

Convolutional Neural Network

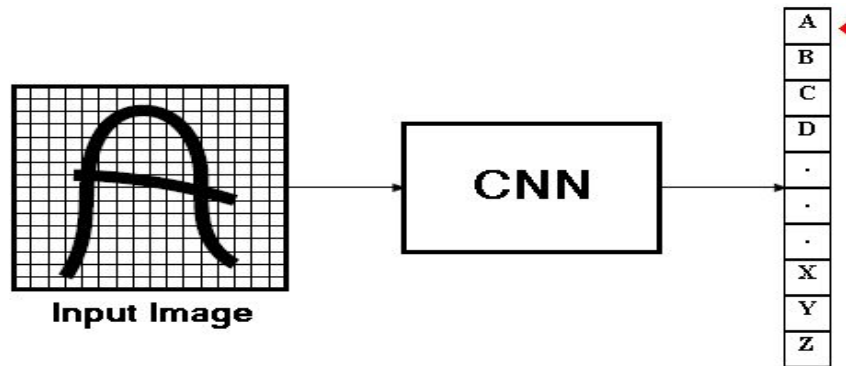
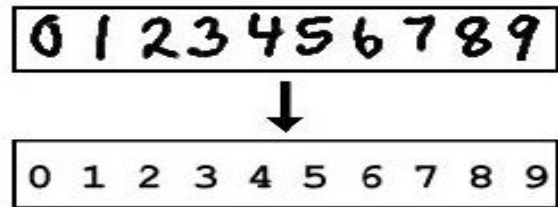
Sử dụng nguồn của Geoffrey Hinton, Abin -
Roosgard & Christopher Olah

Mục lục

- Giới thiệu
- Hạn chế của những neural net trước
- Convolutional neural networks
- LeNet 5
- Bất lợi
- Ứng dụng

Giới thiệu Convolutional neural networks

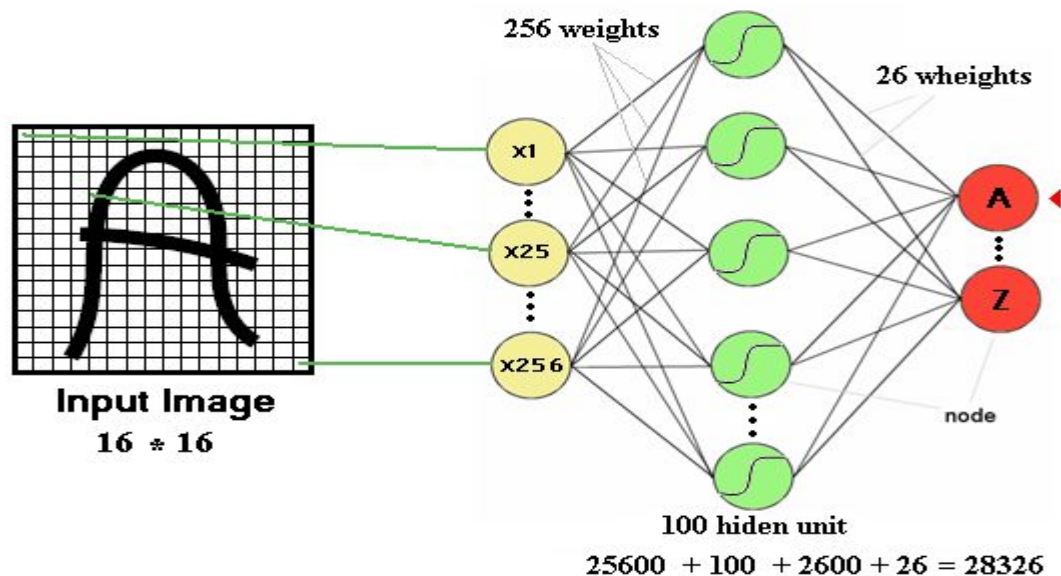
- Dùng để xử lý tín hiệu và hình ảnh
- Cải tiến hơn neural net nói chung:
 - Hiệu suất
 - Độ chính xác
 - Bất biến đối với biến dạng nhỏ đầu vào



Hạn chế của những
neural net trước

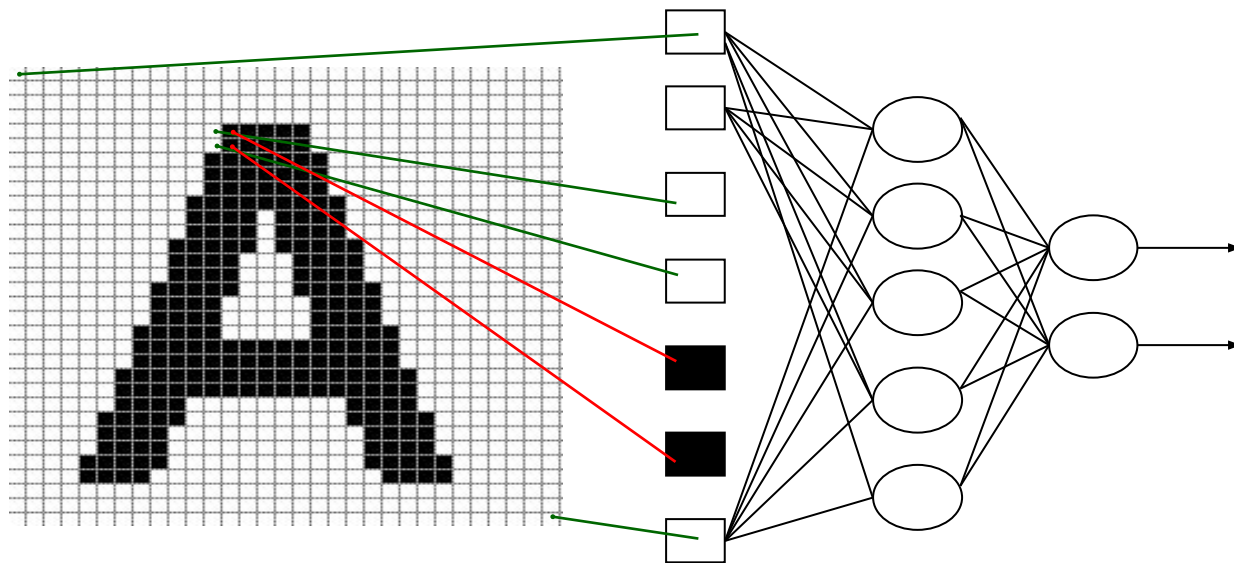
Hạn chế của những neural net trước

- Số neuron cần học trở nên cực kì lớn khi xử lý ảnh



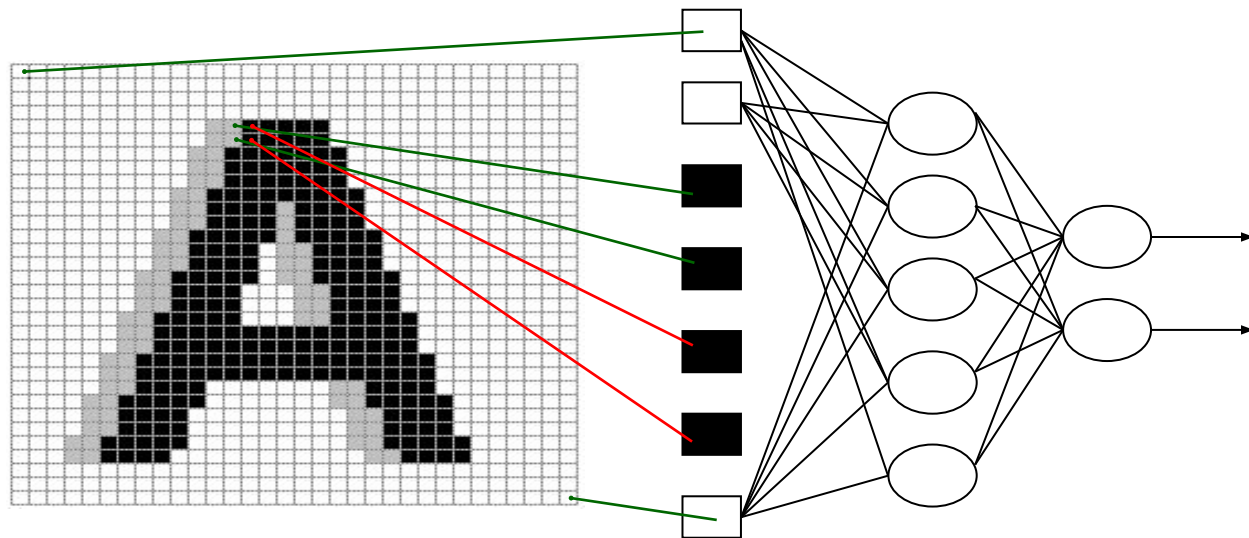
Hạn chế của những neural net trước

- Topo của đầu vào hoàn toàn bị bỏ qua

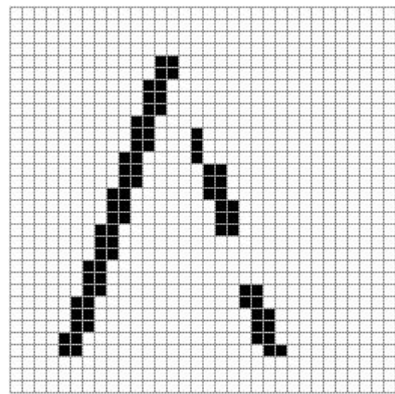
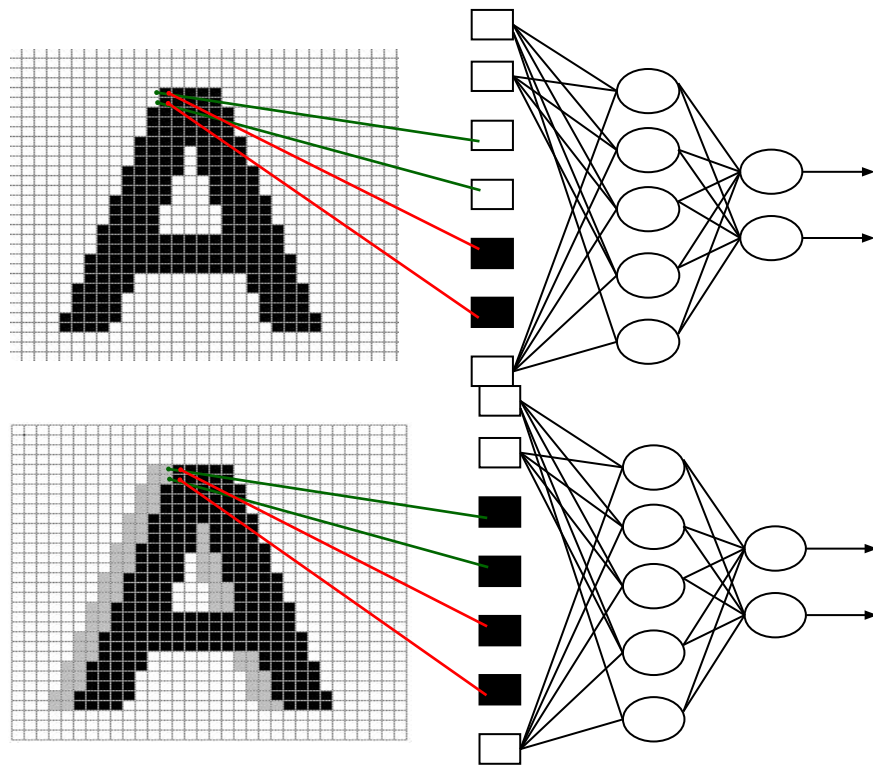


Ví dụ như

- Dịch chuyển
- Phóng to thu nhỏ
- Xoay

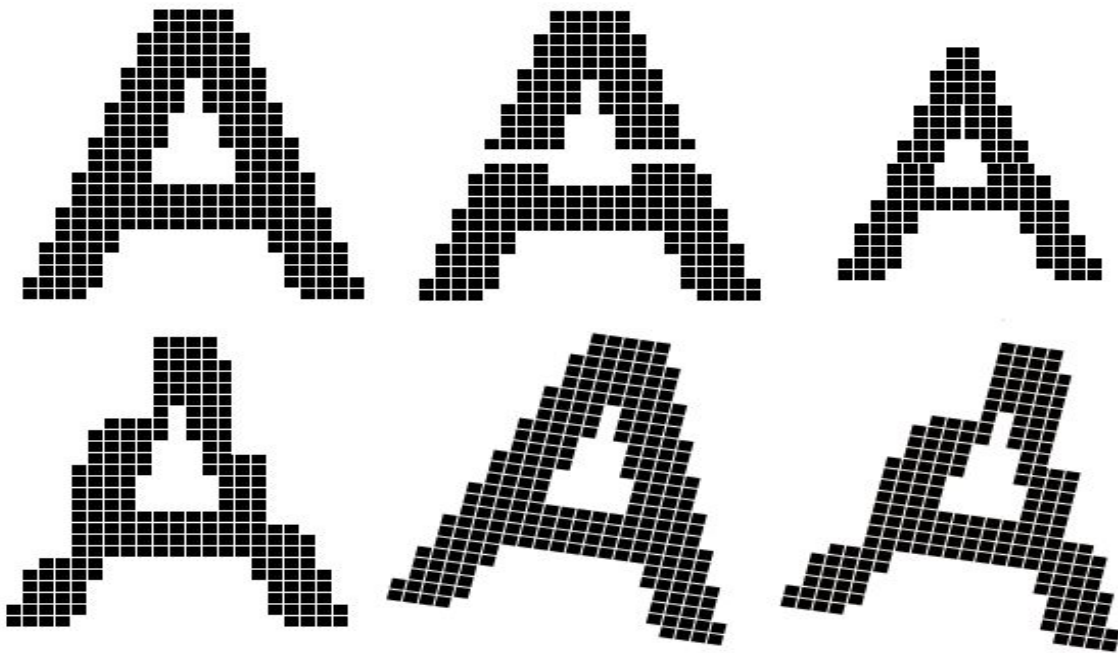


Dịch chuyển



- 154 pixel thay đổi từ việc dịch 2 pixel sang trái:
- 77 từ đen sang trắng
 - 77 từ trắng sang đen

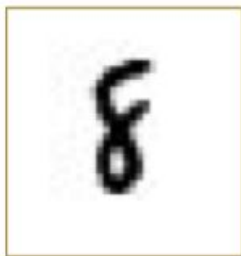
Phóng to thu nhỏ và xoay



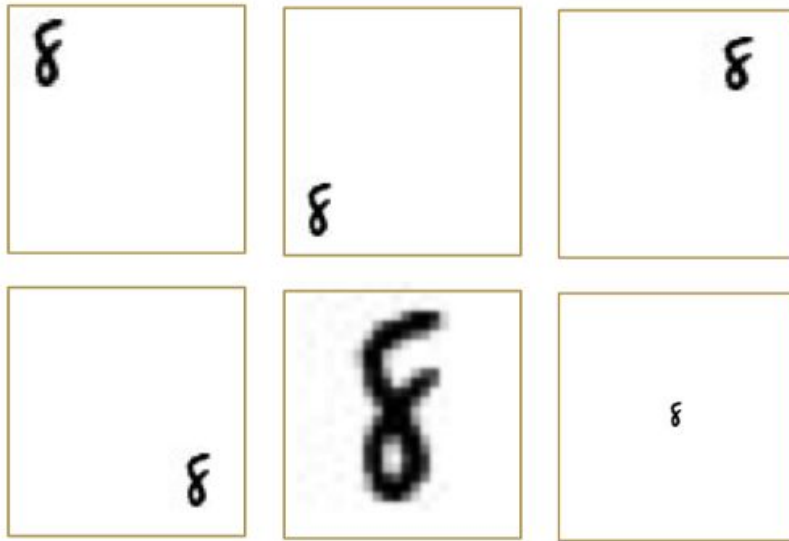
Ý tưởng

Tạo nhiều ảnh cho training data

Original Training Image



More training images generated by a script

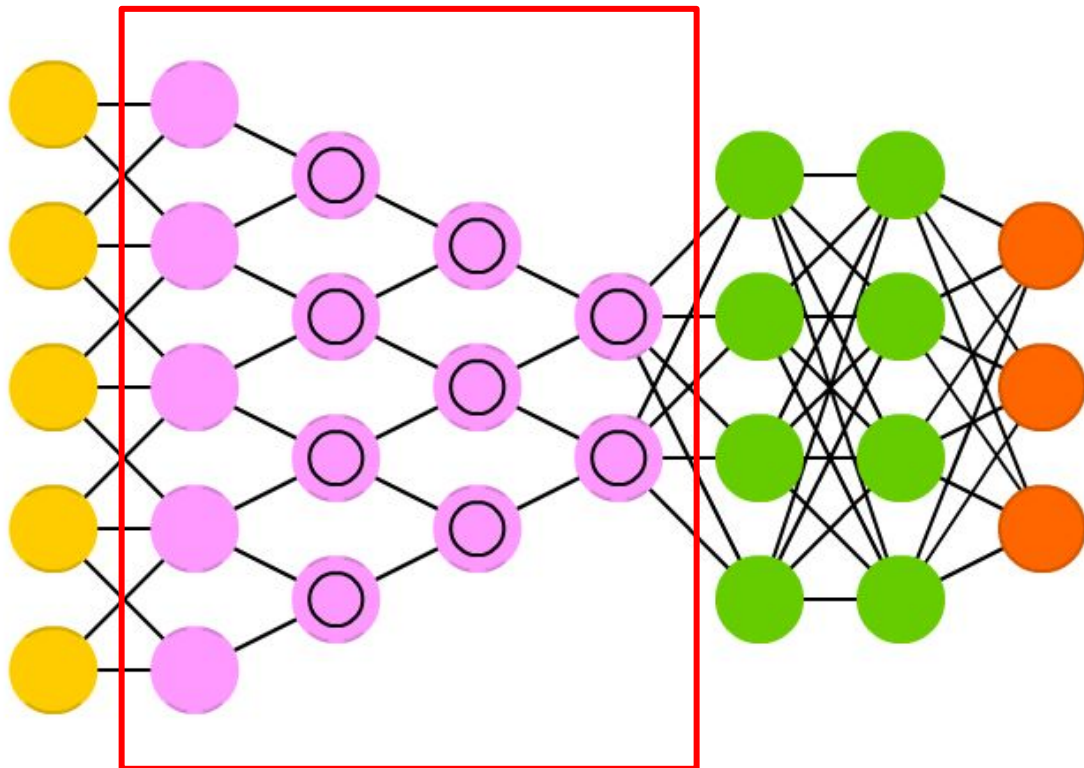


Convolutional Neural Network (CNN)

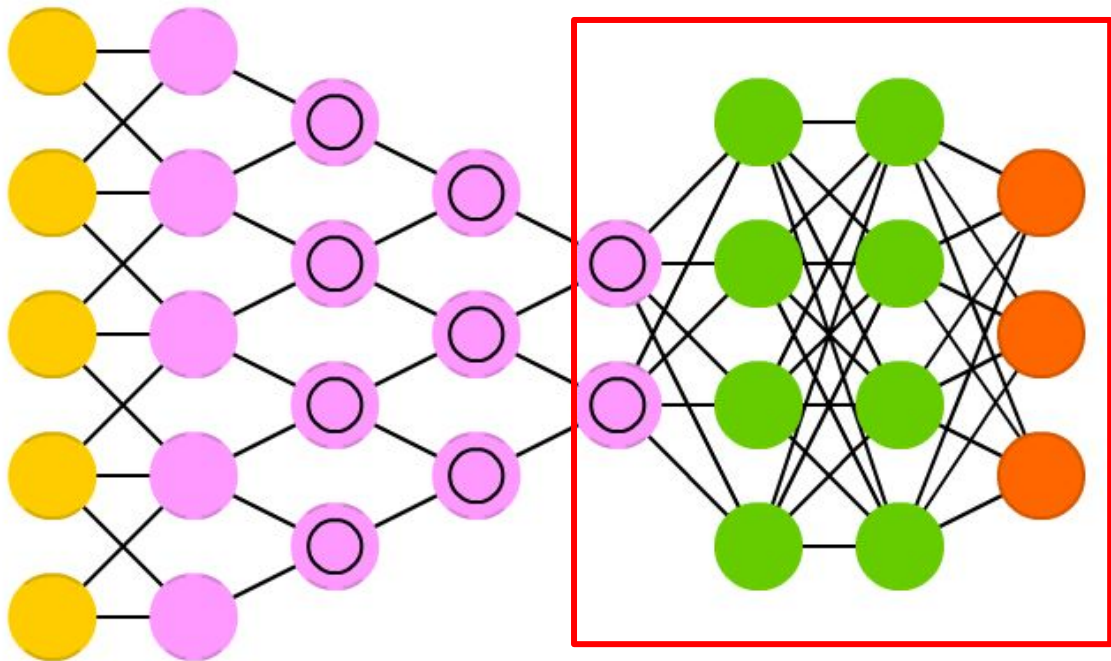
CNN là một dạng đặc biệt của multi-layer neural network

Từ DNN đến CNN

- Convolution layer
- Subsampling



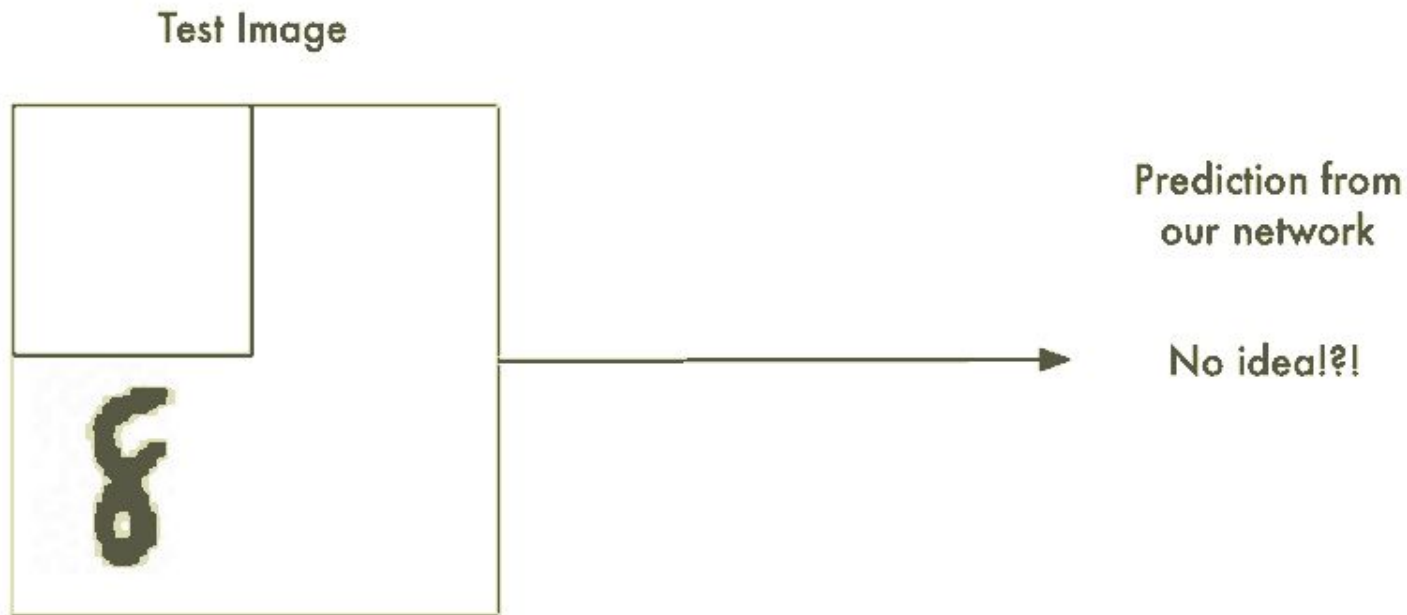
Từ DNN đến CNN



Full connection = DNN!

Bước convolution

Ý tưởng: Tìm vật bằng cách scan từng ô hình nhỏ



Bước convolution

Đoán xem đây có phải là hình ảnh của 1 đứa trẻ?



Ví dụ

Biến ảnh thành những ô ảnh nhỏ. Những ô ảnh gần nhau sẽ trùng nhau một phần



Ví dụ

Mỗi ô ảnh nhỏ sẽ được đi qua một filter và trả về một array

Original Input Image

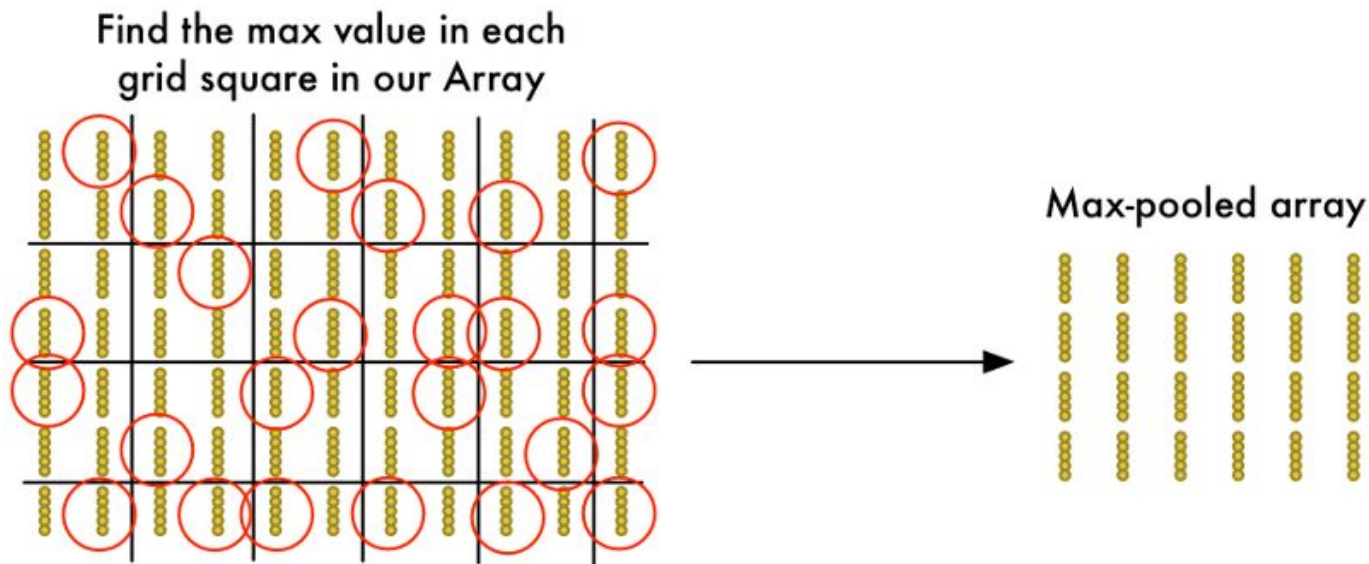


Array resulting from
convolution in Step 3



Bước subsampling

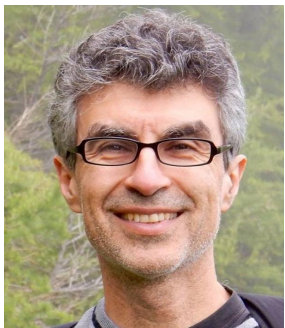
Bước này thu nhỏ kích thước của Array thu được sau bước convolution



Lịch sử



- Yann LeCun
 - Giáo sư môn khoa học máy tính ở viện toán Courant, trường đại học New York
 - Chỉ đạo nhóm Trí tuệ nhân tạo ở Facebook
- Yoshua Bengio
 - Giáo sư môn khoa học máy tính trường đại học Montreal
- Năm 1995, hai người này công bố khái niệm Convolutional neural network



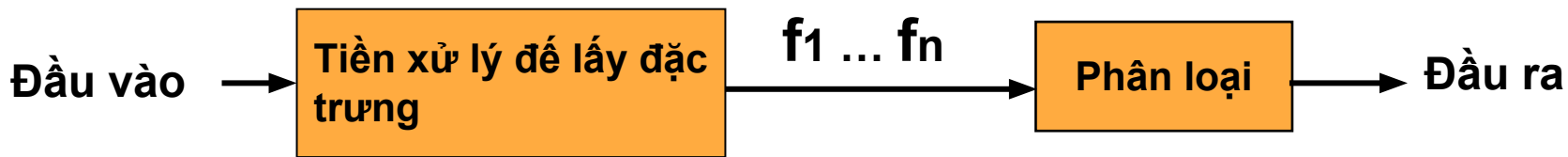
- CNN lấy cảm hứng từ những phát hiện trong Thần kinh học về những tế bào thần kinh nhạy cảm **cục bộ** và có lựa chọn định hướng trong vỏ não thị giác
- Họ thiết kế một kết cấu mạng lưới ngàm chiết xuất những đặc trưng có liên quan

- CNN là một dạng lưới feed-forward mà có thể chiết xuất những **tính chất topo từ ảnh**
- Giống như đa số các neural net khác, nó được học bằng thuật toán back-propagation

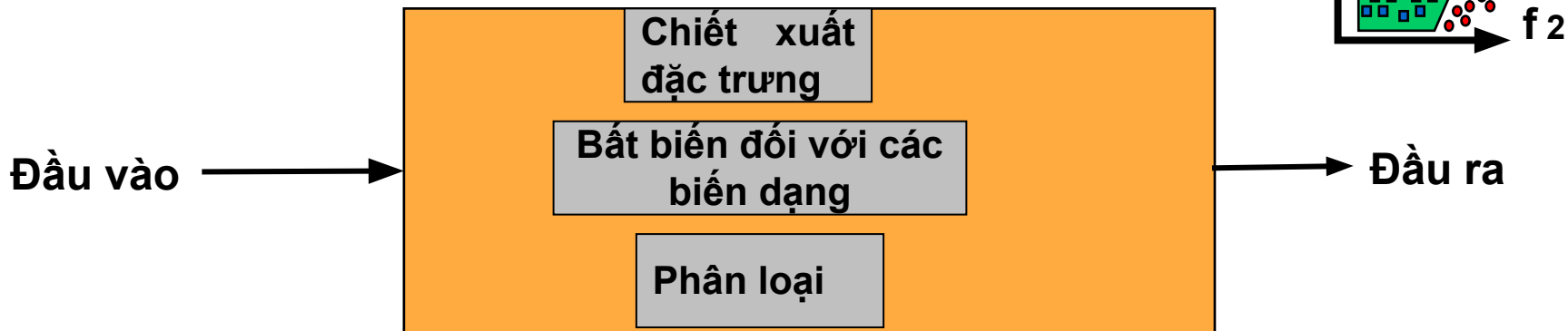
- CNN được thiết kế để nhận diện những khuôn mẫu thị giác từ những điểm ảnh mà **không cần xử lý ảnh nhiều**
- CNN có thể nhận diện những khuôn mẫu có **biên độ biến đổi lớn**, ví dụ như trong nhận diện chữ viết tay

So sánh với các thuật toán phân loại khác

Các thuật toán phân loại khác

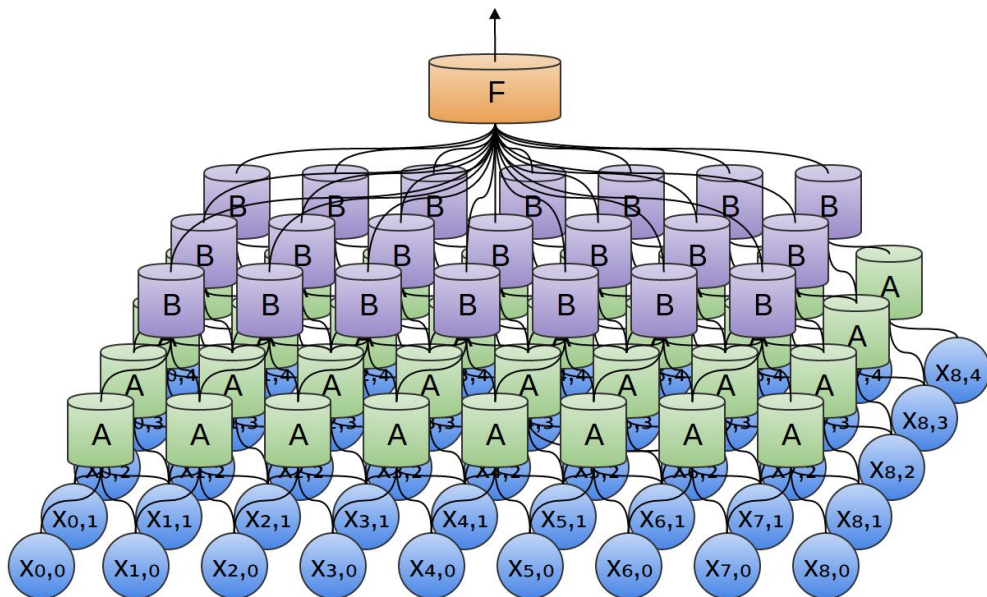


CNN



Thiết kế của CNN

- Có thể coi CNN là một dạng neural net có sử dụng nhiều bản giống hệt nhau của một vài neuron



Động lực

- Cho phép mang lưới dùng nhiều neuron trong mô hình lớn hơn, trong khi giữ số lượng tham số tương đối nhỏ

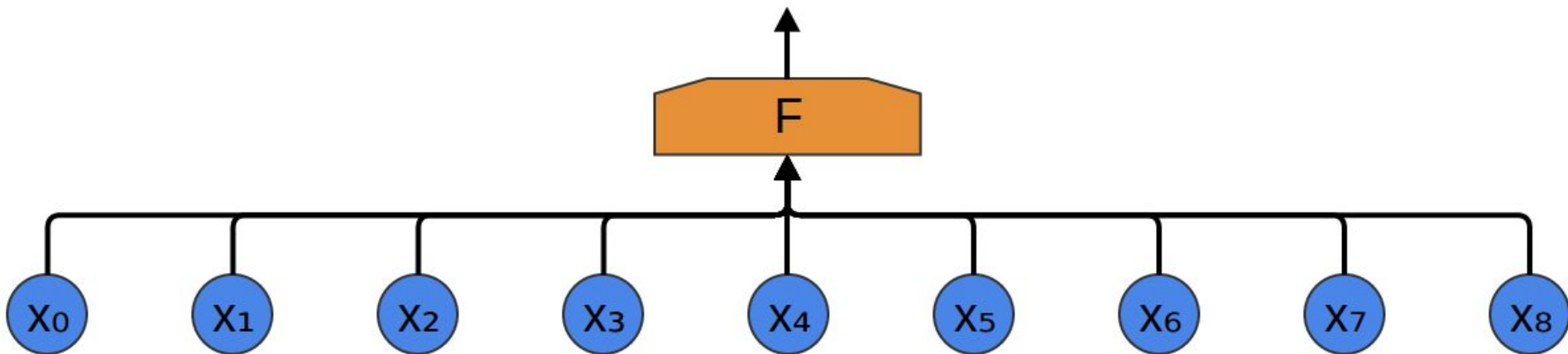
Ý tưởng lớn

- Tối ưu hóa hiệu suất bằng cách
 - Lợi dụng tính đối xứng
 - Lợi dụng khuôn mẫu
 - Lợi dụng tính cục bộ
- Trừu tượng hóa: tái sử dụng đối tượng tính toán

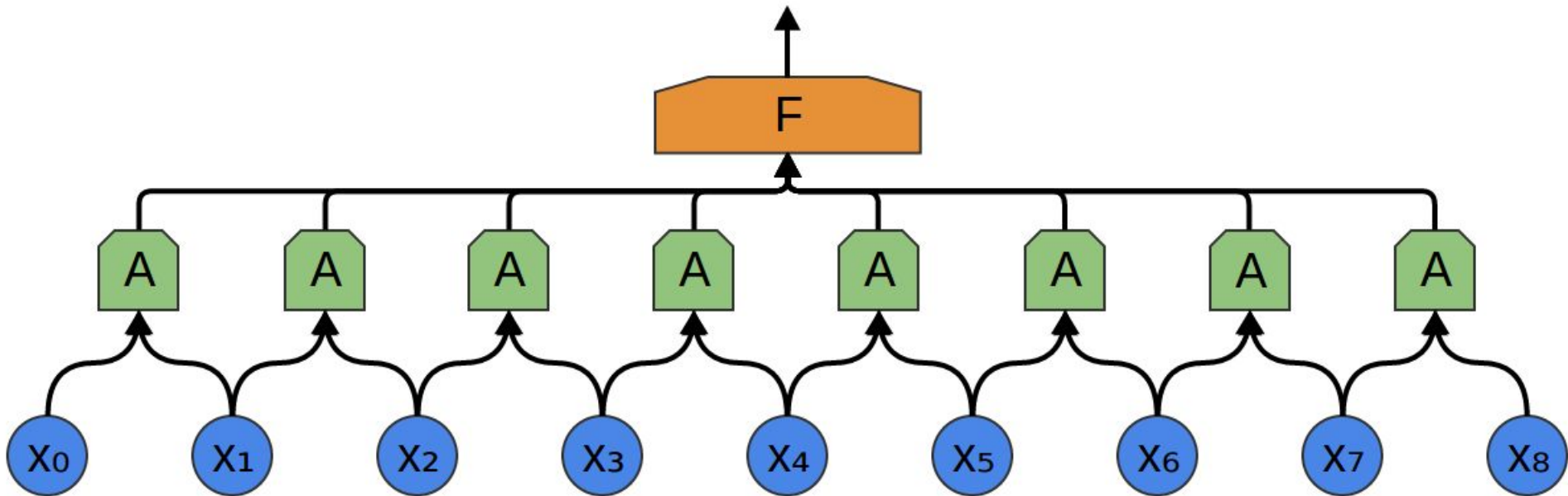
Ví dụ: tín hiệu âm thanh



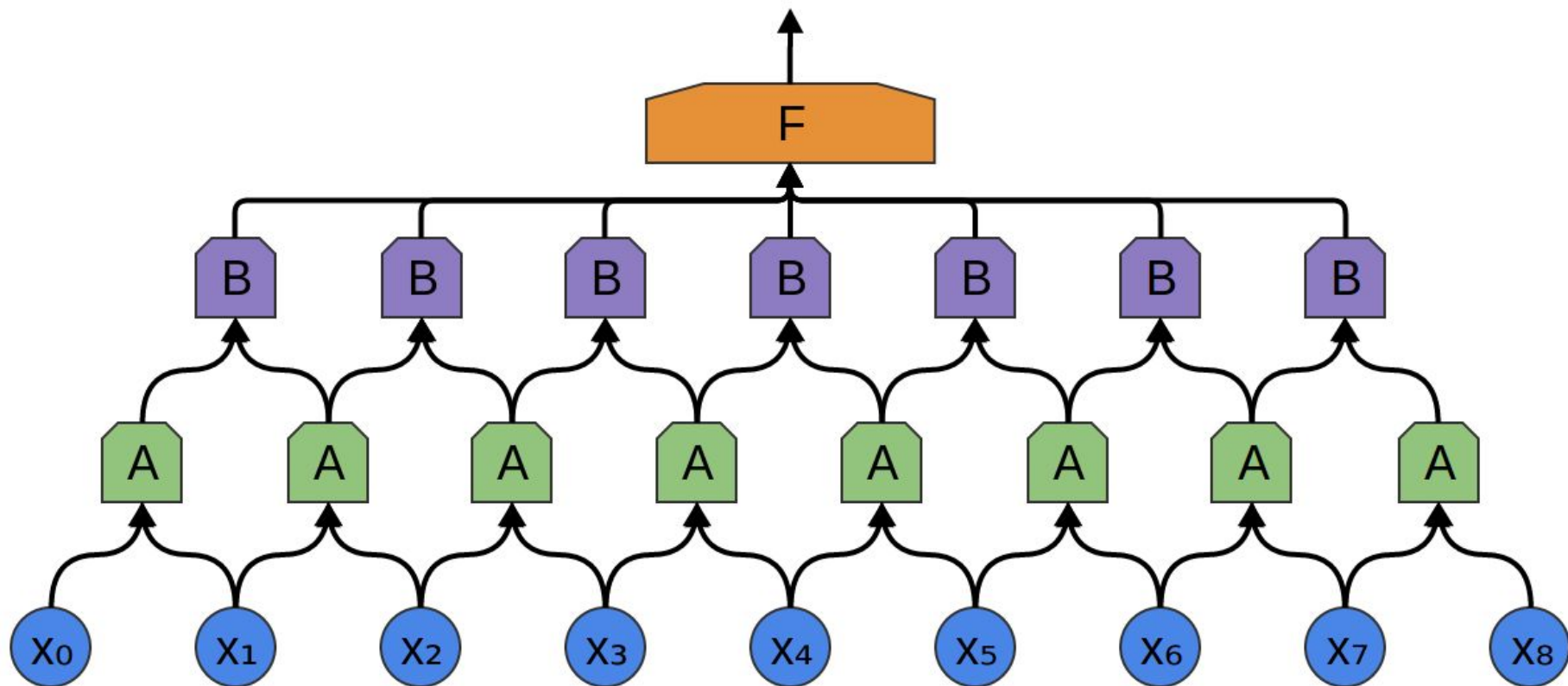
Thử fully connected NNs



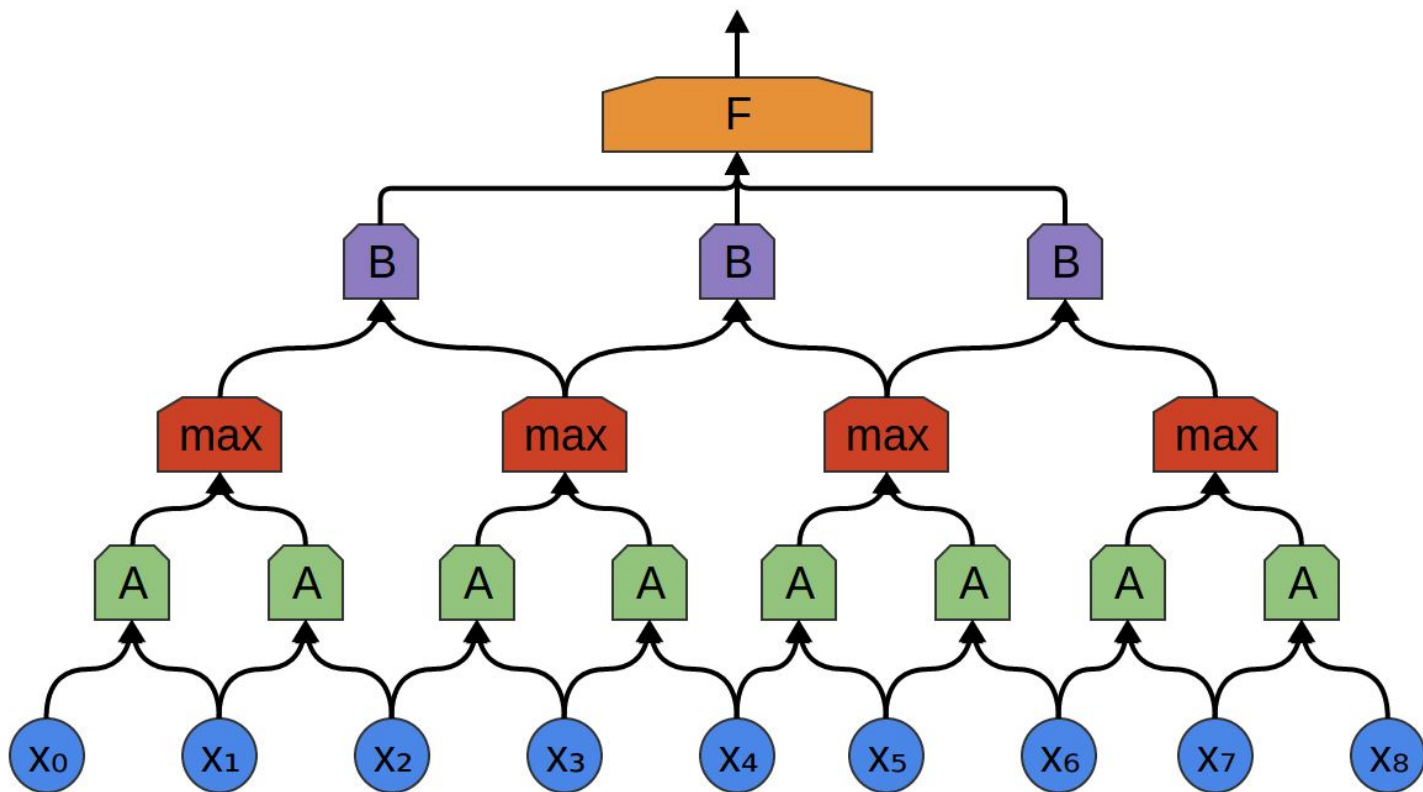
Thêm vào một lớp convolutional



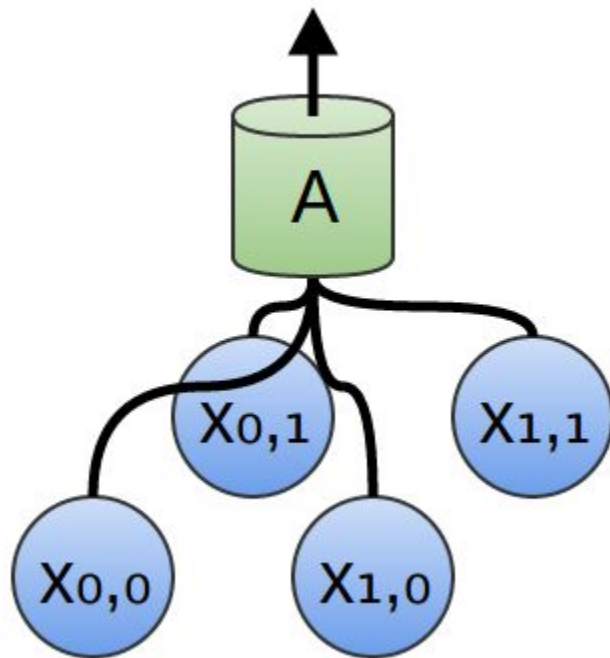
Thêm một lớp nữa



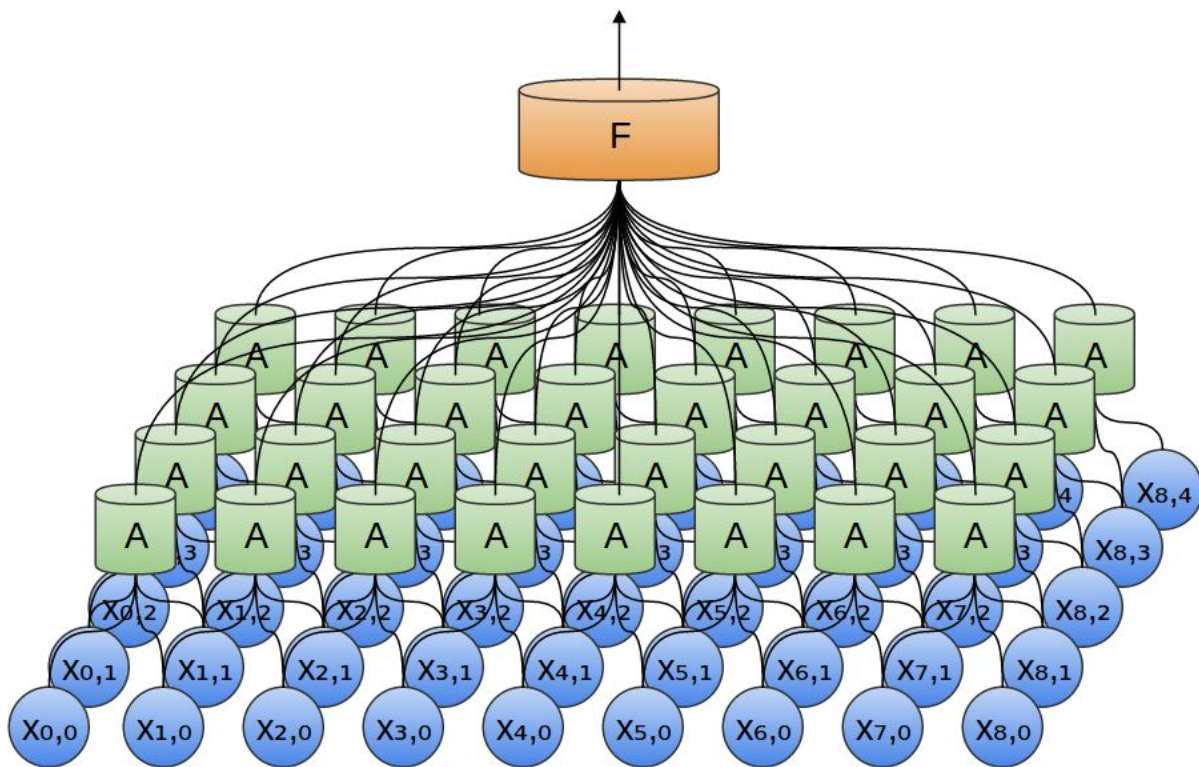
Những tầng trừu tượng hóa cao hơn: max pools



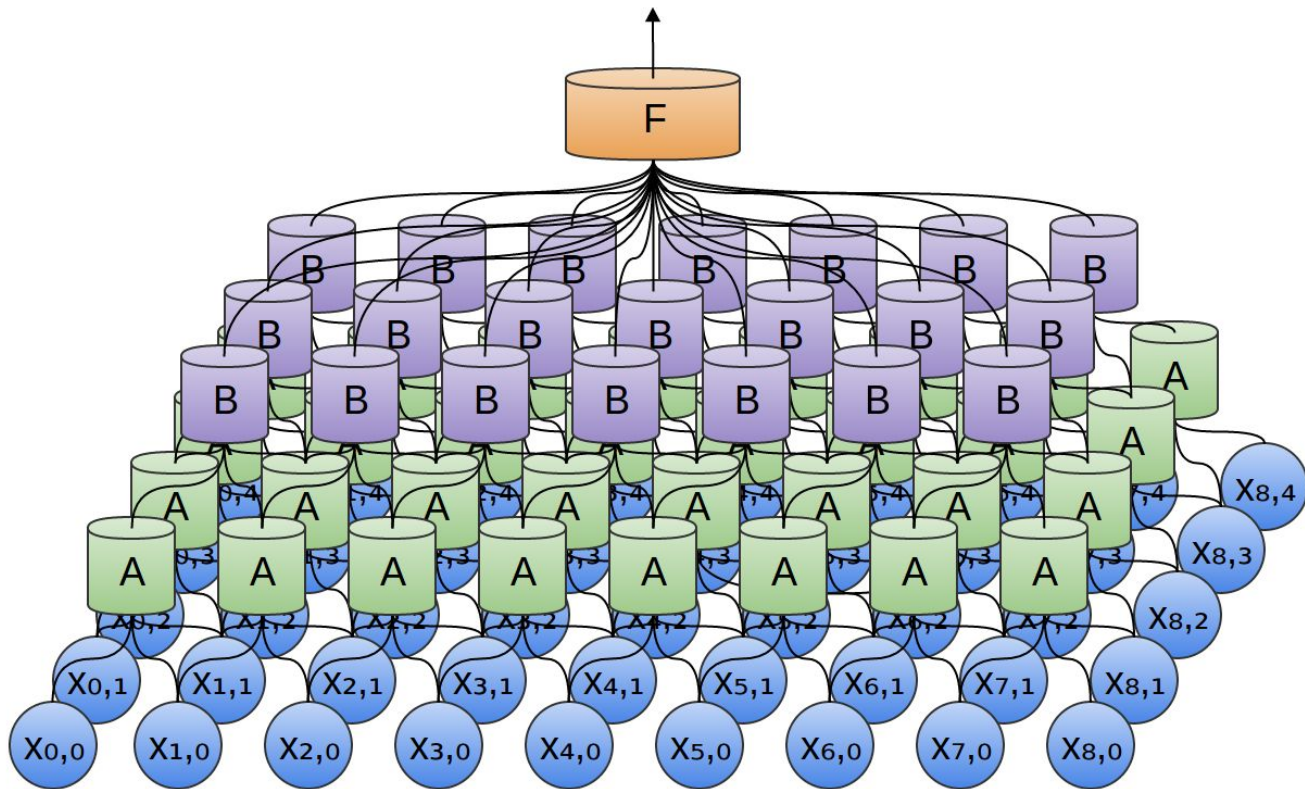
2D CNN



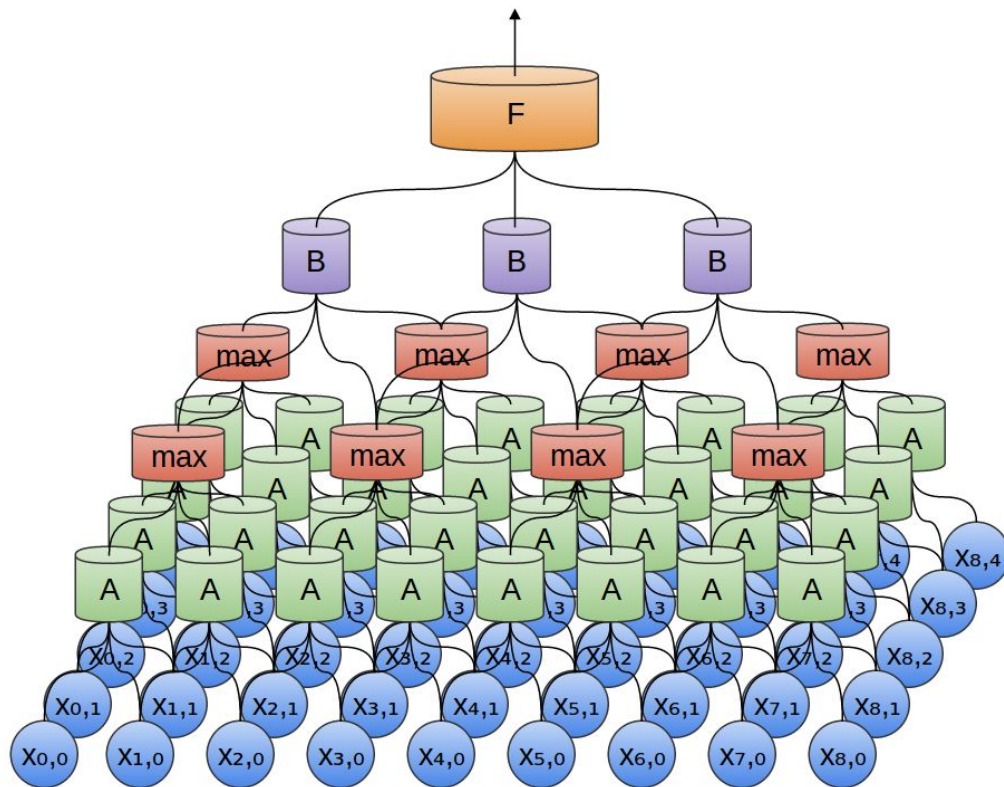
2D CNN



Thêm một lớp nữa



Thêm max pool

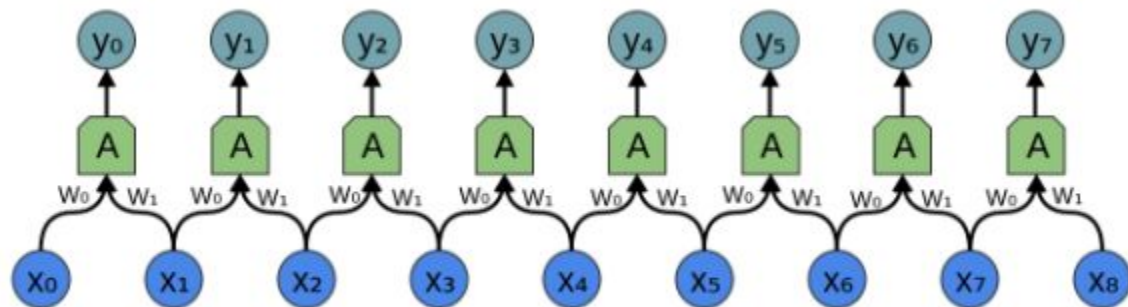


Toán tử tích chập (Convolution) một chiều

$$\begin{aligned}(f * g)[n] &\stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m] g[n - m] \\ &= \sum_{m=-\infty}^{\infty} f[n - m] g[m].\end{aligned}$$

Toán tử tích chập (Convolution) hai chiều

$$(f * g)(c_1, c_2) = \sum_{a_1, a_2} f(a_1, a_2) \cdot g(c_1 - a_1, c_2 - a_2)$$



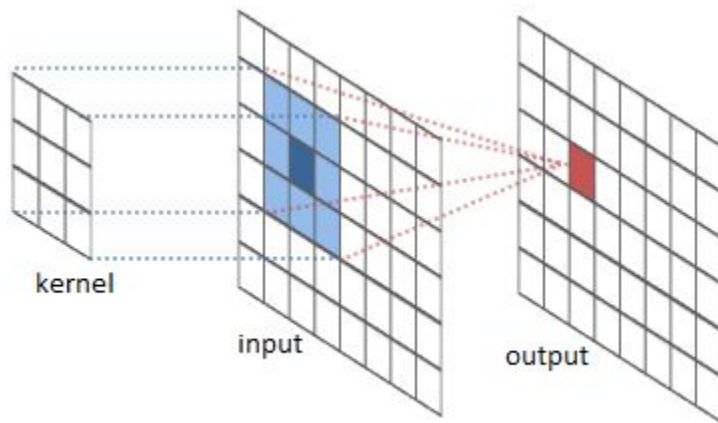
$$y_0 = \sigma(W_0x_0 + W_1x_1 - b)$$

$$y_1 = \sigma(W_0x_1 + W_1x_2 - b)$$

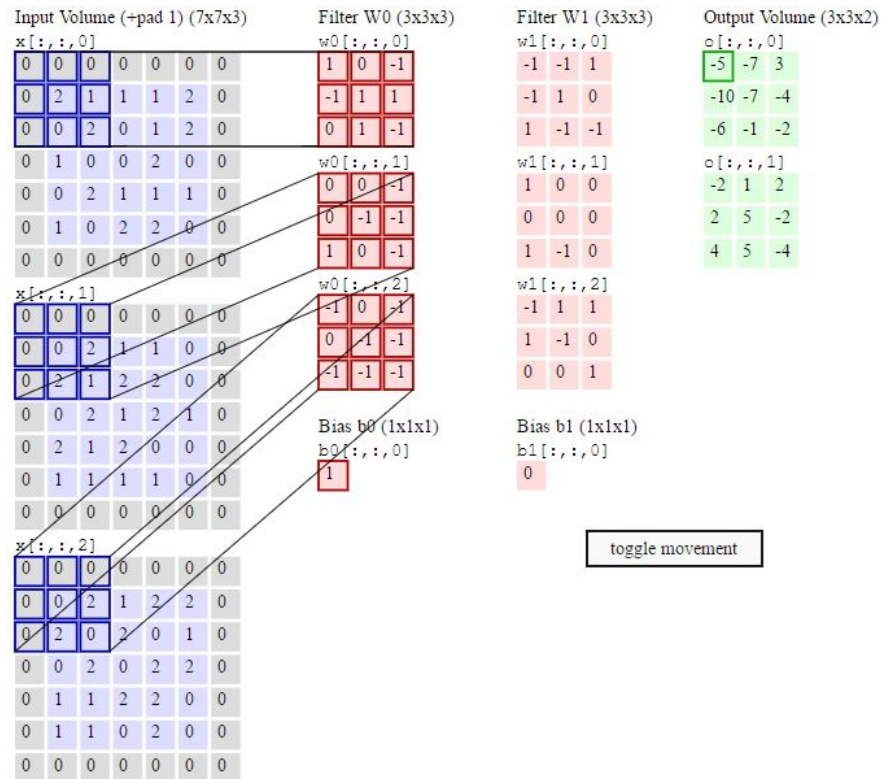
Từ đó tạo ma trận

$$W = \begin{bmatrix} w_0 & w_1 & 0 & 0 & \dots \\ 0 & w_0 & w_1 & 0 & \dots \\ 0 & 0 & w_0 & w_1 & \dots \\ 0 & 0 & 0 & w_0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

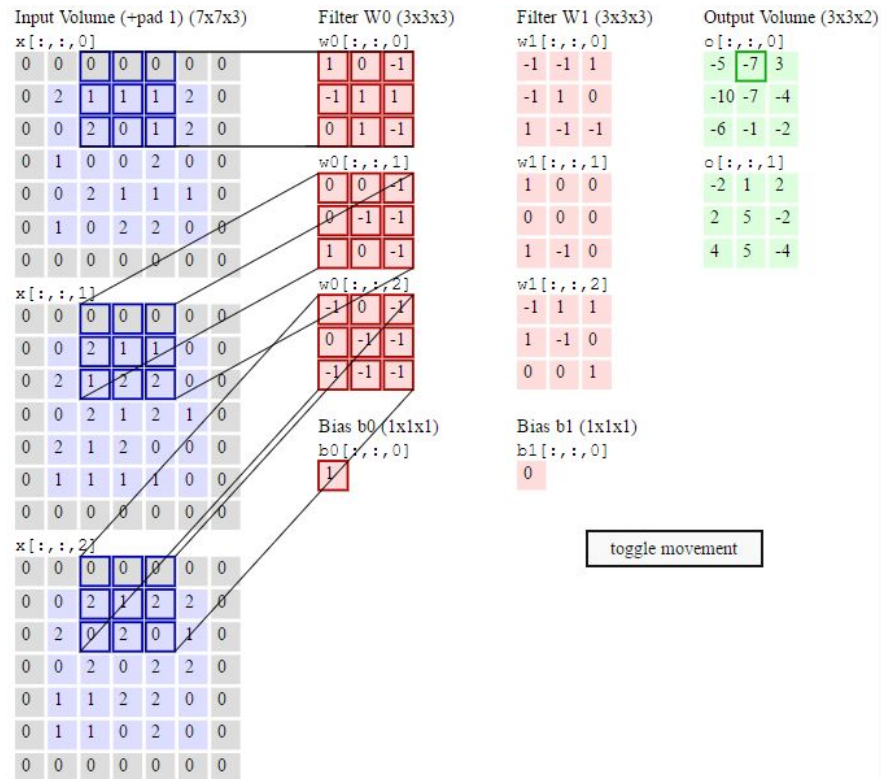
- Tương tự như xê dịch neuron qua chuỗi pixel đầu vào



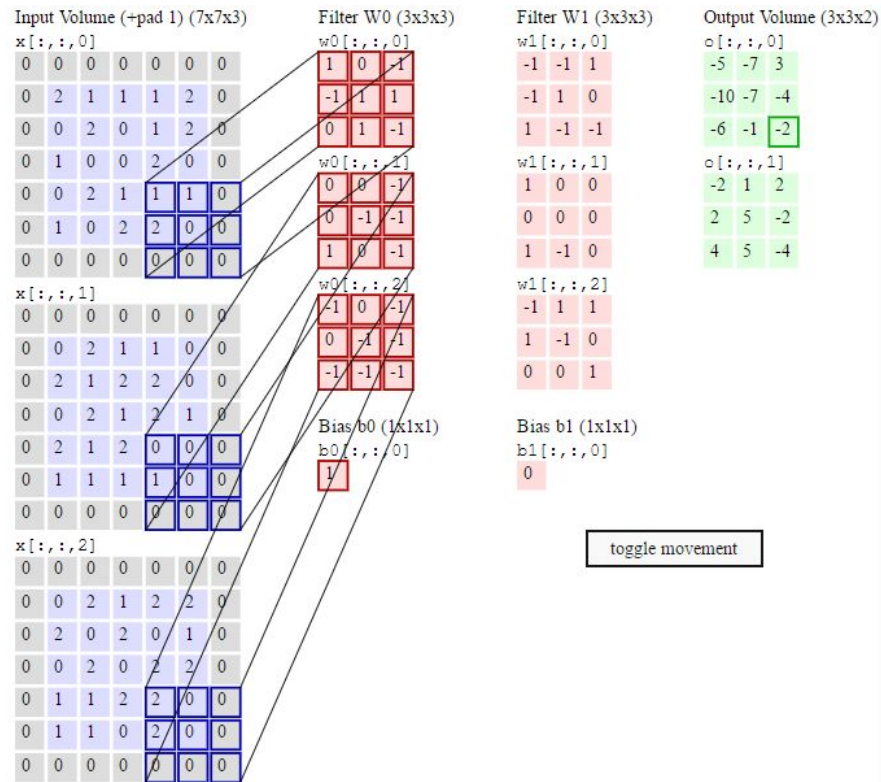
Minh họa lớp convolution



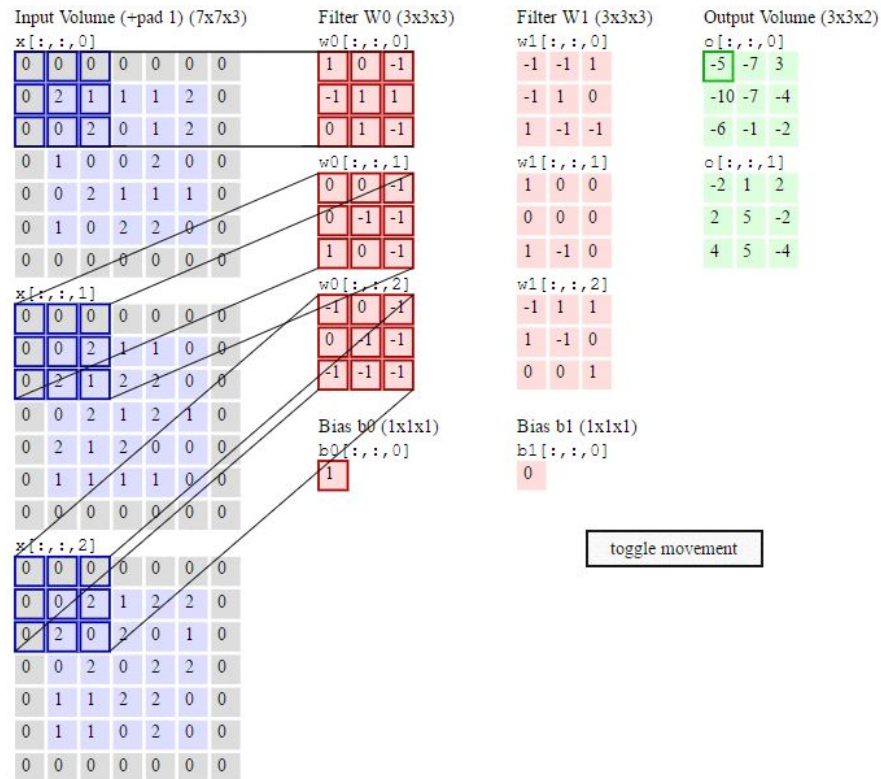
Minh họa lớp convolution



Minh họa lớp convolution

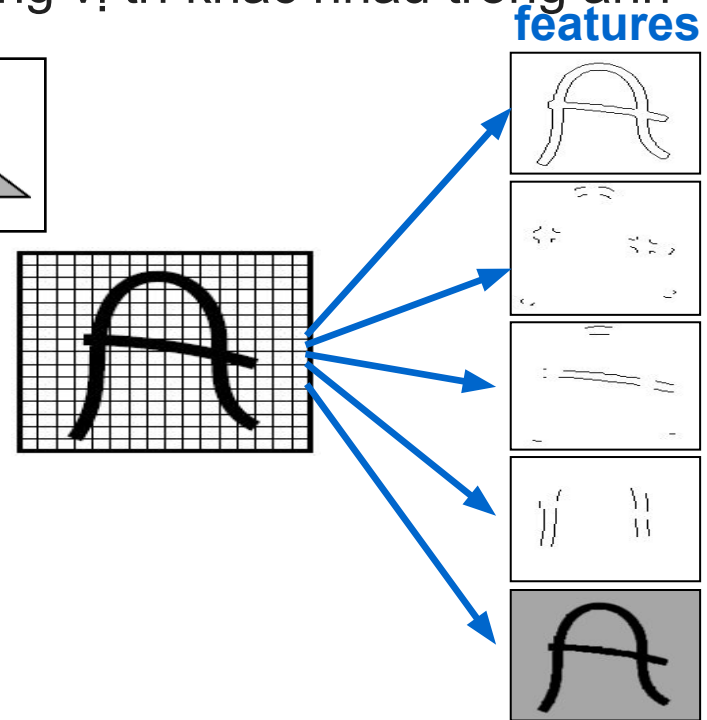
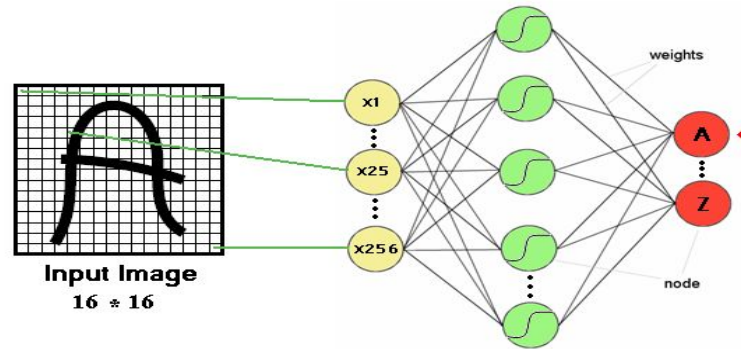
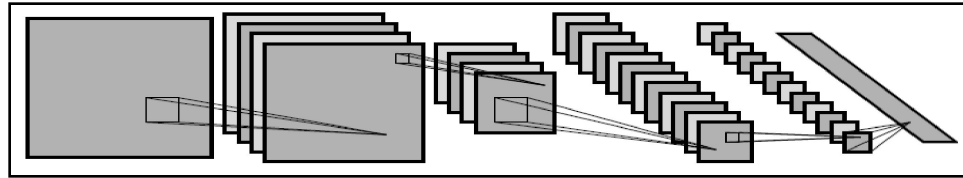


Minh họa lớp convolution

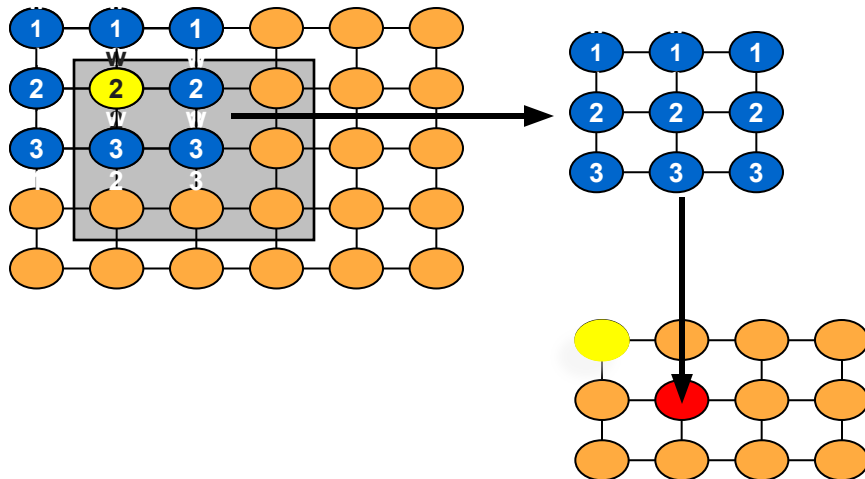
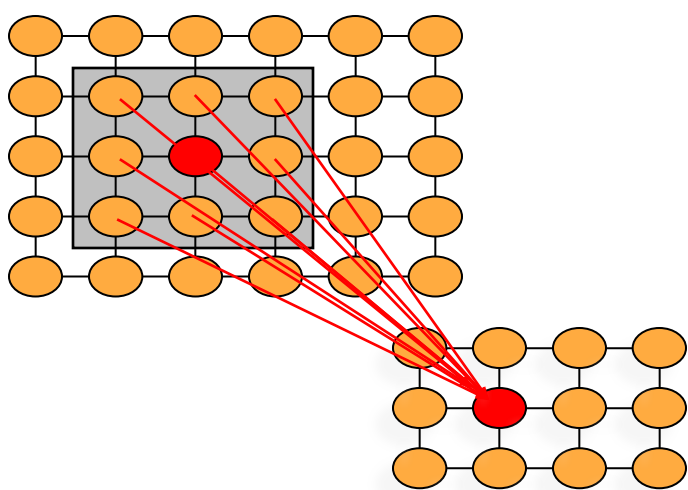
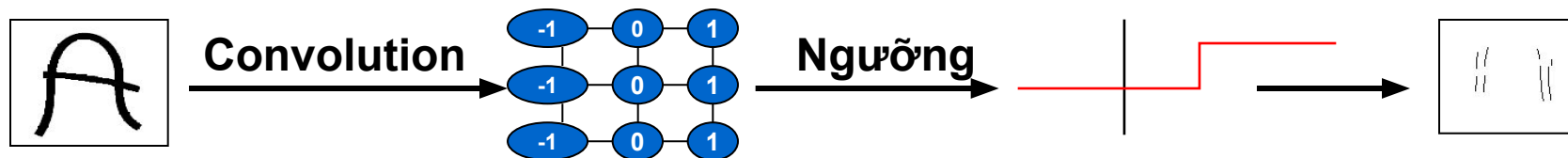


Lớp convolution

- Phát hiện cùng một đặc trưng ở những vị trí khác nhau trong ảnh

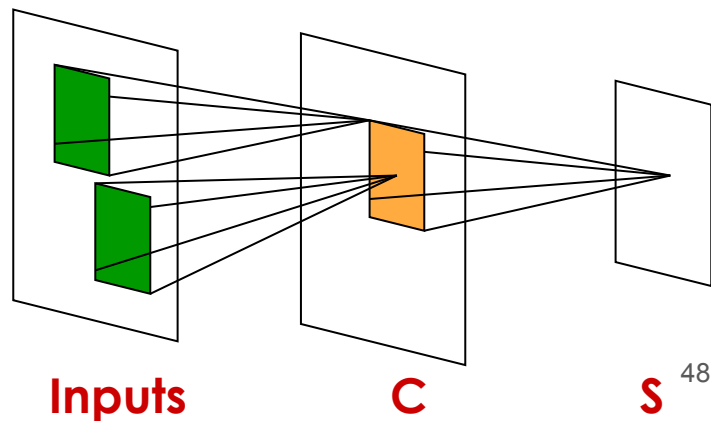


Chiết xuất đặc trưng



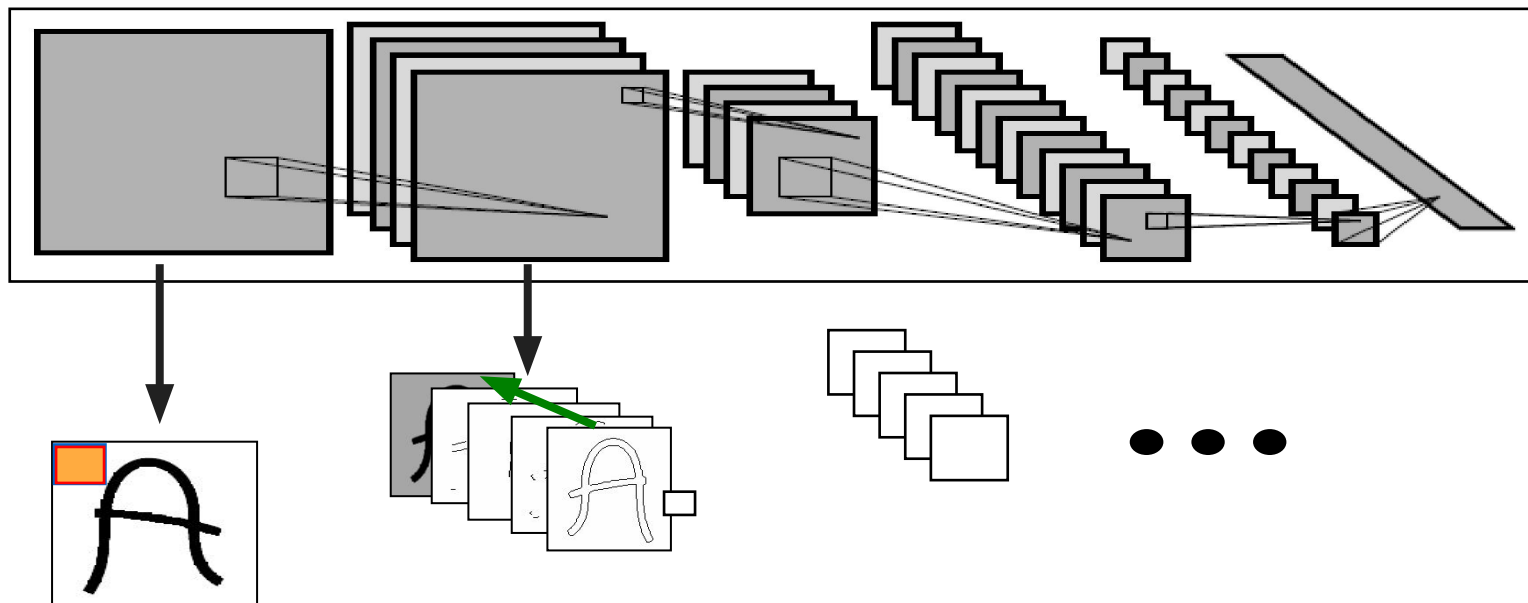
Chiết xuất đặc trưng

- Tất cả các neuron của một đặc trưng đều dùng chung trọng số, nhưng không chung thiên lệch
- Bằng cách này tất cả neuron đều phát hiện cùng một đặc trưng ở những vùng khác nhau trên ảnh
- Giảm số tham số cần học





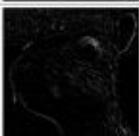




Chiết xuất đặc trưng

- Khi một neuron trong lưới được bật, điều này nghĩa là có tương thích với khuôn mẫu

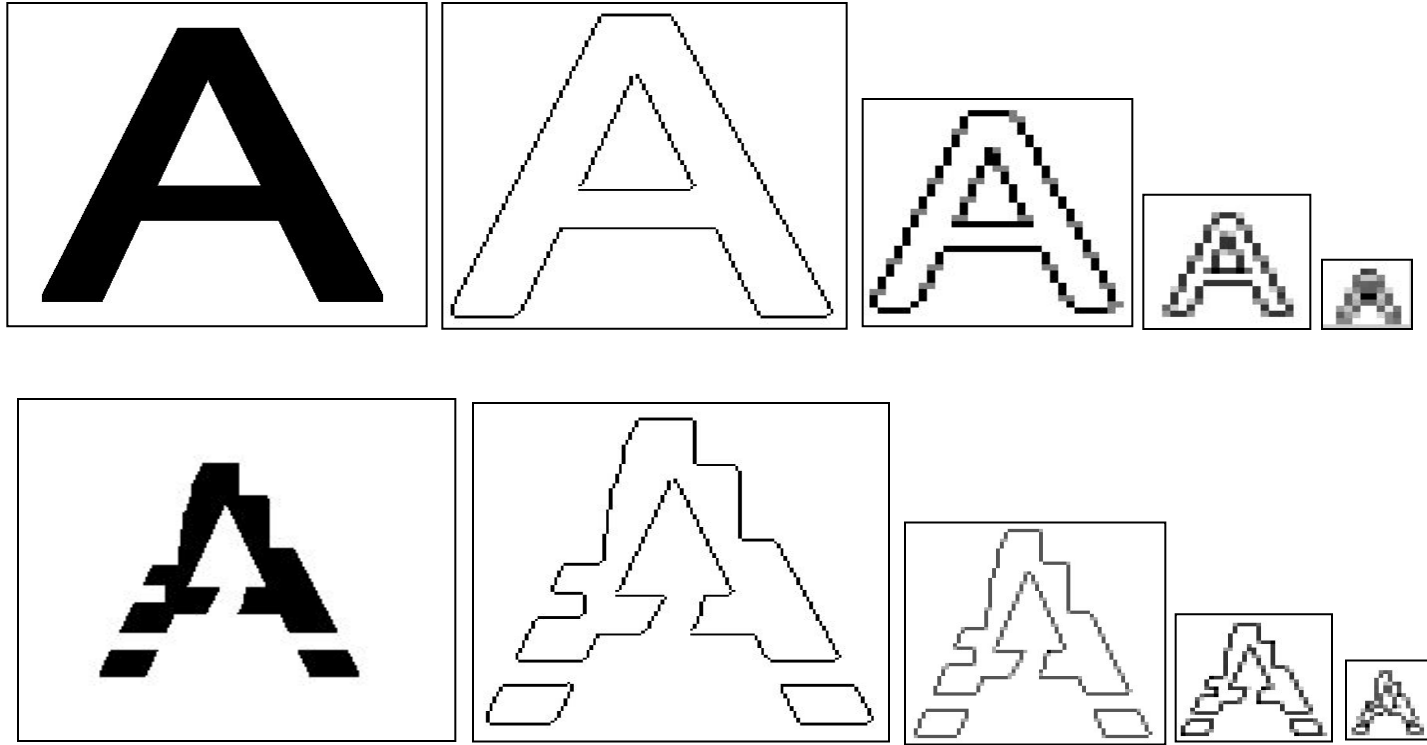


Kênh

- Mỗi ảnh cần nhiều khuôn mẫu để nắm bắt hết những điểm đặc biệt của ảnh
- Trong mỗi lớp convolution, sử dụng nhiều kernel một lúc
- Mỗi kernel tạo ra một kênh
- Ví dụ lớp đầu tiên có thể có cả 7 kênh như hình bên phải

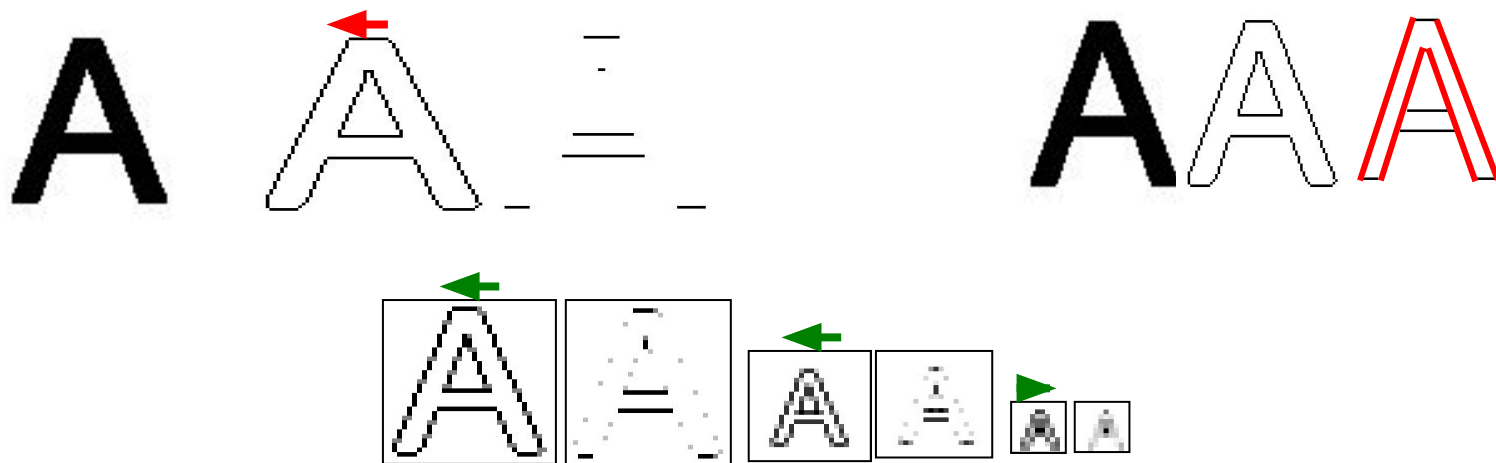
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
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)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Lớp subsampling

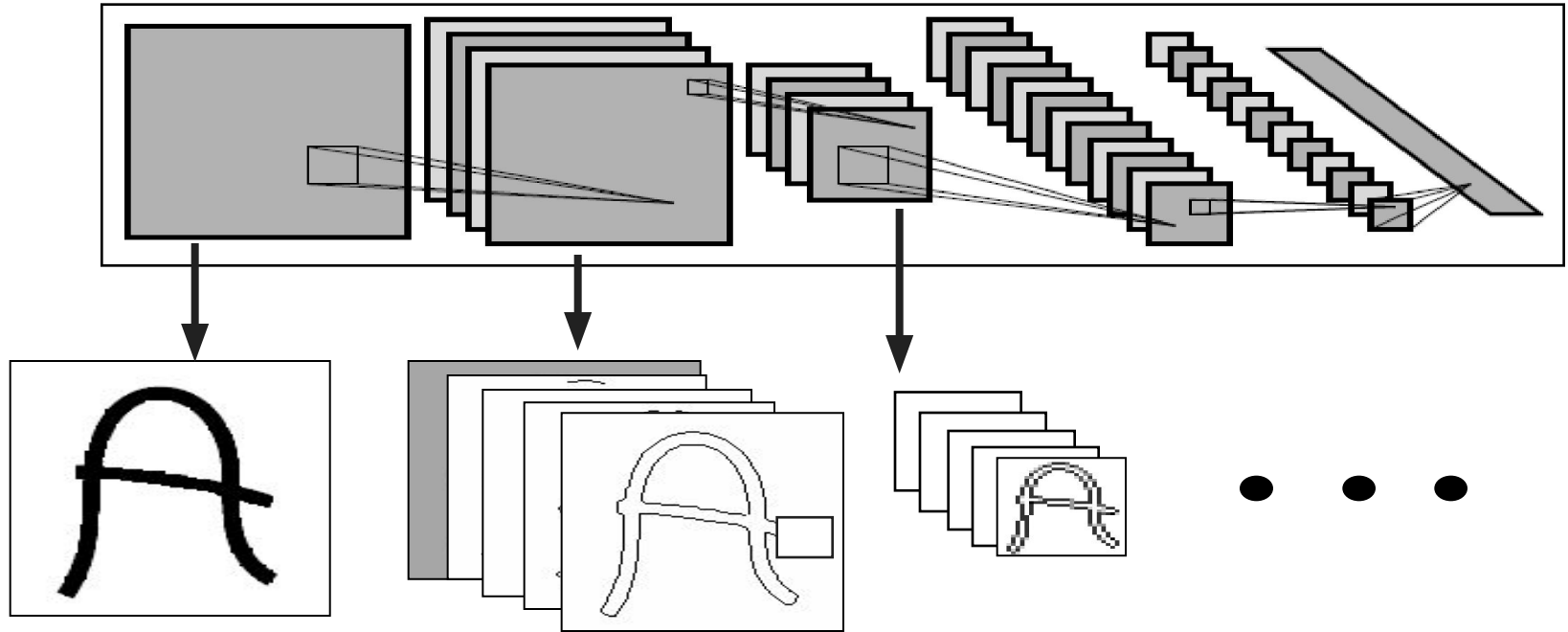


Lớp subsampling

- Lớp subsampling làm giảm độ phân giải không gian của mỗi đặc trưng
- Bằng cách này, ta đạt được một phần nào đó bất biến đối với xê dịch và biến dạng.



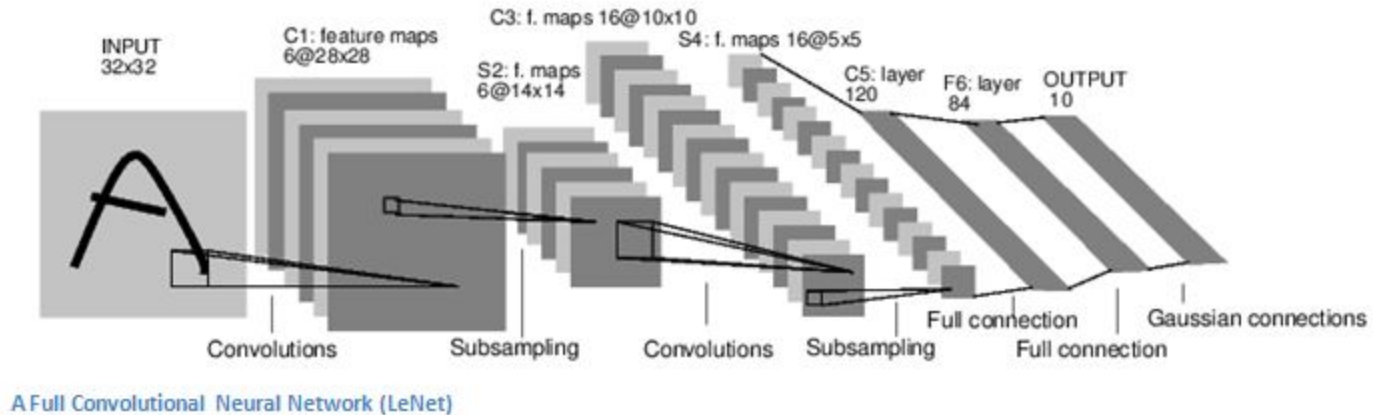
Kết hợp convolution và subsampling



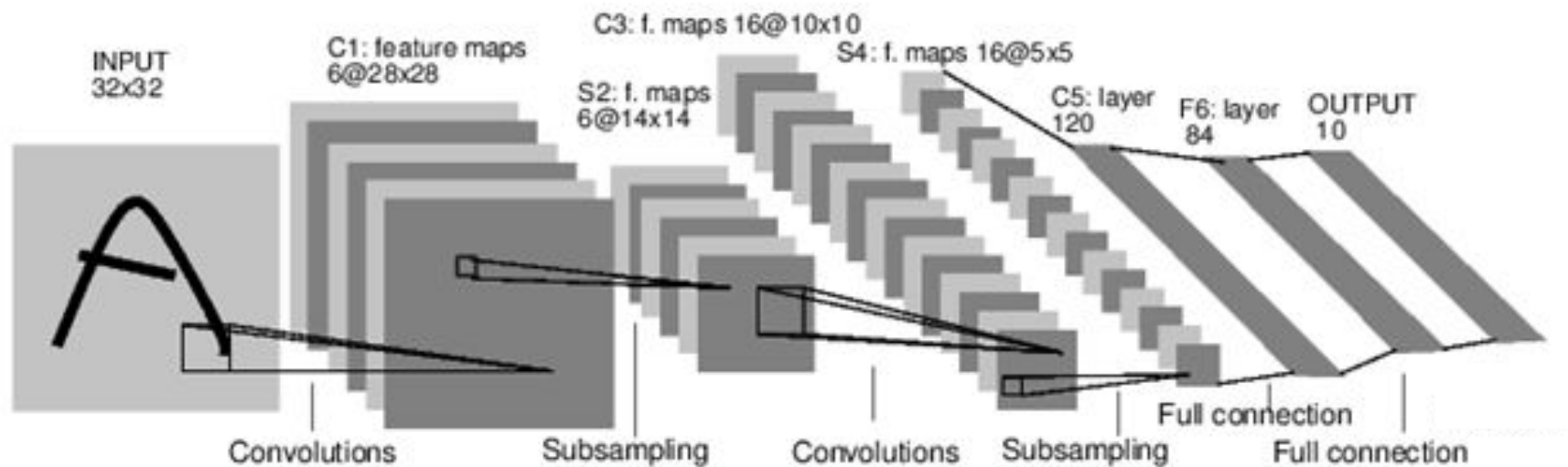
LeNet 5

LeNet5

- Phát minh của LeCun.
- Ảnh nguyên 32×32 pixel.

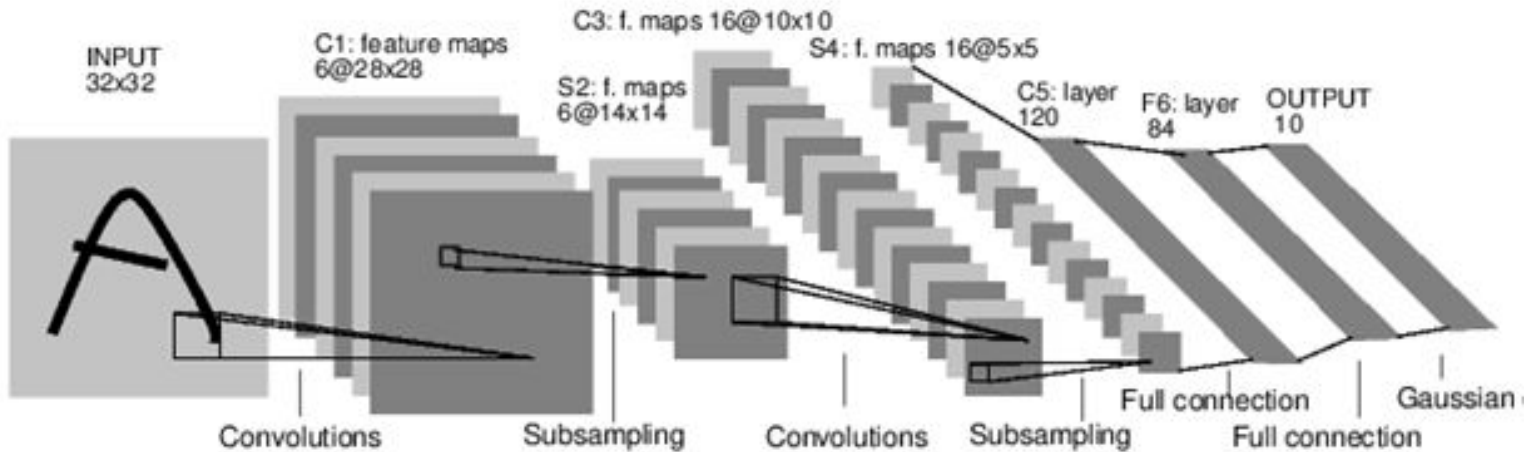


LeNet5



A Full Convolutional Neural Network (LeNet)

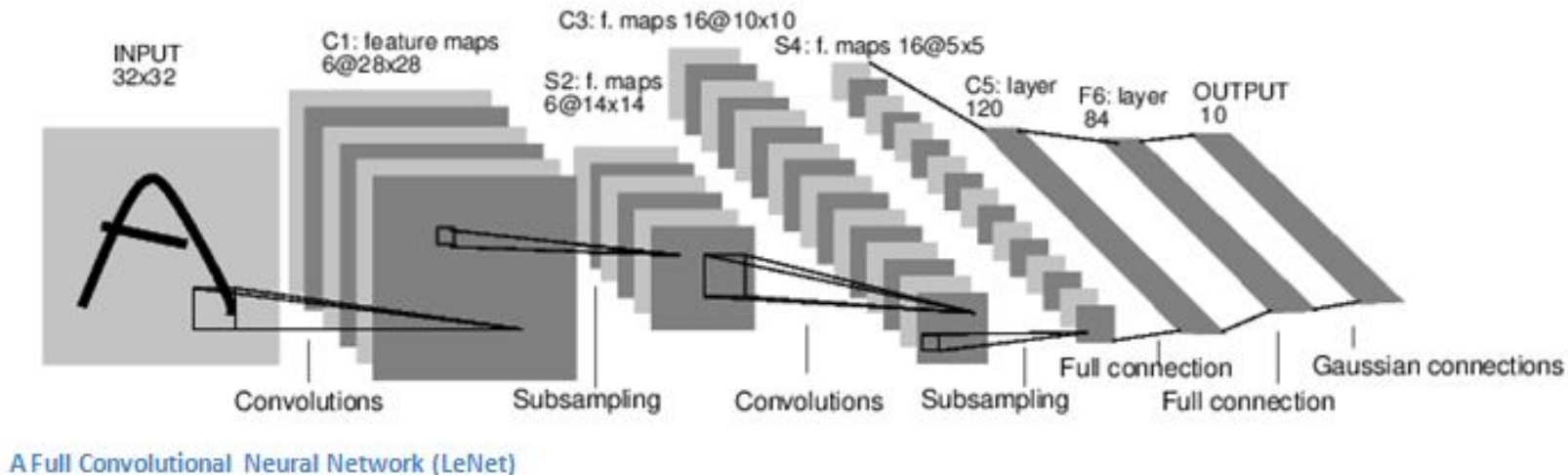
- C1,C3,C5 : Convolution
 - Ma trận convolution 5×5 , stride 1×1 .
- S2 , S4 : Subsampling.
 - Subsampling 2×2 , stride 2×2 .
- F6 : neural net bình thường.



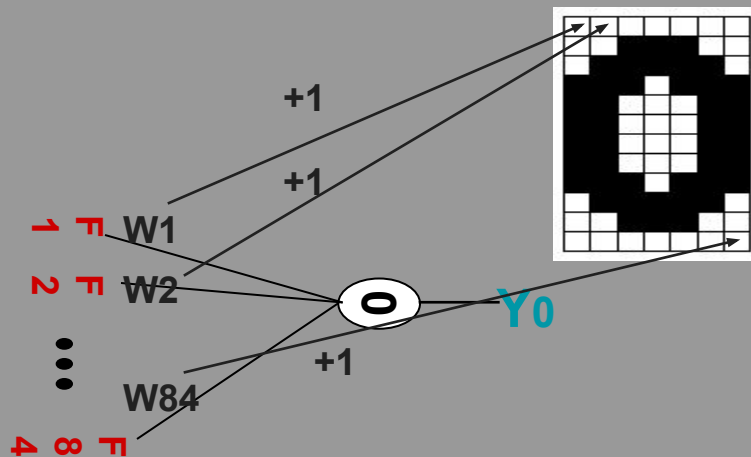
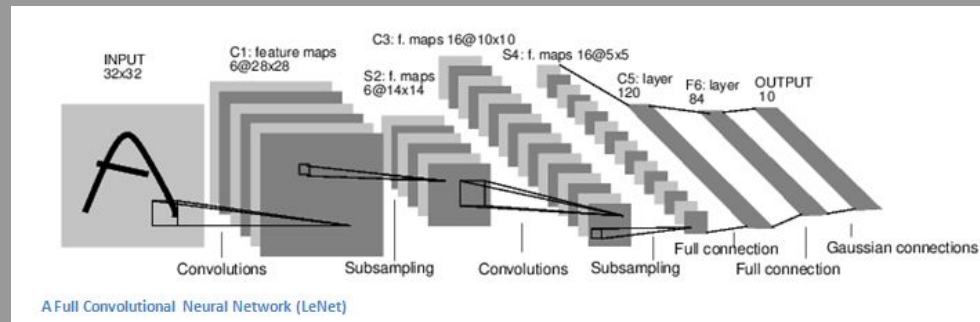
A Full Convolutional Neural Network (LeNet)

LeNet5

Hàm kích hoạt từ đầu đến lớp F6 là hàm sigmoid

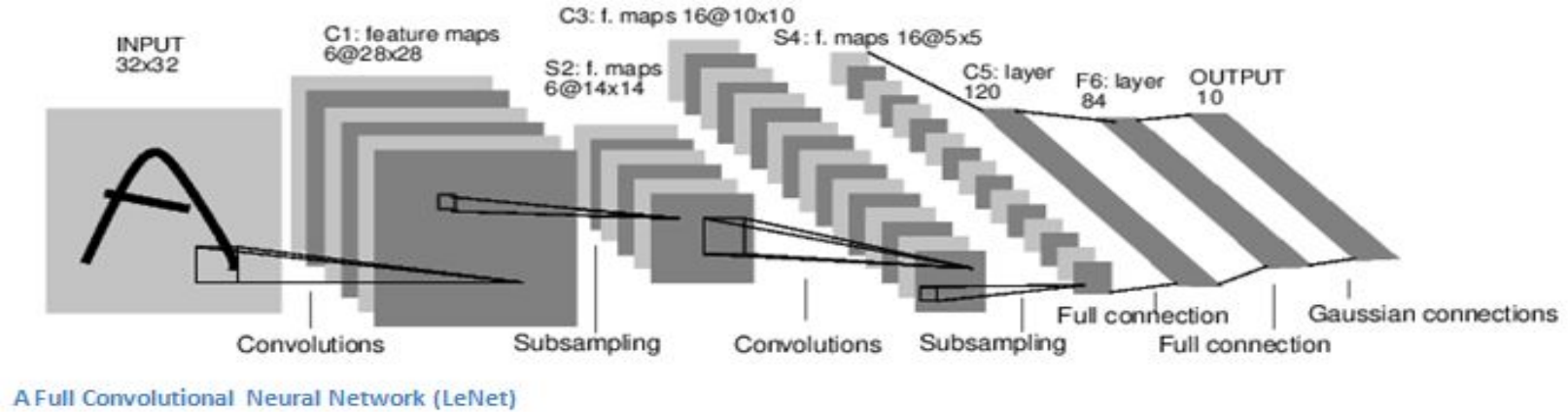


LeNet5: Lớp F6

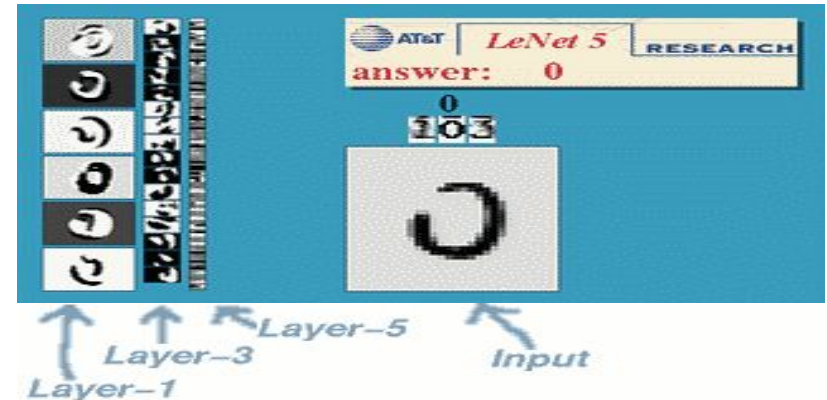


$$Y_j = \sum_{i=1}^{84} (F_i - W_{ij})^2, j = 0, \dots, 9$$

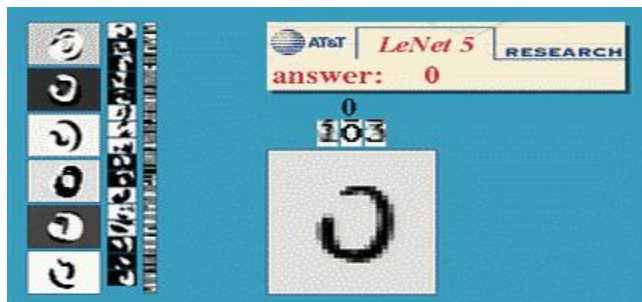
LeNet5: những con số



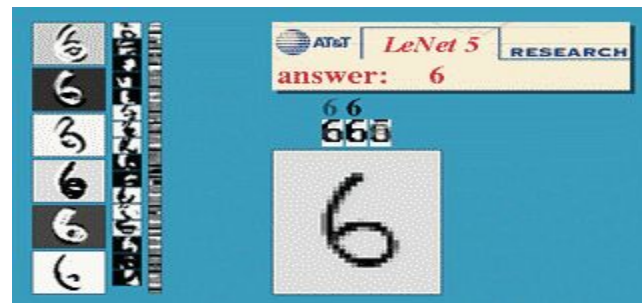
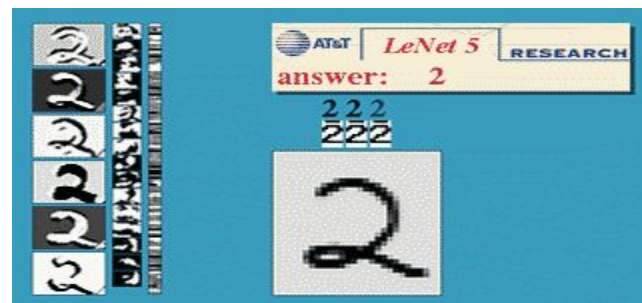
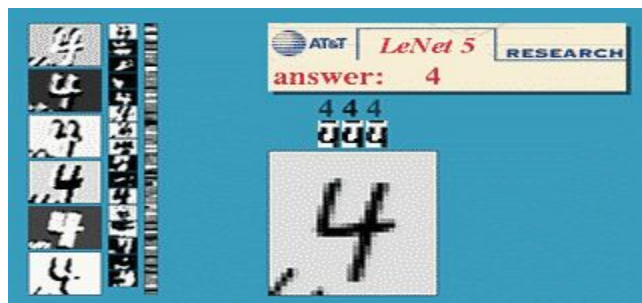
- Khoảng 187,000 liên kết.
- Khoảng 14,000 trọng số cần học.



LeNet5



LeNet5



Bất lợi

Bất lợi

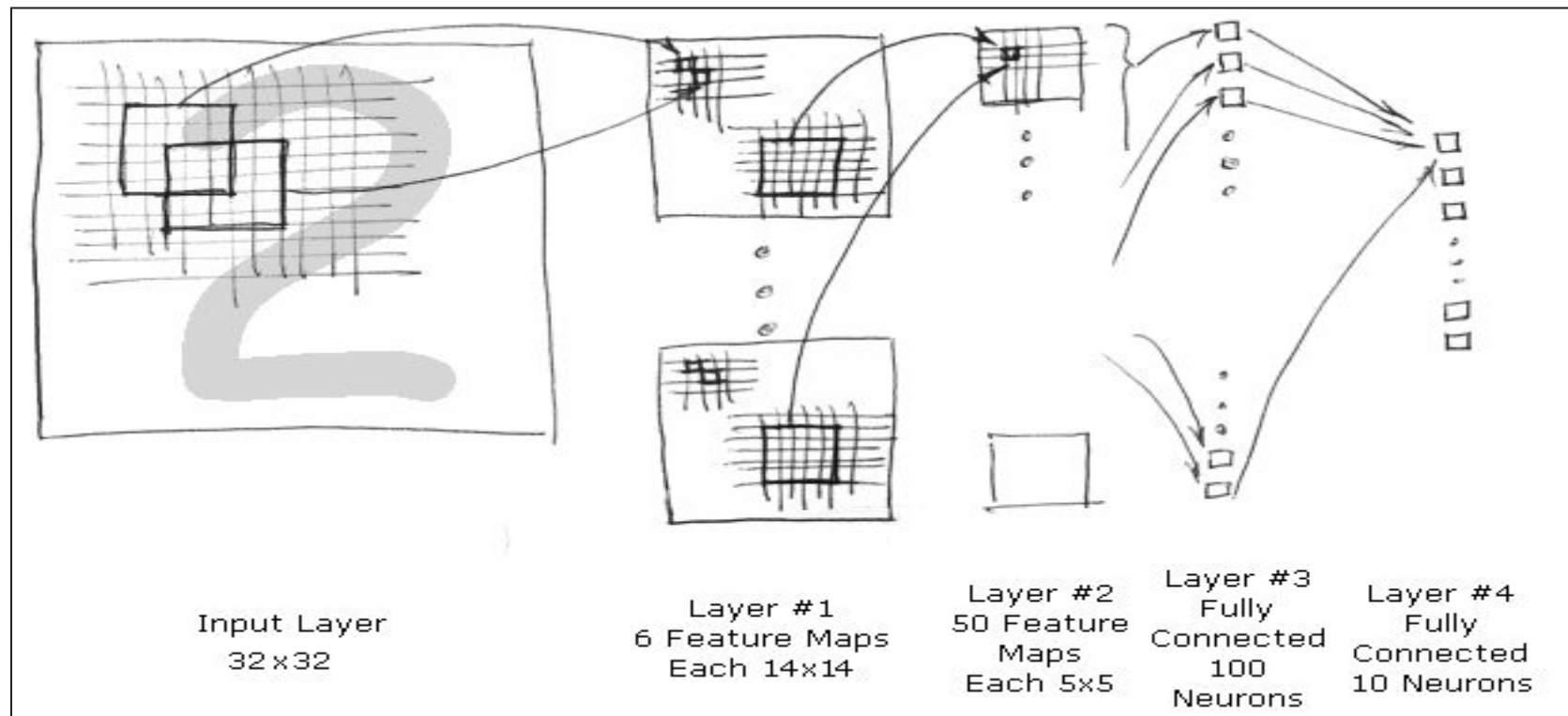
- Về bộ nhớ và năng suất thì CNN không lớn hơn lưới neural 2 lớp nhiều
- Khi chạy, tích chập cần nhiều tính toán và tốn đến 67% thời gian chạy
- So với lưới neural cùng cỡ thì CNN chậm hơn khoảng 3 lần

Bất lợi


















- **Tích chập cần 4 vòng lồng nhau:**
 - 2 vòng trên ảnh đầu vào (2D)
 - 2 vòng trên kernel (2D)
- **Kernel nhỏ làm vòng lặp trong cùng rất chậm vì thường xuyên phải dùng lệnh JMP**
- **Khó lưu trong bộ nhớ (memory cache)**
 - Thuật toán back-propagation cần sử dụng ảnh đầu vào và ảnh kernel theo cả chiều dọc lẫn chiều ngang
 - Ảnh 2D được đại diện theo dãy theo chiều ngang
 - Truy cập theo chiều dọc sẽ bị trượt nhiều

Ứng dụng

Ứng dụng



Ứng dụng

				
3818	6597			
0 => 6 0 => 7				
				
2018	2182	5457		
1 => 7 1 => 3 1 => 8				
				
4176	8059	8094	9664	
2 => 7 2 => 1 2 => 8 2 => 7				
				
1681	2280	4740		
3 => 7 3 => 5 3 => 5				
				
247	2130	8520	8527	9792
4 => 6 4 => 9 4 => 9 4 => 9 4 => 9				

Ứng dụng - Computer vision

- Nhận diện khuôn mặt
- Scene labelling
- ...

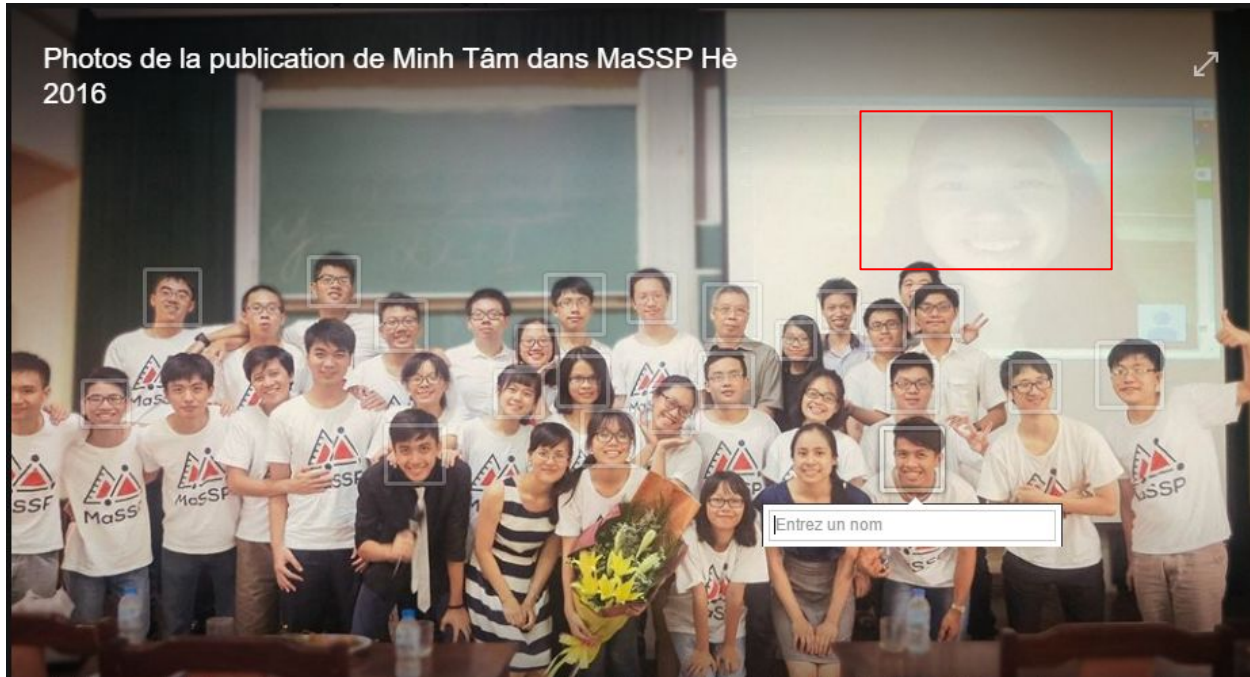
Ứng dụng - Computer vision

Nhận diện khuôn mặt



Ứng dụng - Computer vision

Nhận diện khuôn mặt



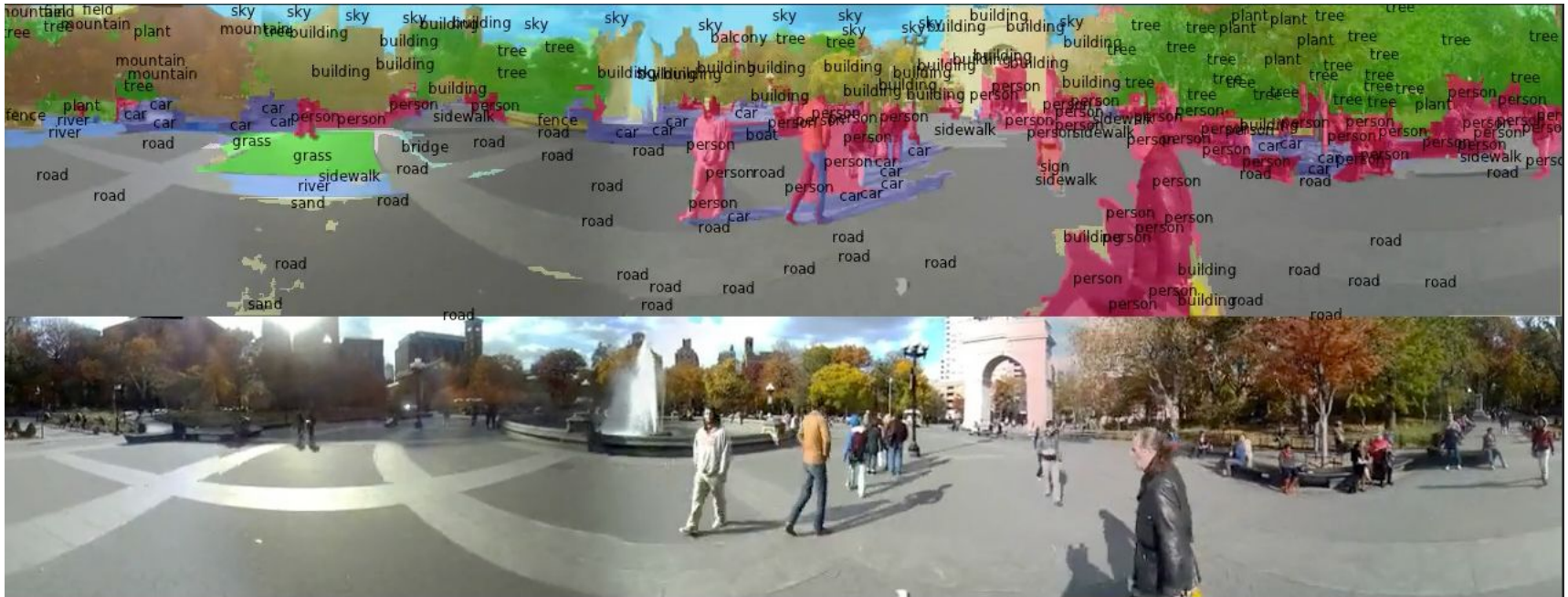
Ứng dụng - Computer vision

Nhận diện khuôn mặt



Ứng dụng - Computer vision

Scene labelling - gắn từng pixel trên ảnh hoặc video với đồ vật (nhà, xe, cây cối) hoặc con người.



Ứng dụng - Natural Language Processing

- Speech Recognition
- Text classification
 - Tutorial: <https://www.tensorflow.org/tutorials/word2vec>

Q&As