

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

— * —

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

**Ứng dụng học sâu trong việc tự động xác định
địa điểm du lịch nổi tiếng**

Sinh viên thực hiện : **Lê Văn Mạnh**
Lớp KSVB2 – K37
Giáo viên hướng dẫn: **GV.Đinh Viết Sang**

HÀ NỘI 6-2019

Mục lục

CHƯƠNG 1 - MỞ ĐẦU	4
1.1. Tính cấp thiết của đề án	4
1.2. Nhiệm vụ của đề án.....	4
1.3. Đối tượng và phạm vi nghiên cứu	4
1.4. Phương pháp thực hiện.....	4
1.5. Ý nghĩa khoa học và thực tiễn.....	4
1.6. Kết quả dự kiến	4
1.7. Bố cục của đề án.....	4
CHƯƠNG 2 - CONVOLUTIONAL NEURAL NETWORK	5
2.1. Giới thiệu về CNN.....	5
2.2. Cấu trúc tổng quan của mạng CNN	5
2.3. CNN hoạt động như thế nào.....	5
2.4. Convolution là gì	6
2.5. Stride và Padding.....	8
2.6. Pooling là gì.....	8
2.7. Fully connected neural network	9
CHƯƠNG 3 – BÀI TOÁN NHẬN DẠNG ĐỐI TƯỢNG.....	9
3.1. Đầu vào và đầu ra của bài toán	9
3.2. Một số kiến trúc mạng convolutional neural network	9
CHƯƠNG 4 – THIẾT KẾ HỆ THỐNG	13
4.1. Biểu đồ ca sử dụng	13
4.2. Biểu đồ hoạt động.....	13
4.3. Biểu đồ tuần tự	13
CHƯƠNG 5 – ĐÁNH GIÁ KẾT QUẢ	14
5.1. Giao diện trưng trình	14
5.2. Minh họa chức năng phát hiện vi phạm luật giao thông	14
5.3. Độ chính xác của hệ thống	14
5.4. Hướng phát triển trong tương lai.....	14

CHƯƠNG 6 - TÀI LIỆU THAM KHẢO	15
-------------------------------------	----

CHƯƠNG 1 - MỞ ĐẦU

1.1. Tính cấp thiết của đề án

Type Here

1.2. Nhiệm vụ của đề án

Type Here

1.3. Đối tượng và phạm vi nghiên cứu

Type Here

1.4. Phương pháp thực hiện

Type Here

1.5. Ý nghĩa khoa học và thực tiễn

Type Here

1.6. Kết quả dự kiến

Type Here

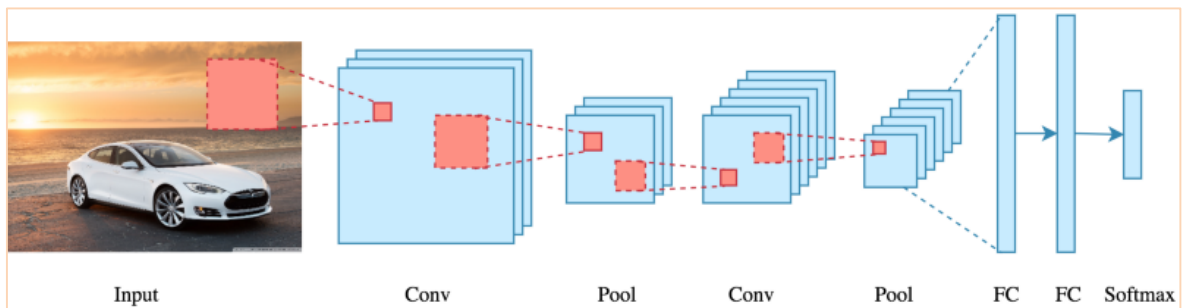
1.7. Bố cục của đề án

CHƯƠNG 2 - CONVOLUTIONAL NEURAL NETWORK

2.1. Giới thiệu về CNN

Convolutional Neural Network (CNN – 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 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.

2.2. Cấu trúc tổng quan của mạng CNN



Hình 2.2.1 Sơ đồ tổng quát CNN

2.3. CNN hoạt động như thế nào

Local receptive fields: Trong mạng neural network truyền thống mỗi một neural trong input layer kết nối với một neural trong hidden layer. Tuy nhiên trong CNN chỉ một vùng xác định trong các neural trong input layer kết nối với một neural trong hidden layer. Những vùng xác định nêu trên gọi là Local receptive fields. Sự kết nối giữa input layer và hidden được chính là việc từ Local receptive fields trên một ảnh đầu vào được biến đổi thông qua một phép toán được gọi là convolution để thu được một điểm trên hidden layer.

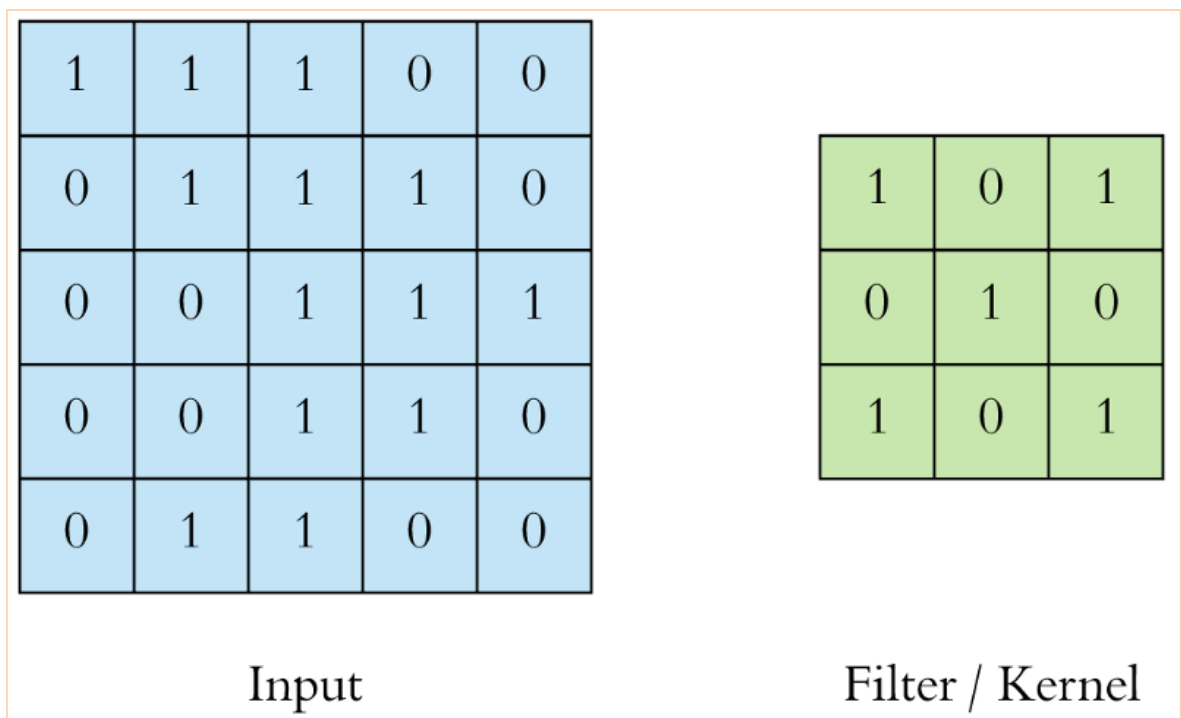
Shared weights và biases: Giống với mạng neural network truyền thống CNN cũng có tham số weights và biases. Các tham số này được học trong suốt quá trình training và liên tục cập nhật giá trị với mỗi mẫu mới (new training example). Tuy nhiên, các trọng số trong CNN là giống nhau đối với mọi neural trong cùng một lớp (layer). điều này có nghĩa là tất cả các hidden

neural trong cùng một lớp đang cùng tìm kiếm trung một đặc trưng (ví dụ như cạnh của ảnh) trong các vùng khác nhau của ảnh đầu vào.

Activation và pooling: Activation là một bước biến đổi giá trị đầu ra của mỗi neural thông qua việc sử dụng một số hàm ví dụ hàm ReLU. Giá trị thu được sau phép biến đổi là giá trị dương nhất có thể của output, trong trường hợp output mang giá trị âm thì giá trị nhận được là 0.

pooling là một bước nhằm giảm số chiều của ma trận, các thức phổ biến nhất là từ một vùng trên ma trận ta chọn ra số có giá trị lớn nhất làm kết quả thu được sau bước pooling (max pooling)

2.4. Convolution là gì

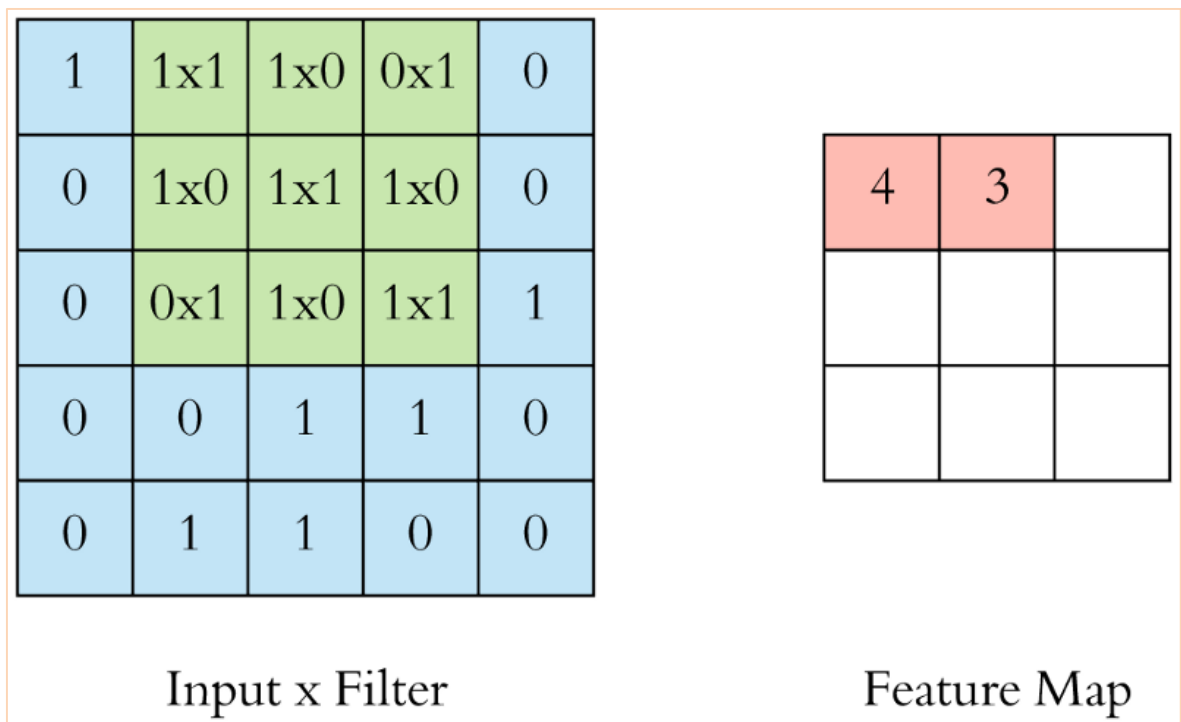


Hình 2.4.1 Convolutional là gì

Khối cơ bản tạo lên CNN là convolutional layer. Convolution là một phép toán học để kết hợp hai khối thông tin với nhau. Trong trường hợp này, convolution được áp dụng trên dữ liệu đầu vào (ma trận) và sử dụng một mặt nạ gọi là convolution filter để tạo ra một mảng mới gọi là feature map.

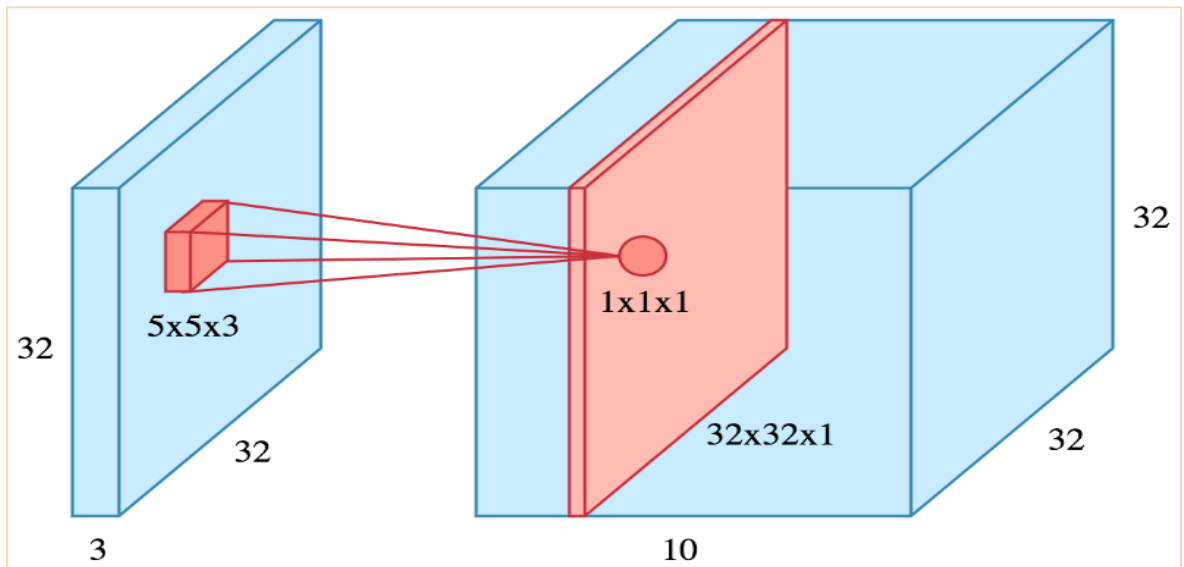
Việc thực hiện phép toán convolution được mô tả như hình dưới đây với đầu vào là một mảng hai chiều 5x5 phần tử là filter có kích thước là 3x3 phần tử.

Cửa sổ filter sẽ được trượt từ trái qua phải, từ trên xuống dưới. Tại mỗi vị trí của cửa sổ filter ta thực hiện nhân tương ứng từng phần tử trong ma trận đầu vào với từng phần tử trong filter, sau đó cộng tổng các tích với nhau ta thu được kết quả là một phần tử trên feature map. Quá trình thực hiện được mô tả trong hình minh họa sau đây.



Hình 2.4.2 Convolutional và mảng hai chiều

Trên đây là mô tả thực hiện phép toán convolution với ma trận hai chiều với một filter duy nhất. Trong thực tế đối với ảnh RGB ta thực hiện convolution với ma trận ba chiều ví dụ như ảnh RGB và với cùng một ảnh đầu vào ta áp dụng phép toán convolution với nhiều filter khác nhau. Mỗi một filter được áp dụng cho ta một feature layer. Nhiều feature layer xếp chồng lên nhau ta thu được một convolution layer. Ví dụ sau thể hiện ảnh có kích thước 32x32 và có ba kênh màu, ta sử dụng 10 filter và thu được convolution layer là một ma trận 32x32x10.



Hình 2.4.3 Convolutional và mảng ba chiều

2.5. Stride và Padding

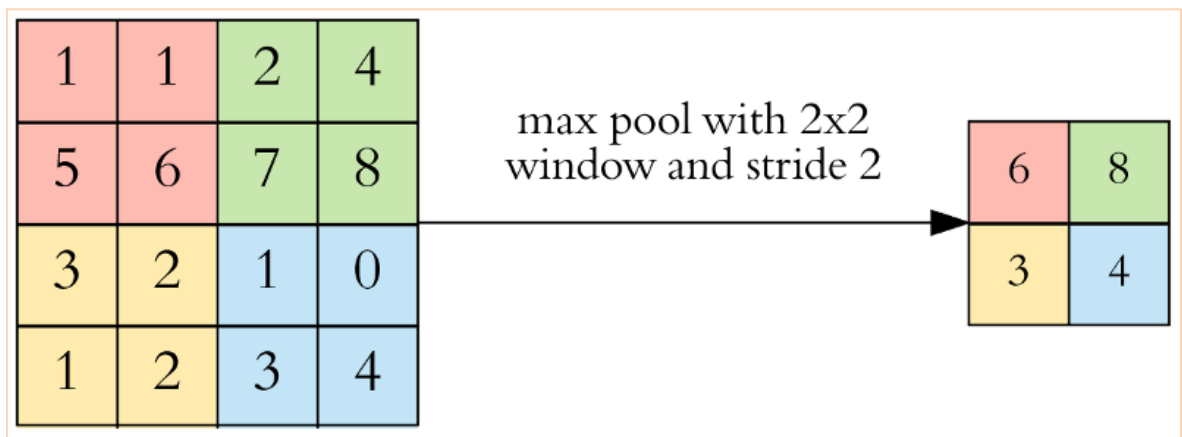
Stride là số bước nhảy của mỗi lần dịch chuyển convolution filter, trong ví dụ đầu tiên về convolution ta nhận thấy kích thước của feature map nhỏ hơn kích thước của ma trận đầu vào. Để kích thước của feature map bằng kích thước của ma trận đầu vào ta cần phải bổ xung thêm một số điểm bao quanh ma trận đầu vào thường là thêm các phần tử 0 vào xung quanh ma trận đầu vào, thao tác trên được gọi là padding.

Khi thực hiện phép toán convolution với đầu vào là ma trận vuông có kích thước là $n \times n$, stride là s , kích thước filter là $f \times f$, vùng padding có kích thước là p ta có kích thước của feature map thu được là:

$$Output\ size = \left(\frac{n + 2p - f}{s} + 1 \right) \times \left(\frac{n + 2p - f}{s} + 1 \right)$$

2.6. Pooling là gì

Sau khi thực hiện phép toán convolution chúng ta thường sử dụng pooling nhằm giảm số chiều của dữ liệu. Loại pooling thông dụng nhất là max pooling tức là trong một vùng được chọn (pooling window) được chọn của ma trận, ta lấy phần tử có kích thước lớn nhất, cũng giống với convolution thì pooling window cũng được định nghĩa kích thước (size) và bước nhảy (stride). Dưới đây là ví dụ việc áp dụng max pooling sử dụng 2×2 window và stride là 2.



Hình 2.6.1 Minh họa max pooling

2.7. Fully connected neural network

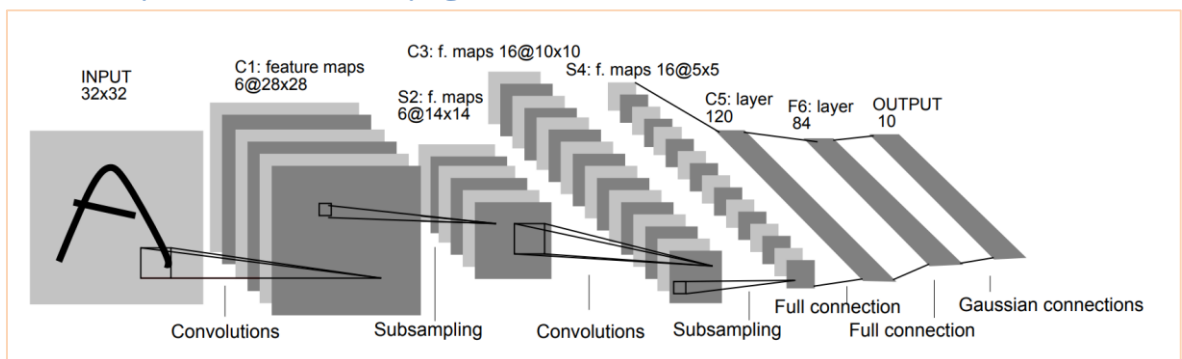
Sau các lớp convolution + pooling layers chúng ta sẽ gắn thêm một mạng Artificial Neural Network nhằm phục vụ quá trình training và nhận diện đối tượng. Thông qua quá trình học CNN tự động sẽ cập nhật lại giá trị cho các filter, weight matrix W và bias b .

CHƯƠNG 3 – BÀI TOÁN NHẬN DẠNG ĐỐI TƯỢNG

3.1. Đầu vào và đầu ra của bài toán

Type here

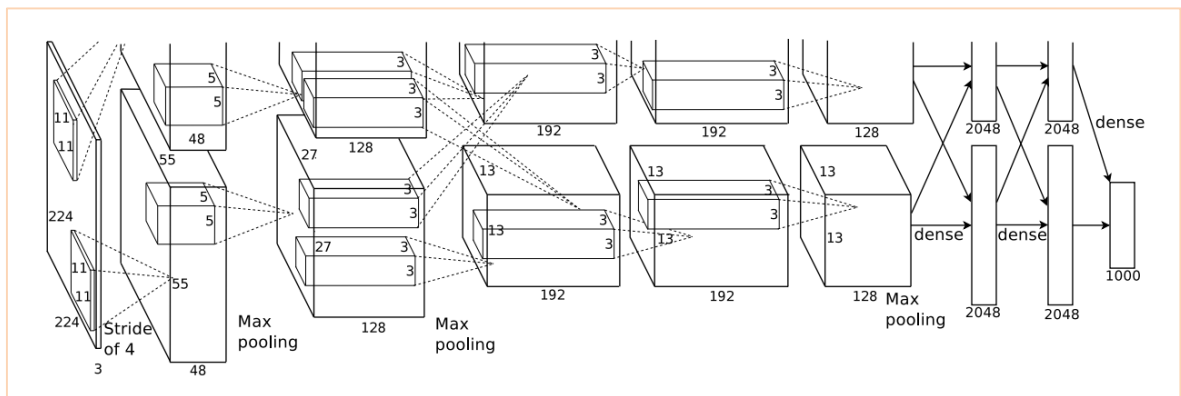
3.2. Một số kiến trúc mạng convolutional neural network



Hình 4.2.1 Sơ đồ kiến trúc mạng LeNet-5

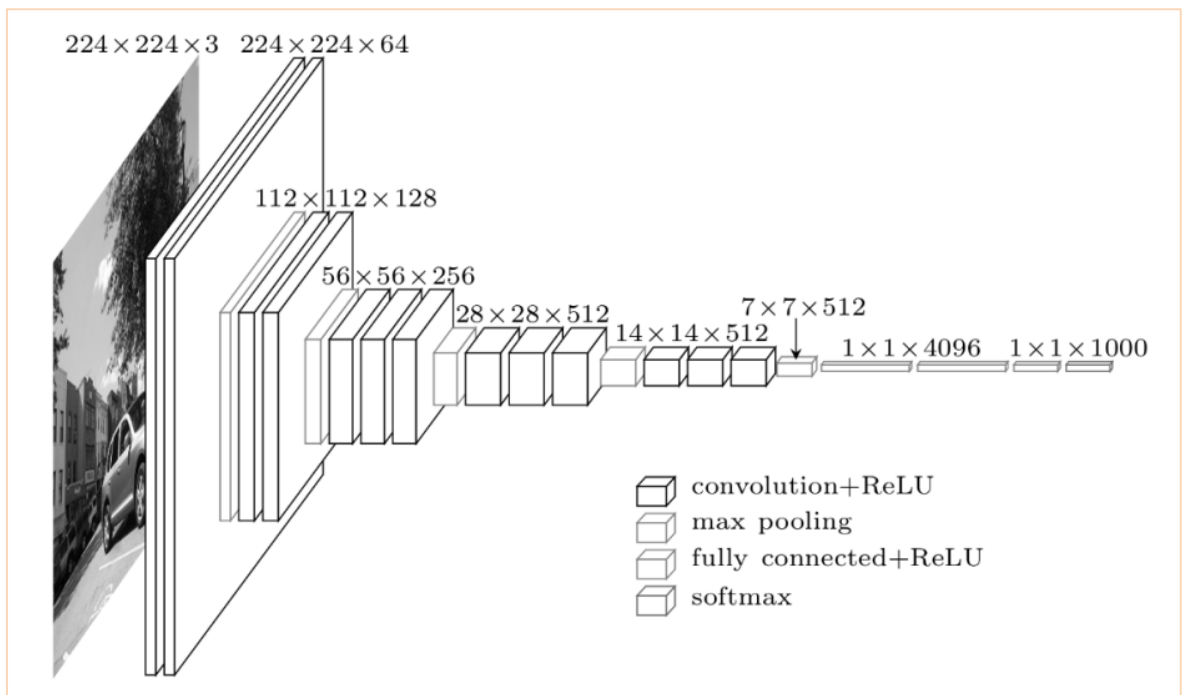
Ảnh đầu vào có kích thước 32x32x1. Layer C1 chứa 6 feature map kích thước 28x28, mỗi phần tử tại lớp này được kết nối với 5x5 phần tử tại lớp đầu vào. Số lượng parameters cần học là $(5 * 5 + 1) * 6 = 156$. Lớp S2 chứa 6 feature map với kích thước 14x14, mỗi phần tử tại lớp này kết nối với 2x2 phần tử tại lớp C1. Số lượng parameters cần học là $6 * 2 = 12$. Lớp C3 chứa 16 feature

map kích thước là 10x10, mỗi phần tử kết nối với 5x5 phần tử tại layer S2. Số lượng parameters cần học là $(5 * 5 * 6 + 1) * 16 = 1516$. Lớp S4 chứa 16 feature map kích thước 5x5, mỗi phần tử kết nối với 2x2 tương ứng ở lớp C3. Số lượng parameters cần học là $16 * 2 = 32$. Lớp C5 chứa 120 feature map mỗi phần tử tại lớp này kết nối với 5x5 phần tử lại lớp S4. C5 có $120 * (16 * 25 + 1) = 48120$. F6 là một lớp của một full connected neural network có 84 phần tử và $84 * (120 + 1) = 10164$ parameters cần học. Lớp cuối cùng là đầu ra của neural network. Các trọng số (còn được gọi là các parameters cần học) được cập nhật bởi thuật toán backpropagation.



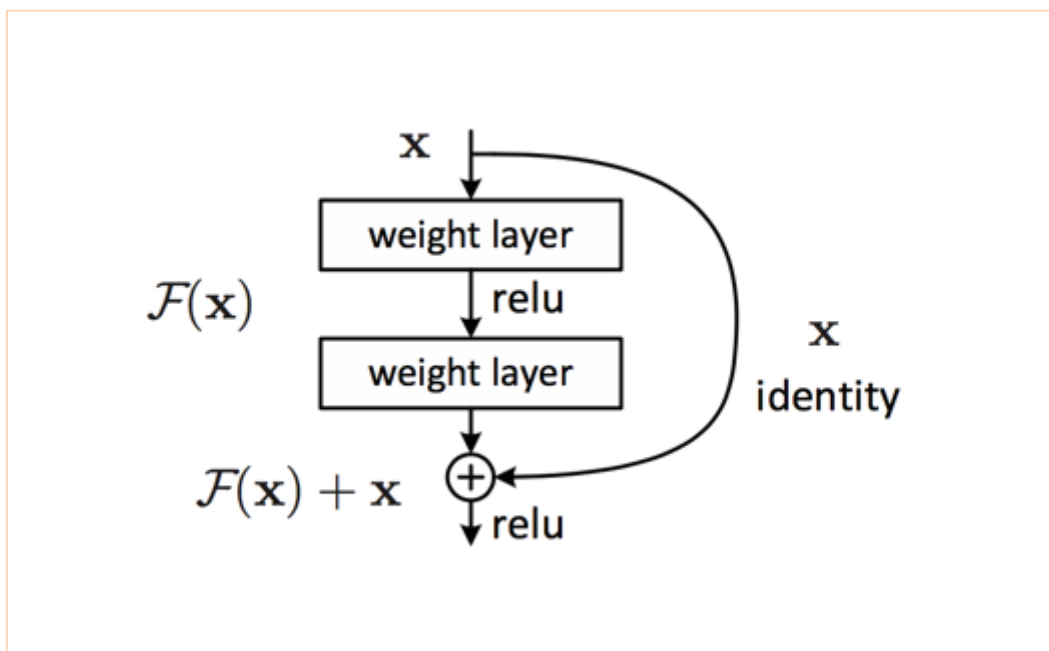
Hình 4.2.2 Sơ đồ kiến trúc mạng AlexNet

Mạng có 8 lớp với 5 lớp đầu là các convolutional layer và ba lớp tiếp theo là ba lớp thuộc mạng fully-connected neural. Đầu vào của mạng là ảnh có kích thước 224x224x3 pixel. Lớp convolutional layer đầu tiên là kết quả của việc thực hiện phép toán convolution và 96 kernel với kích thước 11x11x3 và bước nhảy tride bằng 4 pixels. Lớp thứ hai của mạng được tạo bởi việc thực hiện phép toán convolution với 256 kernel kích thước là 5x5x48. Lớp ba, bốn, năm là các lớp convoluion được kết nối từ layer trước sang layer sau không thông qua phép toán pooling. Lớp thứ ba có 384 kernels với kích thước 3x3x128. Lớp thứ tư có 384 kernels với kích thước 3x3x192. Lớp thứ 5 có 256 kernels với kích thước 3x3x192. Cuối cùng mạng được kết nối với một mạng fully-connected neural với 4096 neurons trong mỗi lớp.



Hình 4.2.3 Sơ đồ kiến trúc mạng VGG16

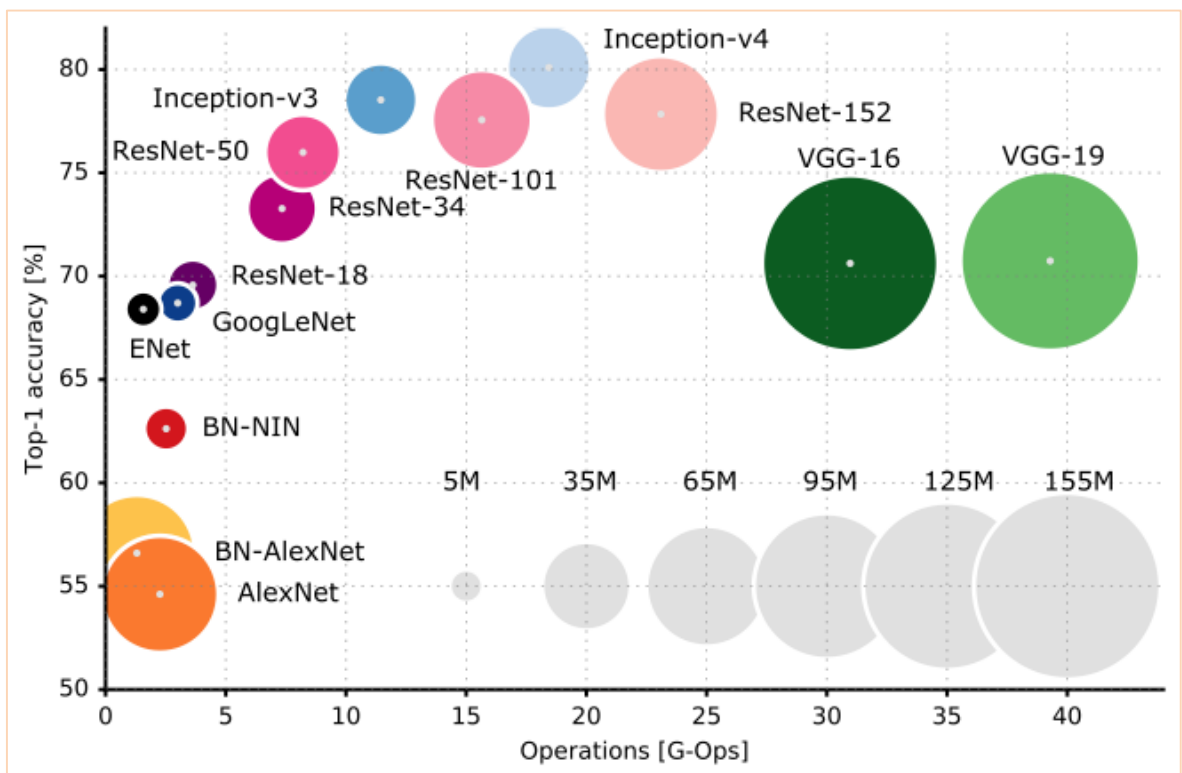
ResNet



Hình 4.2.4 Sơ đồ phần tử Residual Block



Hình 4.2.5 Sơ đồ kiến trúc mạng ResNet



Hình 4.2.6 So sánh độ chính xác và khối lượng tính toán

CHƯƠNG 4 – THIẾT KẾ HỆ THỐNG

4.1. Biểu đồ ca sử dụng

4.2. Biểu đồ hoạt động

4.3. Biểu đồ tuần tự

CHƯƠNG 5 – ĐÁNH GIÁ KẾT QUẢ

5.1. Giao diện trưng trình

5.2. Minh họa chức năng phát hiện vi phạm luật giao thông

5.3. Độ chính xác của hệ thống

5.4. Hướng phát triển trong tương lai

CHƯƠNG 6 - TÀI LIỆU THAM KHẢO

- [1] <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>
- [2] <https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html>
- [3] <https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524>
- [4] <https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/>
- [5] <http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/>
- [6] <https://www.kaggle.com/shivamb/cnn-architectures-vgg-resnet-inception-tl>