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

1.1. Tính cấp thiết của đồ án

Hiện nay để biết về một địa danh hay một điểm du lịch thông thường người dùng sẽ lên các trang tìm kiếm ví dụ google.com, bing.com …sau đó gõ từ khóa tên hoặc địa điểm du lịch muốn tới và sau đó đọc các thông tin liên quan tới địa điểm du lịch. Tuy nhiên, với sự bùng nổ của các mạng xã hội, các ứng dụng di động cùng với sự đa dạng về loại dữ liệu đặc biệt là dữ liệu ảnh nhiều trường hợp người dùng chỉ có một bức ảnh về địa điểm du lịch hoặc muốn tới một nơi có phong cảnh đẹp như trong bức ảnh mình đang có. Điều trên dẫn tới nhu cầu tìm kiếm thông tin địa danh thông qua hình ảnh ngày càn phổ biến.

1.2. Nhiệm vụ của đồ án

Đồ án được thực hiện nhằm mục đích làm cho việc tìm kiếm địa điểm du lịch và danh lam thắng cảnh của Việt Nam trở lên dễ dàng và thuận tiện. Trong phạm vi đồ án này em sẽ xây dụng một hệ thống cho phép người dùng tìm kiếm địa điểm du lịch bằng hình ảnh. Giúp đưa ra thông tin về địa điểm du lịch cũng như các địa danh khác có phong cảnh tương tự.

1.3. Đối tượng và phạm vi nghiên cứu

Đồ án thực hiện trên dữ liệu ảnh về các địa điểm du lịch nổi tiếng, mỗi địa danh sẽ chứa khoảng 1000 ảnh kèm với thông tin về địa lý liên quan tới địa danh đó. Do giới hạn về thời gian và nền tảng phần cứng nên hệ thống xây dựng để nhận diện và gợi ý 64 địa điểm du lịch khác nhau trên lãnh thổ Việt Nam.

1.4. Phương pháp thực hiện

Với bài toán nhân diện thông tin qua ảnh việc lập trình truyền thống sẽ khó có độ chính xác cao do độ phức tạp và đặc thù của thông tin dữ liệu. Qua một thời gian tìm hiểu công nghệ, em lựa chọn phương pháp xây dựng hệ thống với phần lõi nhận diện ảnh sẽ sử dụng trí tuệ nhân tạo để đạt độ chính xác cao nhất có thể.

1.5. Ý nghĩa khoa học và thực tiễn

Người dùng khi xem ảnh có thể ngay lập tức tìm kiếm thông tin địa danh thông qua hình ảnh mình đang xem. Mọi thứ sẽ trờ lên nhanh chóng và thuận tiện cho người sử dụng góp phần thúc đẩy ngành dịch vụ và du lịch của Việt Nam.

1.6. Kết quả dự kiến

Xây dựng mạng neuron nhân tạo nhận diện 64 địa danh thông qua hình ảnh với độ chính xác trên 80%, triển khai hệ thống trên nền tăng web cho người dùng truy cập và upload ảnh lên sau đó trả lại kết quả cho người dùng trên giao diện website.

1.7. Bố cục của đồ án

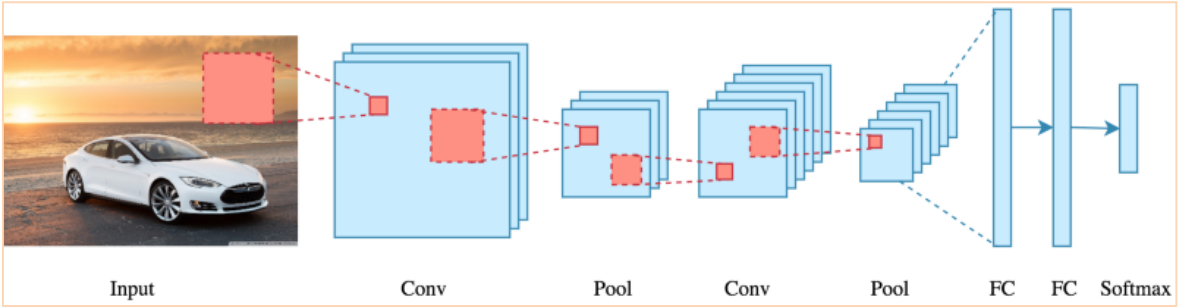
Đồ án được trình bày gồm các phần như sau:

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 Sở đồ tổng quát CNN

2.3. CNN hoạt động như thế nào

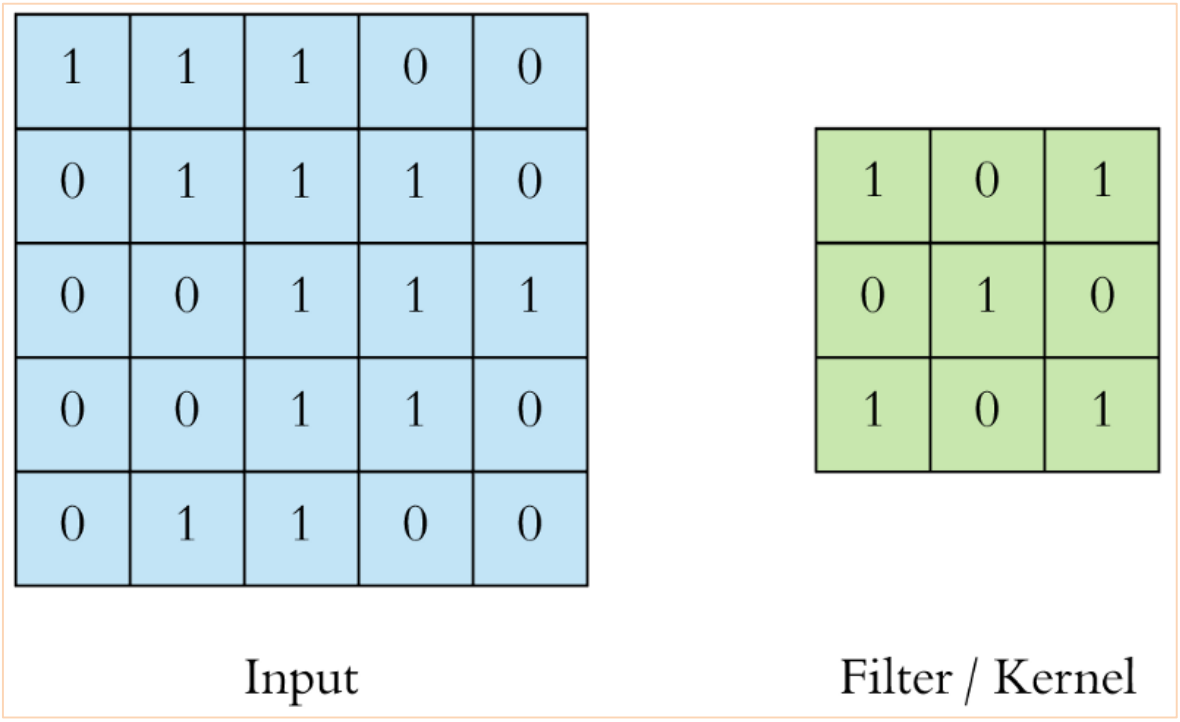
Local receptive fields: Trong mạng neural network truyền thống mỗi một neural trong input layer kết nối với một neural trọng hidden layer. Tuy nhiên trong CNN chỉ một vùng xác định trọng các neural trong input layer kết nối với một neural trong hidden layer. Những vùng xác định nêu trên gọi là Local receptive fields. Sự kết nỗi giữa input layer và hidden được chính là viêc 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 trainning và liên tục cập nhật giá trị với mỗi mẫu mới (new trainning example). Tuy nhiên, các trọng số trong CNN là giống nhau đối với mọi neural trọng cùng một lớp (layer). điều này có nghĩ 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 hàn ví dụ hàm ReLU. Giá trị thu được sau phép biến đổi là giá trị dương nhất có thể của output, trong trường hợp ouput 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

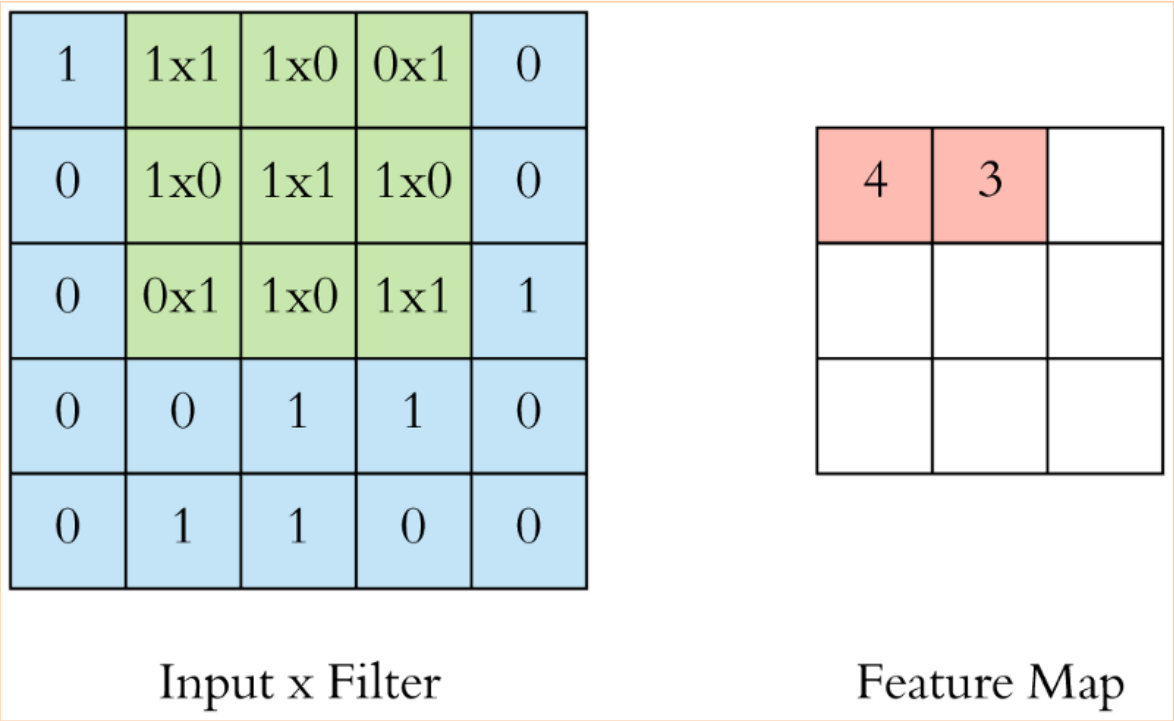


Hình 2.4.1 Convolutional là gì

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 lạ gọi là convolution filter để tạo ra một mảng mới gọi là feature map.

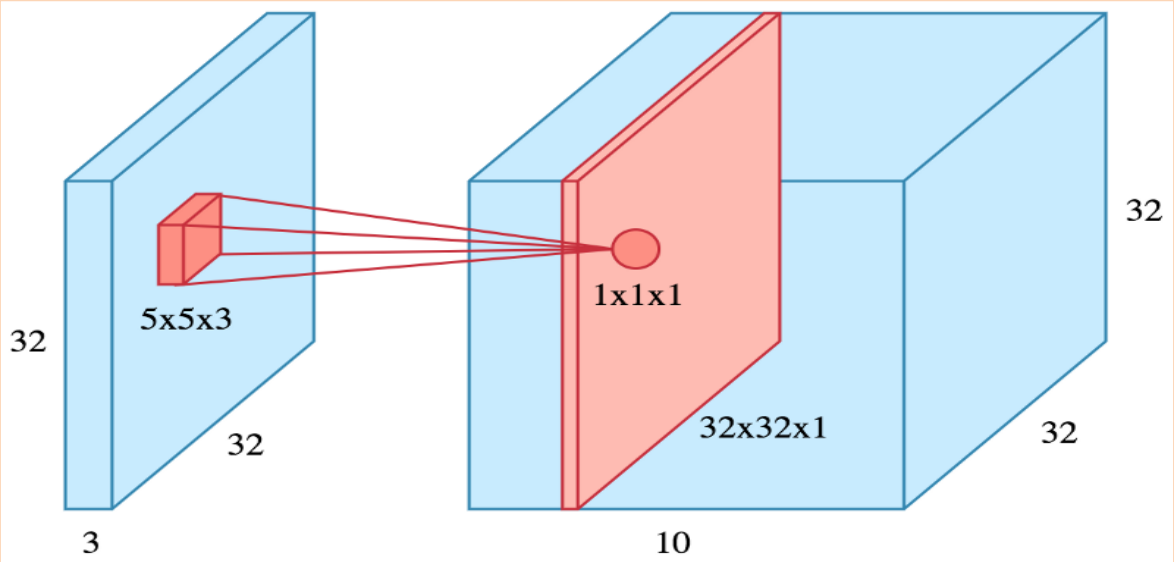
Việc thực hiện phép toán convolution được mô tả như hình dưới dây với đầu vào là một mảng hai chiều 5x5 phần tử là filter có kích thước là 3x3 phần tử. Của sổ filter sẽ được trượt từ trái qua phải, từ trên xuống dưới. Tại mỗi vị trí của cửa sổ filter ta thực hiện nhân tương ứng từng phần tử trong ma 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 trồng lên nhau ta thu được một convolution layer. Ví dụ sau thể hiện ảnh có kích thước 32x32 và có ba kênh mầu, 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

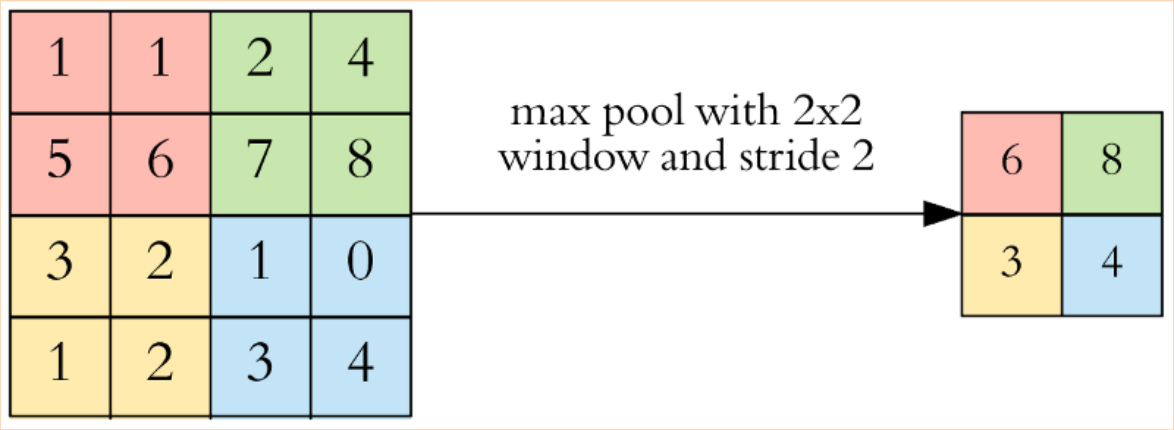
Stride là số bước nhảy của mỗi lần dịch chuyển convolution filter, trong ví dụ đầu tiên về convolution ta nhận thấy kích thước của feature map nhỏ hơn kích thước của ma trận đầu vào. Để kích thước của feature map bằng kích thước của ma trận đầu vào ta cần phải bổ xung thêm một số điểm bao quanh ma trận 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 nxn, stride là s, kích thước filter là fxf, vùng padding có kích thước là p ta có kích thước của feature map thu được là:

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

2.6. Pooling là gì

Sau khi thực hiện phép toán convolution chúng ta thường sủ 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 nhảu (tride). Dưới đây là ví dụ việc áp dụng max pooling sử dụng 2x2 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 tranning 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

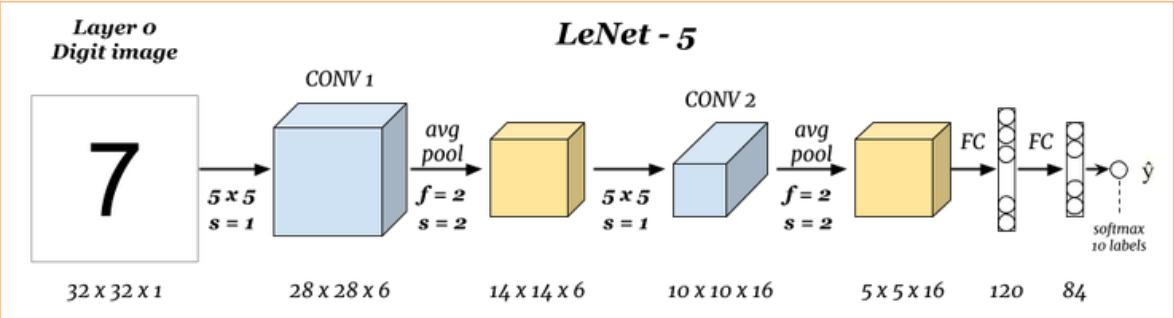
Đầu vào bài toán em chia thành hai dạng đó là đầu vào dữ liệu (dataset) dùng cho việc huấn luyện mạng neuron và đầu vào của bài toán cần phải nhận diện.

Dữ liệu đầu vào dùng cho huấn luyện mạng hay còn gọi là dataset là tập ảnh có kích tướng 480x480x3, đây là ảnh RGB về các địa danh du lịch nổi tiếng của Việt Nam. Như đã nói ở phần mởi đầu trong phần pham vi của đồ án, dữ liệu đầu vào chứa 64 loại ảnh về 64 địa danh tương ứng khác nhau. Các ảnh được lưu chữ trong từng thư mục khác nhau với mỗi thư mục chứa khoảng 1000 ảnh. Quá trình huấn luyện mạng sẽ chia tập dữ liệu thành hai phần, trong đó 70% dùng cho việc huấn luyện mạng và 30% dùng cho việc kiểm định độ chính xác của mạng. Trong quá trình huấn luyện và kiểm định độ chính xác của ảnh, các ảnh sẽ được thay đổi kích thước một cách hợp lý để phù hợp với thiết kế của mạng neuron. Do tập ảnh với số lượng 1000 cho mỗi địa danh là chưa đủ lớn để tăng số lượng ảnh cho việc huấn luyện em sẽ sử dụng một số kĩ thuật như lật, xoay ảnh gốc để thu được một ảnh khác nhằm tăng số lượng ảnh cho việc huấn luyện.

Dữ liệu đầu vào dành cho việc nhận dạng địa danh là ảnh của địa danh có kích thước bất kìa với định dạng ảnh yêu cầu là RGB.

3.2. Một số kiến trúc mạng convolutional neural network

LeNet-5 là một mạng cổ điển và cơ bản nhất cho bài toán nhận diện ảnh, được ứng dụng cho bài toán nhận diện số và chữ viết tay với đặc điêm dữ liệu đầu là ảnh có độ chi tiết đơn giản và kích thước nhỏ. Kiến trúc được công bố trong bài báo của Y. Lecun, L. Bottou, Y. Bengio and P. Haffner vào năm 1998.



Hình 4.2.1 Sơ đồ kiến trúc mạng LeNet-5

Sau đây là bảng tóm tắt kiến trúc của mạng, gồm các lớp cùng với các tham số cần phải học tương ứng.

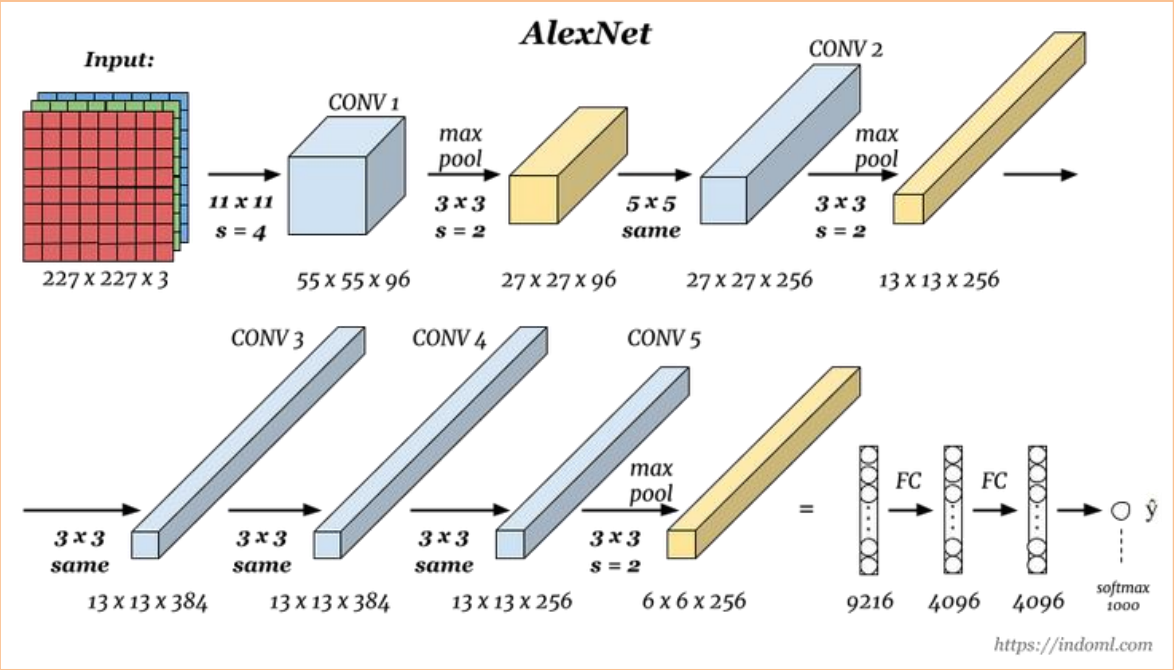
Layer	Filters	Biases	Stride	Padding	Tensor	Parameteres
Input	0	0	0	0	32x32x1	0
Conv-1	5x5x1x6	6	1	0	28x28x6	156
MaxPool -1	2x2	0	2	0	14x14x6	0
Conv-2	5x5x6x16	16	1	0	10x10x16	2416
MaxPool -2	2x2	0	2	0	5x5x16	0
FC-1	400x120	120	0	0	120	48120
FC-2	120x84	84	0	0	84	10164
RBF	0	0	0	0	10	0
Total						60856

Bảng 3.2.1 Kiến trúc mạng LeNet-5

Đầu vào là ảnh Gray với kích thước 32x32 pixel và đâu ra là khoảng các Euclidean giữa mỗi vector đầu vào vector trọng số tức tensor đầu ra của FC-2.

Ta có thể thấy đâu là mạng rất đơn giản với kích thước đầu vào cũng như số lượng các tham số phải học. Tiếp theo chúng ta tìm hiểu một kiến trúc phổ biến khác nhưng phức tạp hơn mạng LeNet-5.

Mạng AlexNet. Dùng để phân loại 1000 loại ảnh có kích thước 227x227x3. Kiến trúc được công bố bởi Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever vào năm 2012



Hình 4.2.2 Sơ đồ kiến trúc mạng AlexNet

Sau đây là bảng tóm tắt các thông số của mạng chứa thông tin về kích thước tensor đầu ra của từng lớp cùng với số lượng tương ứng các tham số mà mạng cần phải học.

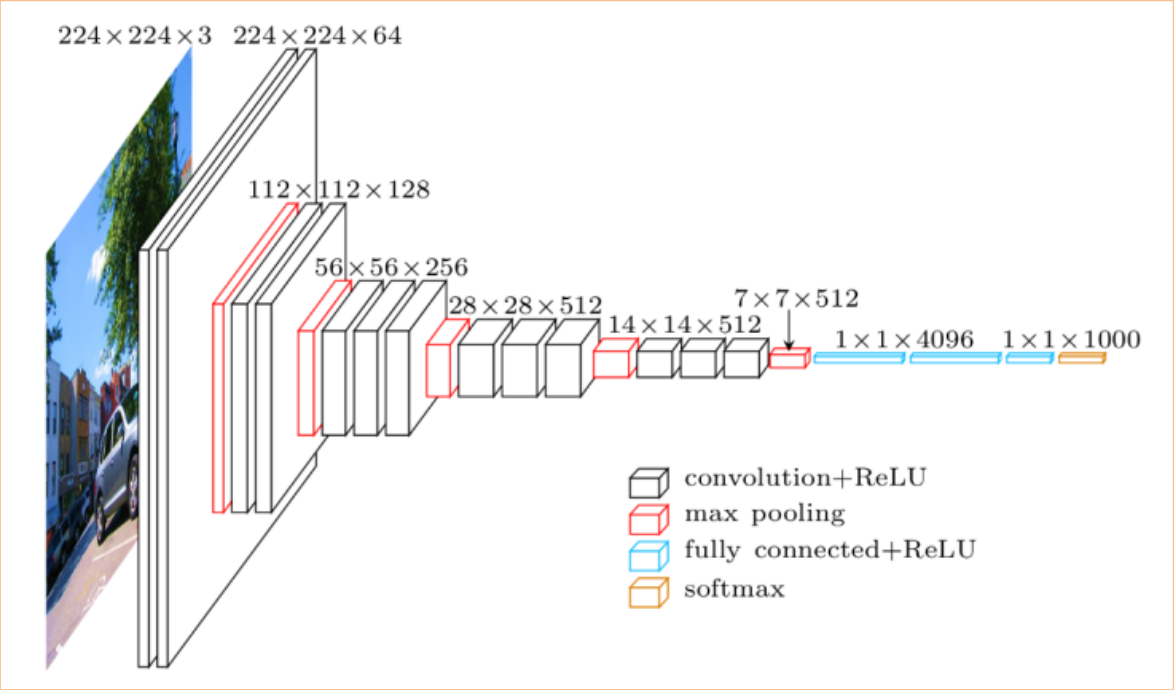
Layer	Filters	Biases	Stride	Padding	Tensor	Parameteres
Input	0	0	0	0	227x227x3	0
Conv-1	11x11x3x96	96	4	0	55x55x96	34944
MaxPool-1	3x3	0	2	0	27x27x96	0
Norm-1	0	0	0	0	27x27x97	0

Conv-2	5x5x96x256	256	1	2	27x27x256	614656
MaxPool-2	3x3	0	2	0	13x13x256	0
Norm-2	0	0	0	0	13x13x256	0
Conv-3	3x3x256x384	384	1	1	13x13x384	885120
Conv-4	3x3x384x384	384	1	1	13x13x384	1327488
Conv-5	3x3x384x256	256	1	1	13x13x256	884992
MaxPool-3	3x3	0	2	0	6x6x256	0
FC-1	9216x4096	4096	0	0	4096	37752832
FC-2	4096x4096	4096	0	0	4096	16781312
FC-3	4096x1000	1000	0	0	1000	4097000
Output	0	0	0	0	1000	
Total						62378344

Bảng 3.2.2 kiến trúc mạng AlexNet

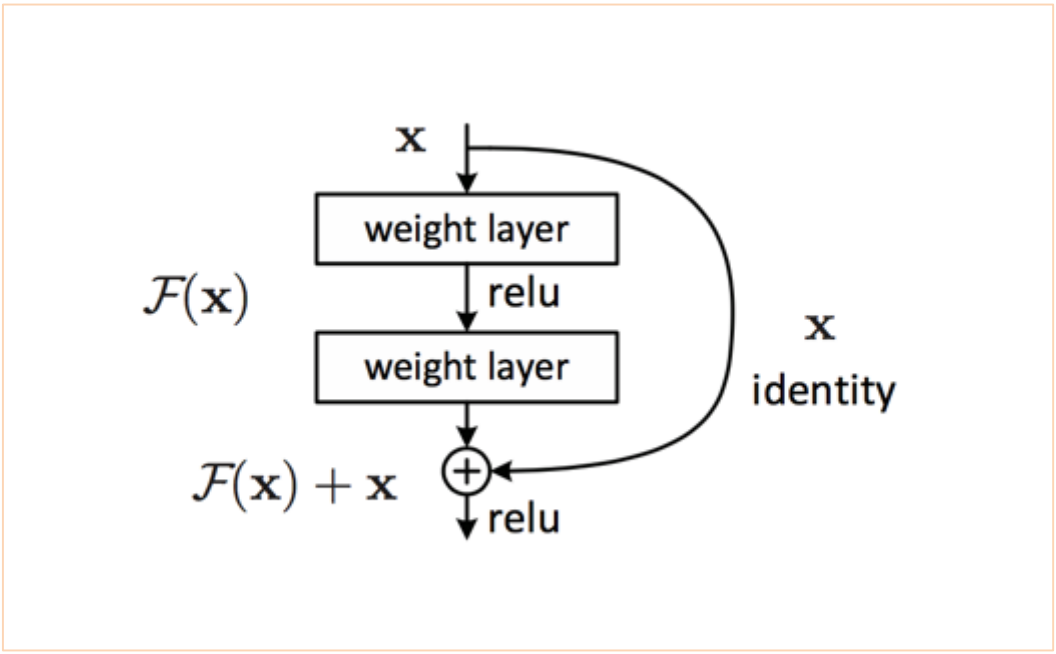
Mạng với đầu vào là ảnh có kích thước 227x227x3 và kết quả đầu cần thực hiện là phân loại 100 ảnh khác nhau, tổng số tham số cần phải học của mạng là 62378344 lớn hơn so với mạng LeNet-5. Các trọng số được cập nhập thông qua quá trình tranning mạng bởi thuật toán backpropagation.

Mạng VGG-16 được công bố trong bài báo của Karen Simonyan and Andrew Zisserman vào năm 2014.

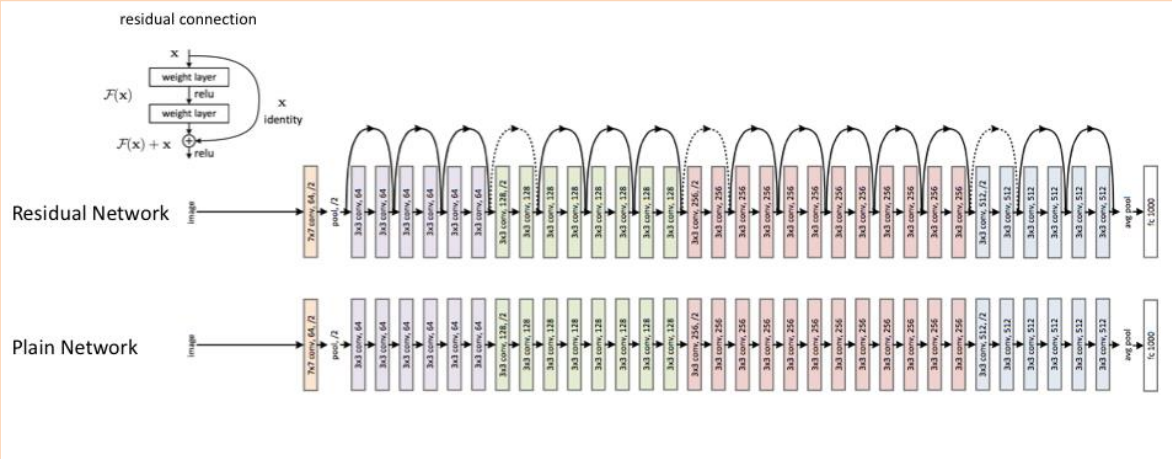


Hình 4.2.3 Sơ đồ kiến trúc mạng VGG16

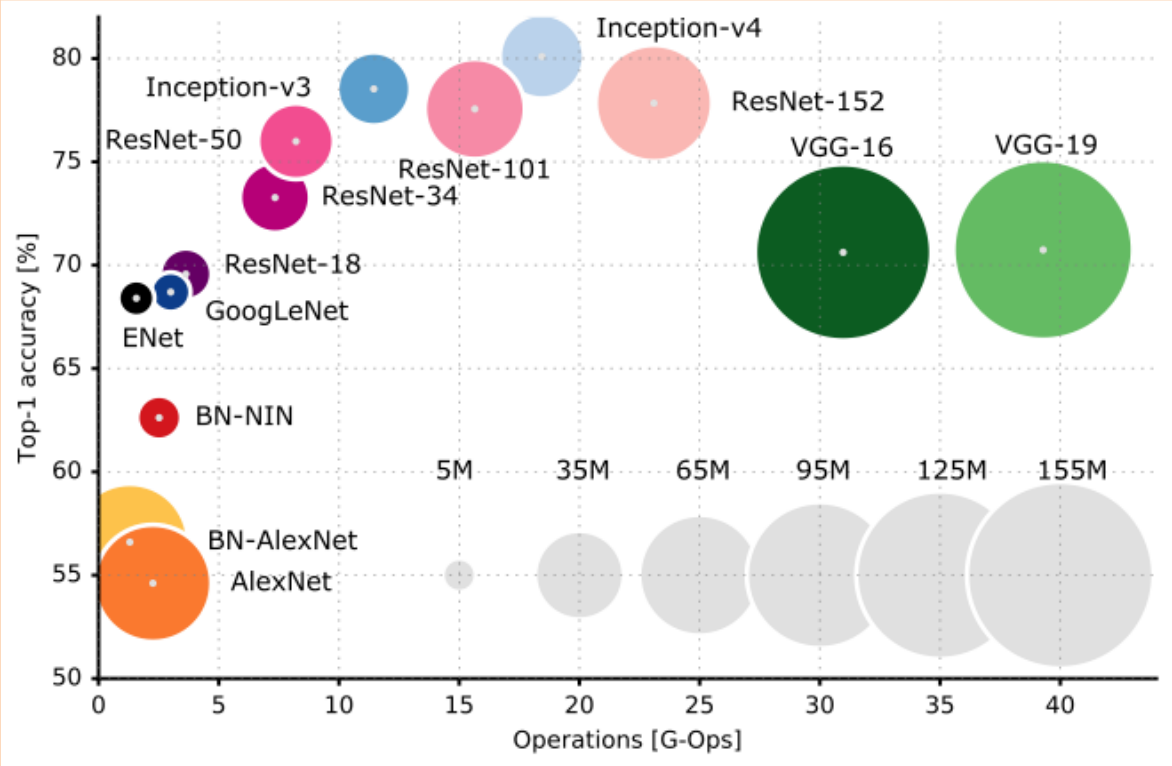
Mạng ResNet được công bố bởi Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun vào năm 2015. Đặc điểm của mạng là số lượng layer lớn, với phần tử cơ bản có tên gọi là Residual Block được mô tả trong hình dưới đây.



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 lương tính toán

Hình trên được lấy từ kaggle.com, chúng ta có thể thấy được sự so sánh về khối lượng tính toán và độ chính xác giữa các cách xây dựng mạng CNN phổ biến hiện nay, trong đó ResNet 50 có độ chính xác khá cao trong khi khối lượng tính toán không nhiều. AlexNet cần khối lượng tính toán thấp nhưng độ chính xác tương đối thấp so với các kiến trúc mạng khác.

3.3. Thử nghiện trên trên mạng LeNet-5 mở rộng

LeNet-5 là mạng cơ bản nhất trong việc nghiên cứu về convolutional neuron network. Mạng trên được thực hiện nhằm nhận diện các đối tượng cơ bản như số và chữ viết tay, các đối tượng đó đơn giản nên kích thước ảnh đầu vào nhỏ và dễ dàng nhận diện. Khi vào bài toán nhận diện địa danh có đặc điểm là kích thước ảnh nhận diện sẽ lớn và độ phức tạp trong chi tiết của ảnh lớn, em đã lấy ý tưởng thiết kế từ mạng này và mởi rộng số lớp convolutional cũng như kích thước của input và filter cho phù hợp với dữ liệu bài toán đang cần xử lý.

No	Layer	Input size	kernel size	stride	Bias	Activation	Learn parameter
1	convolutional	128 x 128 x 3	5 x 5 x 3 x 8	1	8	relu	608
	max pooling	128 x 128 x 8	2 x 2	2	-	-	0
2	convolutional	64 x 64 x 8	5 x 5 x 3 x 16	1	16	relu	1216
	max pooling	64 x 64 x 16	2 x 2	2	-	-	0
3	convolutional	32 x 32 x 16	5 x 5 x 3 x 32	1	32	relu	2432
	max pooling	32 x 32 x 32	2 x 2	2	-	-	0
	flatten	-	-	-	-	-	0
4	fully connect	8192	8192 x 128	-	128	-	1048704
5	fully connect	128	128x64	-	64	-	8256
							1061216

Bảng 3.3.1 thông tin thiết kế mạng lenet-5 mở rộng

Mạng được thiết kế với đầu vào là ảnh có kích thước 128x128 với 3 kênh màu RGB. Và đầu ra của mạng là một vector 64 phần tử, mỗi phần tử là giá trị sác xuất ảnh đầu vào rơi vào địa danh có mã tương ứng.

Sau đây là việc triển khai, training và thực hiện kiểm định độ chính xác của mạng bằng thư viện Tensorflow.

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 trương trình

5.2. Minh họa chức năng tìm kiếm địa danh bằng hình ảnh

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

[7] https://www.learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/

[8] https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/