đến 255).

### 2.3. Mô hình mạng neural tích chập

Trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi nơ-ron ở layer tiếp theo sinh ra từ filter áp đặt lên một vùng ảnh cục bộ của nơ-ron layer trước đó.

Hình 1.11. Mô hình mạng neural tích chập



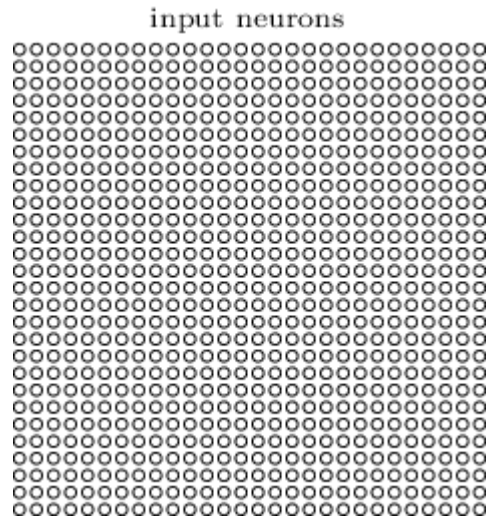
CNNs có tính bất biến và tính kết hợp cục bộ (Location Invariance and Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling).

## 2.4. Xây dựng mạng neural tích chập

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.

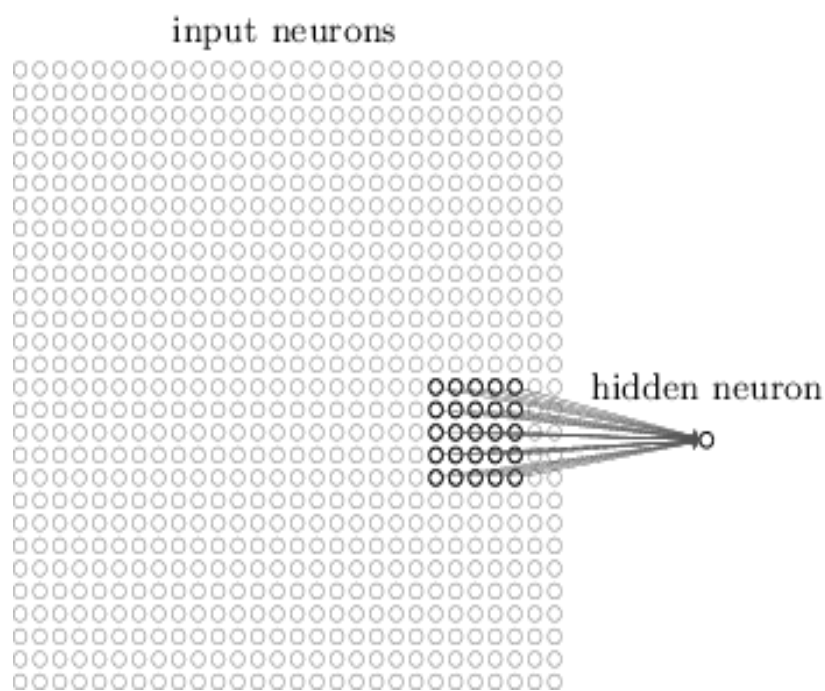
Trường tiếp nhận cục bộ (**Local receptive fields**): Trong các tầng kết nối đầy đủ được chỉ ra trước đây, đầu vào đã được mô tả là một đường thẳng đứng chứa các neuron. Trong mạng tích chập, ta sẽ thay thế các đầu vào là  $28 \times 28$  neuron, giá trị tương ứng với  $28 \times 28$  cường độ điểm ảnh mà chúng ta sử dụng:





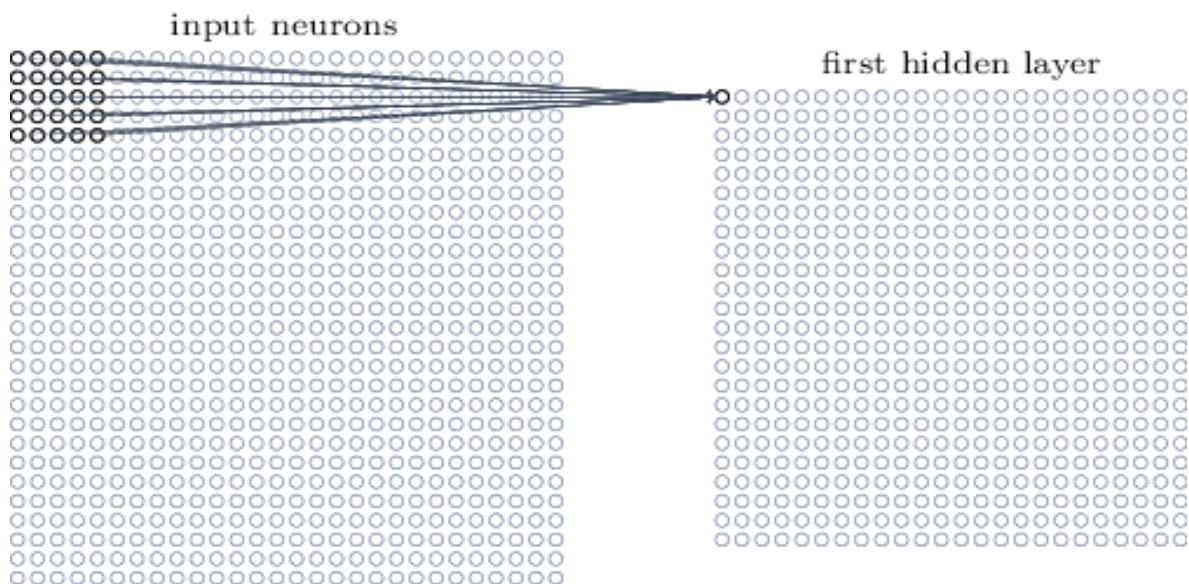
Như thường lệ chúng ta sẽ kết nối các điểm ảnh đầu vào cho các nơron ở tầng ẩn. Nhưng chúng ta sẽ không kết nối mỗi điểm ảnh đầu vào cho mỗi neuron ẩn. Thay vào đó, chúng ta chỉ kết nối trong phạm vi nhỏ, các vùng cục bộ của bức ảnh.

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



Vùng đó trong bức ảnh đầu vào được gọi là vùng tiếp nhận cục bộ cho nơron ẩn. Đó là một cửa sổ nhỏ trên các điểm ảnh đầu vào. Mỗi kết nối sẽ học một trọng số. Và nơron ẩn cũng sẽ học một độ lệch (overall bias). Bạn có thể hiểu rằng nơron lớp ẩn cụ thể là 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 nơron 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:



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

Trọng số và độ lệch (Shared weights and biases) : Mỗi một neuron ẩn có một độ lệch (bias) và  $5 \times 5$  trọng số liên kết với trường tiếp nhận cục bộ. Những gì chúng ta vẫn chưa đề cập đến là chúng ta sẽ sử dụng các trọng số và độ lệch tương tự cho mỗi nơron ẩn  $24 \times 24$ . Nói cách khác, đối với những neuron ẩn thứ  $j, k$ , đầu ra là:

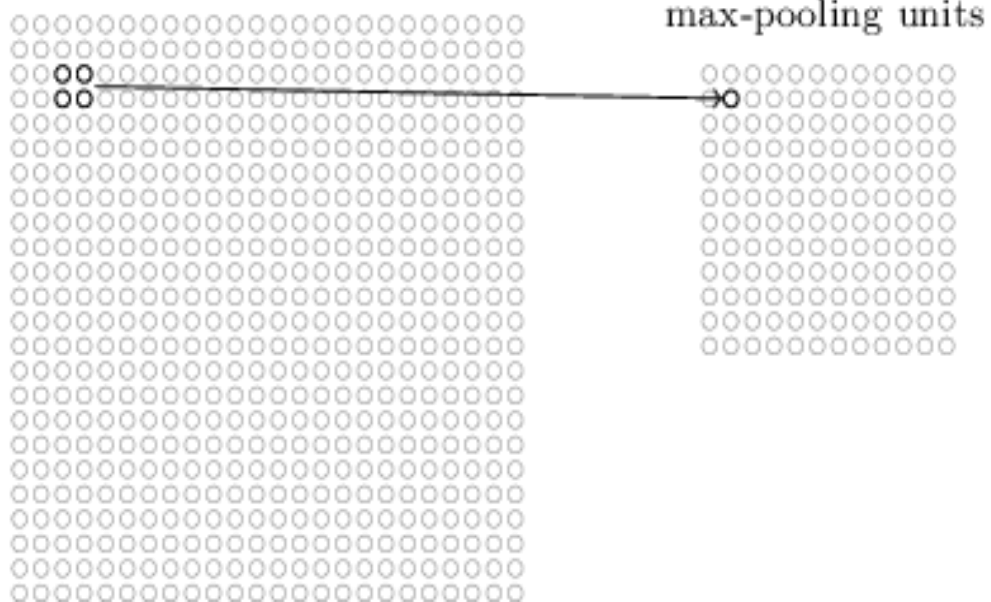
$$\sigma \left( b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l,k+m} \right).$$

Ở đây,  $\sigma$  là hàm kích hoạt neuron - có lẽ là hàm sigmoid chúng ta sử dụng trong các chương trước.  $b$  là giá trị chung cho độ lệch.  $w_{l,m}$ ,  $m$  là một mảng  $5 \times 5$  của trọng số chia sẻ. Và, cuối cùng, chúng ta sử dụng  $a_{xy}$  biểu thị giá trị kích hoạt đầu vào tại vị trí  $x, y$ .

**Lớp chứa hay lớp tổng hợp (Pooling layer):** Ngoài các lớp tích chập vừa mô tả, mạng nơron 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.

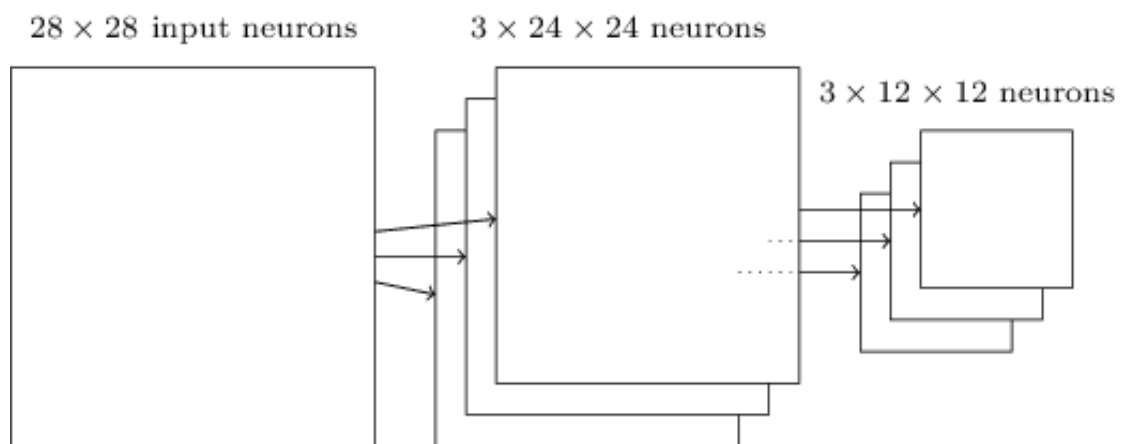
Ví dụ, mỗi đơn vị trong lớp pooling có thể thu gọn một vùng  $2 \times 2$  nơron trong lớp trước. Một thủ tục pooling phổ biến là max-pooling. Trong max-pooling, một đơn vị pooling chỉ đơn giản là kết quả đầu ra kích hoạt giá trị lớn nhất trong vùng đầu vào  $2 \times 2$ , như minh họa trong sơ đồ sau:

hidden neurons (output from feature map)



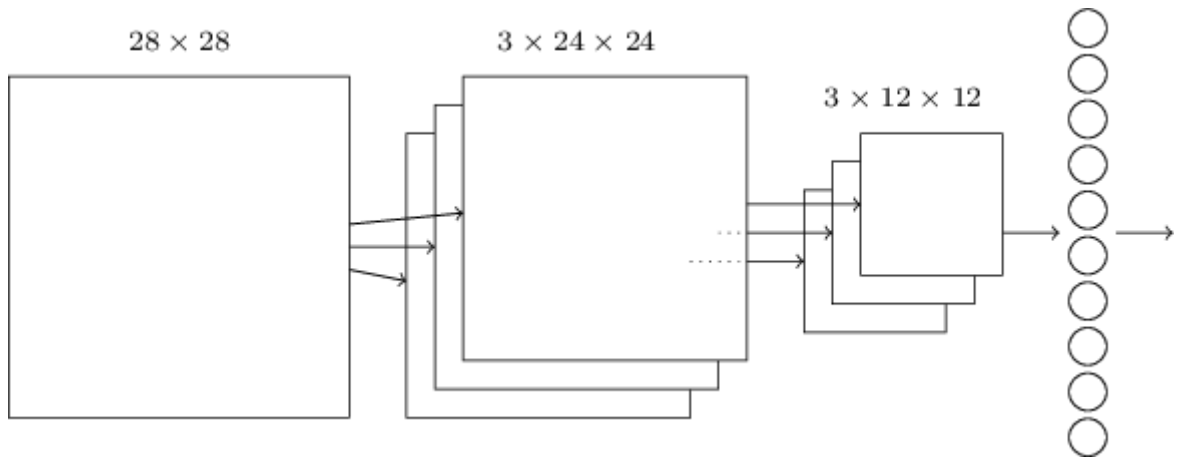
Lưu ý rằng bởi vì chúng ta có  $24 \times 24$  nơon đầu ra từ các lớp tích chập, sau khi pooling chúng ta có  $12 \times 12$  nơon.

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 sau:



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. Trực 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.

**Đặt tất cả chúng lại với nhau (Putting it all together):** Bây giờ 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 nhìn vào, nhưng có thêm một lớp 10 nơon đầu ra, tương ứng với 10 giá trị có thể cho các số MNIST ('0', '1', '2', v.v...):



Lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối. Đó là, lớp này nối mọi nơron từ lớp max-pooled tới mọi nơron của tầng ra. Kiến trúc kết nối đầy đủ này cũng giống như chúng ta sử dụng trong các chương trước.

## **CHƯƠNG 2: TỔNG QUAN VỀ NHẬN DẠNG BIỂN SỐ XE**

### **2.1 Khái niệm về hệ thống nhận dạng biển số xe.**

#### **2.1.1 Khái niệm**

Hệ thống nhận dạng biển số xe là hệ thống có khả năng phân tích hình ảnh và xác định vùng chứa biển số trên xe, thông qua video, thiết bị ghi hình và hình ảnh.

#### **2.1.2 Lịch sử và phát triển.**

Năm 1992, công nghệ ALPR (Automatic License Plate Number) hay còn gọi là tự động nhận dạng biển số xe, được phát triển tại Đại học Cambridge ở Vương quốc Anh để ứng phó với chủ nghĩa khủng bố.

Đến năm 1996, công nghệ ALPR đã được hoàn thiện tại mỗi cổng phía tây Vương quốc Anh để đọc tất cả các biển đăng ký xe từ Ireland. Công nghệ ALPR tiếp tục được nghiên cứu và phát triển tại Anh. Kể từ tháng ba năm 2006, hầu hết các con đường, các trung tâm thị trấn, cảng, trạm xăng của London đã được lắp đặt camera chạy phần mềm ALPR.

Trên thế giới hiện nay, bài toán nhận dạng biển số xe được nghiên cứu và phát triển một cách sâu rộng. Nhiều tác giả với các công trình nghiên cứu được công bố với tỉ lệ nhận dạng ngày càng chính xác.

#### **2.1.3. Cách thức hoạt động của hệ thống nhận dạng biển số xe.**

Camera thu nhận ảnh được đặt tại một vị trí cố định sao cho có thể quét được hình ảnh xe một cách rõ ràng và chụp lại hình ảnh đối tượng xe có chứa biển số. Ảnh này được đưa vào phần mềm nhận dạng để trích ra chính xác biển số xe có trong ảnh, sau đó một thuật toán OCR (Optical Character Recognition) được sử dụng để lấy từng ký tự và chuyển đổi thành định dạng mà máy tính có thể phân biệt được các chữ và số như dạng text...

#### **2.1.4. Phân loại các ứng dụng nhận dạng biển số xe.**

##### ***Loại 1: Giới hạn vùng nhìn***

Đầu vào: Ảnh thu trực tiếp từ các thiết bị ghi nhận ảnh kỹ thuật số. Ảnh được ghi nhận thường chỉ giới hạn trong vùng có biển số xe.

Nguyên lý hoạt động: Các phương tiện giao thông phải chạy với một tốc độ đủ chậm để máy ghi nhận hình ảnh có thể thu được ảnh vùng biển số xe.

Ứng dụng: Những ứng dụng nhận dạng biển số xe loại này thường được dùng tại các trạm kiểm soát, các trạm thu phí, các bãi gửi xe tự động, các trạm gác cổng.

### ***Loại 2: Không giới hạn vùng nhìn***

Đầu vào: Ảnh đầu vào thu được từ các thiết bị ghi hình tự động, không phụ thuộc vào góc độ, các đối tượng xung quanh, ảnh không cần bắt buộc chỉ chụp vùng chứa biển số xe, mà có thể ảnh tổng hợp như chứa thêm các đối tượng như người, cây đường phố..., miễn là vùng biển số xe phải đủ rõ để có thể thực hiện nhận dạng được ký tự trong vùng đó.

#### **2.1.5. Ứng dụng thực tiễn tại Việt Nam**

Thu phí giao thông, kiểm soát xe tại các đường biên giới, các trạm gác cổng.

##### **2.1.6.1. Quy định về màu sắc và các ký tự trên biển số.**

- Biển trắng chữ đen dành cho dân sự.
- Màu trắng 2 chữ, 5 số là biển dành cho người nước ngoài.
- NG là xe ngoại giao.
- NN là xe của các tổ chức, cá nhân nước ngoài: Trong đó 3 số ở giữa là mã quốc gia, 2 số tiếp theo là số thứ tự.

#### **2.2. Phương pháp nhận dạng biển số xe từ ảnh chụp của camera.**

Có nhiều phương pháp để giải quyết vấn đề này nhưng đều quy về các phương pháp chính sau đây:

- Phương pháp dùng chuyển đổi Hough, phương pháp hình thái học, phương pháp khớp mẫu.

##### **2.2.1. Phương pháp chuyển đổi Hough.**

##### **2.2.2. Phương pháp hình thái học.**

### 2.3. Phương pháp nhận dạng ký tự trong biển số xe.

Phương pháp phổ biến nhất để nhận dạng ký tự là sử dụng mạng nơron (hoặc SVM, K-NN,...), tức là huấn luyện cho máy tính để nhận dạng các ký tự. Tuy nhiên do số lượng ký tự trên biển số là không nhiều nên để đảm bảo tốc độ xử lý, chúng ta cũng có thể sử dụng phương pháp Hình thái học để giải quyết khâu này bởi vì các ký tự đều có những đặc điểm hình thái đặc biệt có thể phân biệt với nhau chẳng hạn như “0” có lỗ trống ở giữa, “8” có 2 lỗ trống hay “X” đối xứng 2 trục dọc và ngang... Khâu này được thực hiện trên cơ sở xây dựng cây nhị phân tối ưu của các đặc điểm hình thái nên đảm bảo tính khoa học và tính chính xác cao.

### 2.4. Phạm vi nghiên cứu và hướng giải quyết.

Một số đặc điểm để nhận dạng biển số xe tại Việt Nam.

#### a) Tiêu chuẩn về kích thước (theo quy định của Bộ Giao Thông Vận Tải)

- Biển ô tô
  - Chiều cao: 110 mm. Chiều rộng: 470 mm (biển dài).
  - Chiều cao: 200 mm. Chiều rộng: 280 mm (biển vuông).
- Biển xe máy:
  - Chiều cao: 140 mm. Chiều rộng: 190 mm.

Như vậy, tỉ lệ Chiều cao / Chiều rộng là:

- $0.18 < \text{Chiều cao}/\text{Chiều rộng} < 0.3$  (biển số có 1 hàng).
- $0.6 < \text{Chiều cao}/\text{Chiều rộng} < 0.85$  (biển số xe có 2 hàng).

Từ các đặc điểm này, ta có thể xác định được vùng nào có khả năng là biển số theo ràng buộc về kích thước.

#### b) Tiêu chuẩn về ký tự.

Theo đo đạc trên biển số thực tế, mỗi ký tự thường có tỷ lệ kích thước về chiều rộng, chiều cao tương ứng với chiều dài và rộng của biển số xe. Ví dụ, chiều cao của mỗi ký tự luôn nhỏ hơn 85% chiều cao của biển số xe và luôn lớn hơn 60% chiều cao của biển xe đối với biển số xe có một hàng, với biển số xe có hai hàng thì chiều cao mỗi ký tự không quá 50% chiều cao của



biển số xe. Chiều rộng của ký tự thường không lớn hơn 20% chiều cao của mỗi ký tự. Mỗi ký tự của biển số xe được xem như là một vùng liên thông con hay contour con (bao đóng). Do đó, chúng ta có thể đếm số contours con thỏa mãn tính chất đó là ký tự. Ở nước ta chỉ có số ký tự trên mỗi biển số xe nằm trong khoảng 6 đến 9 ký tự.

Từ những phân tích trên, có thể tóm tắt phương pháp thực hiện của chương trình nhận dạng như sau:

**Bước 1:** Ảnh đầu vào là ảnh màu BGR, tiền xử lý bằng các thuật toán xử lý ảnh.

**Bước 2:** Tìm các contour trên ảnh xe. (Mỗi contour là 1 vùng bao kín, do vùng biển số là 1 vùng bao kín nên sẽ tương ứng với một contour)

**Bước 3:** Lọc các contour theo các tiêu chí như kích thước, góc, tỉ lệ, số kí tự,...

**Bước 4:** Xử lý kết quả đầu ra để lấy vùng biển số

**Bước 5:** Tách ký tự trên vùng biển số tìm được

**Bước 6:** Đưa tập ký tự đã tách vào mạng Neural để nhận dạng

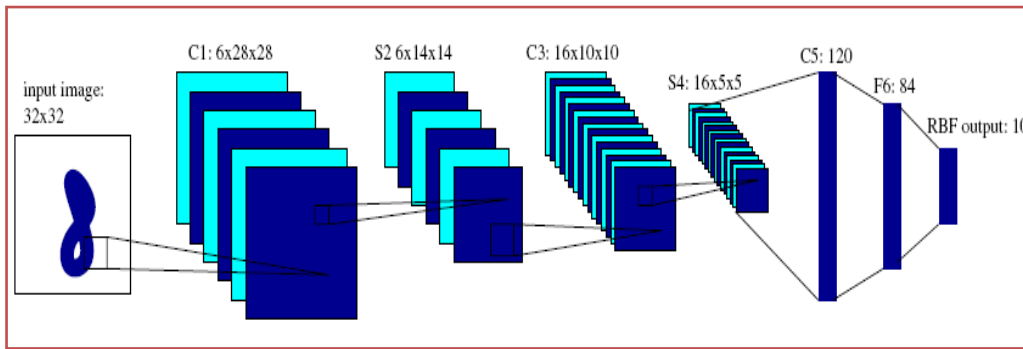
**Bước 7:** Hiện thị kết quả lên giao diện chương trình

## **CHƯƠNG 3: ÁP DỤNG MẠNG NEURAL TÍCH CHẬP TRONG NHẬN DẠNG KÝ TỰ**

### **3.1. Xây dựng mô hình mạng**

Mô hình mạng neural tích chập được xây dựng để nhận dạng các ký tự như sau:

Hình 3.1. Mô hình mạng neural tích chập trong nhận dạng ký tự viết tay



Ảnh đầu vào là 1 bức ảnh thô kích thước 32x32 pixel. Chúng ta sử dụng 6 ma trận chập kích thước 5x5 cho ra 6 ma trận ảnh đặc trưng sau khi chập lần 1 đó là các ma trận ánh xạ đặc trưng ở tầng chập C1, mỗi ma trận ánh xạ đặc trưng này có kích thước 28x28. Tức là ảnh gốc ban đầu được phân tích theo 6 chiều đặc trưng khác nhau với ma trận chập 5x5

Do kích thước các ảnh đặc trưng ở tầng chập C1 có kích thước 28x28 còn lớn, cho nên bước tiếp theo chúng ta thực hiện phép giảm số chiều ở ma trận đặc trưng (down Sampling – hay Pooling – hay subsampling) với hệ số tỷ lệ là 2 sử dụng hàm max:

Ví dụ 2 ô cạnh nhau trong ma trận đặc trưng có giá trị 3,5 tạo thành 1 ô trong ma trận đặc trưng subsampling là :  $\max(3,5)=5$

Như vậy với 6 ma trận đặc trưng kích thước 28x28 ở tầng chập C1 ta tạo được 6 ma trận kích thước 14x14 ở tầng subsampling (S2)

Tiếp tục sử dụng 16 ma trận chập kích thước 5x5 chập với các ma trận ở tầng S2 ta được 16 ma trận ánh xạ đặc trưng kích thước 10x10 ở tầng chập C3

Do kích thước các ảnh đặc trưng ở tầng chập C3 có kích thước 14x14 còn lớn, cho nên bước tiếp theo chúng ta thực hiện phép giảm số chiều ở ma trận đặc trưng (down Sampling – hay Pooling – hay subsampling) với hệ số tỷ lệ là 2 sử dụng hàm max. Kết quả với 16 ma trận đặc trưng kích thước 10x10 ở tầng chập C3 ta tạo được 16 ma trận kích thước 5x5 ở tầng subsampling (S4)

Tiếp tục sử dụng 120 ma trận chập kích thước 5x5 chập với các ma trận ở tầng S4 ta được 120 ma trận ánh xạ đặc trưng kích thước 1x1 ở tầng chập C5

Do các đặc trưng ở tầng chập C5 là các điểm đặc trưng 1x1, cho nên ta không thực hiện phép toán subsampling nữa.

Tiếp theo ta sử dụng phép toán max để giảm kích thước ở tầng chập C5 do tầng C5 có tới 120 node đặc trưng, ta dùng hàm max giảm xuống còn 84 node ở tầng F6.

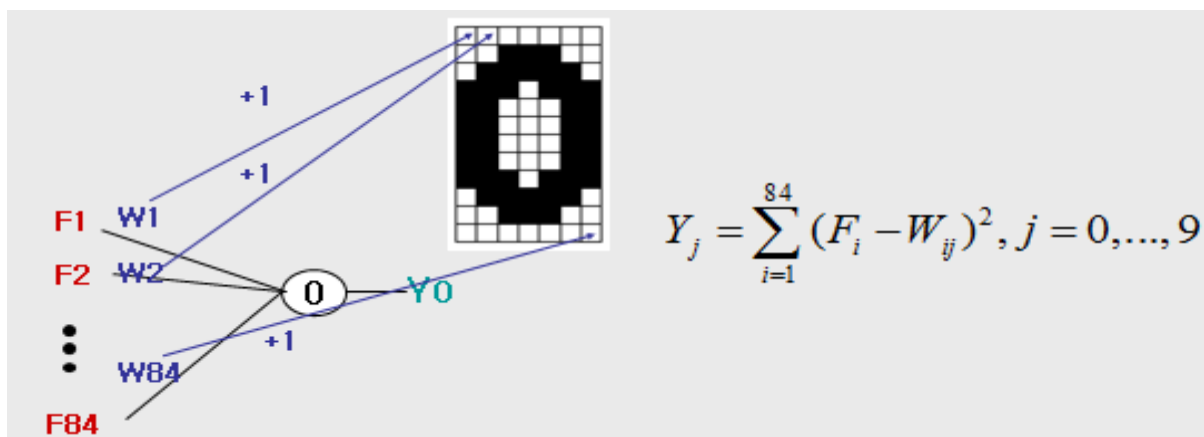
Lưu ý rằng đối với mỗi khối ở các tầng từ đầu tới F6 ta sử dụng hàm kích hoạt Sigmoid dạng :

$$f(x) = \frac{1}{1 + e^{-x}}$$

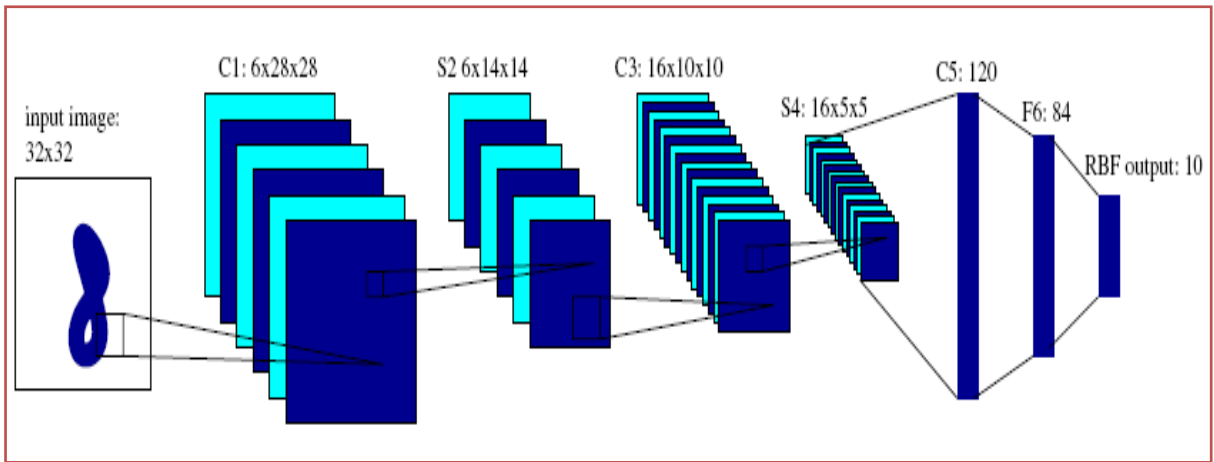
để tính toán giá trị ra của các node mạng, tức là  $y' = f(A * I)$  trong đó A là ảnh chập, I là ma trận chập, và  $y'$  là giá trị 1 node trên ma trận ánh xạ đặc trưng

Với 84 node ở tầng F6, ở đây sử dụng mô hình mạng neural truyền thẳng với các kết nối Fully Connection, với 10 outputs được thiết kế như sau:

Hình 3.2. Minh họa Fully Connection

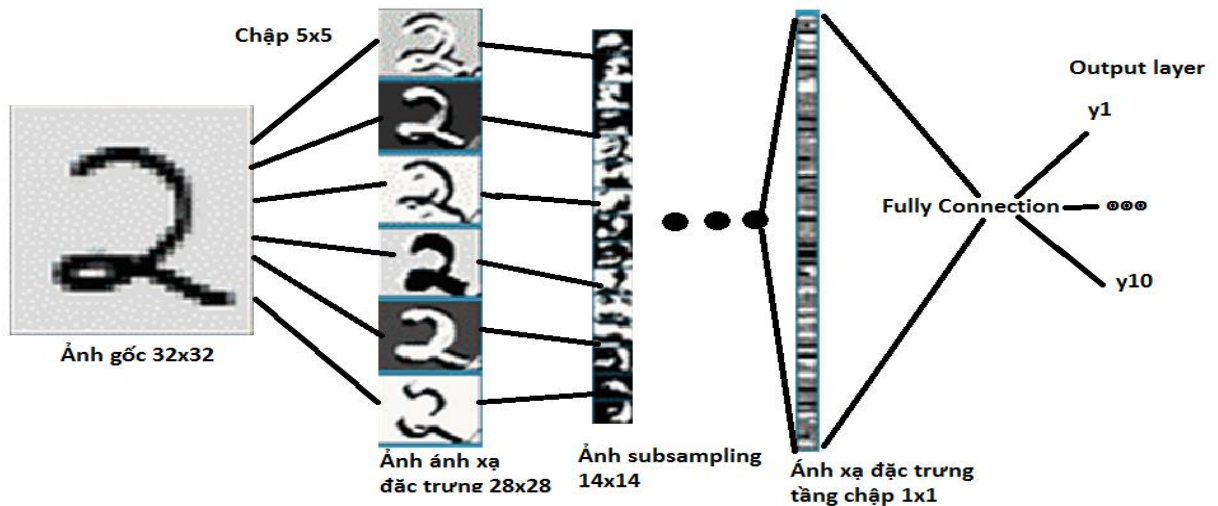


Với mỗi giá trị ra được tính như công thức trên



Ta có thể hình dung mô hình nhận dạng chữ viết được thực hiện như sau:

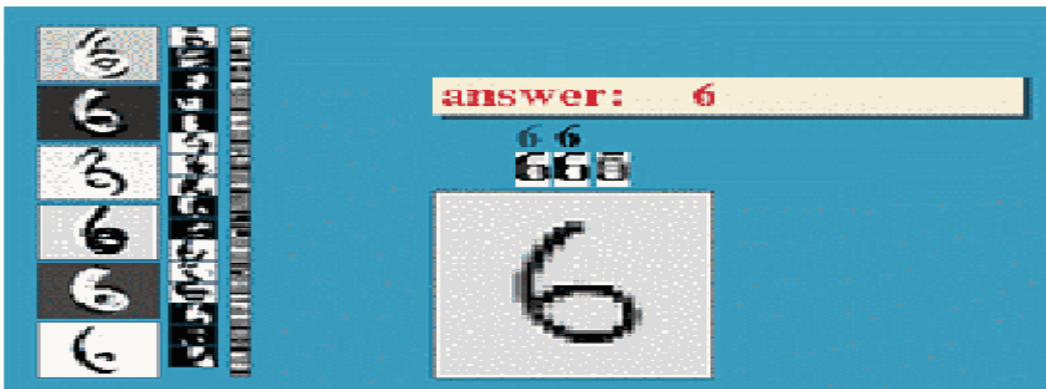
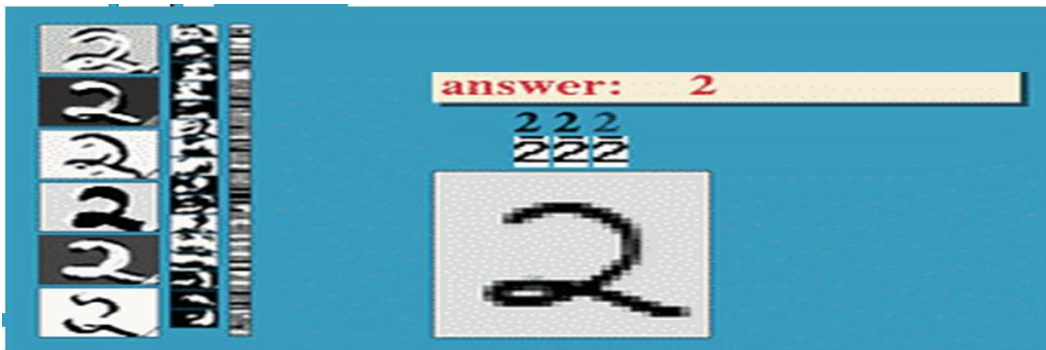
Hình 3.3. Minh họa các bước tích chập



### 3.2. Kết quả nhận dạng ký tự viết tay

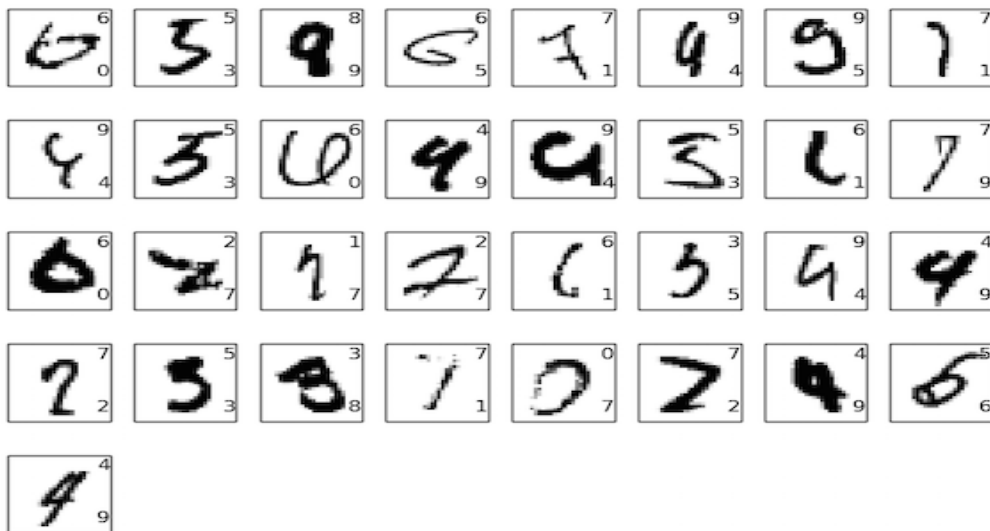
Ta xây dựng phần mềm nhận dạng biến số xe dựa trên mạng neural tích chập, với phần nhận dạng ký tự được trình bày ở trên, ta có kết quả thực nghiệm nhận dạng chữ viết tay với phần mềm mạng neural tích chập:

Hình 3.4. Kết quả thực nghiệm CNN



Quá trình thử nghiệm với 1000 chữ viết tay với phần mềm ta thấy rằng kết quả chính xác với 99.6%, một số kết quả sai do chúng ta viết không thể phân biệt được , ví dụ như với bộ chữ viết sau:

Hình 3.6. Một số mẫu chữ lỗi



Với phần mềm nhận dạng chữ viết tay kết quả thu được gần như có độ chính xác tuyệt đối, ta áp dụng nhận dạng chữ viết tay trong việc nhận dạng biển số xe

### 3.3. Kết quả thực nghiệm nhận dạng biển số xe với mạng Neural tích chập

Chạy ứng dụng nhận dạng biển số, đưa lần lượt 376 ảnh dữ liệu về xe để kiểm tra kết quả phát hiện biển số, tách ký tự và nhận dạng ký tự của chương trình, ghi nhận kết quả.

Tỉ lệ biển nhận dạng đúng vùng biển số: 372/376 ~ 98%



Biển bị bóng, phản chiếu

Biển số bị tối, thiếu ánh sáng

Biển dơ, mù, ó vàng

Hình 4.2. Một số biển không phát hiện được biển số

Tỉ lệ biển tách đúng và đầy đủ ký tự: 330/335 ~ 98%



Hình 3.3. Một số biển không tách đúng ký tự. Chữ số bị dính với các vật bên ngoài như đinh ốc, ký tự bị mờ nét, mất nét, loang lổ,...

Kết quả nhận dạng chung của ứng dụng từ khâu phát hiện biển số, đến tách ký tự và nhận dạng ký tự đạt xấp xỉ 65% với dữ liệu có nhiều ảnh không đạt tiêu chuẩn như bị bóng mờ, quá dơ, nhòe,... Với ảnh biển số sạch sẽ, rõ ràng, không chứa các phụ kiện gắn trên biển, tỉ lệ nhận dạng có thể đạt tới hơn 70%.

### 3.4. Kết luận

Demo nhận dạng với phương pháp trình bày trong luận văn này đã đạt được một số điểm như:

- Nắm được các vấn đề cơ bản của ảnh số và xử lý ảnh số

- Sử dụng tương đối tốt thư viện EmguCV cho C# để xử lý ảnh.
- Nắm được các đặc điểm của một bài toán nhận dạng nói chung và nhận dạng biển số xe nói riêng
  - Tìm hiểu được một số phương pháp nhận dạng biển số xe, nắm được ưu nhược điểm của từng phương pháp.
  - Demo và Test thử thành công trên các bộ dữ liệu mẫu với tỉ lệ nhận dạng chính xác 99%.

**Ưu điểm:** thuật toán cài đặt nhanh, tìm vùng biển số và cách ly ký tự với tỉ lệ thành công cao (ở những biển số thông thường), tìm được ở những ảnh tự nhiên, vùng biển số bị nghiêng.

**Bên cạnh đó, còn tồn tại một số điểm hạn chế chưa giải quyết được như:**

- Tỉ lệ nhận dạng còn phụ thuộc nhiều vào điều kiện ánh sáng, phản chiếu hoặc che bóng.
- Với những biển số có đường viền phức tạp thì mức độ nhận dạng không cao. Chỉ giới hạn ảnh chụp trong góc 40 độ để nhận dạng được tốt, nếu góc lớn hơn thì khả năng nhận dạng được giảm.
- Ảnh xe trong bóng tối hay ban đêm không có đèn flash thì khả năng nhận dạng thấp do thiếu sáng.

### **3.5. Hướng phát triển của bài toán:**

- Nâng cao hiệu quả chương trình, tách ly các ký tự trong biển số trong các trường hợp biển số bị nhiễu nhiều, mất mát thông tin do nhiễu từ điều kiện môi trường, tìm vùng biển số trong ảnh có độ tương phản giữa biển số và nền thấp. Đặc biệt là biển xe có nền màu đỏ chữ trắng.

- Phát triển chương trình thành module phần cứng. Có khả năng tương thích với các thiết bị quan sát như camera.

- Nghiên cứu theo hướng một ứng dụng cụ thể như : giám sát phương tiện giao thông, xử lý vi phạm giao thông, quản lý xe tại các bãi giữ xe, các kho vật tư....



## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] <http://neuralnetworksanddeeplearning.com/chap6.html>
- [2] <http://neuralnetworksanddeeplearning.com/chap2.html>
- [3].Y. LeCun and Y. Bengio.“Convolutional networks for images, speech, and time-series.” In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.
- [4].Fabien Lauer, ChingY. Suen, Gérard Bloch,”A trainable feature extractor for handwritten digit recognition“,Elsevier, october 2006.
- [5].Patrice Y. Simard, Dave Steinkraus, John Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," International Conference on Document Analysis and Recognition (ICDAR), IEEE
- [6] Chirag N. Paunwala & Suprava Patnaik. “A Novel Multiple License Plate Extraction Techniquefor Complex Background in Indian Traffic Conditions”, Sarvajanic College of Engineering and Technology, 2010
- [7] Choo Kar Soon, Kueh Chiung Lin, Chung Ying Jeng and Shahrel A. Suandi , “Malaysian Car Number Plate Detection and Recognition System”, 2012
- [8] Tran Duc Duan, Duong Anh Duc, Tran Le Hong Du, “Combining Hough Transform and Contour Algorithm for detecting Vehicles License-Plates”, University of Natural Sciences, 2004
- [9] Nobuyuki Otsu, "A threshold selection method from gray-level histograms", 1979
- [10] Suman K. Mitra. “Recognition of Car License Plate using Morphology”, hirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India
- [11] Các tài liệu về EmguCV tại [www.emgu.com](http://www.emgu.com) OPenCV tại [www.opencv.com](http://www.opencv.com)

