

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LÊ CẨM HÀ

**NGHIÊN CỨU MẠNG NƠON CNN VÀ ỨNG DỤNG
TRONG BÀI TOÁN PHÂN LOẠI ẢNH**

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

THÁI NGUYÊN - 2020

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LÊ CẨM HÀ

**NGHIÊN CỨU MẠNG NƠON CNN VÀ ỨNG DỤNG
TRONG BÀI TOÁN PHÂN LOẠI ẢNH**

Chuyên ngành: Khoa học máy tính

Mã số: 8 48 01 01

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

Giáo viên hướng dẫn: TS.Nguyễn Đình Dũng

THÁI NGUYÊN - 2020

LỜI CẢM ƠN

Luận văn này được hoàn thành tại Trường Đại học Công nghệ Thông tin và Truyền thông dưới sự hướng dẫn của TS. Nguyễn Đình Dũng. Tác giả xin bày tỏ lòng biết ơn tới các thầy cô giáo thuộc Trường Đại học Công nghệ Thông tin và Truyền thông, các thầy cô giáo thuộc Viện Công nghệ Thông tin – Viện Hàn lâm Khoa học và Công nghệ Việt Nam đã tạo điều kiện, giúp đỡ tác giả trong quá trình học tập và làm luận văn tại Trường, đặc biệt tác giả xin bày tỏ lòng biết ơn tới TS. Nguyễn Đình Dũng đã tận tình hướng dẫn và cung cấp nhiều tài liệu cần thiết để tác giả có thể hoàn thành luận văn đúng thời hạn.

Xin chân thành cảm ơn anh chị em học viên cao học và bạn bè đồng nghiệp đã trao đổi, khích lệ tác giả trong quá trình học tập và làm luận văn tại Trường Đại học Công nghệ Thông tin và Truyền thông – Đại học Thái Nguyên.

Cuối cùng tác giả xin gửi lời cảm ơn đến gia đình, những người đã luôn bên cạnh, động viên và khuyến khích tôi trong quá trình thực hiện đề tài.

Thái Nguyên, tháng 10 năm 2020

Học viên cao học

Lê Cẩm Hà

LỜI CAM ĐOAN

Tôi xin cam đoan luận văn này do chính tôi thực hiện, dưới sự hướng dẫn khoa học của TS. Nguyễn Đình Dũng, các kết quả lý thuyết được trình bày trong luận văn là sự tổng hợp từ các kết quả đã được công bố và có trích dẫn đầy đủ, kết quả của chương trình thực nghiệm trong luận văn này được tác giả thực hiện là hoàn toàn trung thực, nếu sai tôi hoàn toàn chịu trách nhiệm.

Thái Nguyên, tháng 10 năm 2020

Học viên

Lê Cẩm Hà

MỤC LỤC

LỜI CẢM ƠN	i
LỜI CAM ĐOAN	ii
DANH MỤC CÁC HÌNH ẢNH	vii
DANH MỤC BẢNG BIỂU	ix
MỞ ĐẦU.....	1
1. Tính khoa học và cấp thiết của đề tài.....	1
2. Đối tượng và phạm vi nghiên cứu của đề tài	2
3. Phương pháp luận nghiên cứu.....	2
4. Nội dung và bố cục của luận văn	2
CHƯƠNG 1 TỔNG QUAN BÀI TOÁN PHÂN LOẠI ẢNH SỐ	3
1.1 Tổng quan xử lý ảnh số.....	3
1.1.1 Một số khái niệm cơ bản trong xử lý ảnh	3
1.1.2 Tổng quan về một hệ thống xử lý ảnh.....	4
1.1.3 Một số thao tác cơ bản trong xử lý ảnh.....	5
1.2 Biểu diễn ảnh trong máy tính.....	7
1.2.1 Ảnh màu	7
1.2.2 Ảnh xám	10
1.3 Phép tích chập trong xử lý ảnh.....	10
1.4 Lý thuyết phân loại ảnh số	13
1.4.1 Các khái niệm cơ bản	13
1.4.2 Phương pháp số phân loại ảnh	15
1.4.3 Phương pháp phân loại theo cấu trúc:.....	17
1.5 Một số thuật toán tiêu biểu trong phân loại ảnh.....	19
1.5.1 Thuật toán KNN	19
1.5.2 Thuật toán sử dụng mạng Nơ ron	20
1.5.3 Thuật toán SVM	21
1.6 Kết luận chương 1	21
CHƯƠNG 2 MẠNG NƠ RON CNN VÀ ỨNG DỤNG TRONG PHÂN LOẠI ẢNH	23

2.1	Các khái niệm chung về mạng nơron.....	23
2.1.1	Mạng nơron sinh học	23
2.1.2	Mạng nơron nhân tạo	24
2.1.3	Mô hình toán học và kiến trúc mạng nơron.....	27
2.1.4	Phân loại mạng nơron.....	30
2.1.5	Huấn luyện mạng nơron	31
2.2	Mạng nơron CNN.....	32
2.2.1	Giới thiệu	32
2.2.2	Kiến trúc mạng CNN.....	33
2.2.3	Ứng dụng CNN trong phân loại ảnh.....	37
2.3	Xây dựng mạng CNN cho phân loại ảnh	38
2.3.1	Trường tiếp nhận cục bộ (Local receptive fields)	38
2.3.2	Trọng số chia sẻ và độ lệch (Shared weights and biases).....	42
2.3.3	Lớp chứa hay lớp tổng hợp (Pooling layer)	42
2.3.4	Cách chọn tham số cho CNN	45
2.4	Cập nhật một số hướng nghiên cứu về bài toán phân loại ảnh sử dụng mạng nơron CNN.....	45
2.4.1	Các nghiên cứu trên thế giới.....	45
2.4.2	Các nghiên cứu trên trong nước	46
2.5	Kết luận chương	48
CHƯƠNG 3 XÂY DỰNG CHƯƠNG TRÌNH MÔ PHỎNG ỨNG DỤNG MẠNG CNN TRONG PHÂN LOẠI ẢNH		49
3.1	Đặt vấn đề.....	49
3.2	Bài toán nhận dạng chữ viết tay	50
3.2.1	Mô tả bài toán	50
3.2.2	Các bước thực hiện	51
3.2.3	Một số kết quả đạt được	57
3.3	Bài toán giải mã Capcha.....	61
3.3.1	Mô tả bài toán	61
3.3.2	Các bước thực hiện	65

3.3.3 Một số kết quả đạt được	67
3.4 Kết luận chương	68
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	70
TÀI LIỆU THAM KHẢO.....	72

DANH MỤC CÁC TỪ VIẾT TẮT

Từ hoặc cụm từ	Từ tiếng Anh	Từ tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
ANN	Artificial Neural Network	Mạng noron nhân tạo
CV	Computer Vision	Thị giác máy tính
CNN	Convolutional Neural Network	Mạng noron tích chập
DL	Deep Learning	Học sâu
CAPCHA	Completely Automated Public Turing test to tell Computers and Humans Apart	Phép thử Turing công cộng hoàn toàn tự động để phân biệt máy tính với người
MCR	Miss Classification Rate	Tỷ lệ nhận dạng sai
RMSE	Root Mean Square Error	Sai số bình phương trung bình
MLP	Multilayer Neural Network	Mạng noron nhiều lớp
MNIST	Modified National Institute of Standards and Technology database	Cơ sở dữ liệu về chữ số viết tay
ReLU	Rectified Linear Units	Hàm tinh chỉnh các đơn vị tuyến tính

DANH MỤC CÁC HÌNH ẢNH

Hình 1.1. Các giai đoạn chính trong xử lý ảnh	5
Hình 1.2. Minh họa hệ màu RGB	8
Hình 1.3. Ví dụ về ảnh màu	8
Hình 1.4. Biểu diễn ảnh theo tensor 3 chiều	9
Hình 1.5. Ví dụ về ảnh xám	10
Hình 1.6. Minh họa phép tích chập trong xử lý ảnh	11
Hình 1.7. Ma trận đầu ra Y khi chập ảnh X với kernel W	11
Hình 1.8. Stride=1, padding=1	12
Hình 1.9. Stride=2, padding=1	12
Hình 1.10. Một số bộ lọc Kernel trong xử lý ảnh	13
Hình 1.11. Phương pháp lưới	16
Hình 1.12. Phương pháp cung	16
Hình 1.13. Biểu diễn mẫu bằng tập kí hiệu	18
Hình 1.14. Minh họa thuật toán KNN	19
Hình 2.1. Cấu trúc cơ bản của nơron sinh học	23
Hình 2.2. Nơron nhân tạo	25
Hình 2.3. Mô hình toán học mạng nơron nhân tạo	27
Hình 2.4. Nơron 1 đầu vào với hàm hoạt hoá là hàm hardlimit	29
Hình 2.5. Phân loại mạng nơ ron	30
Hình 2.6. Học có giám sát	31
Hình 2.7. Học không có giám sát	31
Hình 2.8. Học tăng cường	32
Hình 2.9. Cách máy tính “nhìn” một hình [16]	32
Hình 2.10. Mạng nơ-ron thông thường (trái) và CNN (phải)	34
Hình 2.11. Kiến trúc mạng CNN	34
Hình 2.12. Max pooling kích thước 2×2	36
Hình 2.13. Lớp kết nối đầy đủ	36
Hình 2.14. Các bước phân loại ảnh sử dụng mạng CNN	37

Hình 2.16. Lớp input gồm 28x28 nơ ron cho nhận dạng chữ từ tập dữ liệu MNIST	38
Hình 2.17. Kết nối vùng 5x5 nơ ron input với nơ ron lớp ẩn	39
Hình 2.18. Vị trí bắt đầu của trường tiếp nhận cục bộ	39
Hình 2.19. Vị trí thứ 2 của trường tiếp nhận cục bộ và nơ ron lớp ẩn	40
Hình 2.20. Trường tiếp nhận cục bộ với ba bản đồ đặc trưng	40
Hình 2.21. Trường tiếp nhận cục bộ với 20 bản đồ đặc trưng	41
Hình 2.22. Ví dụ về Max pooling 2x2	43
Hình 2.23. Max pooling với ba bản đồ đặc trưng	43
Hình 2.24. Một kiến trúc mạng CNN cho nhận dạng chữ viết từ dữ liệu MNIST ..	44
Hình 3.1. Giao diện chính của chương trình mô phỏng	49
Hình 3.2. Chữ viết tay số “5” từ bộ dữ liệu MNIST	50
Hình 3.3. Giao diện thiết kế mạng CNN	55
Hình 3.4. Mạng CNN cơ bản	55
Hình 3.5. Tiến trình luyện mạng với kernel 7 x 7 and 8 bản đồ đặc trưng.	56
Hình 3.6. Giao diện chương trình nhận dạng chữ viết tay.	60
Hình 3.7. Một số mẫu captcha	62
Hình 3.8. Một số kết quả tấn công captcha	63
Hình 3.9. Hai cách tiếp cận để nhận dạng captcha bằng CNN	64
Hình 3.10. Kiểu dữ liệu captcha dùng trong bài toán nhận dạng	65
Hình 3.11. Ký tự W và Q bị dính với nhau	65
Hình 3.12. Giãn nở ký tự trong captcha để dễ phát hiện vùng liên thông	66
Hình 3.13. Phát hiện thành phần liên thông	66
Hình 3.14. Một mẫu captcha có 2 ký tự dính liền nhau	66
Hình 3.15. Vùng nhận dạng liên tục nhận 2 ký tự vào 1 ảnh cắt, chưa tốt	66
Hình 3.16. Kết quả sau khi dùng thủ thuật cắt đôi vùng nhận các ký tự liền nhau ..	66
Hình 3.17. Ví dụ tập các ảnh ký tự đã được cắt và xếp theo thư mục	67
Hình 3.18. Chương trình mô phỏng nhận dạng mã Captcha	68

DANH MỤC BẢNG BIỂU

Bảng 2.1. Một số dạng hàm hoạt hóa trong mạng nơron nhân tạo	29
Bảng 3.1. Các tham số hoạt động của mạng CNN cơ bản	57
Bảng 3.2. Các tham số hoạt động của mạng CNN ba lớp ẩn	58
Bảng 3.3. So sánh kết quả của một số phương pháp trên bộ dữ liệu MNIST	61

MỞ ĐẦU

1. Tính khoa học và cấp thiết của đề tài

Ứng dụng của công nghệ phân loại hiện nay đang phát triển rất mạnh ở rất nhiều lĩnh vực như: học thuật, kinh doanh, bảo mật, y tế... và các ở các đối tượng như: nhà nghiên cứu xã hội, chính phủ và các tổ chức phi lợi nhuận khác. Vì các tổ chức này sở hữu một lượng lớn dữ liệu không có cấu trúc và việc xử lý dữ liệu sẽ trở nên dễ dàng hơn rất nhiều nếu như các dữ liệu này được chuẩn hóa bởi các chủ đề/nhân. Nền tảng công nghệ để thực hiện bài toán phân loại chính là trí tuệ nhân tạo (Artificial Intelligence – AI) và học sâu (Deep Learning - DL).

Trong ngành Thị giác máy tính (Computer Vision - CV), nhờ những thành tựu của lĩnh vực học sâu mà trong những năm gần đây, ta đã chứng kiến được nhiều thành tựu vượt bậc. Các 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.

Từ lâu các nhà khoa học đã nhận thấy những ưu điểm của bộ óc con người và tìm cách bắt chước để thực hiện trên những máy tính, tạo cho nó có khả năng học tập, nhận dạng và phân loại. Vì vậy các nhà khoa học đã nghiên cứu và sáng tạo ra mạng nơron nhân tạo. Nó thực sự được chú ý và nhanh chóng trở thành một hướng nghiên cứu mới triển vọng đặc biệt là lĩnh vực nhận dạng, dự đoán và phân loại.

Convolutional Neural Network (Mạng nơ-ron tích chập - CNN) 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. Việc nghiên cứu về mạng nơron cũng như mạng CNN (tích chập) và sử dụng mô hình CNNs trong phân lớp ảnh (Image Classification) là một bài toán đầy hấp dẫn và có khả năng áp dụng để giải quyết nhiều vấn đề trong thực tế.

Được sự gợi ý của thầy giáo hướng dẫn tôi đã chọn đề tài: “Nghiên cứu mạng nơron CNN và ứng dụng trong bài toán phân loại ảnh” làm luận văn tốt nghiệp của mình. Mục tiêu chính của luận văn là tìm hiểu về bài toán phân loại hình ảnh trong CV và cách thực hiện bằng mạng CNN cho hai ứng dụng (bài toán nhận dạng chữ viết tay và bài toán giải mã Capcha).

2. Đối tượng và phạm vi nghiên cứu của đề tài

- Đối tượng nghiên cứu: Luận văn nghiên cứu kỹ thuật phân loại ảnh sử dụng mạng CNN
- Phạm vi nghiên cứu: Luận văn tập trung nghiên cứu trên hai bài toán (bài toán nhận dạng chữ viết tay và bài toán giải mã Capcha) dựa trên các bộ dữ liệu ảnh có sẵn được cộng đồng khoa học quốc tế công nhận.

3. Phương pháp luận nghiên cứu

- ***Phương pháp nghiên cứu lý thuyết:*** Tổng hợp, nghiên cứu các tài liệu về bài toán phân loại ảnh, mạng nơ ron CNN; Tìm hiểu các kiến thức liên quan. Ứng dụng mạng nơ ron CNN bài toán nhận dạng chữ viết tay và bài toán giải mã Capcha.

- ***Phương pháp nghiên cứu thực nghiệm:*** Sau khi nghiên cứu lý thuyết, luận văn sẽ tập trung vào xây dựng phần mềm mô phỏng việc phân loại dữ liệu ảnh trong hai bài toán nêu trên; Đánh giá kết quả sau khi thử nghiệm

- ***Phương pháp trao đổi khoa học:*** Thảo luận, xemina, lấy ý kiến chuyên gia.

4. Nội dung và bố cục của luận văn

Ngoài phần mở đầu, kết luận và hướng phát triển, luận văn được bố cục thành ba chương chính như sau:

Chương 1 Tổng quan bài toán phân loại ảnh số: Nghiên cứu các khái niệm cơ bản trong xử lý ảnh số, tập trung sâu vào phân loại ảnh số, một số thuật toán tiêu biểu được sử dụng trong phân loại ảnh số.

Chương 2 Mạng nơ ron CNN và ứng dụng trong phân loại ảnh: Nghiên cứu về mạng nơ ron nhân tạo, tập trung vào mạng nơ ron CNN và các ứng dụng của mạng này trong thực tế, đặc biệt trong phân lớp dữ liệu ảnh

Chương 3 Xây dựng chương trình mô phỏng ứng dụng mạng CNN trong phân loại ảnh: Chương này giới thiệu về hai bài toán nhận dạng chữ viết tay và giải mã Capcha. Xây dựng các mô hình mạng nơ ron CNN để giải quyết hai bài toán này dựa trên tập mẫu dữ liệu ảnh có sẵn được cộng đồng khoa học quốc tế công nhận. Đánh giá hiệu năng của mô hình mạng CNN thu được với một số phương pháp công bố trước đó.

CHƯƠNG 1

TỔNG QUAN BÀI TOÁN PHÂN LOẠI ẢNH SỐ

1.1 Tổng quan xử lý ảnh số

1.1.1 Một số khái niệm cơ bản trong xử lý ảnh

▪ *Ảnh số*

Ảnh số thực tế là biểu diễn số học của hình ảnh trong máy tính, thường là biểu diễn nhị phân. Có thể phân ảnh số thành 2 loại: ảnh xám và ảnh màu.

Ảnh xám thực chất là một hàm hai chiều của cường độ sáng $f(x,y)$, trong đó x và y là các tọa độ không gian và giá trị của hàm f tại một điểm (x,y) tỷ lệ với cường độ sáng của ảnh tại điểm đó. Nếu chúng ta có một ảnh màu thì f là một vector mà mỗi thành phần của vector đó chỉ ra cường độ sáng của ảnh tại điểm (x,y) đó tương ứng với dải màu [2].

Mỗi thành phần của mảng (x,y) được gọi là một điểm ảnh (pixel: picture element) và là phần tử nhỏ nhất cấu tạo nên ảnh. Điểm ảnh được hiểu như 1 dấu hiệu hay cường độ sáng tại một tọa độ xác định trong không gian. Hình ảnh được xem như là 1 tập hợp các điểm. Với cùng kích thước nếu sử dụng càng nhiều điểm ảnh thì bức ảnh càng đẹp, càng mịn và càng thể hiện rõ hơn chi tiết của ảnh người ta gọi đặc điểm này là độ phân giải.

▪ *Cường độ sáng của một ảnh tại một vị trí điểm ảnh*

Mỗi điểm ảnh của một ảnh tương ứng với một phần của một đối tượng vật lý tồn tại trong thế giới thực. Đối tượng vật lý này được chiếu sáng bởi một vài tia sáng mà tia sáng này bị phản xạ một phần hay hấp thụ một phần khi chiếu lên đối tượng vật lý đó. Phần ánh sáng phản xạ lại đi tới các bộ cảm biến được sử dụng để tạo ảnh cảm nhận và tạo ra các giá trị ghi nhận được đối tượng đối với từng điểm ảnh. Giá trị thu nhận được phụ thuộc vào phổ ánh sáng phản xạ. Giá trị cường độ sáng của các điểm ảnh khác nhau chỉ có ý nghĩa tương đối mà không có ý nghĩa trong các toán hạng tuyệt đối [2].

▪ **Số bits cần thiết để lưu trữ một ảnh**

Ở đây chúng ta chỉ quan tâm tới ảnh xám, nếu ảnh được lưu trữ dưới dạng một mảng hai chiều với kích thước $N \times N$ và có 2^m mức xám thì số bits cần thiết để lưu trữ ảnh là:

$$b = N \times N \times 2^m \quad (1.1)$$

Ví dụ như, một ảnh cỡ 512×512 với 256 (tức $m=8$) mức xám thì cần số bits lưu trữ là: $512 \times 512 \times 256 = 2.097.152$ bits.

▪ **Độ phân giải ảnh**

Độ phân giải ảnh biểu diễn mức độ chi tiết của ảnh mà chúng ta có thể nhìn rõ đối tượng. Khi thay đổi các giá trị m và N trong phương trình thì sẽ có các hiện tượng thay đổi khác nhau. Xong thực nghiệm cho thấy khi giữ nguyên kích thước ảnh N và tăng số mức xám m lên thì sẽ thể hiện rõ hơn mức độ chi tiết trong ảnh.

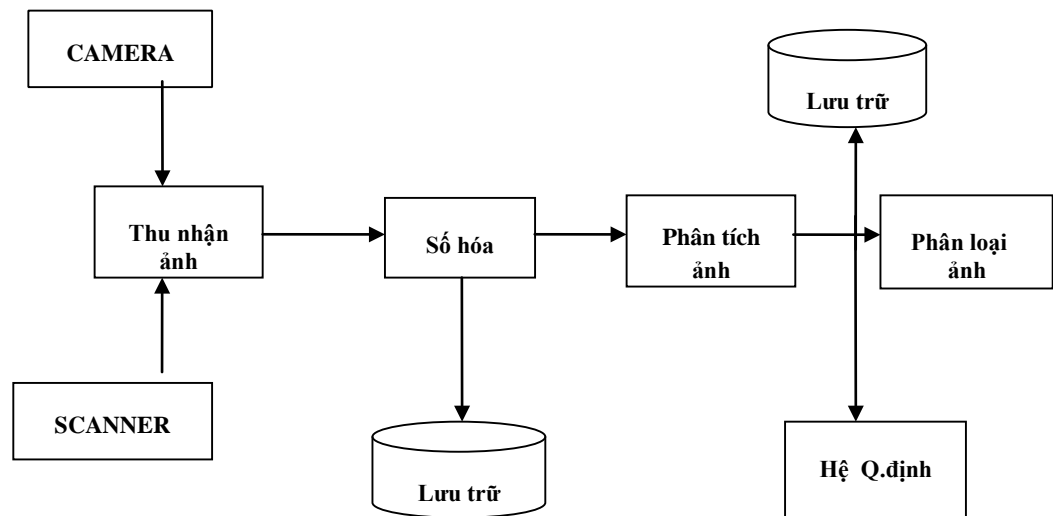
1.1.2 Tổng quan về một hệ thống xử lý ảnh

Xử lý ảnh là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh [2] .

Cũng như xử lý dữ liệu bằng đồ họa, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo ra bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật để biến đổi, để truyền tải hoặc mã hoá các ảnh tự nhiên. Mục đích của xử lý ảnh gồm:

- Biến đổi ảnh, làm tăng chất lượng ảnh.
- Tự động nhận dạng, đoán nhận, đánh giá các nội dung của ảnh.

Các bước cần thiết trong xử lý ảnh được mô tả chi tiết trong Hình 1.1 bao gồm các bước sau:



Hình 1.1. Các giai đoạn chính trong xử lý ảnh

Đầu tiên là quá trình thu nhận ảnh. Ảnh có thể thu nhận được qua camera. Thường khi thu nhận ảnh qua camera là tín hiệu tương tự (loại camera ống kính CCIR), nhưng cũng có thể là tín hiệu số hóa (loại CCD- Charge Coupled Device). Ảnh cũng có thể thu nhận từ vệ tinh qua các bộ cảm ứng (sensor), hay ảnh tranh được quét trên scanner. Tiếp theo là quá trình số hóa (Digitalizer) để biến đổi tín hiệu tương tự sang tín hiệu rời rạc (lấy mẫu) và số hóa bằng lượng hóa, trước khi chuyển sang giai đoạn xử lý, phân tích hay lưu trữ lại. Trước hết là công việc tăng cường ảnh để nâng cao chất lượng ảnh. Do những nguyên nhân khác nhau: có thể do chất lượng thiết bị thu nhận ảnh, do nguồn sáng hay do nhiễu, ảnh có thể bị suy biến do vậy cần phải tăng cường và khôi phục lại ảnh để làm nổi bật một số đặc tính chính của ảnh, hay làm cho ảnh gần giống nhất với trạng thái gốc – trạng thái trước khi bị biến dạng. Giai đoạn tiếp theo là phát hiện các đặc tính như biên, phân vùng ảnh, trích chọn các đặc tính...v.v...

Cuối cùng tùy theo mục đích của ứng dụng, sẽ là giai đoạn nhận dạng, phân loại hay các quyết định khác.

1.1.3 Một số thao tác cơ bản trong xử lý ảnh

▪ Biểu diễn ảnh

Trong biểu diễn ảnh, người ta thường dùng các phần tử đặc trưng của ảnh là pixel. Nhìn chung có thể một hàm hai biến chứa các thông tin như biểu diễn của một

ảnh. Các mô hình biểu diễn cho ta một mô tả logic hay định lượng các tính chất của hàm này. Trong biểu diễn ảnh cần chú ý đến tính trung thực hoặc các tiêu chuẩn “thông minh” để đo chất lượng ảnh hoặc tính hiệu quả của các kỹ thuật xử lý.

Một số mô hình thường được dùng trong biểu diễn ảnh: mô hình bài toán, mô hình thống kê. Trong mô hình bài toán, ảnh hai chiều được biểu diễn nhờ các hàm hai biến trực giao gọi là các hàm cơ sở. Còn mô hình thống kê, một ảnh được coi như một phần tử của một tập hợp đặc trưng bởi các đại lượng như: kỳ vọng toán học, hiệp biến, phương sai, moment.

▪ ***Biến đổi ảnh (Image Transform)***

Thuật ngữ biến đổi ảnh thường dùng để nói tới một lớp các ma trận đơn vị và các kỹ thuật dùng để biến đổi ảnh.

Biến đổi ảnh nhằm làm giảm các nguyên nhân của ảnh để việc xử lý hiệu quả hơn. Như làm rõ hơn các thông tin mà người dùng quan tâm nhưng người dùng phải chấp nhận mất đi một số thông tin cần thiết.

▪ ***Phân tích ảnh***

Phân tích ảnh liên quan đến việc xác định các độ đo định lượng của 1 ảnh để đưa ra một mô tả đầy đủ về ảnh.

Quá trình phân tích ảnh thực chất bao gồm nhiều công đoạn nhỏ. Trước hết là công việc tăng cường ảnh để nâng cao chất lượng ảnh, giai đoạn tiếp theo là phát hiện các đặc tính như phát hiện biên, phân vùng ảnh, trích chọn các đặc tính...v.v..

▪ ***Tăng cường ảnh – khôi phục ảnh***

Tăng cường ảnh là một bước quan trọng, tạo tiền đề cho xử lý ảnh. Nó gồm các kỹ thuật như: lọc độ tương phản, khử nhiễu, nổi màu...

Khôi phục ảnh là nhằm loại bỏ các suy giảm trong ảnh.

▪ ***Xử lý biên ảnh***

Biên là vấn đề chủ yếu trong phân tích ảnh vì các điểm trích chọn trong quá trình phân tích ảnh đều dựa vào biên. Mỗi điểm ảnh có thể là biên nếu ở đó có sự thay

đôi đột ngột về mức xám. Tập hợp các điểm biên tạo thành biên hay đường bao quanh của ảnh.

▪ ***Phân vùng ảnh***

Phân vùng là bước then chốt trong xử lý ảnh. Giai đoạn này nhằm phân tích ảnh thành những thành phần có tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là mức xám, cùng màu hay độ tương phản.

▪ ***Nhận dạng ảnh***

Nhận dạng ảnh là quá trình liên quan đến các mô tả đối tượng mà người ta muốn đặc tả nó. Quá trình nhận dạng thường đi sau quá trình trích chọn các đặc tính chủ yếu của đối tượng. Có hai kiểu mô tả đối tượng:

Mô tả tham số (nhận dạng theo tham số).

Mô tả theo cấu trúc (nhận dạng theo cấu trúc).

Trên thực tế người ta đã áp dụng kỹ thuật nhận dạng khá thành công với nhiều đối tượng khác nhau như: nhận dạng ảnh vân tay, nhận dạng chữ viết.

▪ ***Nén ảnh***

Dữ liệu ảnh cũng như các dữ liệu khác cần phải lưu trữ hay truyền đi trên mạng mà lượng thông tin để biểu diễn cho một ảnh là rất lớn. Do đó làm giảm lượng thông tin hay nén dữ liệu là một nhu cầu cần thiết.

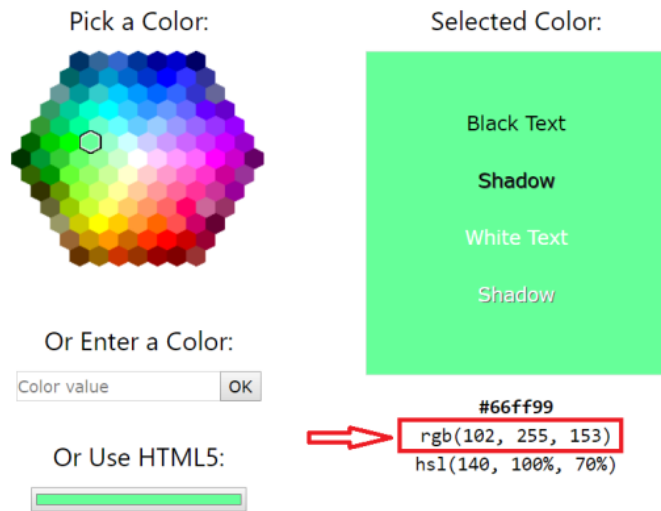
Nén dữ liệu là quá trình làm giảm lượng thông tin “dư thừa” trong dữ liệu gốc và do vậy lượng thông tin thu được sau khi nén thường nhỏ hơn dữ liệu gốc rất nhiều.

1.2 Biểu diễn ảnh trong máy tính

1.2.1 Ảnh màu

▪ ***Hệ màu RGB***

RGB viết tắt của red (đỏ), green (xanh lục), blue (xanh lam), là ba màu chính của ánh sáng khi tách ra từ lăng kính. Khi trộn ba màu trên theo tỉ lệ nhất định có thể tạo thành các màu khác nhau.



Hình 1.2. Minh họa hệ màu RGB

Hình 1.2 minh họa việc chọn màu thường thấy trong các chương trình máy tính. Khi ta chọn một màu thì sẽ ra một bộ ba số tương ứng (r,g,b) màu được chọn. Ở đây là rgb(102, 255, 153), nghĩa là r=102, g=255, b=153.

- ***Biểu diễn ảnh màu***



Hình 1.3. Ví dụ về ảnh màu

Ảnh màu (Hình 1.3) là một ma trận các pixel mà mỗi pixel biểu diễn một điểm màu. Mỗi điểm màu được biểu diễn bằng bộ 3 số (r,g,b). Để tiện cho việc xử lý ảnh thì sẽ tách ma trận pixel ra 3 channel red, green, blue.

Bức ảnh trên Hình 1.3 có kích thước 800 pixel * 600 pixel, bức ảnh này có thể biểu diễn dưới dạng một ma trận kích thước 600 * 800 như (1.2).

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,800} \\ w_{2,1} & w_{2,2} & \dots & w_{2,800} \\ \dots & \dots & \dots & \dots \\ w_{600,1} & w_{600,2} & \dots & w_{600,800} \end{bmatrix} \quad (1.2)$$

Trong đó mỗi phần tử w_{ij} là một pixel. Tuy nhiên để biểu diễn 1 màu ta cần 3 thông số (r,g,b) nên gọi $w_{ij} = (r_{ij}, g_{ij}, b_{ij})$ ta có thể để biểu diễn dưới dạng ma trận như sau:

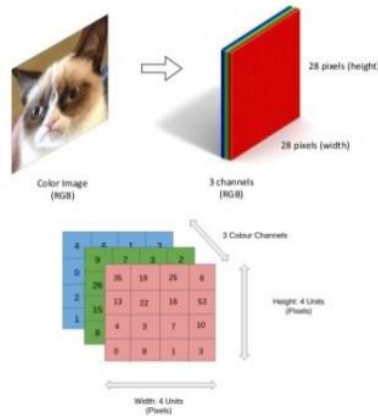
$$\begin{bmatrix} (r_{1,1}, g_{1,1}, b_{1,1}) & (r_{1,2}, g_{1,2}, b_{1,2}) & \dots & (r_{1,800}, g_{1,800}, b_{1,800}) \\ (r_{2,1}, g_{2,1}, b_{2,1}) & (r_{2,2}, g_{2,2}, b_{2,2}) & \dots & (r_{2,800}, g_{2,800}, b_{2,800}) \\ \dots & \dots & \dots & \dots \\ (r_{600,1}, g_{600,1}, b_{600,1}) & (r_{600,2}, g_{600,2}, b_{600,2}) & \dots & (r_{600,800}, g_{600,800}, b_{600,800}) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,800} \\ r_{2,1} & r_{2,2} & \dots & r_{2,800} \\ \dots & \dots & \dots & \dots \\ r_{600,1} & r_{600,2} & \dots & r_{600,800} \end{bmatrix}, \begin{bmatrix} g_{1,1} & g_{1,2} & \dots & g_{1,800} \\ g_{2,1} & g_{2,2} & \dots & g_{2,800} \\ \dots & \dots & \dots & \dots \\ g_{600,1} & g_{600,2} & \dots & g_{600,800} \end{bmatrix}, \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,800} \\ b_{2,1} & b_{2,2} & \dots & b_{2,800} \\ \dots & \dots & \dots & \dots \\ b_{600,1} & b_{600,2} & \dots & b_{600,800} \end{bmatrix},$$

Mỗi ma trận được tách ra (r, g, b) được gọi là 1 channel nên ảnh màu được gọi là 3 channel: channel red, channel green, channel blue.

Ảnh màu trên máy tính sẽ được biểu diễn dưới dạng tensor 3 chiều chồng lên nhau. Hình 1.4 mô tả biểu diễn một ảnh màu kích thước 28*28 trên máy tính. Trong đó, ảnh được biểu diễn dưới dạng tensor 3 chiều kích thước 28*28*3 do có 3 ma trận (channel) màu red, green, blue kích thước 28*28 chồng lên nhau.

color image is 3rd-order tensor



Hình 1.4. Biểu diễn ảnh theo tensor 3 chiều

1.2.2 Ảnh xám



Hình 1.5. Ví dụ về ảnh xám

Hình 1.5 mô tả ảnh xám của bức ảnh màu trong Hình 1.3. Tương tự ảnh màu, ảnh xám cũng có kích thước 800 pixel * 600 pixel, có thể biểu diễn dưới dạng một ma trận kích thước 600 * 800 như (1.2).

Tuy nhiên mỗi pixel trong ảnh xám chỉ cần biểu diễn bằng một giá trị nguyên trong khoảng từ [0,255] thay vì (r,g,b) như trong ảnh màu. Giá trị 0 là màu đen, 255 là màu trắng và giá trị pixel càng gần 0 thì càng tối và càng gần 255 thì càng sáng. Do đó khi biểu diễn ảnh xám trong máy tính chỉ cần một ma trận là đủ.

1.3 Phép tích chập trong xử lý ảnh

Phép tích chập (Convolution) là kỹ thuật quan trọng trong xử lý ảnh, được sử dụng chính yếu trong các phép toán trên ảnh như: đạo hàm ảnh, làm trơn ảnh, trích xuất biên cạnh trong ảnh... Kí hiệu phép tính convolution là \otimes : $Y = X \otimes W$

Theo toán học, tích chập là phép toán tuyến tính, cho ra kết quả là một hàm bằng việc tính toán dựa trên hai hàm đã có (X và W). Để cho dễ hình dung mình sẽ lấy ví dụ trên ảnh xám, tức là ảnh được biểu diễn dưới dạng ma trận X kích thước $m \times n$ [2] .

Công thức tích chập giữa hàm ảnh $X(x, y)$ và bộ lọc $W(x, y)$ (kích thước $m \times n$):

$$Y(x, y) = X(x, y) \otimes W(x, y) = \sum_{u=-m/2}^{m/2} \sum_{v=-n/2}^{n/2} X(u, v) W(x-u, y-v) \quad (1.3)$$

▪ Kernel

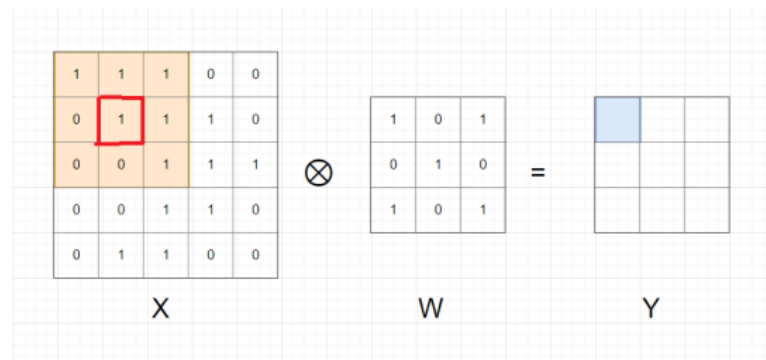
Ta định nghĩa **kernel** là một ma trận vuông kích thước $k \times k$ trong đó k là số lẻ. k có thể bằng 1, 3, 5, 7, 9, ... Ví dụ kernel kích thước 3×3 như (1.4)

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (1.4)$$

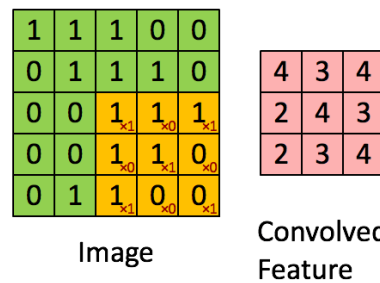
Với mỗi phần tử x_{ij} trong ma trận X lấy ra một ma trận có kích thước bằng kích thước của kernel W có phần tử x_{ij} làm trung tâm (đây là vì sao kích thước của kernel thường lẻ) gọi là ma trận A . Sau đó tính tổng các phần tử của phép tính element-wise của ma trận A và ma trận W , rồi viết vào ma trận kết quả Y .

Hình 1.6 minh họa việc thực hiện phép tích chập trong xử lý ảnh số khi tính tại x_{22} (ô khoanh đỏ trong hình), ma trận A cùng kích thước với W , có x_{22} làm trung tâm có màu nền da cam. Sau đó tính y_{11} :

$$y_{11} = x_{11} * w_{11} + x_{12} * w_{12} + x_{13} * w_{13} + x_{21} * w_{21} + x_{22} * w_{22} + x_{23} * w_{23} + x_{31} * w_{32} + x_{32} * w_{32} + x_{33} * w_{33} = 4 \quad (1.5)$$



Hình 1.6. Minh họa phép tích chập trong xử lý ảnh



Hình 1.7. Ma trận đầu ra Y khi chập ảnh X với kernel W

Làm tương tự với các phần tử còn lại trong ma trận ta thu được kết quả là ma trận Y đầu ra (Hình 1.7). Ma trận Y này có kích thước nhỏ hơn ma trận X. Kích thước của ma trận Y là $(m-k+1) * (n-k+1)$.

▪ Padding

Như đã phân tích ở trên, mỗi lần thực hiện phép tính convolution xong, kích thước ma trận Y đều nhỏ hơn X. Nếu muốn ma trận Y thu được có kích thước bằng ma trận X ta phải thêm giá trị 0 ở viền ngoài ma trận X. Phép tính này gọi là convolution với padding=1. Padding=k nghĩa là thêm k vector 0 vào mỗi phía của ma trận (Hình 1.8).

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

Hình 1.8. Stride=1, padding=1

▪ Stride




Như ở trên ta thực hiện tuần tự các phần tử trong ma trận X, thu được ma trận Y cùng kích thước ma trận X, ta gọi là stride=1. Tuy nhiên, nếu stride=k ($k > 1$) thì ta chỉ thực hiện phép tính convolution trên các phần tử $x_{1+i*k, 1+j*k}$. Hình 1.9 minh họa trường hợp stride=2.

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

Hình 1.9. Stride=2, padding=1

Hiệu đơn giản là bắt đầu từ vị trí x_{11} sau đó nhảy k bước theo chiều dọc và ngang cho đến hết ma trận X . Kích thước của ma trận Y lúc này là 3×3 đã giảm đi đáng kể so với ma trận X . Tổng quát cho phép tính convolution của ma trận X kích thước $m \times n$ với kernel kích thước $k \times k$, stride = s , padding = p ra ma trận Y với kích thước là $\left(\frac{m-k+2p}{s}+1\right) \times \left(\frac{n-k+2p}{s}+1\right)$. Stride thường dùng để giảm kích thước của ma trận sau phép tính convolution.

▪ **Ý nghĩa của phép tính convolution**

Operation	Kernel w	Image result $g(x,y)$
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)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Hình 1.10. Một số bộ lọc Kernel trong xử lý ảnh

Mục đích của phép tính convolution trên ảnh là làm mờ, làm nét ảnh; xác định các đường;... Mỗi kernel khác nhau thì sẽ phép tính convolution sẽ có ý nghĩa khác nhau. Hình 1.10 minh họa một số bộ lọc Kernel được sử dụng trong các thao tác xử lý ảnh như làm nét ảnh, xác định biên ảnh, làm mờ ảnh.

1.4 Lý thuyết phân loại ảnh số

1.4.1 Các khái niệm cơ bản

- Mẫu và mô tả mẫu

Người ta mô tả tất cả những kích thước vật lý có thể thu nhận được trong thế giới xung quanh ta bằng các mẫu (*pattern*). Phân loại mẫu chính là việc xử lý mô tả và diễn dịch các mẫu. Các mẫu thường được mô tả bằng một tập các thuộc tính đặc trưng của đối tượng. Giả sử các p_i là đại lượng biểu diễn đặc trưng của đối tượng đang xét thì ta có thể biểu diễn một mẫu là $P = \{p_1, p_2, \dots, p_n\}$. Để mô tả mẫu từ các đặc trưng của đối tượng người ta có thể sử dụng hai phương pháp sau:

- Phương pháp số
- Phương pháp cấu trúc
- Khoảng cách mẫu:

Khoảng cách mẫu là một khái niệm được xây dựng để đánh giá các đối tượng có ở “gần nhau” hay không. Khi khoảng cách nhỏ hơn một mức ngưỡng nào đó thì ta có thể coi như hai đối tượng là đồng dạng với nhau và chúng sẽ ở cùng một lớp. Trường hợp khoảng cách lớn hơn mức ngưỡng chúng sẽ thuộc về hai lớp phân biệt.

- Lớp mẫu và phân lớp mẫu:

Không gian mẫu là một tập các mẫu trộn lẫn nhau. Nhờ vào quá trình phân lớp (*classification*) của bài toán phân loại mà các mẫu được nhóm lại thành từng lớp mẫu (*class*) riêng biệt. Các lớp mẫu phân biệt chứa các mẫu đồng dạng với nhau. Mỗi lớp mẫu sẽ được gán một cái tên. Như vậy khi mẫu của một đối tượng được quá trình phân lớp gán vào một lớp mẫu nào đó thì cũng có nghĩa là đối tượng đó đã được phân loại.

- Không gian mẫu và không gian diễn dịch

Các đặc trưng cơ bản của các đối tượng tạo nên các thành phần biểu diễn mẫu. Tập hợp các mẫu của các đối tượng sẽ tạo nên không gian mẫu. Còn tập các tên gọi của các đối tượng tạo thành không gian diễn dịch. Nói một cách khác tập các mẫu chuẩn sẽ tạo thành không gian diễn dịch. Như vậy quá trình nhận dạng mẫu là quá trình ánh xạ f từ tập không gian mẫu $\Pi = \{P_1, P_2, \dots, P_N\}$ sang tập không gian diễn dịch $\Omega = \{n_1, n_2, \dots, n_S\}$ (S: số tên gọi cho các đối tượng cần nhận dạng).

- Nhận dạng được giám sát và không được giám sát:

Bài toán nhận dạng có thể chia thành hai dạng chính là nhận dạng có giám sát và nhận dạng không giám sát.

Khi ta đã biết trước được tập các tên gọi sẽ gán cho đối tượng cần nhận dạng, tức là không gian diễn dịch đã được xác định thì ta có nhận dạng được giám sát. Với cách nhận dạng này, ta sẽ dùng một tập thư viện các mẫu chuẩn để “huấn luyện” cho hệ thống nhận dạng trước khi đưa vào sử dụng. Quá trình huấn luyện sẽ phân lớp tập mẫu tạo thành các lớp mẫu chuẩn. Việc nhận dạng các mẫu thực tế chính là việc so sánh các mẫu đó với các mẫu chuẩn để đưa các mẫu này vào các lớp mẫu chuẩn đã tạo ra.

Ngược lại, khi tập không gian diễn dịch là chưa xác định cụ thể thì ta có nhận dạng không được giám sát. Loại nhận dạng này yêu cầu phải tự định ra được các lớp mẫu và xác định được các đặc trưng của từng lớp mẫu. Bản chất của quá trình phân lớp ở đây là phân chia các mẫu theo những qui tắc định trước. Hoạt động phân lớp như vậy còn được gọi là tự học hay tự tổ chức (*self-organization*).

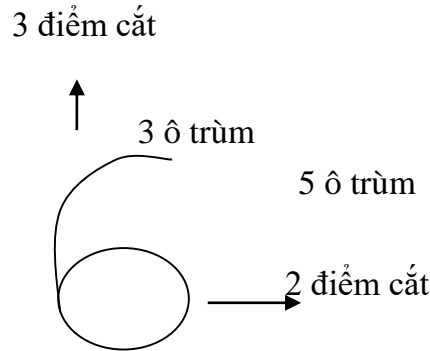
1.4.2 Phương pháp số phân loại ảnh

Phương pháp số phân loại ảnh biểu diễn các mẫu dưới dạng các giá trị số và quá trình phân lớp tập mẫu chính là quá trình thực hiện việc sắp xếp các giá trị số này thành từng lớp riêng biệt.

- Trích chọn đặc trưng mẫu:

Nhiệm vụ đặt ra cho bước trích chọn đặc trưng mẫu là phải rút ra được các đặc trưng riêng của từng đối tượng ảnh trong tập mẫu. Sau đó mỗi đặc trưng của đối tượng được mô tả bằng các giá trị số và các giá trị này sẽ tạo thành vectơ mô tả tập mẫu. Để tìm ra các đặc trưng riêng của đối tượng ta có thể xét đến các đặc trưng đơn giản như đặc trưng về hình học, topo...Ngoài ra có thể dùng một số phương pháp đặc biệt để có thể phát hiện được các đặc trưng phức tạp hơn của đối tượng đặc biệt là với các trường hợp mẫu có hình dạng phức tạp. Có thể kể ra đây một số phương pháp như sau:

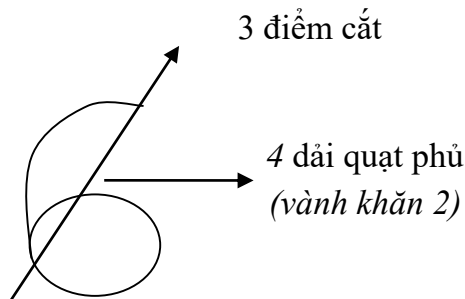
- Phương pháp lưới:



Hình 1.11. Phương pháp lưới

Một lưới vuông chuẩn được chụm lên đối tượng. Số lượng điểm cắt của mỗi nan lưới ngang và dọc với đối tượng sẽ là đặc trưng của đối tượng. Đồng thời số lượng mắt lưới vuông có trù lên đối tượng theo từng chiều dọc và ngang cũng được sử dụng làm đặc trưng của đối tượng. Mỗi đường dọc và ngang của lưới sẽ được gán cho một trọng số nhất định. Với phương pháp này, việc tiêu chuẩn hoá đối tượng rất quan trọng vì nó sẽ giúp cho việc xác định kích thước của lưới chuẩn được sử dụng

▪ Phương pháp cung:



Hình 1.12. Phương pháp cung

Có thể loại bỏ ảnh hưởng của hướng đối tượng trong phương pháp lưới bằng cách thay thế lưới vuông chuẩn bằng lưới hình vành khăn, tức là các nan lưới là các đường tròn đồng tâm (có thể coi đây là phương pháp lưới dùng trong hệ toạ độ cực). Điểm tâm của các vòng tròn này chính là trọng tâm của đối tượng và ta cần xác định điểm này trước tiên. Từ điểm này ta sẽ kẻ các đường bán kính chuẩn chia đều các đường tròn thành các cung. Số lượng các điểm cắt với đối tượng dọc theo một bán kính sẽ là đặc trưng của đối tượng. Tương tự như phương pháp lưới, số lượng cung

của mỗi vòng tròn phủ lên đối tượng xét cũng coi như là đặc trưng của đối tượng. Mỗi bán kính và vành khăn sẽ được gán một trọng số.

- Kỹ thuật phân lớp mẫu:

Các đặc trưng của đối tượng được biểu diễn bởi các giá trị số và các giá trị này được xem là các thành phần của các vector biểu diễn mẫu. Khi ta đưa vào hệ thống một tập các mẫu chuẩn thì quá trình trích chọn đặc trưng sẽ tạo nên các vector mẫu chuẩn phân bố trong không gian mẫu và với mỗi vector mẫu chuẩn này thì ta biết được ánh xạ từ nó sang không gian diễn dịch, tức là biết tên của nó. Như vậy các vector mẫu chuẩn đã được phân thành các lớp mà mỗi lớp ứng với một tên. Những lớp này ta gọi là lớp chuẩn.

Khi đưa các mẫu chưa xác định (mẫu cần phân loại) vào hệ thống thì việc nhận dạng mẫu chính là tìm ra một quy tắc để sắp xếp vector biểu diễn mẫu đó vào một lớp chuẩn nào đó. Để có thể đạt được mục đích này thì trước hết cần phải tạo được một sự phân định rõ ràng giữa các lớp chuẩn, tức là trong không gian mẫu phải có một sự phân hoạch rõ ràng. Trong thực tế thì không gian mẫu không phải lúc nào cũng đạt được đến sự phân tách hoàn toàn sau quá trình phân lớp mẫu. Nguyên nhân là do chúng ta chưa chọn được bộ đặc trưng tối ưu để phân tách đối tượng. Bởi vậy mà có thể xảy ra trường hợp một vector mẫu nào đó sẽ rơi vào vùng chồng lên nhau của 2 hay nhiều lớp mẫu chuẩn. Trong trường hợp này ta phải chọn lớp có xác suất cao hơn hoặc phải đánh dấu để chỉ ra rằng mẫu đó không phân lớp được. Quá trình xây dựng các lớp mẫu chuẩn như thế gọi là quá trình học. Việc xây dựng một thư viện mẫu chuẩn có vai trò rất quan trọng cho khả năng nhận dạng của hệ thống.

1.4.3 Phương pháp phân loại theo cấu trúc:

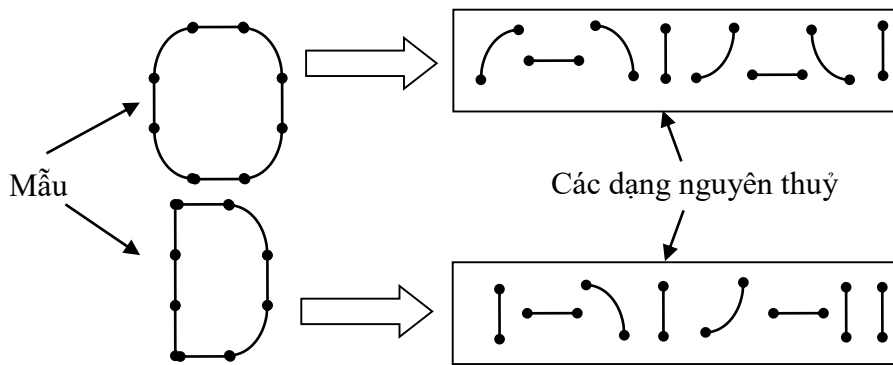
Bên cạnh phương pháp số, phương pháp cấu trúc là một trong những phương pháp truyền thống để nhận dạng mẫu. Trong khi ở phương pháp số, người ta thực hiện việc gán ý nghĩa cho các mẫu riêng biệt thì trong phương pháp cấu trúc lại xem xét các đối tượng như là các cấu trúc phức tạp tổ hợp từ các dạng nguyên thủy đơn giản và mối quan hệ giữa các dạng nguyên thủy này. Việc mô hình hoá các quá trình như vậy là tương đối khó khăn, bởi vậy mà phương pháp nhận dạng theo cấu trúc vẫn chưa được dùng phổ biến như phương pháp số.

- Xây dựng bộ ký hiệu

Trước hết, ta cần xây dựng một tập các dạng nguyên thủy và các mẫu sẽ được biểu diễn bởi các dạng nguyên thủy này và mối quan hệ giữa các dạng nguyên thủy đó. Các dạng nguyên thủy phải được chọn sao cho khi dạng nguyên thủy này được sắp xếp theo một trật tự nào đó đối với nhau thì ta sẽ tạo ra được tất cả các dạng cấu trúc từ đơn giản đến phức tạp của tập các đối tượng cần nhận dạng. Các dạng nguyên thủy có thể được chọn ví dụ như đoạn thẳng, cung, điểm ngoặt, điểm kết thúc..

Để biểu diễn dạng nguyên thủy và quan hệ giữa chúng một cách thuận tiện ta dùng một bộ ký hiệu. Mỗi ký hiệu sẽ được đặc trưng cho một dạng nguyên thủy và một mẫu như vậy sẽ được biểu diễn bằng một chuỗi ký hiệu. Với việc biểu diễn dạng nguyên thủy bởi một bộ ký hiệu thì việc xử lý các mẫu sẽ đơn giản hơn rất nhiều.

- Trích chọn đặc trưng cấu trúc:



Hình 1.13. Biểu diễn mẫu bằng tập kí hiệu

Quá trình trích chọn đặc trưng mẫu có thể hiểu là quá trình chuyển đổi tập mẫu sang các chuỗi ký hiệu. Tất cả các ký hiệu của một chuỗi phải được định nghĩa từ trước. Kết quả của quá trình trích chọn đặc trưng mẫu sẽ phân tập mẫu ra thành từng các phân lớp dựa theo cấu trúc của các chuỗi ký hiệu. Mỗi phân lớp sẽ được đặc trưng bằng một nguyên mẫu đại diện.

- Kỹ thuật phân lớp mẫu:

Quá trình phân lớp mẫu là quá trình đánh giá sự tương tự của các mẫu với các nguyên mẫu đại diện cho từng phân lớp tức là ta phải so sánh chuỗi kí hiệu của mẫu với chuỗi kí hiệu của các nguyên mẫu đại diện. Căn cứ vào kết quả thu được ta sẽ phân chia được các mẫu mới vào từng phân lớp chuẩn.

1.5 Một số thuật toán tiêu biểu trong phân loại ảnh

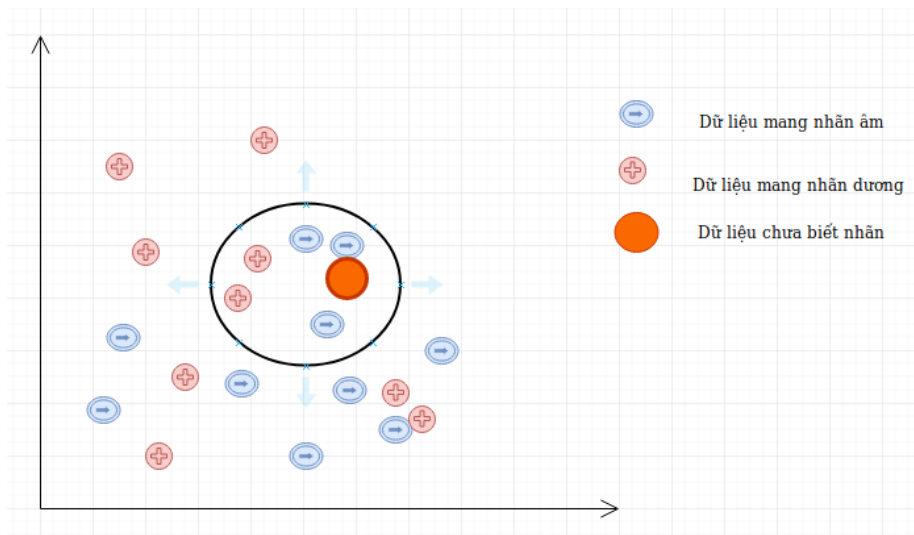
Để thực hiện phân loại ảnh có rất nhiều phương pháp như KNN, mạng Noron, SVM...

1.5.1 Thuật toán KNN

▪ Nguyên lý chung

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới. Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

Giả sử ta có D là tập các dữ liệu đã được phân loại thành 2 nhãn (+) và (-) được biểu diễn trên trục tọa độ như hình vẽ và một điểm dữ liệu mới A chưa biết nhãn. Vậy làm cách nào để chúng ta có thể xác định được nhãn của A là (+) hay (-)?



Hình 1.14. Minh họa thuật toán KNN

Có thể thấy cách đơn giản nhất là so sánh tất cả các đặc điểm của dữ liệu A với tất cả tập dữ liệu học đã được gán nhãn và xem nó giống cái nào nhất, nếu dữ liệu (đặc điểm) của A giống với dữ liệu của điểm mang nhãn (+) thì điểm A mang nhãn

(+), nếu dữ liệu A giống với dữ liệu nhãn (-) hơn thì nó mang nhãn (-), trông có vẻ rất đơn giản nhưng đó là những gì mà KNN làm.

Trong trường hợp của KNN, thực tế nó không so sánh dữ liệu mới (không được phân lớp) với tất cả các dữ liệu khác, thực tế nó thực hiện một phép tính toán học để đo khoảng cách giữa dữ liệu mới với tất cả các điểm trong tập dữ liệu học D để thực hiện phân lớp. Phép tính khoảng cách giữa 2 điểm có thể là Euclidian, Manhattan, trọng số, Minkowski, ...

- *Ưu điểm*
 - Thuật toán đơn giản, dễ dàng triển khai.
 - Độ phức tạp tính toán nhỏ.
 - Xử lý tốt với tập dữ liệu nhiễu
- *Nhược điểm*
 - Với K nhỏ dễ gặp nhiễu dẫn tới kết quả đưa ra không chính xác
 - Cần nhiều thời gian để thực hiện do phải tính toán khoảng cách với tất cả các đối tượng trong tập dữ liệu.
 - Cần chuyển đổi kiểu dữ liệu thành các yếu tố định tính.

1.5.2 Thuật toán sử dụng mạng Nơ ron

▪ *Nguyên lý chung*

Mạng nơron nhân tạo (Artificial Neural Networks) mô phỏng lại mạng nơron sinh học là một cấu trúc khối gồm các đơn vị tính toán đơn giản được liên kết chặt chẽ với nhau trong đó các liên kết giữa các nơron quyết định chức năng của mạng. Về cơ bản mạng Neural là một mạng các phân tử (gọi là neural) kết nối với nhau thông qua các liên kết (các liên kết này được gọi là trọng số liên kết) để thực hiện một công việc cụ thể nào đó. Khả năng xử lý của mạng neural được hình thành thông qua quá trình hiệu chỉnh trọng số liên kết giữa các neural, nói cách khác là học từ tập hợp các mẫu huấn luyện.

- *Ưu điểm*
 - Dễ cài đặt cùng với khả năng học và tổng quát hoá rất cao.
 - Tốc độ xử lý nhanh

- Linh hoạt và dễ bảo trì
- *Nhược điểm*
 - Tính chậm và xác suất không cao không có quy tắc tổng quát để xác định cấu trúc mạng và các tham số học tối ưu cho một (lớp) bài toán nhất định.
 - Tiêu chuẩn thu thập cơ sở dữ liệu huấn luyện còn khắt khe.
 - Đòi hỏi thời gian xử lý cao với mạng một mạng Neural lớn.

1.5.3 Thuật toán SVM

▪ *Nguyên lý chung*

Cho trước một tập huấn luyện, các ảnh được biểu diễn dưới dạng vector. Trong không gian vector, mỗi vector được biểu diễn bởi một điểm. Phương pháp SVM sẽ tìm một siêu phẳng quyết định để phân chia không gian vector thành hai lớp. Chất lượng của siêu phẳng này phụ thuộc vào khoảng cách giữa các vector, tức là phụ thuộc vào các đặc trưng của ảnh.

▪ *Ưu điểm:*

- Cho kết quả nhận dạng với độ chính xác cao
- Bài toán huấn luyện SVM thực chất là bài toán QP trên một tập lồi, do đó SVM luôn có nghiệm toàn cục và duy nhất, đây chính là điểm khác biệt rõ nhất giữa SVM so với phương pháp mạng Neural, vì mạng Neural vốn tồn tại nhiều điểm cực trị địa phương.

▪ *Nhược điểm:*

- Hạn chế lớn nhất của SVM là tốc độ phân lớp rất chậm, tùy thuộc vào số lượng các véc tơ hỗ trợ.
- Giai đoạn huấn luyện SVM đòi hỏi bộ nhớ rất lớn, do đó các bài toán huấn luyện với số lượng mẫu lớn sẽ gặp trở ngại trong vấn đề lưu trữ. Hiệu quả phân lớp của SVM phụ thuộc vào hai yếu tố: giải bài toán QP và lựa chọn hàm nhân.

1.6 Kết luận chương 1

Nội dung của luận văn là nghiên cứu áp dụng mạng CNN cho phân loại ảnh. Chính vì vậy, chương 1 đưa ra các kiến thức tổng quan về bài toán phân loại ảnh với mục đích tổng kết các lý thuyết liên quan đến quá trình xử lý ảnh số thông thường

Đặc biệt tập trung vào phép tính tích chập trong xử lý ảnh do nó có mối quan hệ biện chứng với nguyên lý của mạng CNN. Mối quan hệ này sẽ được phân tích rõ trong chương 2. Bên cạnh đó, luận văn cũng đã tìm hiểu về các phương pháp phân loại ảnh thường được sử dụng. Nội dung này kết hợp với kiến thức về mạng CNN trong chương 2 sẽ là nền tảng cho các việc xây dựng các ứng dụng nhận dạng ở chương 3.

CHƯƠNG 2

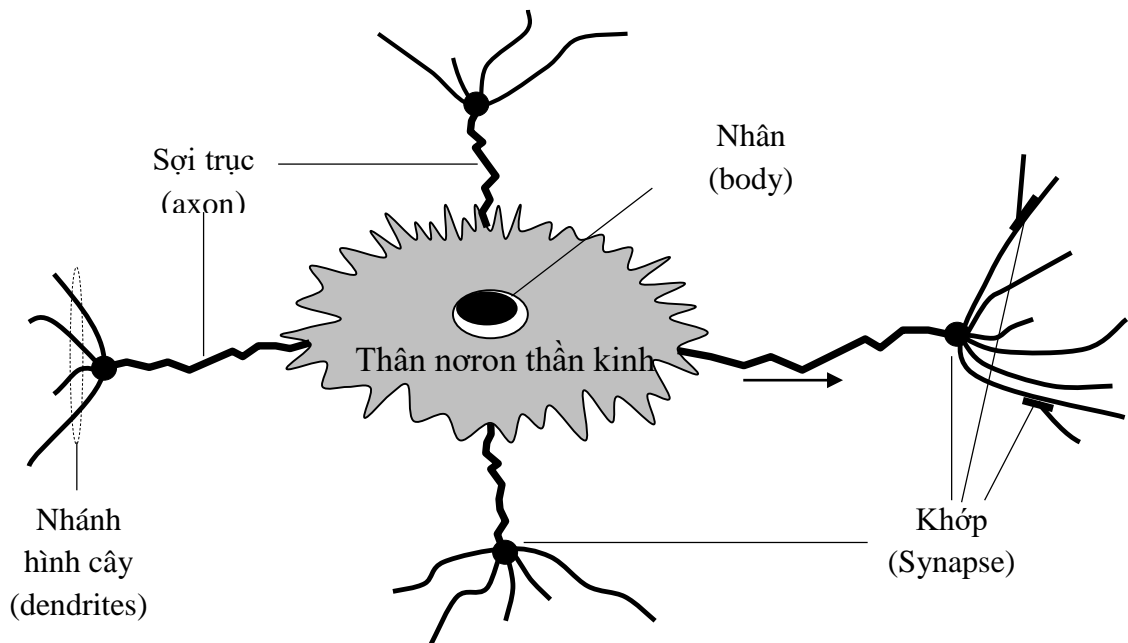
MẠNG NƠI RON CNN

VÀ ỨNG DỤNG TRONG PHÂN LOẠI ẢNH

2.1 Các khái niệm chung về mạng nơron

2.1.1 Mạng nơron sinh học

Não bộ con người là một mạng lưới khoảng 10^{11} tế bào thần kinh hay còn gọi là nơron. Chúng có cấu trúc và chức năng tương đối đồng nhất. Các nhà nghiên cứu sinh học về bộ não con người đã đưa ra kết luận rằng các nơron là đơn vị đảm nhiệm những chức năng nhất định trong hệ thần kinh bao gồm não, tủy sống và các dây thần kinh [1] , [9] . Hình 2.1 chỉ ra cấu tạo của hệ thống tế bào sinh học này.



Hình 2.1. Cấu trúc cơ bản của nơron sinh học

Cấu trúc của một nơron được chia thành 3 phần chính: Phần thân, hệ thống dây thần kinh tiếp nhận và sợi trục thần kinh ra. Hệ thống dây thần kinh tiếp nhận tạo thành một mạng lưới dày đặc xung quanh thân tế bào (chiếm diện tích khoảng 0.25 mm^2). Chúng là đầu vào để đưa các tín hiệu điện đến thân tế bào. Thân tế bào có nhân bên trong sẽ tổng hợp các tín hiệu vào và sẽ làm thay đổi điện thế của bản thân nó. Khi điện thế này vượt quá một mức ngưỡng thì nhân tế bào sẽ kích thích đưa một

xung điện ra sợi trục thần kinh ra. Sợi trục thần kinh ra có thể dài một vài centimet đến vài met. Nó có thể phân thành nhiều nhánh theo dạng hình cây để nối với các dây thần kinh vào của nhiều tế bào khác hoặc có thể nối trực tiếp đến thân tế bào của duy nhất một noron. Việc kết nối này được thực hiện nhờ các khớp nối. Số khớp nối của mỗi noron có thể lên tới hàng trăm ngàn. Người ta tính toán rằng mạng lưới dây thần kinh ra và các khớp nối chiếm khoảng 90% diện tích bề mặt noron. Các tín hiệu điện truyền trên các sợi dây thần kinh cũng như hiệu điện thế của nhân tế bào là kết quả của quá trình phản ứng và giải phóng của các chất hữu cơ được đưa ra từ các khớp nối dẫn đến dây thần kinh vào. Xung điện đưa ra sợi trục axon sẽ truyền tới các khớp nối với đầu vào của các noron khác và sẽ kích thích giải phóng các chất truyền điện. Tuy theo việc tăng hay giảm hiệu điện thế mà người ta chia thành hai loại khớp nối là khớp nối kích thích và khớp nối ức chế. Cường độ tín hiệu mà một tế bào thần kinh nhận được phụ thuộc chủ yếu vào mức độ liên kết của khớp nối. Các nghiên cứu chỉ ra rằng quá trình học của mạng noron sinh học chính là việc thay đổi mức độ liên kết của các khớp nối. Chính cấu trúc mạng noron và mức độ liên kết của các khớp nối đã tạo nên hức năng của hệ thần kinh con người. Quá trình phát triển của hệ thần kinh là một quá trình “*học*” liên tục. Ngay từ khi chúng ta sinh ra, một số cấu trúc thần kinh đơn giản đã được hình thành. Sau đó các cấu trúc khác lần lượt được xây dựng thêm nhờ quá trình học. Do đó cấu trúc mạng noron liên tục biến đổi để ngày càng phát triển hoàn thiện.

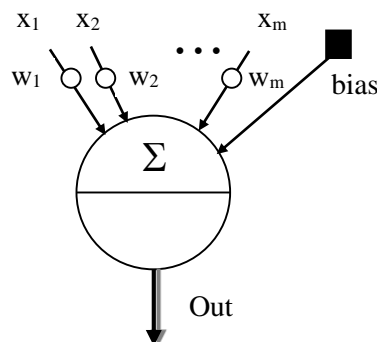
Một vấn đề đặt ra là dựa trên những kết quả nghiên cứu về hệ thần kinh con người chúng ta có thể mô phỏng, xây dựng lên các hệ thần kinh nhân tạo nhằm phục vụ cho một chức năng nào đó không. Nghiên cứu trả lời câu hỏi này đã đưa ra một hướng phát triển mới: Mạng noron nhân tạo.

2.1.2 Mạng noron nhân tạo

2.1.2.1 Noron nhân tạo

Noron nhân tạo là sự rút gọn hết sức đơn giản của noron sinh học. Nó có thể thực hiện nhờ chương trình máy tính hoặc bằng mạch phần cứng. Mỗi noron thực hiện hai chức năng là chức năng đầu vào và chức năng kích hoạt đầu ra. Do đó ta có thể

coi mỗi nơon như là một đơn vị xử lý. Nó được xây dựng mô phỏng theo cấu trúc của các nơon sinh học. Mỗi nơon có một số đầu vào giống như các dây thần kinh tiếp nhận. Các đầu vào này làm nhiệm vụ tiếp nhận thông tin từ các nơon khác hoặc từ tập số liệu gốc vào. Tương tự như nơon sinh học, mỗi đầu vào của nơon nhân tạo có ảnh hưởng khác nhau đối với tín hiệu ra của nơon (còn gọi là kết xuất của nơon). Điều này được thực hiện nhờ các hệ số được gán cho từng đầu vào- w_i : trọng số của đầu vào thứ i . Giá trị của w_i có thể dương hay âm tương tự như việc có hai loại khớp nối trong mạng nơon sinh học. Nếu w_i có giá trị dương thì tương đương với khớp nối kích thích còn nếu w_i âm thì tương đương với khớp nối ức chế. Thân nơon sẽ làm nhiệm vụ tổng hợp các tín hiệu đầu vào xử lý để đưa một tín hiệu ra đầu ra của nơon. Quá trình xử lý, tính toán này sẽ được đề cập cụ thể ở phần sau. Đầu ra của nơon nhân tạo tương tự như sợi trục axon của nơon sinh học. Tín hiệu ra cũng có thể tách ra thành nhiều nhánh theo cấu trúc hình cây để đưa đến đầu vào của các nơon khác.



Hình 2.2. Nơon nhân tạo

2.1.2.2 Mạng nơon nhân tạo

Các nơon nhân tạo được tổ chức thành mạng nơon nhân tạo. Các nơon thường được sắp xếp trong mạng thành từng lớp. Đầu ra của mỗi nơon sẽ được nối đến đầu vào của một số nơon khác theo một cấu trúc phù hợp. Tuy nhiên cấu trúc mạng nơon nhân tạo chưa thể đạt được độ phức tạp như mạng nơon sinh học. Mạng nơon nhân tạo hiện chỉ mới là sự mô phỏng hết sức đơn giản cấu trúc của mạng nơon sinh học.

Giữa mạng nơon nhân tạo và mạng nơon sinh học có 3 điểm chung là

- Mạng được xây dựng bằng các phần tử tính toán đơn giản liên kết lại với nhau một cách phức tạp và hoạt động theo nguyên tắc song song.

- Chức năng của mạng được xác định qua cấu trúc mạng, quá trình xử lý bên trong các phần tử và mức độ liên kết giữa các phần tử.

- Mức độ liên kết giữa các phần tử được xác định thông qua quá trình học của mạng (hay còn gọi là quá trình huấn luyện mạng).

Điểm khác nhau về căn bản giữa Mạng nơron nhân tạo và mạng nơron sinh học là ở tốc độ tính toán, độ phức tạp và tính song song. Tuy xét về tốc độ xử lý của các máy tính hiện đại là cao hơn rất nhiều so với tốc độ xử lý của não bộ con người nhưng bộ não lại có thể đồng thời kích hoạt toàn bộ các nơron để làm nhiều công việc khác nhau. Điều này mạng nơron nhân tạo không thể thực hiện được. Với sự phát triển nhanh chóng của khoa học như hiện nay thì ta có thể hi vọng sẽ có những bước đột phá mới trong lĩnh vực mô phỏng mạng nơron sinh học.

2.1.2.3 Các ứng dụng của mạng nơron

Mạng nơron thích hợp với các ứng dụng so sánh và phân loại mẫu, dự báo và điều khiển. Dưới đây là một số ứng dụng cụ thể của công nghệ mạng nơron [8] .

- Không gian vũ trụ: Trình điều khiển máy bay không người lái, chế độ tự bay nâng cao; mô phỏng các đường bay và các bộ phận của máy bay; hệ thống điều khiển của máy bay và hệ thống phát hiện sai hỏng.

- Dự đoán tài chính kinh tế: Dự đoán giá cả biến động cổ phiếu. Dự đoán cấp số thời gian trong thị trường tài chính. Các ứng dụng về điều hành vốn. Dự đoán thị trường ngoại hối. Đánh giá dự đoán rủi ro. Dự đoán tình hình kinh tế. Đánh giá hiệu suất vốn vay và vốn đầu tư.

- Hoạt động ngân hàng: Dự đoán khả năng phá sản. Hệ thống thẻ đọc ngân hàng, thẻ tín dụng.

- Hệ thống phòng thủ: Hệ thống điều khiển vũ khí dò tìm mục tiêu, nhận dạng mục tiêu. Điều khiển đường đạn. Xử lý và nhận dạng tín hiệu ảnh, radar, siêu âm

- Điện tử viễn thông: Dự đoán chuỗi mã. Bố trí mạch tích hợp trên chip. Phân tích lỗi mạch tích hợp. Nhìn bằng máy. Nhận dạng và tổng hợp tiếng nói. Nhận dạng chữ viết tay và chữ ký. Xử lý ảnh, nén ảnh và nén số liệu. Các dịch vụ thông tin tự

động. Dịch ngôn ngữ nói thời gian thực. Hệ thống xử lý thanh toán của khách hàng. Định tuyến và chuyển mạch cho mạng ATM

- Quá trình sản xuất và người máy: Điều khiển quá trình sản xuất. Thiết kế và phân tích sản phẩm. Chuẩn đoán và giám sát quá trình máy móc. Hệ thống kiểm định chất lượng. Hệ thống lập kế hoạch và điều hành. Điều khiển vận động và hệ thống nhìn của robot.

- Y tế: Phân tích tế bào ung thư vú. Phân tích điện não đồ. Thiết kế bộ phận thay thế. Tối ưu hoá thời gian cấy ghép. Dò tìm và đánh giá các hiện tượng y học

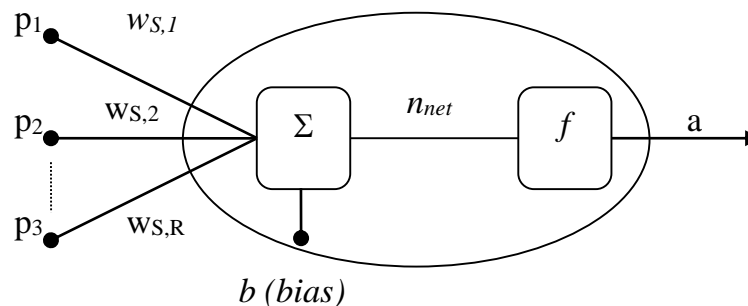
- Vận tải: Hệ thống chuẩn đoán phanh xe tải. Hệ thống định tuyến và lịch trình cho các phương tiện giao thông.

- Giải trí: Các hiệu ứng chuyên động, các trò chơi

2.1.3 Mô hình toán học và kiến trúc mạng nơron

2.1.3.1 Mô hình toán học của một nơron nhân tạo

Dựa trên những kiến thức cơ bản về nơron nhân tạo như đã trình bày ở phần trên, ta có thể xây dựng một mô hình toán học của nơron nhân tạo như Hình 2.3 dưới đây



Hình 2.3. Mô hình toán học mạng nơron nhân tạo

Các tín hiệu vào (còn gọi là mẫu vào) p_i ($i=1..R$) được đưa tới đầu vào của nơron S tạo thành ma trận tín hiệu vào P. Mỗi đầu vào của nơron S sẽ có một trọng số kí hiệu là $w_{s,i}$ ($i=1..R$) và các trọng số này tạo thành một ma trận trọng số đầu vào W của nơron. Mức ngưỡng θ của nơron có thể được biểu diễn trong mô hình toán học bằng hệ số bias b (gọi là thế hiệu dịch). Ta có $b=-\theta$. Hàm thế sau khớp nối (Post

Synaptic Potential function - PSP) là tổng của các tín hiệu vào có trọng số và hệ số bias. Như vậy tín hiệu vào là n_{net} sẽ được tính theo công thức sau:

$$n_{net} = w_{s,1}p_1 + w_{s,2}p_2 + + w_{s,R}p_R + b \quad (2.1)$$

Viết dưới dạng ma trận sẽ là:

$$n_{net} = WP + b \quad (2.2)$$

Xem các biểu thức trên thì ta có thể coi hệ số bias như trọng số của một đầu vào với tín hiệu bằng 1. Có một số loại nơron có thể bỏ qua hệ số bias này.

Hàm hoạt hoá (hay còn gọi là hàm truyền đạt) được kí hiệu là f sẽ biến đổi tín hiệu đầu vào net thành tín hiệu đầu ra nơron a. Ta có biểu thức:

$$a=f(n_{net})=f(WP+b) \quad (2.3)$$

Thông thường thì hàm đầu ra sẽ được chọn bởi người thiết kế tùy theo mục đích của mạng. Các trọng số và hệ số bias là các thông số điều chỉnh được của mạng nơron. Chúng được điều chỉnh bởi một số luật học. Như vậy quan hệ giữa đầu ra và các đầu vào của nơron sẽ tùy thuộc vào việc nơron đó được dùng cho các mục đích cụ thể nào.

2.1.3.2 Cấu trúc mạng nhân tạo

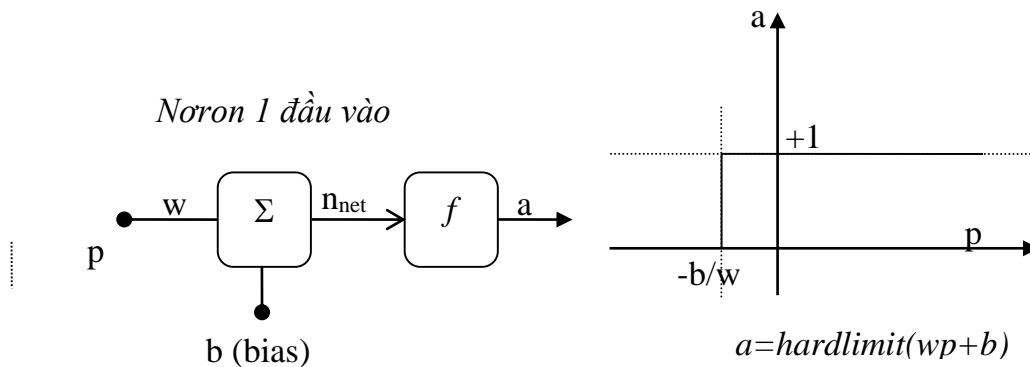
Mạng nơron thường được cấu tạo thành các lớp gồm lớp vào, lớp ra và các lớp ẩn. Các nơron trong một lớp chỉ nối với các nơron lớp tiếp theo, không cho phép có các liên kết giữa các nơron trong cùng một lớp

Lớp vào là lớp nhận thông tin từ số liệu gốc. Thông tin này được đưa đến đầu vào của một số hay toàn bộ các nơron của lớp tiếp theo (lớp ẩn). Như vậy mỗi nút của lớp ẩn sẽ nhận được tín hiệu của một số các nút lớp vào. Các giá trị này sẽ được nhân với hệ số nhân (trọng số) của các nút ẩn và đưa vào hàm thể sau khớp nối thực hiện chức năng đầu vào để tạo tín hiệu duy nhất net. Chức năng kích hoạt đầu ra được thực hiện bằng hàm hoạt hoá. Hàm này sẽ nhận tín hiệu đầu vào net để tạo ra tín hiệu đầu ra của nơron (kết xuất của nơron lớp ẩn). Tín hiệu ra của các nút ẩn lại được đưa đến các nút của lớp tiếp theo. Quá trình xử lý tương tự cho đến khi tín hiệu được đưa ra tại các nút lớp ra. Đây chính là tín hiệu đầu ra của mạng. Nó chính là giá trị của các biến cần tìm.

Mạng nơron có thể tổ chức theo kiểu liên kết đầy đủ tức là đầu ra của các nơron lớp trước sẽ có liên kết với tất cả các nơron ở lớp tiếp theo hoặc ngược lại theo kiểu không đầy đủ-mỗi đầu ra chỉ liên kết với một số nơron của lớp tiếp theo tùy theo chức năng của mạng.

2.1.3.3 Hàm truyền (Hàm hoạt hoá)

Hàm hoạt hoá có thể là một hàm tuyến tính hoặc phi tuyến của tín hiệu đầu vào $net-n_{net}$, nó được chọn để thoả mãn một số đặc điểm kỹ thuật của bài toán mà mạng nơron cần giải quyết [9]. Hình 2.4 cho thấy quan hệ giữa tín hiệu vào p và tín hiệu ra a của nơron một đầu vào với hàm kích hoạt là hàm *Hardlimit*.



Hình 2.4. Nơron 1 đầu vào với hàm hoạt hoá là hàm hardlimit

Bảng 2.1. Một số dạng hàm hoạt hóa trong mạng nơron nhân tạo

Tên hàm	Quan hệ đầu vào đầu ra
Hard-limit (<i>Hardlimit</i>)	$a=0 \quad net < 0$ $a=1 \quad net \geq 0$
Hard-limit đối xứng (<i>Symmetrical Hardlimit</i>)	$a=-1 \quad net < 0$ $a=1 \quad net \geq 0$
Đường thẳng <i>Linear</i>	$a=net$
Hàm logistic (<i>log-sigmoid</i>)	$a = \frac{1}{1 + e^{-net}}$
Hàm Hypecbol xích ma (<i>Hyperloic Tangent Sigmoid</i>)	$a = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}}$
Cạnh tranh (<i>Competitive</i>)	$a=1 \quad \text{nơron có net lớn nhất}$ $a=0 \quad \text{còn lại}$

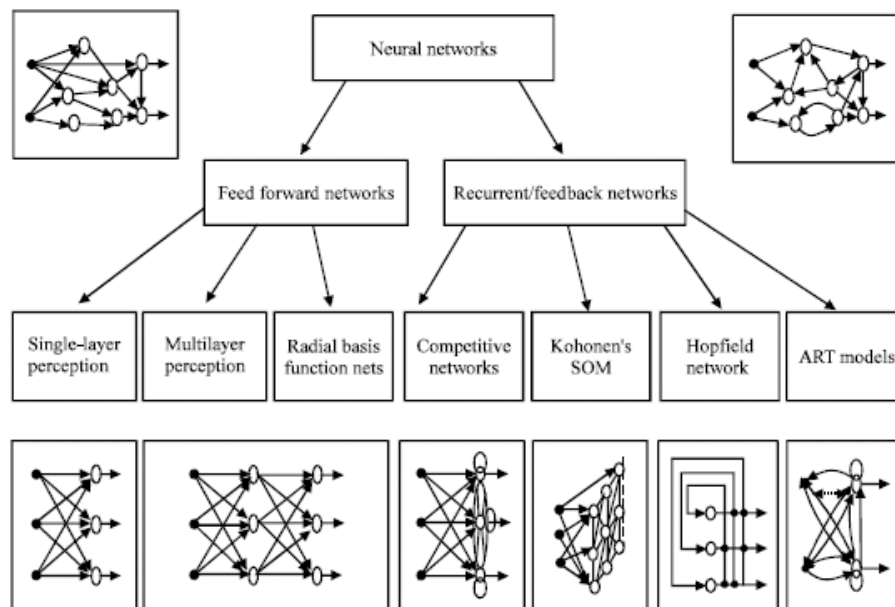
2.1.4 Phân loại mạng nơ ron

Có rất nhiều cách phân loại mạng nơron nhân tạo. Dựa vào các đặc trưng mạng nơron nhân tạo được phân loại như Hình 2.5 sau:

2.1.4.1 Phân loại mạng theo số lớp trong mạng

- Mạng một lớp: Là tập hợp các phần tử nơron có đầu vào và đầu ra trên cùng một phần tử. Nếu mạng nối đầu ra của các phần tử này với đầu vào của phần tử kia gọi là mạng tự liên kết (Auto associative).

- Mạng nhiều lớp: Gồm một lớp đầu vào và một lớp đầu ra riêng biệt. Các lớp nằm giữa lớp đầu vào và lớp đầu ra gọi là lớp ẩn (Hidden Layer).



Hình 2.5. Phân loại mạng nơ ron

2.1.4.2 Phân loại theo đường truyền tín hiệu

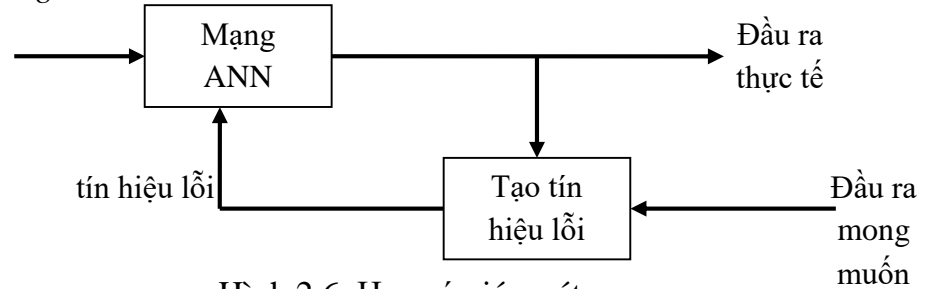
- Mạng truyền thẳng: Là mạng hai hay nhiều lớp mà quá trình truyền tín hiệu từ đầu ra lớp này đến đầu vào lớp kia theo một hướng.

- Mạng phản hồi: Là mạng mà trong đó một hoặc nhiều đầu ra của các phần tử lớp sau truyền ngược tới đầu vào của lớp trước.

- Mạng tự tổ chức: Là mạng có khả năng sử dụng những kinh nghiệm của quá khứ để thích ứng với những biến đổi của môi trường (không dự báo trước). Loại mạng này thuộc nhóm hệ học, thích nghi không cần có tín hiệu chỉ đạo từ bên ngoài.

2.1.5 Huấn luyện mạng nơron

2.1.5.1 Học có giám sát

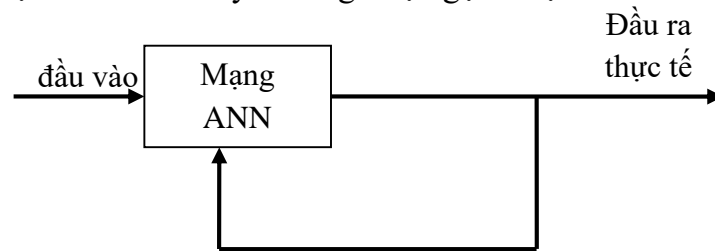


Hình 2.6. Học có giám sát

Với học có giám sát, mạng được cung cấp một tập K mẫu học $\{(P_K, T_K)\}$ với P_K là vectơ tín hiệu vào sẽ được đưa vào mạng và theo yêu cầu thì vectơ tín hiệu ra tương ứng sẽ phải là T_K . Thực tế thì vectơ đầu ra lại là Z_K và sẽ có một sai số so với T_K . Sai số này được giám sát và truyền trở lại hệ thống để hiệu chỉnh các trọng số liên kết và các hệ số bias của mạng. Quá trình đưa các mẫu học vào mạng được lặp đi lặp lại và mỗi lần như vậy các trọng số và hệ số bias luôn được hiệu chỉnh, cho đến khi mạng đạt một tiêu chuẩn nào đó thì dừng lại.

2.1.5.2 Học không có giám sát

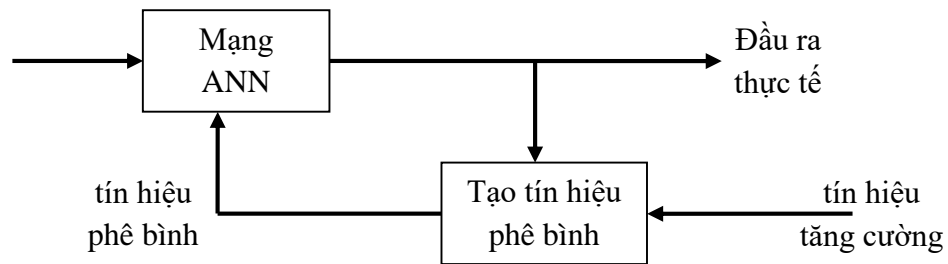
Khác với kiểu huấn luyện có giám sát, ở đây tập huấn luyện chỉ bao gồm các vectơ tín hiệu đầu vào $\{P_K\}$. Huấn luyện là quá trình hệ thống tự tìm ra các nhóm hợp của số liệu vào. Điều này thường được gọi là sự tự tổ chức hay sự thích ứng.



Hình 2.7. Học không có giám sát

Huấn luyện không được giám sát khá phức tạp. Việc huấn luyện cho mạng Kohonel là một ví dụ. Các tín hiệu đầu ra không được biết chính xác và việc hiệu chỉnh trọng số ứng với một mẫu tín hiệu vào để đầu ra của nơron “chiến thắng” lớn hơn hoặc gần giá trị mong muốn, còn tín hiệu đầu ra của các nơron lân cận sẽ được giảm đi.

2.1.5.3 Học tăng cường



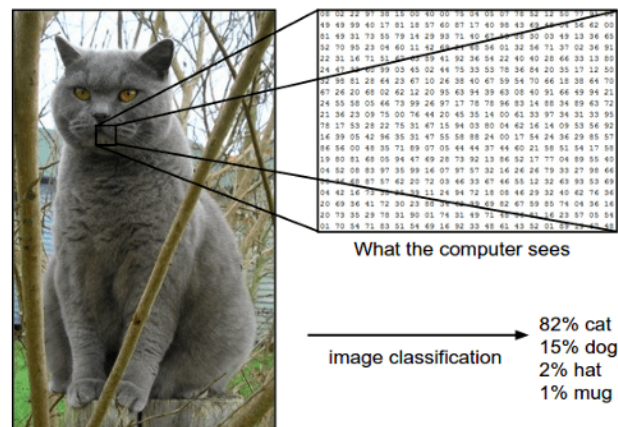
Hình 2.8. Học tăng cường

Học tăng cường là một biến thể của học có giám sát. Nó là quá trình học dựa trên việc cập nhật trọng số dựa vào một tín hiệu phê bình nào đó gọi là tín hiệu tăng cường. Tín hiệu này được đưa đến từ môi trường bên ngoài và được sử dụng như là một đại lượng ước lượng từ đó đưa đến cho mạng những chỉ dẫn yêu cầu để mạng cập nhật điều chỉnh tập trọng số cho thích hợp nhất.

2.2 Mạng nơron CNN

2.2.1 Giới thiệu

Tương tự như việc trẻ em học cách nhận diện đối tượng, chúng ta cần cho thuật toán học rất nhiều hình ảnh trước khi nó có thể đưa ra phân loại cho hình ảnh đầu vào mà nó chưa từng thấy [14] .



Hình 2.9. Cách máy tính “nhìn” một hình [16]

Máy tính “nhìn” theo cách khác con người. Trong thế giới máy tính chỉ có những con số. Mỗi hình ảnh có thể được biểu diễn dưới dạng mảng 2 chiều những con số được gọi là các pixel.

Mặc dù máy tính nhìn nhận theo cách khác con người, chúng ta vẫn có thể dạy máy tính nhận diện các mẫu như con người. Điều quan trọng là chúng ta cần nghĩ về hình ảnh theo một cách khác đi.

Để dạy thuật toán nhận diện đối tượng trong hình ảnh, ta sử dụng một loại mạng ANN, đó là CNN. Tên của nó được dựa trên phép tính quan trọng được sử dụng trong mạng- tích chập.

Mạng CNN lấy cảm hứng từ não người. Nghiên cứu trong những thập niên 1950 và 1960 của D.H Hubel và T.N Wiesel trên não của động vật đã đề xuất một mô hình mới cho việc cách mà động vật nhìn nhận thế giới. Trong báo cáo, hai ông đã diễn tả 2 loại tế bào nơ-ron trong não và cách hoạt động khác nhau: tế bào đơn giản (simple cell – S cell) và tế bào phức tạp (complex cell – C cell).

Các tế bào đơn giản được kích hoạt khi nhận diện các hình dáng đơn giản như đường nằm trong một khu vực cố định và một góc cạnh của nó. Các tế bào phức tạp có vùng tiếp nhận lớn hơn và đầu ra của nó không nhạy cảm với những vị trí cố định trong vùng.

Trong thị giác, vùng tiếp nhận của một nơ-ron tương ứng với một vùng trên võng mạc nơi mà sẽ kích hoạt nơ-ron tương ứng.

Năm 1980, Fukushima đề xuất mô hình mạng nơ-ron có cấp bậc gọi là neocognitron. Mô hình này dựa trên khái niệm về S cell và C cell. Mạng neocognitron có thể nhận diện mẫu dựa trên việc học hình dáng của đối tượng.

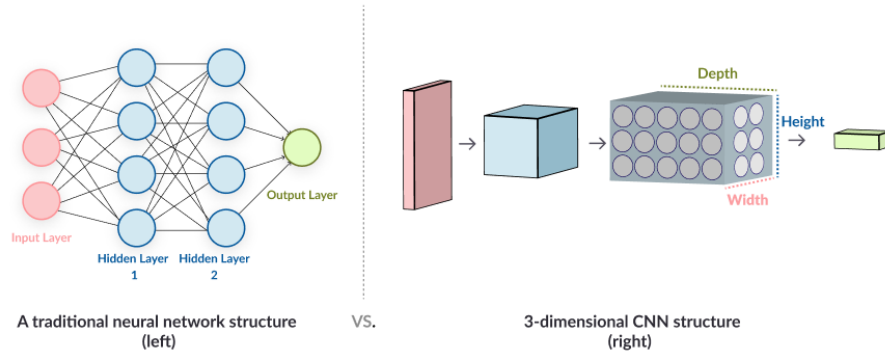
Sau đó vào năm 1998, mạng CNN được giới thiệu bởi Bengio, Le Cun, Bottou và Haffner. Mô hình đầu tiên của họ được gọi tên là LeNet-5. Mô hình này có thể nhận diện chữ số viết tay.

2.2.2 Kiến trúc mạng CNN

Mạng CNN có kiến trúc khác với Mạng Nơ-ron thông thường. Mạng ANN bình thường chuyển đổi đầu vào thông qua hàng loạt các tầng ẩn. Mỗi tầng là một tập các nơ-ron và các tầng được liên kết đầy đủ với các nơ-ron ở tầng trước đó. Và ở tầng cuối cùng sẽ là tầng kết quả đại diện cho dự đoán của mạng.

Đầu tiên, mạng CNN được chia thành 3 chiều: rộng, cao, và sâu. Kế đến, các nơ-ron trong mạng không liên kết hoàn toàn với toàn bộ nơ-ron kế đến nhưng chỉ liên

kết tới một vùng nhỏ. Cuối cùng, một tầng đầu ra được tối giản thành véc-tơ của giá trị xác suất.

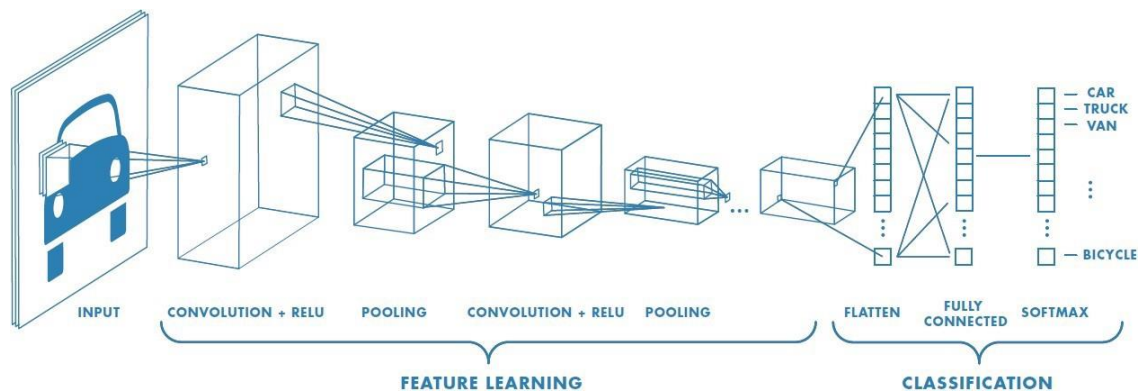


Hình 2.10. Mạng nơ-ron thông thường (trái) và CNN (phải)

Mạng CNN gồm hai thành phần:

Phần tầng ẩn hay phần rút trích đặc trưng: trong phần này, mạng sẽ tiến hành tính toán hàng loạt phép tích chập và phép hợp nhất (pooling) để phát hiện các đặc trưng. Ví dụ: nếu ta có hình ảnh con ngựa vằn, thì trong phần này mạng sẽ nhận diện các sọc vằn, hai tai, và bốn chân của nó.

Phần phân lớp: tại phần này, một lớp với các liên kết đầy đủ sẽ đóng vai trò như một bộ phân lớp các đặc trưng đã rút trích được trước đó. Tầng này sẽ đưa ra xác suất của một đối tượng trong hình.



Hình 2.11. Kiến trúc mạng CNN

2.2.2.1 Trích rút đặc trưng

▪ Lớp tích chập

Tích chập là một khối quan trọng trong CNN. Thuật ngữ tích chập được dựa trên một phép hợp nhất toán học của hai hàm tạo thành hàm thứ ba. Phép toán này kết hợp hai tập thông tin khác nhau.

Trong trường hợp CNN, tích chập được thực hiện trên giá trị đầu vào của dữ liệu và kernel/filter (thuật ngữ này được sử dụng khác nhau tùy tình huống) để tạo ra một bản đồ đặc trưng (feature map).

Ta thực hiện phép tích chập bằng cách trượt kernel/filter theo dữ liệu đầu vào. Tại mỗi vị trí, ta tiến hành phép nhân ma trận và tính tổng các giá trị để đưa vào bản đồ đặc trưng. Thao tác này đã được minh họa cụ thể trong Hình 1.8 ở mục 1.3.

Trong thực tế, tích chập được thực hiện trên không gian 3 chiều. Vì mỗi hình ảnh được biểu diễn dưới dạng 3 chiều: rộng, cao, và sâu. Chiều sâu ở đây chính là giá trị màu sắc của hình (RGB).

Ta thực hiện phép tích chập trên đầu vào nhiều lần khác nhau. Mỗi lần sử dụng một kernel/filter khác nhau. Kết quả ta sẽ thu được những bản đồ đặc trưng khác nhau. Cuối cùng, ta kết hợp toàn bộ bản đồ đặc trưng này thành kết quả cuối cùng của tầng tích chập.

▪ Lớp ReLU

Tương tự như mạng nơ-ron thông thường, ta sử dụng một hàm kích hoạt (activate function) để có đầu ra dưới dạng phi tuyến. Trong trường hợp CNN, đầu ra của phép tích chập sẽ đi qua hàm kích hoạt nào đó ví dụ như hàm tính chỉnh các đơn vị tuyến tính (Rectified linear units - ReLU).

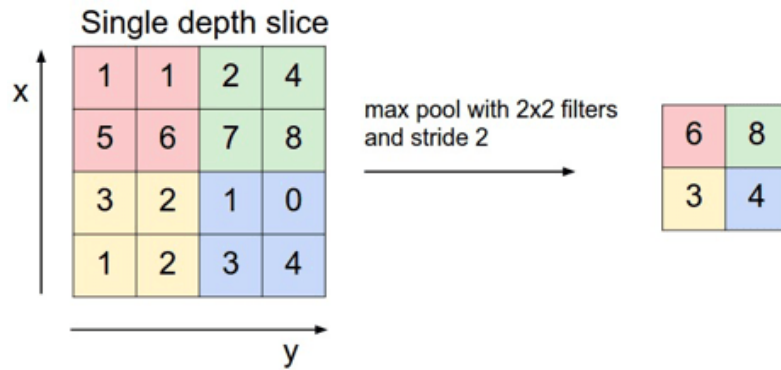
Trong quá trình trượt kernel/filter trên dữ liệu đầu vào, ta sẽ quy định một bước nhảy (stride) với mỗi lần di chuyển. Thông thường ta lựa chọn thường chọn bước nhảy là 1. Nếu kích thước bước nhảy tăng, kernel/filter sẽ có ít ô trùng lặp.

Bởi vì kích thước đầu ra luôn nhỏ hơn đầu vào nên ta cần một phép xử lý đầu vào để đầu ra không bị co giãn. Đơn giản ta chỉ cần thêm một lề nhỏ vào đầu vào. Một lề (padding) với giá trị 0 sẽ được thêm vào xung quanh đầu vào trước khi thực hiện phép tích chập.

▪ Lớp pooling

Thông thường, sau mỗi tầng tích chập, ta sẽ cho kết quả đi qua một tầng hợp nhất (pooling layer). Mục đích của tầng này là để nhanh chóng giảm số chiều. Việc này giúp giảm thời gian học và hạn chế việc overfitting.

Một phép hợp nhất đơn giản thường được dùng đó là max pooling, phép này lấy giá trị lớn nhất của một vùng để đại diện cho vùng đó. Kích thước của vùng sẽ được xác định trước để giảm kích thước của bản đồ đặc trưng nhanh chóng nhưng vẫn giữ được thông tin cần thiết (Hình 2.12).



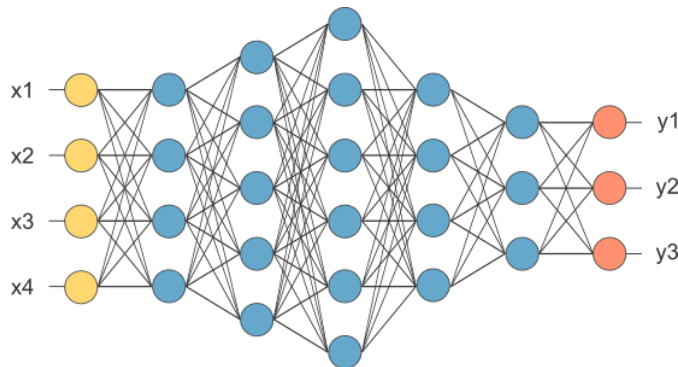
Hình 2.12. Max pooling kích thước 2×2

Như vậy, khi thiết kế phần rút trích đặc trưng của mạng CNN, ta cần chú ý đến 4 siêu tham số quan trọng là: Kích thước kernel/filter, Số lượng kernel/filter, Kích thước bước nhảy (stride), Kích thước lề (padding).

2.2.2.2 Phân lớp

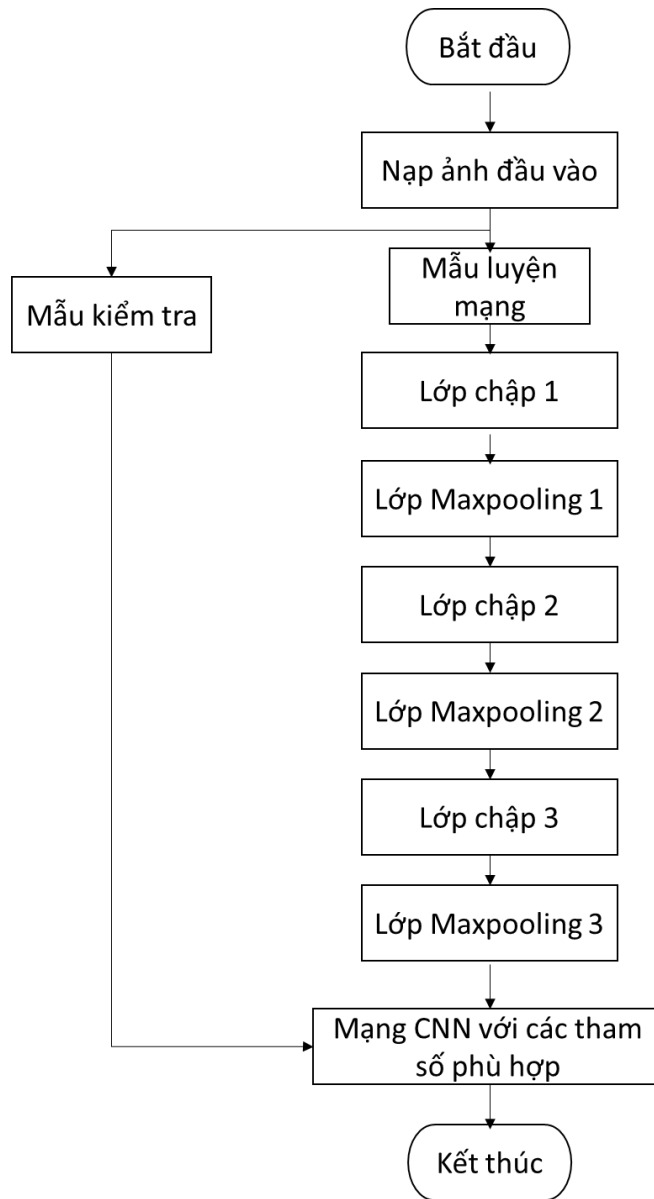
Trong phần phân lớp, ta sử dụng một vài tầng với kết nối đầy đủ để xử lý kết quả của phần tích chập. Vì đầu vào của mạng liên kết đầy đủ là 1 chiều, ta cần làm phẳng đầu vào trước khi phân lớp. Tầng cuối cùng trong mạng CNN là một tầng liên kết đầy đủ, phần này hoạt động tương tự như mạng nơ-ron thông thường.

Kết quả thu được cuối cùng cũng sẽ là một véc-tơ với các giá trị xác suất cho việc dự đoán như mạng nơ-ron thông thường.



Hình 2.13. Lớp kết nối đầy đủ

2.2.3 Ứng dụng CNN trong phân loại ảnh



Hình 2.14. Các bước phân loại ảnh sử dụng mạng CNN

Các bước để thực hiện phân loại hình ảnh dựa trên mạng CNN được mô tả trong Hình 2.14. Đầu tiên, kho dữ liệu ảnh đầu vào được nạp. Ảnh này được chia làm hai phần, một phần dành cho luyện mạng và một phần cho kiểm tra. Trước tiên, ta phải lựa chọn cấu trúc mạng CNN bao gồm số lượng lớp ẩn, các tham số trong mỗi lớp ẩn như kích thước trường tiếp nhận cục bộ, stride, padding. Ảnh luyện mạng sau đó được đưa vào lớp chập 1 để thực hiện tích chập trên ảnh và thực hiện hàm ReLU. Sau đó, kết quả được đưa đến quá trình thực hiện pooling với tham số pooling size phù

hợp để giảm kích cỡ ảnh. Ảnh sẽ tiếp tục được đưa thêm qua các lớp tích chập nữa cho đến khi đạt được kết quả mong muốn. Kết quả này được dàn phẳng và đưa vào lớp kết nối đầy đủ. Cuối cùng là quá trình thực hiện các activation function và phân loại ảnh. Quá trình luyện mạng sẽ kết thúc sau khi tổng sai số nhỏ hơn một ngưỡng cho phép hoặc sau một số thế hệ cho trước (điều kiện hội tụ). Kết thúc của quá trình luyện mạng là cấu trúc mạng CNN với các tham số phù hợp. Để kiểm tra, các mẫu ảnh kiểm tra được đưa qua mạng CNN rồi thực hiện đánh giá sai số.

2.3 Xây dựng mạng CNN cho phân loại ảnh

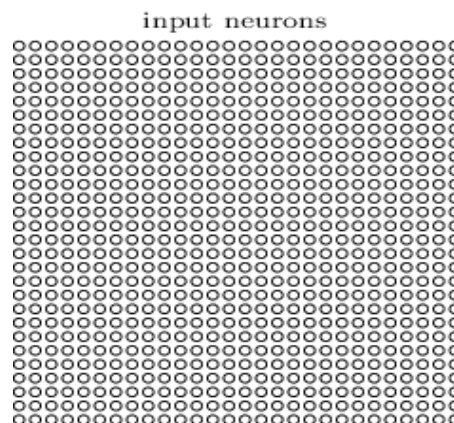
Trước tiên, đối với mỗi điểm ảnh trong ảnh đầu vào, ta mã hóa cường độ của điểm ảnh là giá trị của nơron tương ứng trong tầng đầu vào.

Ví dụ đối với bài toán nhận dạng chữ viết tay từ tập dữ liệu MNIST, mỗi bức ảnh kích thước 28×28 điểm ảnh. Do vậy, mạng có 784 (28×28) nơron đầu vào (Hình 2.15). Sau đó ta huấn luyện trọng số (weight) và độ lệch (bias) để đầu ra của mạng như ta mong đợi là xác định chính xác ảnh các chữ số 0, 1, 2...8, 9.

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.

2.3.1 Trường tiếp nhận cục bộ (Local receptive fields)

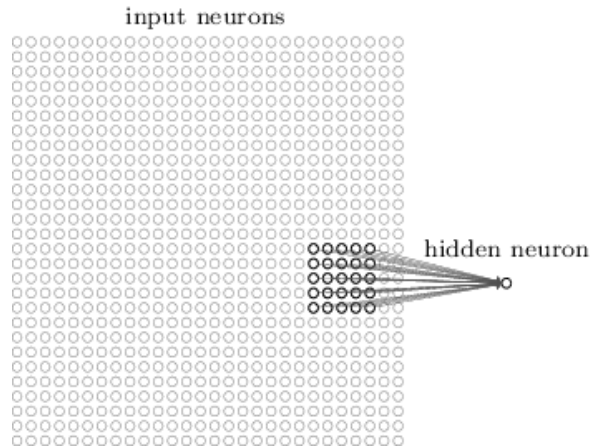
Như thường lệ chúng ta sẽ kết nối các điểm ảnh đầu vào cho các nơron ở tầng ẩn. Nhưng chúng ta sẽ không kết nối mỗi điểm ảnh đầu vào cho mỗi neuron ẩn. Thay vào đó, chúng ta chỉ kết nối trong phạm vi nhỏ, các vùng cục bộ của bức ảnh.



Hình 2.15. Lớp input gồm 28×28 nơron cho nhận dạng chữ từ tập dữ liệu MNIST

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 5×5 , tương ứng với 25 điểm

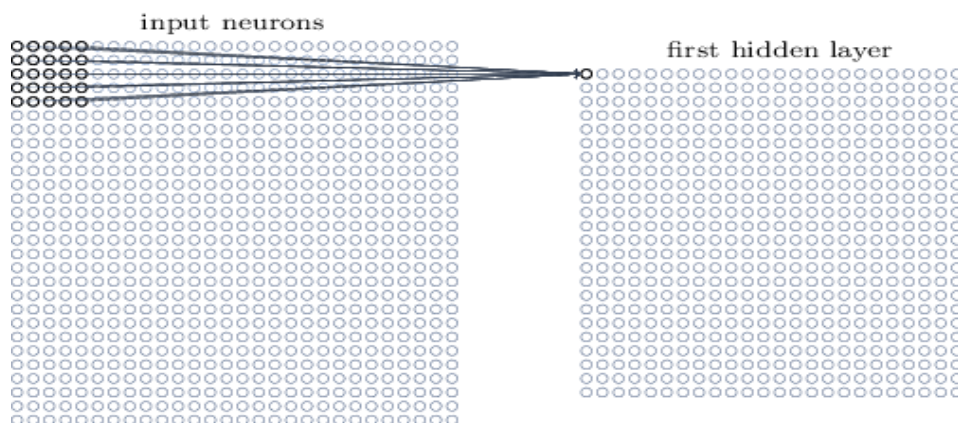
ảnh đầu vào. Vì vậy, đối với một nơron ẩn cụ thể, chúng ta có thể có các kết nối như Hình 2.16 sau:



Hình 2.16. Kết nối vùng 5x5 nơ ron input với nơ ron lớp ẩn

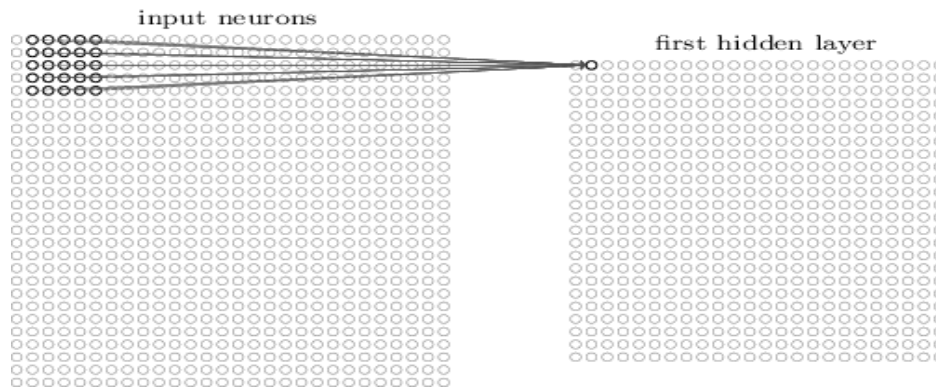
Vùng đó trong bức ảnh đầu vào được gọi là vùng tiếp nhận cục bộ cho nơron ẩn. Đó là một cửa sổ nhỏ trên các điểm ảnh đầu vào. Mỗi kết nối sẽ học một trọng số và nơron ẩn cũng sẽ học một độ lệch (overall bias). Ta có thể hiểu rằng, nơron lớp ẩn cụ thể học để phân tích trường tiếp nhận cục bộ cụ thể của 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 (Hình 2.17):



Hình 2.17. Vị trí bắt đầu của trường tiếp nhận cục bộ

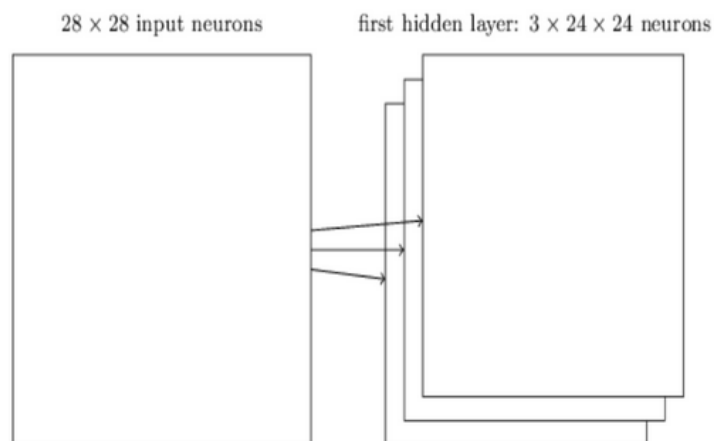
Sau đó, chúng ta trượt trường tiếp nhận cục bộ trên bởi một điểm ảnh bên phải (tức là bằng một nơron), để kết nối với một nơron ẩn thứ hai (Hình 2.18):



Hình 2.18. Vị trí thứ 2 của trường tiếp nhận cục bộ và nơ ron lớp ẩn

Cứ như vậy, ta sẽ xây dựng các lớp ẩn đầu tiên. Lưu ý rằng nếu chúng ta có một ảnh đầu vào 28×28 và 5×5 trường tiếp nhận cục bộ thì ta sẽ có 24×24 nơ ron trong lớp ẩn. Có được điều này là do chúng ta chỉ có thể di chuyển các trường tiếp nhận cục bộ ngang qua 23 nơ ron (hoặc xuống dưới 23 nơ ron), trước khi chạm với phía bên phải (hoặc dưới) của ảnh đầu vào.

Với bài toán nhận dạng ảnh người ta thường gọi ma trận lớp ẩn 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.

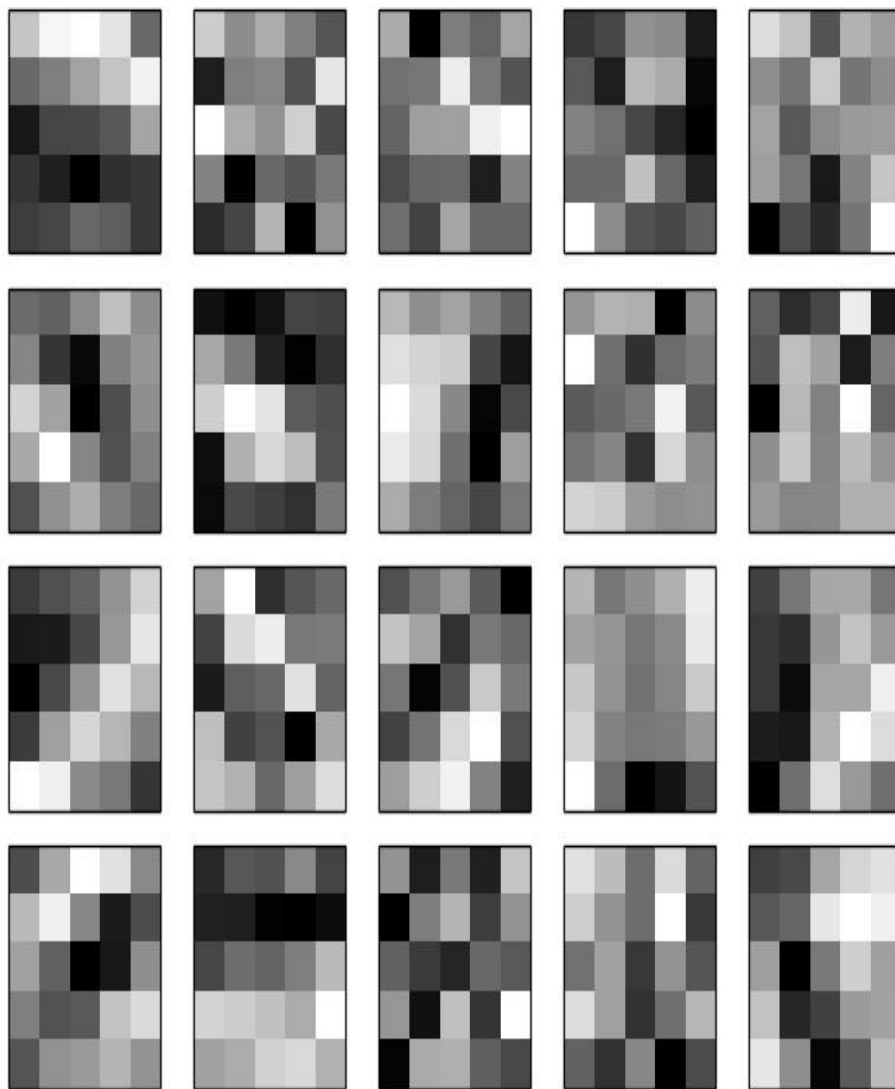


Hình 2.19. Trường tiếp nhận cục bộ với ba bản đồ đặc trưng

Trong ví dụ ở Hình 2.19, có 3 bản đồ đặc trưng. Mỗi bản đồ đặc trưng được xác định bởi một tập 5×5 trọng số chia sẻ, và một độ lệch chia sẻ duy nhất. Kết

quả là các mạng có thể phát hiện 3 loại đặc trưng khác nhau, với mỗi đặc trưng được phát hiện trên toàn bộ ảnh.

Trong thực tế mạng CNN có thể sử dụng nhiều bản đồ đặc trưng hơn. Một trong những mạng chập đầu tiên là LeNet-5, sử dụng 6 bản đồ đặc trưng, mỗi bản đồ được liên kết đến một trường tiếp nhận cục bộ 5×5 , để phát hiện các kí tự MNIST. Vì vậy, các ví dụ minh họa ở trên là thực sự khá gần LeNet-5. Trong một số nghiên cứu gần đây sử dụng lớp tích chập với 20 và 40 bản đồ đặc trưng.



Hình 2.20. Trường tiếp nhận cục bộ với 20 bản đồ đặc trưng

Trên đây là 20 ảnh tương ứng với 20 bản đồ đặc trưng khác nhau (hay còn gọi là bộ lọc, hay là nhân). Mỗi bản đồ được thể hiện là một hình khối kích thước

5×5 , tương ứng với 5×5 trọng số trong trường tiếp nhận cục bộ. Khối trắng có nghĩa là một trọng số nhỏ hơn, vì vậy các bản đồ đặc trưng đáp ứng ít hơn để tương ứng với điểm ảnh đầu vào. Khối sẫm màu hơn có nghĩa là trọng số lớn hơn, do đó, các bản đồ đặc trưng đáp ứng nhiều hơn với các điểm ảnh đầu vào tương ứng.

Có thể thấy rằng, trường tiếp nhận cục bộ 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.

2.3.2 Trọng số chia sẻ và độ lệch (*Shared weights and biases*)

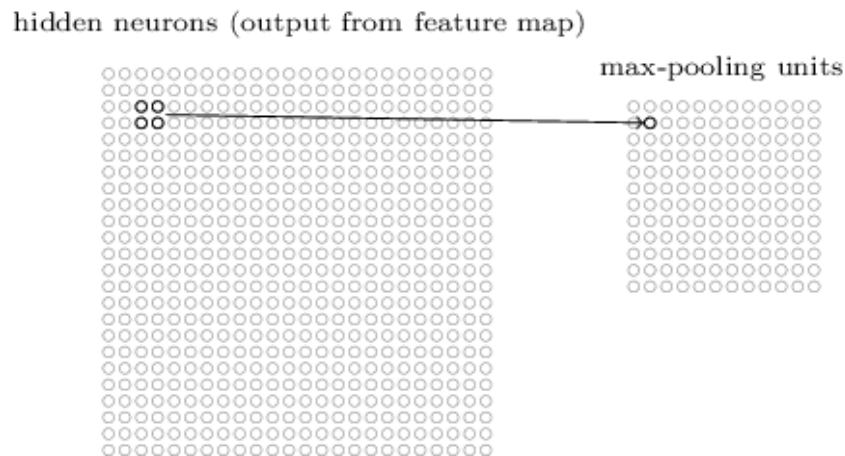
Đầu tiên, các trọng số cho mỗi filter (kernel) phải giống nhau. Tất cả các nơ-ron trong lớp ẩn đầu sẽ phát hiện chính xác feature tương tự chỉ ở các vị trí khác nhau trong hình ảnh đầu vào. Chúng ta gọi việc map từ input layer sang hidden layer là một feature map. Ta cần tìm ra mối quan hệ giữa số lượng Feature map với số lượng tham số.

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 $28 \times 28 = 784$ neuron đầu vào như vậy ta chỉ có 30 neuron 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.

2.3.3 Lớp chứa hay lớp tổng hợp (*Pooling layer*)

Ngoài các lớp tích chập vừa mô tả, mạng nơ-ron 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.

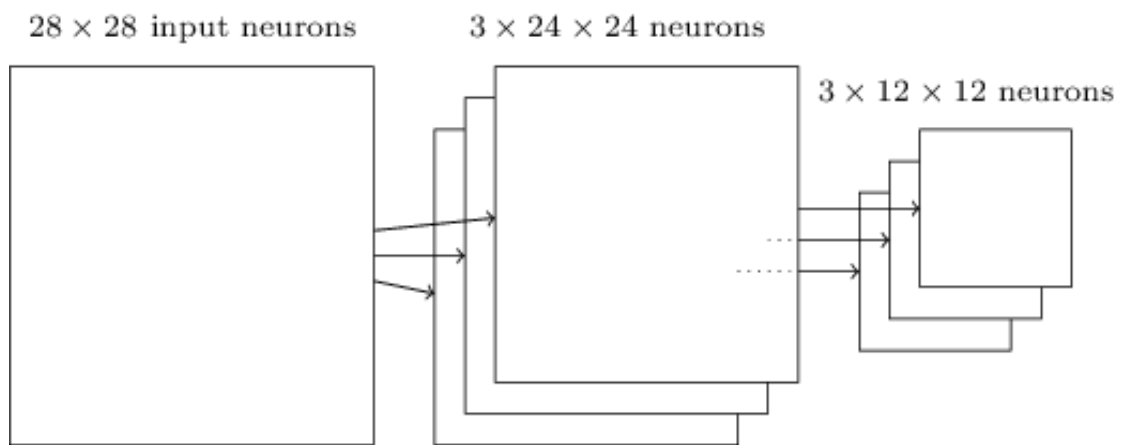
Ví dụ, mỗi đơn vị trong lớp pooling có thể thu gọn một vùng 2×2 nơ-ron trong lớp trước. Một thủ tục pooling phổ biến là max-pooling. Trong maxpooling, một đơn vị pooling chỉ đơn giản là kết quả đầu ra kích hoạt giá trị lớn nhất trong vùng đầu vào 2×2 , như minh họa trong sơ đồ sau:



Hình 2.21. Ví dụ về Max pooling 2x2

Lưu ý rằng bởi vì chúng ta có 24×24 nơron đầu ra từ các lớp tích chập, sau khi pooling chúng ta có 12×12 nơron.

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:

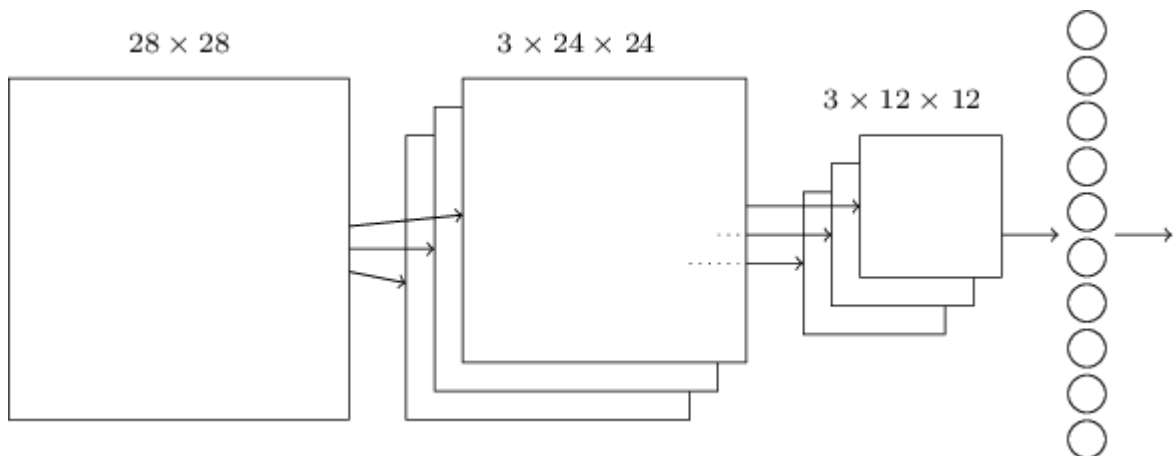


Hình 2.22. Max pooling với ba bản đồ đặc trưng

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.

Max-pooling không phải là kỹ thuật duy nhất được sử dụng để pooling. Một phương pháp phổ biến khác được gọi là L2 pooling. Ở đây, thay vì lấy giá trị kích hoạt tối đa (maximum activation) của một vùng 2×2 nơron, chúng ta lấy căn bậc hai của tổng các bình phương của kích hoạt trong vùng 2×2 . Trong khi các chi tiết thì khác nhau, nhưng về trực giác thì tương tự như max-pooling: L2 pooling là một cách để cô đọng thông tin từ các lớp tích chập. Trong thực tế, cả hai kỹ thuật đã được sử dụng rộng rãi. Và đôi khi người ta sử dụng các loại pooling khác.

Như vậy, 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 phân tích ở trên, nhưng có thêm một lớp 10 nơron đầu ra, tương ứng với 10 giá trị có thể cho các số MNIST ('0', '1', '2', v.v...):



Hình 2.23. Một kiến trúc mạng CNN cho nhận dạng chữ viết từ dữ liệu MNIST

Mạng bắt đầu với 28×28 nơron đầu vào, được sử dụng để mã hóa các cường độ điểm ảnh cho ảnh MNIST. Sau đó là một lớp tích chập sử dụng 5×5 trường tiếp nhận cục bộ và 3 bản đồ đặc trưng. Kết quả là một lớp $3 \times 24 \times 24$ nơron lớp ẩn. Bước tiếp theo là một lớp max-pooling, áp dụng cho 2×2 vùng qua 3 bản đồ đặc trưng (feature maps). Kết quả là một lớp $3 \times 12 \times 12$ nơron đặc trưng ở tầng ẩn.

Lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối. Lớp này nối mọi nơron từ lớp max-pooled tới mọi nơron của tầng ra.

2.3.4 Cách chọn tham số cho CNN

Hiệu quả hoạt động của mạng CNN phụ thuộc rất nhiều vào việc lựa chọn các tham số sau:

- 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

Trong thực tế, tùy vào ứng dụng cụ thể mà ta chọn các tham số khác nhau. Thông thường ta sẽ thực hiện nhiều lần việc train test để chọn ra được param tốt nhất (Phương pháp thử sai.).

2.4 Cập nhật một số hướng nghiên cứu về bài toán phân loại ảnh sử dụng mạng nơ ron CNN

2.4.1 Các nghiên cứu trên thế giới

CNN thường được sử dụng trong các hệ thống nhận dạng hình ảnh [11], [12] [13]. Vào năm 2012, một tỷ lệ lỗi 0,23% trên cơ sở dữ liệu MNIST đã được báo cáo. Một bài báo khác về việc sử dụng CNN để phân loại hình ảnh đã báo cáo rằng quá trình luyện mạng "nhanh đến mức đáng ngạc nhiên"; trong cùng một bài báo, các kết quả được công bố tốt nhất tính đến năm 2011 đã đạt được trong cơ sở dữ liệu MNIST và cơ sở dữ liệu NORB. Sau đó, một CNN tương tự có tên AlexNet đã giành chiến thắng trong Thử thách nhận dạng hình ảnh quy mô lớn ImageNet 2012 [12].

Khi áp dụng cho nhận dạng khuôn mặt, CNN đã đạt được mức giảm lớn về tỷ lệ lỗi. Một bài báo khác đã báo cáo tỷ lệ nhận dạng 97,6% trên "5.600 ảnh tĩnh của hơn 10 đối tượng". CNN được sử dụng để đánh giá chất lượng video một cách khách quan sau khi huấn luyện thủ công; hệ thống kết quả có lỗi bình phương trung bình gốc rất thấp.

Thử thách nhận dạng hình ảnh quy mô lớn ImageNet là một chuẩn mực trong phân loại và phát hiện đối tượng, với hàng triệu hình ảnh và hàng trăm lớp đối tượng. Trong ILSVRC 2014, một thách thức nhận dạng hình ảnh quy mô lớn, hầu hết mọi nhóm được xếp hạng cao đều sử dụng CNN làm khung cơ bản. Người chiến thắng GoogLeNet (nền tảng của DeepDream) đã tăng độ chính xác trung bình trung bình của phát hiện đối tượng lên 0,439329 và giảm lỗi phân loại xuống 0,06656, kết quả tốt nhất cho đến nay. Mạng CNN này sử dụng hơn 30 lớp. Hiệu suất của các CNN trong các thử nghiệm ImageNet gần bằng với con người. Các thuật toán tốt nhất vẫn phải vật lộn với các vật thể nhỏ hoặc mỏng, chẳng hạn như một con kiến nhỏ trên thân cây hoa hoặc một người cầm một chiếc bút lông trong tay. Họ cũng gặp rắc rối với hình ảnh đã bị méo với các bộ lọc, một hiện tượng ngày càng phổ biến với máy ảnh kỹ thuật số hiện đại. Ngược lại, những loại hình ảnh đó hiếm khi gây rắc rối cho con người. Con người, tuy nhiên, có xu hướng gặp rắc rối với các vấn đề khác. Ví dụ, chúng không giỏi trong việc phân loại các đối tượng thành các loại hạt mịn như giống chó hoặc loài chim cụ thể, trong khi mạng CNN có thể xử lý việc này [17] .

Vào năm 2015, một CNN nhiều lớp đã chứng minh khả năng phát hiện khuôn mặt từ nhiều góc độ khác nhau, bao gồm lộn ngược, ngay cả khi bị che khuất một phần, với hiệu suất cạnh tranh. Mạng được đào tạo trên cơ sở dữ liệu gồm 200.000 hình ảnh bao gồm các khuôn mặt ở nhiều góc độ và định hướng khác nhau và hơn 20 triệu hình ảnh không có khuôn mặt. Họ đã sử dụng lô 128 hình ảnh trên 50.000 lần lặp [17] .

2.4.2 Các nghiên cứu trên trong nước

Có thể nói, trong những năm gần đây, các nghiên cứu về học sâu và đặc biệt là mạng CNN được công bố với số lượng rất lớn. Khó có thể thống kê hết được các công trình đã xuất bản. Trong lĩnh vực phân loại hình ảnh, qua việc tham khảo tài liệu, học viên có thể kể đến một số công trình tiêu biểu sau:

Năm 2016, tác giả Lê Thị Thu Hằng đã báo cáo luận văn thạc sĩ về “Nghiên cứu về mạng neural tích chập và ứng dụng cho bài toán nhận dạng biển số xe” [5] .

Trong đó, tác giả cũng đã sử dụng một cấu trúc mạng CNN cho việc nhận dạng chữ số từ ảnh chụp biển số xe với tỉ lệ nhận dạng chính xác 99%.

Năm 2017, trong luận văn “Nhận dạng và phân loại hoa quả trong ảnh màu” [4], tác giả Nguyễn Đắc Thành cũng đề xuất sử dụng mạng CNN AlexNet để phân biệt 40 loại hoa quả trong ảnh màu với độ chính xác 98,67%.

Năm 2018, tác giả Huỳnh Văn Nhứt báo cáo luận văn thạc sĩ “Nhận dạng chữ số viết tay sử dụng kỹ thuật học sâu” [6]. Trong luận văn này, tác giả sử dụng mô hình CNN trong công việc xây dựng nhận dạng ký tự số viết tay đạt được kết quả thực nghiệm dựa trên 10.000 tập mẫu với độ chính xác trên 99%.

Năm 2019, tác giả Nguyễn Văn Doanh và Phạm Thế Bảo đề xuất sử dụng mạng CNN cho nhận dạng mặt người [3]. Họ đã thử nghiệm trên 02 cơ sở dữ liệu là ORL, CyberSoft. Thử nghiệm trên ORL và CyberSoft, họ phát hiện ra trọng số toàn cục của phương pháp bỏ phiếu. Sử dụng trọng số này, các tác giả đã đạt độ chính xác cao 99,375%, 99,5% trên cơ sở dữ liệu ORL, CyberSoft.

Trong một nghiên cứu mới nhất năm 2020 [7], các tác giả nghiên cứu về mạng CNN sử dụng mô hình VGG16 ứng dụng trong việc xây dựng hệ thống nhận dạng khuôn mặt tự động từ video. Độ chính xác nhận dạng khuôn mặt của mô hình trong điều kiện lý tưởng đã đạt hoặc vượt qua cả con người. Từ những kết quả đã thử nghiệm của mô hình cho thấy, có thể xây dựng các ứng dụng dựa trên phân loại và nhận dạng khuôn mặt, như: hệ thống chấm công tự động, điểm danh tự động trong các cơ sở đào tạo, và các hệ thống kiểm soát an ninh, phòng chống tội phạm.

Qua việc khảo sát, cập nhật các công trình công bố liên quan đến việc sử dụng mạng nơ ron CNN cho phân loại tra, ta có thể rút ra một số kết luận như sau:

- Để thực hiện bài toán phân loại ảnh, thông thường phải áp dụng các thuật toán phù hợp cho hai khâu trích chọn đặc trưng và phân lớp đối tượng. Tuy nhiên, với việc sử dụng mạng CNN ta có thể đồng thời thực hiện hai khâu trên. Kết quả thu được là tốt hơn nhiều so với các phương pháp truyền thống (không sử dụng DeepLearning).

- Để nâng cao chất lượng hoạt động, mạng nơ ron CNN cần phải được lựa chọn các thông số phù hợp với từng ứng dụng cụ thể.

Từ kết luận trên, phần tiếp theo của luận văn sẽ nghiên cứu tìm ra cấu trúc phù hợp của mạng CNN cho hai bài toán nhận dạng chữ viết tay và giải mã captcha.

2.5 Kết luận chương

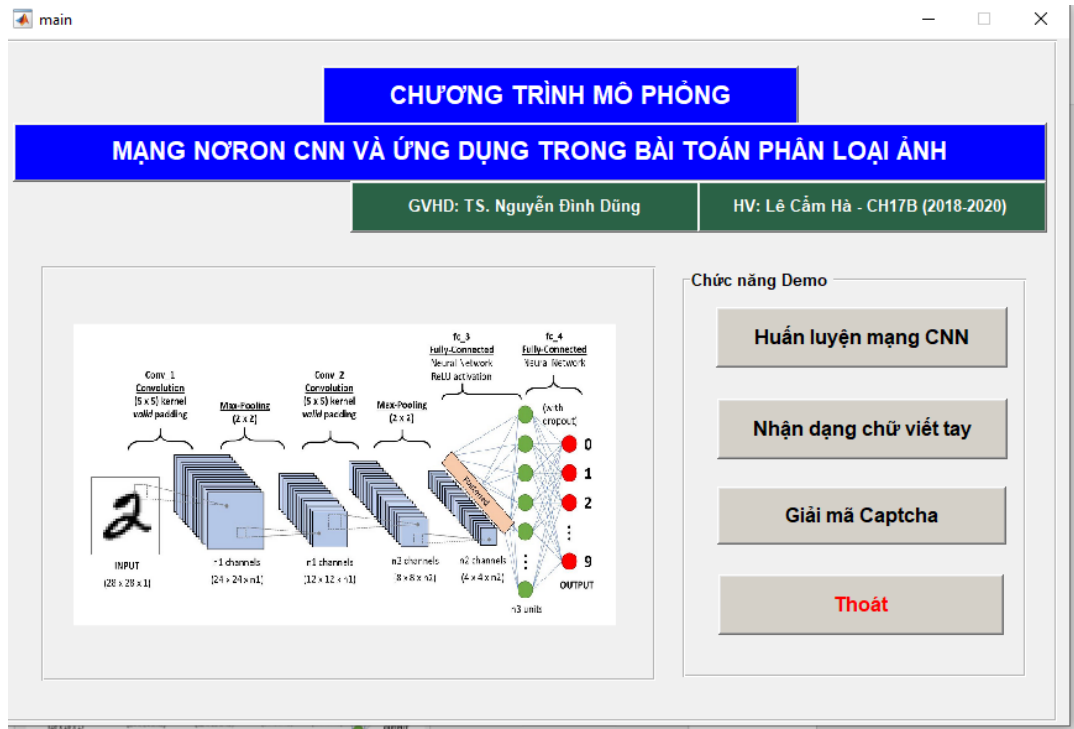
Có thể thấy rằng, mạng nơ ron CNN là một công cụ hữu hiệu trong việc xử lý các lớp bài toán phi tuyến phức tạp. Đặc biệt là trong bài toán phân loại ảnh. Chính vì vậy, nội dung chương 2 đã trình bày các kiến thức tổng quan cho việc xây dựng mạng nơ ron CNN và thực hiện các thuật toán học nhằm điều chỉnh trọng số của các node mạng sao cho sai số đầu ra là nhỏ nhất. Bên cạnh đó, nội dung chương cũng đã cập nhật các công trình công bố gần đây nhất trong và ngoài nước về mạng CNN nhằm khẳng định tính hợp lý của việc ứng dụng mạng nơ ron CNN giải quyết hai bài toán nhận dạng chữ viết tay và giải mã captcha. Điều này sẽ được kiểm chứng thông qua việc xây dựng phần mềm mô phỏng trong chương 3 của luận văn.

CHƯƠNG 3

XÂY DỰNG CHƯƠNG TRÌNH MÔ PHỎNG ỨNG DỤNG MẠNG CNN TRONG PHÂN LOẠI ẢNH

3.1 Đặt vấn đề

Trên cơ sở các kiến thức về mạng nơ ron và xử lý ảnh đã được trình bày trong các chương trước, chương 3 của luận văn sẽ đi sâu vào việc xây dựng chương trình mô phỏng nhằm kiểm chứng lý thuyết.



Hình 3.1. Giao diện chính của chương trình mô phỏng

Từ Hình 3.1 có thể thấy hai bài toán mà luận văn tập trung vào bao gồm:

- Ứng dụng mạng CNN cho nhận dạng chữ viết tay: Trong phần này, dựa trên bộ dữ liệu MNIST về chữ viết tay. Luận văn sẽ đề xuất một số kiến trúc mạng CNN, sau đó tiến hành luyện mạng và đánh giá. Các tham số thu được của quá trình luyện mạng đối với kiến trúc mạng CNN có hiệu suất cao nhất sẽ được sử dụng trong phần mềm mô phỏng để minh họa việc nhận dạng. Đồng thời, kiến trúc này cũng sẽ được áp dụng cho bài toán nhận dạng mã Captcha.

- Ứng dụng mạng CNN cho bài toán giải mã CAPTCHA: Trên cơ sở một bộ mẫu Captcha chuẩn, luận văn sẽ tiến hành các thao tác xử lý ảnh để lọc nhiễu, tiền xử lý hình ảnh Captcha, tách ra các chữ cái và xây dựng bộ cơ sở dữ liệu cho quá trình luyện mạng, huấn luyện mạng CNN có cấu trúc tốt nhất ở trên, xây dựng phần mềm mô phỏng từ tham số luyện mạng thu được.

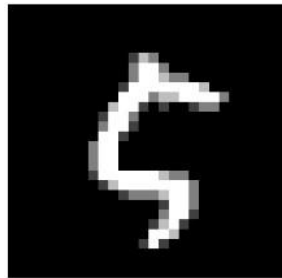
Mỗi bài toán sẽ được trình bày lần lượt theo trình tự thực hiện bao gồm: Mô tả bài toán, cách thức xây dựng kiến trúc mạng, cách thức xây dựng chương trình mô phỏng và đánh giá dựa trên một số kết quả đạt được.

3.2 Bài toán nhận dạng chữ viết tay

3.2.1 Mô tả bài toán

Với khả năng xử lý một lượng lớn đầu vào và xử lý chúng để suy ra các mối quan hệ ẩn và phức tạp, mạng CNN đã đóng một vai trò quan trọng trong xử lý hình ảnh, đặc biệt là nhận dạng ký tự viết tay.

Thách thức chính nảy sinh từ vấn đề nhận dạng chữ số viết tay (Hand Written Digits Recognition - HWDR) nằm ở chỗ các chữ số viết tay (trong cùng một chữ số) khác nhau rất nhiều về hình dạng, độ rộng đường và kiểu, ngay cả khi chúng được chuẩn hóa về kích thước và tập trung chính xác.



Hình 3.2. Chữ viết tay số “5” từ bộ dữ liệu MNIST

Một trong những bộ dữ liệu nổi tiếng được sử dụng trong nghiên cứu về vấn đề HWDR là MNIST (Modified National Institute of Standards and Technology database). MNIST cung cấp hai bộ dữ liệu riêng biệt. Tập dữ liệu đầu tiên chứa 60.000 hình ảnh đào tạo và các chữ số tương ứng của chúng từ 0 đến 9, và tập dữ liệu thứ hai chứa 10.000 hình ảnh thử nghiệm và các chữ số tương ứng của chúng. Mỗi hình ảnh là một ảnh xám 8 bit có kích thước 28 28, Hình 3.2 mô tả một hình ảnh mẫu từ MNIST đại diện cho chữ số 5.

Bộ dữ liệu có sẵn từ trang web của MNIST [18] bao gồm bốn tệp nén, cụ thể là *train-images-idx3-ubyte.gz* cho hình ảnh huấn luyện (9912422 byte); *train-labels-idx1-ubyte.gz* cho nhãn của hình ảnh huấn luyện (28881 byte); *t10k-images-idx3-ubyte.gz* cho hình ảnh kiểm tra (1648877 byte); và *t10k-labels-idx1-ubyte.gz* cho nhãn của hình ảnh kiểm tra (4542 byte).

Phần này của luận văn sẽ thực hiện việc mô phỏng ứng dụng mạng CNN cho bài toán HWDR. Công cụ được sử dụng là MATLAB Deep Learning Toolbox. MATLAB đã tích hợp Neural Network Toolbox for deep learning từ năm 2016. Hộp công cụ này cung cấp một nền tảng hiệu quả để thiết kế và triển khai các mạng nơ-ron học sâu với nhiều tùy chọn cho các thuật toán luyện mạng và mô hình được luyện trước. Hộp công cụ chứa nhiều chức năng hữu ích như *nftool* để điều chỉnh chức năng, *nprtool* để nhận dạng mẫu và *nctool* để phân cụm dữ liệu, v.v. Các chức năng triển khai mạng CNN và mạng bộ nhớ ngắn hạn (LSTM) để phân loại và hồi quy cho hình ảnh, dữ liệu văn bản và chuỗi thời gian cũng có sẵn.

3.2.2 Các bước thực hiện

3.2.2.1 Chuẩn bị dữ liệu

Để xây dựng và đánh giá hoạt động của mạng CNN cho nhận dạng chữ viết tay, luận văn sử dụng MATLAB phiên bản R2019a và cơ sở dữ liệu MNIST. Nền tảng phần cứng cho việc tính toán là PC Windows 10 với CPU Intel 6700k, GPU Nvidia 1080 và RAM 32 GB.

- Nạp dữ liệu

Sử dụng các hàm `loadMNISTImages` và `loadMNISTLabels` tại trang web [18] hoặc [19], bốn bộ dữ liệu có thể được trích xuất từ các tệp cơ sở dữ liệu bằng các lệnh:

```
Tr28 = loadMNISTImages('train-images.idx3-ubyte');
Ltr28 = loadMNISTLabels('train-labels.idx1-ubyte');
Te28 = loadMNISTImages('t10k-images.idx3-ubyte');
Lte28 = loadMNISTLabels('t10k-labels.idx1-ubyte');
```

Trong đó các biến `Tr28` và `Ltr28` là hai ma trận có kích thước lần lượt là 784 x 60000 và 60000 x 1, với mỗi cột của `Tr28` đại diện cho một hình ảnh cho một chữ số

viết tay được định hình thành một vector cột có độ dài 784 và Ltr28 là một cột để biểu thị các nhãn cho các chữ số tương ứng. Để xem các chữ số dưới dạng hình ảnh, cần phải định hình lại các cột của Tr28 trở lại ma trận 28 x 28. Chạy mã bên dưới sẽ hiển thị 100 chữ số đầu tiên từ Tr28:

```
figure
for i = 1:100 subplot(10,10,i)
digit = reshape(Tr28(:,i),[28,28]);
imshow(digit);
title(num2str(labels(i)));
end
```

- Chuyển dữ liệu từ MNIST Database thành file ảnh

Để lưu trữ dữ liệu hình ảnh (như Tr28) vào kho dữ liệu, chúng cần được chuyển đổi trở lại thành hình ảnh (tức là ma trận thay vì vector) và điều này có thể được thực hiện bằng cách sử dụng định hình lại hoặc imwrite. Mã dưới đây tạo một thư mục chính có tên tr và 10 thư mục con riêng biệt trong thư mục chính cho 10 bộ chữ số MNIST theo nhãn của chúng:

```
ltr = Ltr28'; len = length(ltr);
uni_ltr = unique(ltr); cpath = pwd;
for i = 1:length(uni_ltr)
label = num2str(uni_ltr(i)); mkdir(fullfile(cpath,'tr',label));
end
```

Tiếp theo, các mẫu đầu vào được định hình lại thành hình ảnh có kích thước 28 x 28 và sau đó được lưu trữ trong các thư mục con tương ứng ở định dạng .png. Điều này được thực hiện như sau:

```
count = 0; cpath = pwd;
for n = 1:len
count = count+1;
digit = reshape(Tr28(:,n),[28 28]);
label = num2str(ltr(n)); count_str = num2str(count);
fname = fullfile(cpath,'tr',label,[label '_' count_str'.png']);
imwrite(digit,fname); end
```

- Chuyển dữ liệu file ảnh lên kho dữ liệu Matlab

Để tạo kho dữ liệu MATLAB, cần có ba đường dẫn dữ liệu và thiết lập một số thuộc tính bằng các lệnh:

```

cpath = pwd;
tr_path = fullfile(cpath, 'tr',); te_path = fullfile(cpath, 'te',);
ds_path = fullfile(cpath); verbose = true; visualize = false;

```

Sử dụng hàm *imageDatastore*, đoạn mã dưới đây lưu dữ liệu huấn luyện và kiểm tra vào kho dữ liệu tương ứng và được đặt tên lần lượt là *trds* và *teds*:

```

trds = imageDatastore(tr_path, 'IncludeSubfolders',true,...
'FileExtensions','.png','LabelSource','foldernames');
save(fullfile(ds_path, 'trds.mat'), 'trds');

teds = imageDatastore(te_path, 'IncludeSubfolders',true,...
'FileExtensions','.png','LabelSource','foldernames');
save(fullfile(ds_path, 'teds.mat'), 'teds');

```

3.2.2.2 Tạo mạng nơ ron CNN

Hộp công cụ Deep Learning Toolbox cung cấp nhiều tùy chọn để huấn luyện một mạng. Các tùy chọn cho các thuật toán huấn luyện bao gồm giảm độ dốc ngẫu nhiên với động lượng, lan truyền bình phương trung bình gốc và ước lượng mô men thích ứng. Tất cả các thuật toán huấn luyện này đều áp dụng cùng các trọng số ban đầu mặc định, đó là phân phối Gaussian với giá trị trung bình bằng 0 và độ lệch chuẩn là 0,01. Giá trị *bias* ban đầu mặc định được đặt thành 0. Tuy nhiên, nếu cần, các giá trị ban đầu này có thể được đặt lại thủ công thông qua thiết lập mạng.

Mạng CNN trong các mô phỏng của luận văn được huấn luyện sử dụng thuật toán độ dốc giảm ngẫu nhiên với động lượng với trọng số ban đầu và giá trị sai lệch, và tỷ lệ học ban đầu được đặt thành 0,01. Số epoch tối đa là 30 mặc dù với số lượng epoch tăng lên, kết quả ổn định hơn dự kiến. Như có thể thấy từ mã bên dưới, việc đào tạo được thực hiện trong một GPU duy nhất và tiến trình của nó được hiển thị dưới dạng đồ thị.

```

options = trainingOptions('sgdm', ...
'MaxEpochs',epoch,...
'InitialLearnRate',1e-2, ...
'Shuffle','every-epoch',...
'Verbose',false, ...
'Plots','training-progress',...
'ExecutionEnvironment','gpu');

```

Để tính toán độ chính xác nhận dạng, lệnh *classify* được sử dụng như sau:


```

tep = classify(convnet, teds);
tev = teds.Labels;
acc = sum(tep== tev)/numel(tev);
fprintf('accuracy: %2.2f%%, error rate: %2.2f%%\n', acc*100, 100-
acc*100);

```

Trong đó `convnet` là mạng được huấn luyện và `teds` là dữ liệu kiểm tra ở định dạng kho dữ liệu. Hai dòng mã cuối cùng so sánh sự khác biệt giữa dữ liệu thử nghiệm dự đoán với nhãn thực của chúng và hiển thị tỷ lệ lỗi.

Trong quá trình thử nghiệm mô phỏng, luận văn sẽ thực hiện đánh giá kết quả theo hai cấu hình mạng nơ ron. Đó là cấu hình mạng nơ ron cơ bản và cấu hình mạng CNN với ba lớp chập.

- Mạng nơ ron CNN cơ bản

Một CNN cơ bản bao gồm một lớp đầu vào, một lớp chập, một lớp chuẩn hóa, một lớp kích hoạt, một lớp gộp, một lớp được kết nối đầy đủ và một lớp softmax đầu ra để dự đoán nhãn của đầu vào. Mã dưới đây thực hiện một mạng CNN cơ bản như vậy:

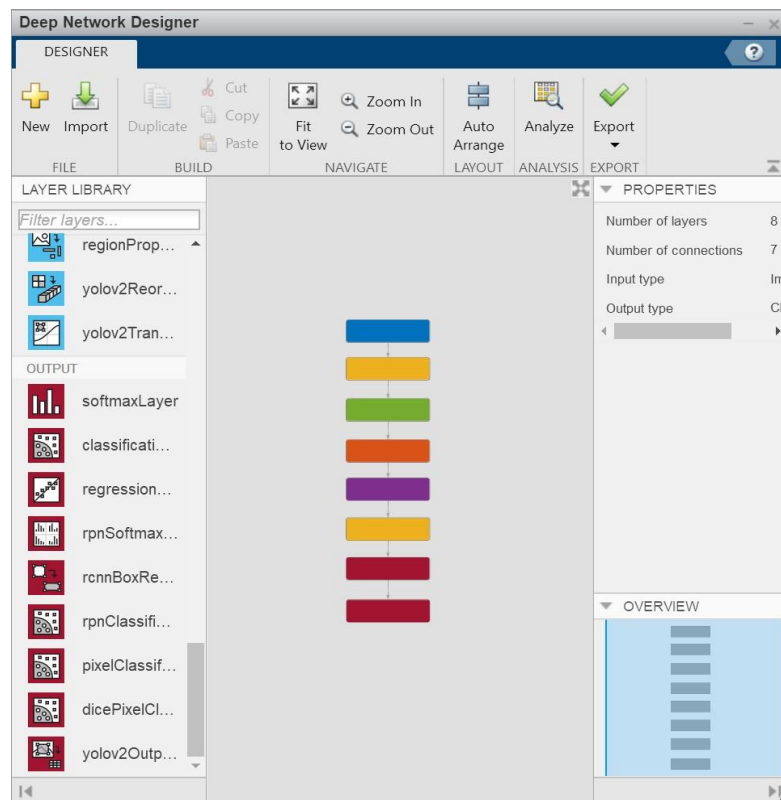
```

imageInputLayer([28 28 1])

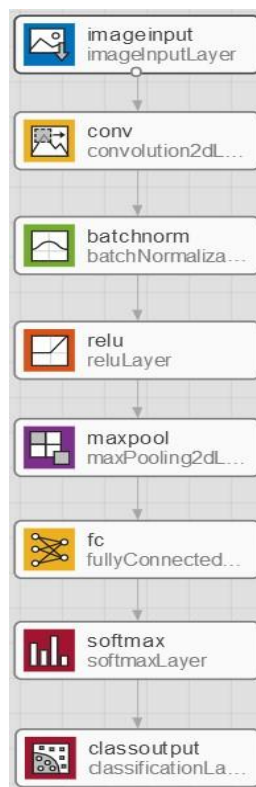
convolution2dLayer(5,3)                %C1
batchNormalizationLayer reluLayer
maxPooling2dLayer(2,'Stride',2)        %S2
fullyConnectedLayer(10)                %F3
softmaxLayer classificationLayer];

```

Để minh họa rõ hơn về các lớp của CNN, luận văn sử dụng thực hiện chức năng *deepNetworkDesigner* trong cửa sổ lệnh tạo ra không gian làm việc như minh họa trong Hình 3.3 trong đó cột bên trái là thư viện lớp, cung cấp các lớp có sẵn. Bên phải là thanh thuộc tính nơi các giá trị tham số có thể được chỉ định. Dưới thanh thuộc tính là tổng quan của mạng. Ta có thể sử dụng ctrl và thanh cuộn để phóng to hoặc thu nhỏ các chi tiết của mạng như trong Hình 3.4. Một khi các lớp được xây dựng, mạng có thể được kiểm tra bằng cách nhấp vào biểu tượng Analyze. Nếu không có lỗi hoặc cảnh báo, mạng đã sẵn sàng để được xuất sang không gian làm việc.



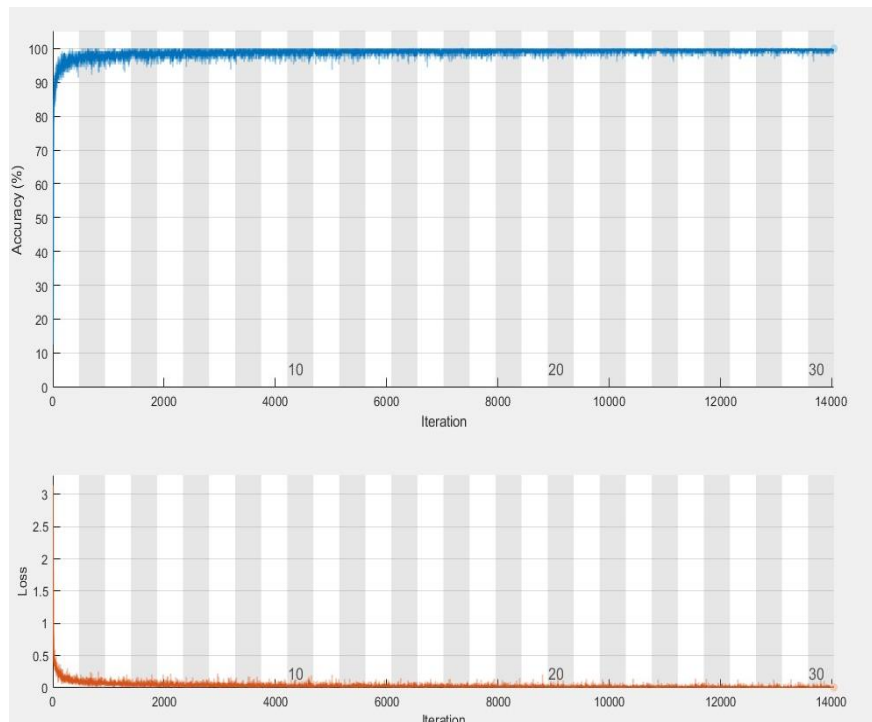
Hình 3.3. Giao diện thiết kế mạng CNN



Hình 3.4. Mạng CNN cơ bản

Về mặt ký hiệu, Ci đại diện cho một lớp chập, Bi đại diện cho một lớp chuẩn hóa hàng loạt, Ai đại diện cho một lớp kích hoạt, Si đại diện cho một lớp mẫu phụ và Fi đại diện cho một lớp được kết nối đầy đủ, trong đó i biểu thị chỉ số lớp. Lớp chuẩn hóa hàng loạt và lớp kích hoạt thường không được coi là lớp CNN, vì vậy khi đếm số lớp, chúng không được tính.

Với mạng CNN cơ bản, lớp đầu vào có hình ảnh có kích thước $28 \times 28 \times 1$, tương ứng với chiều dài, chiều rộng, màu sắc của hình ảnh. Do các chữ số MNIST là ảnh xám nên kích thước của màu được đặt thành 1 trong khi đối với hình ảnh RGB, kích thước của màu sẽ là 3. Trong mạng CNN cơ bản này, chỉ có một lớp chập (C1) được sử dụng, kích thước của cục bộ trường tiếp nhận (giống như của hạt nhân chập) là 5×5 và lớp tạo ra ba bản đồ đặc trưng, mỗi lớp được theo sau bởi một lớp chuẩn hóa hàng loạt (B) để chuẩn hóa đầu ra của lớp chập. Vì hộp công cụ Deep Learning Toolbox không có lớp sigmoid tích hợp, nên lớp ReLU (A) được sử dụng. Lớp tiếp theo là lớp gộp tối đa (S2) để thực hiện lấy mẫu xuống 2×2 và kích thước bước là 2. Các bản đồ đặc trưng suy giảm sau đó được kết nối đầy đủ với lớp 10 nơ-ron (F3), sau đó là lớp softmax và một lớp đầu ra (phân loại).



Hình 3.5. Tiến trình luyện mạng với kernel 7×7 và 8 bản đồ đặc trưng.

Để kiểm tra hiệu suất của mạng, kích thước của trường tiếp nhận cục bộ LRF được đặt thành 3×3 , 5×5 và 7×7 với các kích thước đệm phù hợp, trong mỗi trường hợp, lớp chập tạo ra 3, 6 hoặc 8 bản đồ đặc trưng FM. Hình 3.5 minh họa tiến trình của một trường hợp luyện mạng với kernel 7×7 and 8 bản đồ đặc trưng.

▪ Mạng nơ ron CNN ba lớp ẩn

Để cải thiện mạng CNN cơ bản, luận văn xem xét thêm lớp chập thứ hai (C3) và lớp chập thứ ba (C5) cũng như các lớp gộp tương ứng vào mạng. Kích thước đệm được đặt thành 1 và chiều dài stride được đặt thành 2.

Mã được hiển thị dưới đây thực hiện một CNN với ba lớp chập:

```
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(lrf1, fm1, 'Padding', 1)           %C1
    batchNormalizationLayer      reluLayer
    maxPooling2dLayer(2, 'Stride', s)                   %S2
    convolution2dLayer(lrf2, fm2, 'Padding', 1)         %C3
    batchNormalizationLayer      reluLayer
    maxPooling2dLayer(2, 'Stride', s)                   %S4
    convolution2dLayer(lrf3, fm3, 'Padding', 1)         %C5
    batchNormalizationLayer      reluLayer
    fullyConnectedLayer(10)                             %F6
    softmaxLayer
    classificationLayer];
```

Trong đó *lrf1*, *lrf2*, *lrf3* là trường tiếp nhận cục bộ thứ 1, 2 và 3, *fm1*, *fm2*, *fm3* là bản đồ đặc trưng thứ 1, 2 và 3 và *s* biểu thị độ dài Stride.

3.2.3 Một số kết quả đạt được

Bảng 3.1. Các tham số hoạt động của mạng CNN cơ bản

Kích thước trường tiếp nhận cục bộ	Bản đồ đặc trưng	Độ chính xác (%)	Thời gian thực hiện	
			Thời gian huấn luyện (Phút)	Thời gian kiểm tra (Giây)
3×3	3	97.62	10.21	2.03
3×3	6	98.01	10.79	2.30
3×3	8	98.10	10.83	2.27

5 x 5	3	97.80	10.29	2.25
5 x 5	6	98.25	10.77	2.08
5 x 5	8	98.39	11.10	2.22
7 x 7	3	98.02	10.40	2.14
7 x 7	6	98.58	10.63	1.94
7 x 7	8	98.61	11.08	2.11

Để tìm ra cấu trúc mạng CNN phù hợp cho nhận dạng chữ viết tay MNIST, luận văn kiểm tra ban đầu với cấu trúc mạng cơ bản đã đề xuất ở trên. Trong đó, kích thước trường tiếp nhận cục bộ được lựa chọn thay đổi (3x3, 5x5, 7x7), số lượng bản đồ đặc trưng cũng được kiểm tra qua ba giá trị (3, 6, 8). Kích thước đệm được đặt thành 1 và kích thước spike được đặt thành 1. Số lượng Epoch là 30. Bảng 3.1 tổng hợp hiệu suất của mạng trong các trường hợp, thể hiện ở độ chính xác nhận dạng và thời gian thực hiện.

Từ Bảng 3.1, ta thấy rằng mạng CNN cơ bản cung cấp hiệu suất tốt nhất với độ chính xác dự đoán 98,61% trong 11,08 phút khi LRF được đặt thành 7 x 7, FM được đặt thành 8, kích thước đệm được đặt thành 1 và kích thước stride được đặt thành 1. Số lượng Epoch là 30. Tuy nhiên, trường hợp này cũng cho thời gian luyện mạng và lâu hơn so với các trường hợp khác.

Bảng 3.2. Các tham số hoạt động của mạng CNN ba lớp ẩn

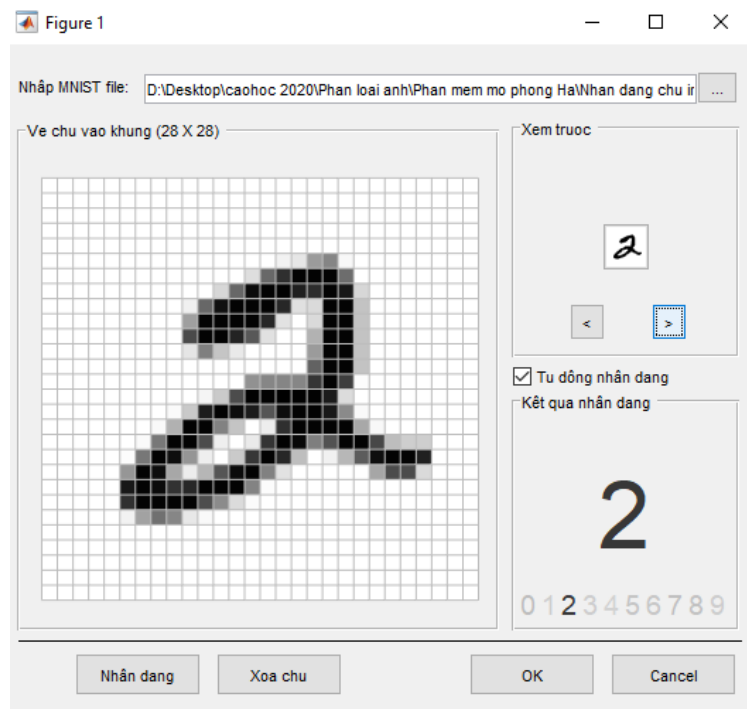
Kích thước trường tiếp nhận cục bộ			Bản đồ đặc trưng			Độ chính xác (%)	Thời gian thực hiện	
1	2	3	1	2	3		Thời gian huấn luyện (Phút)	Thời gian kiểm tra (Giây)
7	5	5	9	9	9	99.14	11.14	1.86
7	5	5	9	9	18	99.13	10.78	2.33
7	5	5	9	9	36	99.12	10.76	1.97
7	5	7	9	9	9	99.27	10.45	1.91

7	5	7	9	9	18	99.15	10.40	2.00
7	5	7	9	9	36	99.32	10.53	2.66
7	5	5	9	18	9	99.26	10.44	1.94
7	5	5	9	18	18	99.28	10.45	2.86
7	5	5	9	18	36	99.35	10.62	2.11
7	5	7	9	18	9	99.16	10.62	1.97
7	5	7	9	18	18	99.29	10.67	1.94
7	5	7	9	18	36	99.32	10.82	2.19
7	7	5	9	9	9	99.19	10.55	1.92
7	7	5	9	9	18	99.18	10.56	2.00
7	7	5	9	9	36	99.32	10.58	2.13
7	7	7	9	9	9	99.29	10.57	2.13
7	7	7	9	9	18	99.22	10.67	1.98
7	7	7	9	9	36	99.34	10.64	2.72
7	7	5	9	18	9	99.20	10.70	2.01
7	7	5	9	18	18	99.29	10.71	2.77
7	7	5	9	18	36	99.27	10.76	2.08
7	7	7	9	18	9	99.27	10.73	1.96
7	7	7	9	18	18	99.37	10.86	2.18
7	7	7	9	18	36	99.43	11.07	2.25

Tiếp theo, luận văn tiếp tục huấn luyện và đánh giá với cấu trúc mạng ba lớp chập. Trong đó, kích thước trường tiếp nhận cục bộ của ba lớp chập này được lựa chọn thay đổi (5x5, 7x7), số lượng bản đồ đặc trưng trong mỗi lớp ẩn cũng được kiểm tra qua ba giá trị (9, 18, 36). Kích thước đệm được đặt thành 1 và kích thước stride được đặt thành 2. Số lượng Epoch là 30 (Bảng 3.2).

Từ Bảng 3.2, có thể thấy rằng CNN đạt được hiệu suất tốt nhất với độ chính xác dự đoán 99,43% trong 11,07 phút khi sử dụng ba lớp chập, mỗi lớp có 7×7 LRF và 9 FM ở C1, 18 FM ở C2 và 36 FM trong C3.

Sau khi xác định được cấu trúc mạng CNN tối ưu, luận văn tiến hành áp dụng tham số thu được sau khi luyện mạng CNN vào xây dựng chương trình mô phỏng. Giao diện của chương trình được mô tả trên Hình 3.5. Chương trình này cho phép có thể minh họa kết quả nhận dạng bằng cách nạp file ảnh dạng MNIST hoặc vẽ ký tự viết tay bất kỳ trên Bảng Icon Edit Pane (28x28). Quá trình hoạt động cho thấy chương trình nhận dạng có độ chính xác cao.



Hình 3.6. Giao diện chương trình nhận dạng chữ viết tay.

Luận văn cũng thực hiện việc đánh giá hoạt động của mạng CNN ba lớp chập thu được với một số phương pháp phân loại ảnh như KNN, SVM (được trình bày trong phần 1.5) trên cùng bộ mẫu MNIST (60000 mẫu cho luyện mạng và 10000 mẫu cho kiểm tra). Các kết quả đạt được của các phương pháp KNN, MLP, SVM được thống kê từ tài liệu [13]. Kết quả cho thấy, mạng CNN với ba lớp chập cho kết quả tốt hơn so với các phương pháp sử dụng mạng lan truyền ngược MLP cũng như phương pháp sử dụng thuật toán KNN. Tuy nhiên, so với mạng CNN được đề xuất trong [13], kết quả về độ chính xác luyện mạng có kém hơn một chút do trong [13], tác giả luyện với 100 epoch. Tuy nhiên, độ chính xác khi kiểm tra 10000 mẫu lại cho kết quả tốt hơn và thời gian luyện mạng nhanh hơn.

Bảng 3.3. So sánh kết quả của một số phương pháp trên bộ dữ liệu MNIST

	Thuật toán phân loại ảnh				
Độ chính xác	MLP	KNN	SVM	CNN [13]	CNN 3 lớp chập
Độ chính xác trên tập mẫu luyện mạng (%)	97.71	99.71	99.71	99.71	99.43
Độ chính xác trên tập mẫu kiểm tra (%)	94.89	96.67	97.91	98.72	99.12
Thời gian luyện mạng (phút)	10	15	14	70	11.07
Thời gian kiểm tra (giây)	6	9	10	20	2.25

3.3 Bài toán giải mã Capcha

3.3.1 Mô tả bài toán

Trong thời đại công nghệ hiện nay đa số người dùng sử dụng máy tính đều quen với việc sử dụng captcha. Captcha có thể được gặp ở bất cứ đâu trên môi trường internet với mục đích chính là phân biệt máy tính với con người để chống lại các hình thức spam. Captcha có nhiều loại: âm thanh, đánh tích, sắp xếp hình ảnh và hình ảnh ký tự bị làm nhiễu [20] .

CAPTCHA là một loại kiểm tra được dùng để xác minh trong máy tính nhằm xác định xem người dùng có phải là một con người thực sự không. CAPTCHA là dãy

ký tự viết tắt các chữ cái đầu tiên của Completely Automated Public Turing test to tell Computers and Humans Apart dịch ra là “Phép thử Turing công cộng hoàn toàn tự động để phân biệt máy tính với người”. CAPTCHA đã từng được trường Đại học Carnegie Mellon cố gắng đăng ký bản quyền nhưng đã bị bác bỏ. CAPTCHA là quá trình một máy chủ yêu cầu người dùng hoàn tất một kiểm tra đơn giản mà máy tính có thể dễ dàng tạo ra những bản thân máy tính không thể giải. Vì vậy, chỉ có người dùng đích thức mới có thể hoàn thành CAPTCHA .

Một hệ thống captcha là một dạng kiểm thử được tạo ra tự động thỏa mãn các điều kiện sau:

- Các máy tính hiện nay không thể giải được một cách chính xác
- Đa số con người có thể giải được
- Người tấn công có thể biết trước các kiểu captcha




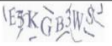

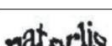
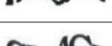
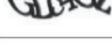



Hình 3.7. Một số mẫu captcha

Hình 3.7 (a) mô tả một mẫu captcha được chương trình EZ-Gimpy tạo ra, đã được Yahoo sử dụng vào những năm 2000. Tuy nhiên, đã có những công nghệ nhận dạng tự động được loại captcha này. Captcha Hình 3.7 (b) làm cho nội dung khó nhận ra hơn bằng cách thêm vào đường gạch ngang và bố trí ký tự không thẳng hàng.

Ngày nay captcha được sử dụng rất phổ biến như một thành phần quan trọng trong quy trình bảo mật an ninh an toàn thông tin. Song hành với điều này các nỗ lực trong việc tự động hóa các cuộc tấn công nhằm vào các website như: quảng cáo quy mô lớn, can thiệp vào các hệ thống bình chọn trực tuyến, tấn công từ chối dịch vụ các website; Tạo ra các liên kết giả để nâng hạng của website trong các máy tìm kiếm; Truy cập thông tin bí mật hoặc lây lan mã độc v.v. Từ đó nhu cầu đánh giá và kiểm định và nâng cấp độ an toàn của mỗi loại captcha trước khi đưa vào sử dụng trong

thực tế là rất cần thiết. Phương pháp cụ thể và phổ biến nhất là nghiên cứu và tìm ra cách tấn công giả định vào các mẫu captcha, hay chính là thực hiện nhận dạng captcha tự động bằng máy không phải con người để phủ định mục đích chính mà captcha được tạo ra. Nhận dạng tự động captcha được chia thành 2 kỹ thuật chính là nhận dạng cứng và nhận dạng mềm.

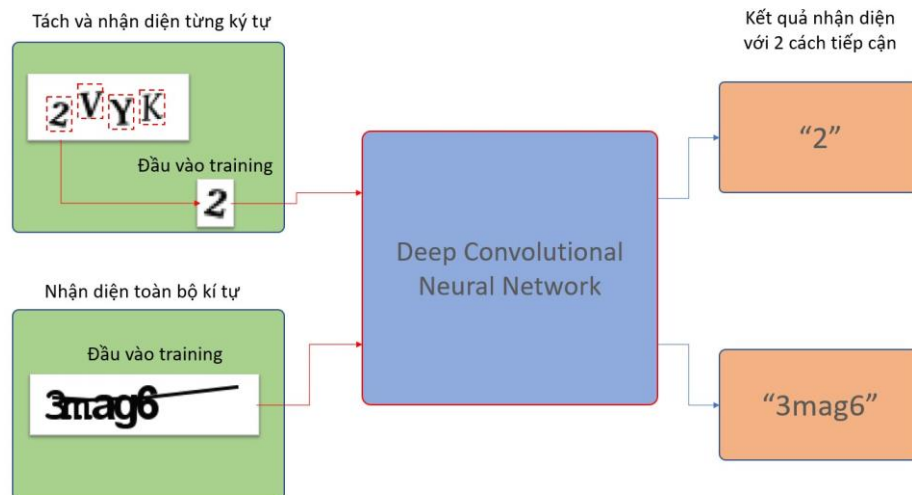
Nhận dạng cứng là một phương pháp nhận dạng tự động mang tính kỹ thuật cao. Phương pháp này tập trung vào các điểm yếu trong quá trình sinh và kiểm tra thông qua bộ giải captcha. Các phương pháp này sử dụng nhiều cách khác nhau để vượt qua bước kiểm tra captcha, trong đó có hai hướng phổ biến nhất là tấn công vào máy khách và tấn công vào máy chủ.

	Ego-share	92.2%	Segmentation: connected region Recognition: SVM	2009
	MSN Yahoo	18% 45%	Segmentation: projection and central	2010
	Megaupload	78%	Segmentation: color filling Combination: nonredundancy Recognition: CNN	2010
	reCAPTCHA-CHA Google	33% 46.75%	Segmentation: character structure feature Recognition: CNN	2011
	Yahoo	54.7%	Segmentation: projection and character feature Recognition: OCR	2012
	Yahoo	36%–89%	Segmentation: color filling Combination: redundancy Recognition: CNN Postprocessing: DFS	2013
	Microsoft	5.56% 57.05%	Different width/location segmenting and template matching	2015
	Microsoft	5%–77.2%	Segmentation: Log-Gabor filter Combination: redundancy Recognition: KNN Postprocessing: DP search	2016
	MSN	27.1%–53.2%	Segmentation: different width Recognition: BPNN	2016

Hình 3.8. Một số kết quả tấn công captcha

Nhận dạng mềm là một phương pháp nhận dạng tự động mang tính học thuật cao. Với mục đích chính là nghiên cứu và xây dựng các phương pháp sử dụng những kiến thức trong các lĩnh vực về trí tuệ nhân tạo cụ thể là thị giác máy tính, học máy thống kê để xây dựng các công cụ tự động nhận dạng và xử lý các loại captcha mà không cần quan tâm tới quy trình sinh và kiểm tra captcha như phương pháp nhận dạng cứng.

Từ những năm 2009 tới nay, việc ứng dụng các mô hình học máy vào nhận dạng tự động captcha ngày một phổ biến và linh hoạt, một số mô hình có thể kể đến bao gồm SVN, KNN, CNN, v.v. Có thể thấy mô hình CNN vẫn được sử dụng phổ biến do tính năng ưu việt và tiềm năng phát triển phong phú của nó đối với bài toán nhận dạng [10] .



Hình 3.9. Hai cách tiếp cận để nhận dạng captcha bằng CNN

Để nhận dạng tự động captcha bằng CNN có 2 hướng tiếp cận để nhận dạng là:

- Tách và nhận dạng từng ký tự trong ảnh captcha: thường sử dụng với các bộ captcha có nhiều đơn giản, các ký tự ít bị dính liền. Thông qua các kỹ thuật tiền xử lý để tách được các ký tự trong captcha ra để nhận dạng.

- Nhận dạng toàn bộ ký tự trong ảnh captcha: là phương pháp khắc phục điểm yếu của phương pháp trên chỉ hiệu quả với captcha có nhiều đơn giản. Phương pháp nhận dạng toàn bộ ký tự có thể áp dụng được với hầu hết các loại captcha từ đơn giản đến phức tạp. Tuy nhiên phương pháp vẫn có những điểm yếu nhất định như thời gian tính toán lâu, thực nghiệm tối ưu hóa tiền xử lý và mô hình CNN với từng kiểu captcha khá phức tạp.

Trong luận văn này, học viên lựa chọn nhận dạng mã Captcha bằng cách tách và nhận dạng từng ký tự trong ảnh Captcha. Nếu các ký tự trong mã captcha được xử lý và đưa về kích thước 28x28 giống như các mẫu MNIST thì hoàn toàn có cơ sở để

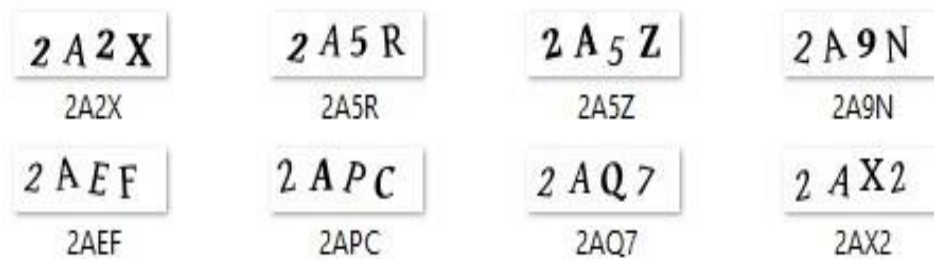
sử dụng cấu trúc mạng CNN 3 lớp ẩn ở trên cho việc luyện mạng.. Vấn đề quan trọng là cần chuẩn bị bộ mẫu dữ liệu đủ lớn.

3.3.2 Các bước thực hiện

3.3.2.1 Chuẩn bị dữ liệu

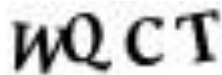
Bộ dữ liệu được chọn để luyện mạng là bộ captcha có kích thước 72×24 chứa 4 kí tự được tạo bởi hỗn hợp 2 bộ kí tự bao gồm (Hình 3.10):

- Các số từ 0 đến 9
- Các chữ cái hoa trong bảng chữ cái tiếng Anh



Hình 3.10. Kiểu dữ liệu captcha dùng trong bài toán nhận dạng

Các chữ tồn tại trên nền trắng sạch sẽ không bị nhiễu, các kí tự không thẳng hàng và chữ cái không đứng thẳng và có sự kết nối nhỏ giữa một số kí tự với nhau mục đích chống lại các phương pháp nhận dạng chữ bằng máy cơ bản như Hình 3.11.



Hình 3.11. Kí tự W và Q bị dính với nhau

Học viên đã xây dựng tập dữ liệu để huấn luyện chứa 10000 mẫu dữ liệu. Và tập kiểm tra chứa 2000 dữ liệu dùng để hiển thị khả năng nhận dạng từng kí tự trong captcha.

3.3.2.2 Tiền xử lý dữ liệu

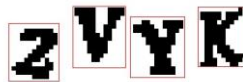
Do bộ dữ liệu captcha khá đơn giản với ít nhiễu nên cách tiếp cận hợp lý nhất là thực hiện tách từng kí tự trong hình ảnh sau đó nhận dạng từng kí tự một và trả ra kết quả captcha. Quá trình thực hiện theo các bước như sau:

Đầu tiên sau khi đọc xong hình ảnh cần chuyển ảnh sang ảnh nhị phân đen trắng bằng phương pháp OTSU và giãn nở ảnh để dễ dàng nhận dạng được các vùng liên thông như Hình 3.12



Hình 3.12. Giãn nở ký tự trong captcha để dễ phát hiện vùng liên thông

Tiếp đó tìm các thành phần liên thông chứa các điểm điểm ảnh cùng màu trong Hình 3.12 và tách ra thành các ảnh riêng biệt để lưu lại làm dữ liệu huấn luyện.



Hình 3.13. Phát hiện thành phần liên thông

Lúc này sẽ có một số trường hợp xảy ra do một số captcha bị làm nhiễu bằng cách có 2 ký tự xếp dính vào nhau ví dụ như mẫu captcha hình 3.10.



Hình 3.14. Một mẫu captcha có 2 ký tự dính liền nhau

Khi xử lý cắt vùng liên thông sẽ có kết quả nhận luôn cả vùng có 2 ký tự như Hình 3.15 :



Hình 3.15. Vùng nhận dạng liên tục nhận 2 ký tự vào 1 ảnh cắt, chưa tốt



Hình 3.16. Kết quả sau khi dùng thủ thuật cắt đôi vùng nhận các ký tự liền nhau

Để xử lý vấn đề này có thể khắc phục được bằng cách sau, do khi 2 ký tự bị liền nhau thì khi nhận khung, chiều dài chắc chắn sẽ tăng lớn hơn chiều cao khung cắt khi đó ta chỉ cần đặt mức điều kiện nếu chiều dài lớn hơn "1.25× chiều cao" thì thực hiện

chia đôi khung thành 2 khung mới để tách kí tự. Khi đó ta sẽ có được kết quả nhận đủ 4 khung ký tự trong ảnh như Hình 3.16.



Hình 3.17. Ví dụ tập các ảnh kí tự đã được cắt và xếp theo thư mục

Sau khi đã tách được từng ký tự ra khỏi ảnh, các ký tự được chuẩn hóa về kích thước 28x28, sau đó dữ liệu được lưu thành các thư mục chứa các mẫu cho các bước luyện và kiểm tra mạng (Hình 3.17).

3.3.2.3 Lựa chọn kiến trúc mạng nơ ron CNN

Về cơ bản, các thao tác xây dựng mạng CNN cho nhận dạng captcha vẫn được thực hiện như đã trình bày trong phần 3.2.2.

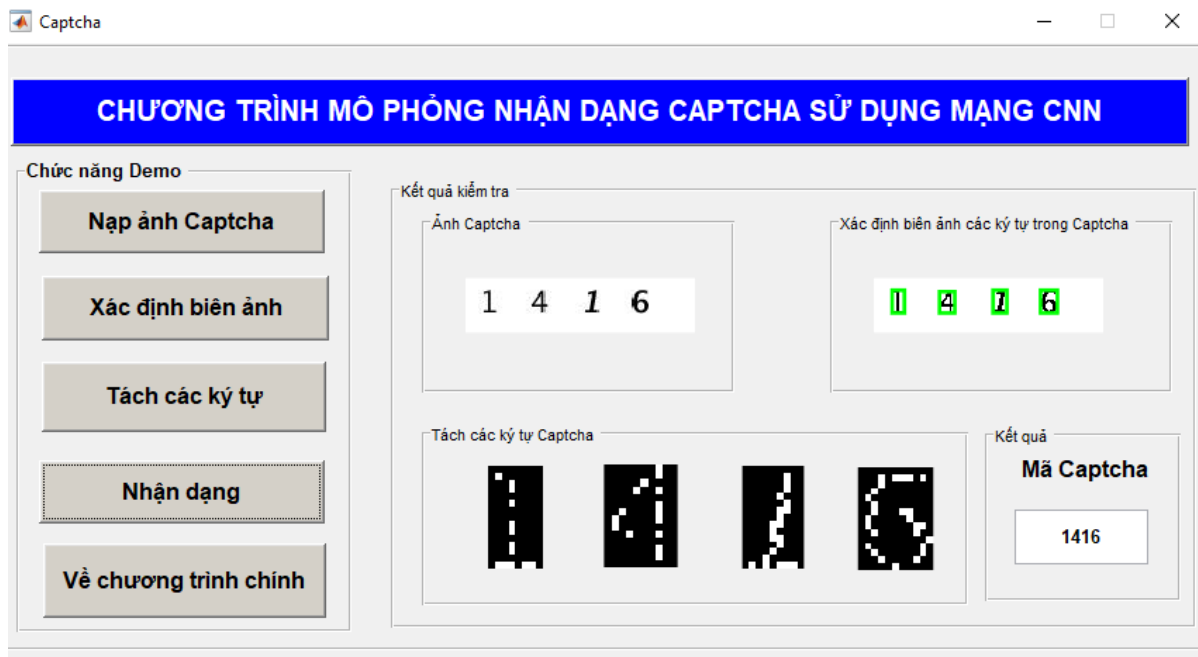
Mã lệnh tạo cấu trúc mạng CNN cho nhận dạng các ký tự được tách ra từ mã Captcha như sau:

```
layers = [
    imageInputLayer([28 28 1])
    convolution2dLayer(lrf1, fm1, 'Padding', 1)           %C1
    batchNormalizationLayer      reluLayer
    maxPooling2dLayer(2, 'Stride', s)                     %S2
    convolution2dLayer(lrf2, fm2, 'Padding', 1)           %C3
    batchNormalizationLayer      reluLayer
    maxPooling2dLayer(2, 'Stride', s)                     %S4
    convolution2dLayer(lrf3, fm3, 'Padding', 1)           %C5
    batchNormalizationLayer      reluLayer
    fullyConnectedLayer(36)                                %F6
    softmaxLayer
    classificationLayer];
```

3.3.3 Một số kết quả đạt được

Khi đưa dữ liệu vào mô hình CNN được xây dựng dựa trên công cụ Deep Learning Toolbox để huấn luyện và kiểm tra. Kết quả thu được rất tốt với hiệu suất nhận dạng đúng toàn bộ các captcha ngẫu nhiên trong tập kiểm tra.

Sau khi có được các tham số của mạng CNN từ quá trình huấn luyện, học viên xây dựng thêm một chức năng mô phỏng việc thực hiện nhận dạng mã captcha. Việc nhận dạng có thể được thực hiện thông qua việc nạp vào các ảnh Capcha nằm trong 12000 mẫu dữ liệu hoặc người dùng tự thêm vào. Kết quả thu được là 99.38 % với các mẫu Captcha thuộc dữ liệu luyện và kiểm tra mạng. Tuy nhiên, với các dữ liệu mới tỷ lệ chính xác giảm hơn. Điều này là do quá trình tách ảnh Captcha thành các ký tự vẫn chưa được tốt.



Hình 3.18. Chương trình mô phỏng nhận dạng mã Captcha

3.4 Kết luận chương

Có thể nói mạng CNN có vai trò rất quan trọng trong việc nâng cao chất lượng phân loại ảnh. So với các công cụ truyền thống trước đây, độ chính xác của phân loại ảnh dùng mạng CNN được cải thiện đáng kể. Tuy nhiên, kết quả chính xác còn phụ thuộc rất nhiều vào các công cụ tiền xử lý ảnh cũng như các tham số trong cấu trúc mạng nơ ron CNN.

Với mục đích thử nghiệm khả năng ứng dụng mạng CNN vào phân loại ảnh, nội dung chương ba đã từng bước xây dựng chương trình nhận dạng chữ viết và nhận dạng Captcha. Đầu tiên, học viên thực hiện tìm ra cấu trúc mạng CNN phù hợp nhất cho nhận dạng chữ viết tay trong bộ mẫu MINST có kích thước 28x28. Theo phương

pháp “thử sai” bằng cách thay đổi số lớp chập, số lượng LRF và FM..học viên đã tìm ra cấu trúc mạng CNN ba lớp chập, mỗi lớp có 7×7 LRF và 9 bản đồ đặc trưng ở lớp chập đầu tiên, 18 bản đồ đặc trưng ở lớp chập thứ 2 và 36 bản đồ đặc trưng ở lớp chập thứ 3. Kích thước đệm bằng 1, độ dài stride bằng 2. Trên cơ sở đã xác định được cấu trúc mạng CNN phù hợp, trong ứng dụng nhận dạng Captcha, sau khi xử lý ảnh và tách ra các ký tự, đưa về dạng chuẩn 28×28 , mạng CNN 3 lớp lại được áp dụng để huấn luyện tạo ra bộ tham số phù hợp cho xây dựng chương trình mô phỏng. Kết quả hoạt động cho thấy tỷ lệ nhận dạng chính xác trên 99 %.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Phân loại ảnh số là một lĩnh vực nghiên cứu hấp dẫn vì có thể áp dụng trong rất nhiều bài toán thực tế. Đây cũng là một bài toán phức tạp nhưng sẽ được giải quyết nếu ta biết ứng dụng các thành tựu nghiên cứu trong các lĩnh vực như xử lý ảnh số, trí tuệ nhân tạo... Trong đó, việc ứng dụng thành quả của Deep learning mà trong đó đặc biệt là mạng CNN cho ta các kết quả thực sự ấn tượng.

Sau một thời gian tìm hiểu nghiên cứu, luận văn đã trình bày được các vấn đề sau:

- Nghiên cứu lý thuyết chung về xử lý ảnh số, tập trung phân tích bài toán phân loại ảnh số, làm rõ các bước trong phân loại ảnh số.
- Nghiên cứu lý thuyết về mạng CNN, cập nhật các ứng dụng mới nhất của mạng CNN trong lĩnh vực phân loại ảnh số.
- Xây dựng chương trình minh họa ứng dụng mạng CNN cho hai bài toán phân loại ảnh tiêu biểu (nhận dạng chữ viết tay và giải mã captcha)

Trong quá trình thử nghiệm chương trình, các kết quả phân loại ảnh số là tương đối tốt (chính xác đến 99%). Tuy nhiên, bài toán giám sát vẫn chỉ dừng lại trong phạm vi nghiên cứu của đề tài là phân lớp từ dữ liệu có sẵn được cộng đồng quốc tế công nhận. Việc lựa chọn cấu trúc mạng CNN vẫn chỉ dựa trên phương pháp “thử sai”. Vì vậy, theo quan điểm của học viên, ***đề tài còn có một số hướng phát triển sau:***

- Nghiên cứu phương pháp tối ưu nhằm xác định cấu trúc, các tham số của mạng CNN cho mỗi ứng dụng cụ thể thay vì phương pháp thử sai.
- Áp dụng các kiến thức về xử lý ảnh nhằm phát triển hai bài toán phân loại ảnh trong luận văn với các dữ liệu đầu vào là ảnh trong thực tế (chữ viết tay trên các bản scan, captcha trên các ảnh chụp từ các giao diện trên trang web có độ khó cao hơn).

Do giới hạn về thời gian nghiên cứu và kiến thức của bản thân, luận văn khó có thể tránh khỏi một số sai sót nhất định. Học viên rất mong nhận được sự đóng góp ý kiến của các thầy cô, các bạn đọc quan tâm để luận văn được hoàn thiện hơn.

Một lần nữa học viên xin được cảm ơn Thầy giáo **TS. Nguyễn Đình Dũng** đã tận tình giúp đỡ, hướng dẫn trong thời gian thực hiện đề tài, cảm ơn sự giúp đỡ của gia đình, bạn bè và các đồng nghiệp trong thời gian qua.

TÀI LIỆU THAM KHẢO

I. Tài liệu tiếng Việt

- [1] Trần Hoài Linh (2015), *Giáo trình mạng neuron và ứng dụng xử lý tín hiệu số*, Nhà XB Bách Khoa.
- [2] Lương Mạnh Bá, Nguyễn Thanh Thủy (2009), *Nhập môn xử lý ảnh số*, Nhà xuất bản Khoa học và Kỹ thuật.
- [3] Nguyễn Văn Danh, Phạm Thế Bảo (2019), *Nhận dạng mặt người bằng học máy chuyên sâu*, Tạp chí giáo dục nghề nghiệp, Vol 65 No ISSN 2354 (2019).
- [4] Nguyễn Đắc Thành (2017), *Nhận dạng và phân loại hoa quả trong ảnh màu*, Luận văn thạc sĩ kỹ thuật phần mềm, Trường ĐH Công nghệ, ĐH Quốc gia Hà nội.
- [5] Lê Thị Thu Hằng (2016), *Nghiên cứu về mạng neural tích chập và ứng dụng cho bài toán nhận dạng biển số xe*, Luận văn thạc sĩ công nghệ thông tin, Trường ĐH Công nghệ, ĐH Quốc gia Hà nội.
- [6] Huỳnh Văn Nhứt (2018), *Nhận dạng chữ số viết tay sử dụng kỹ thuật học sâu*, Luận văn thạc sĩ khoa học máy tính, Trường ĐH Bách Khoa, ĐH Đà Nẵng.
- [7] Đoàn Hồng Quang, Lê Hồng Minh, Thái Doãn Nguyên (2020), *Nhận dạng khuôn mặt trong video bằng mạng neuron tích chập*, Tạp chí Khoa học và Công nghệ Việt Nam, Số 1 năm 2020, pp.8-12.

II. Tài liệu tiếng Anh

- [8] N. Aloysius and M. Geetha (2017), *A review on deep convolutional neural networks*, International Conference on Communication and Signal Processing (ICCSP), Chennai, pp. 0588-0592.
- [9] C. T. S. E. M. Jyh Shing Roger Jang (2002), *Neuro fuzzy and Soft Computing*, Prentice Hall International, Inc.
- [10] Chen, J., Luo, X., Guo, Y., Zhang, Y., Gong, D. (2017), *A Survey on Breaking Technique of Text-Based CAPTCHA*, Hindawi.

- [11] Sharma, Neha & Jain, Vibhor & Mishra, Anju (2018). *An Analysis Of Convolutional Neural Networks For Image Classification*. Procedia Computer Science. 132. 10.1016/j.procs.2018.05.198. pp 377-384
- [12] Alsaffar, Ahmed & Tao, Hai & Talab, Mohammed. (2017). *Review of deep convolution neural network in image classification*. 26-31. 10.1109/ICRAMET.2017.8253139.
- [13] Narender Kumar, Himanshu Beniwal (2018), *Survey on Handwritten Digit Recognition using Machine Learning*, International Journal of Computer Sciences and Engineering, Vol-6, Special Issue-5, June 2018, pp. 96-100.

Các trang Web

- [14] <https://nttuan8.com/bai-3-neural-network/>
- [15] http://en.wikipedia.org/wiki/Image_processing.
- [16] <https://nhdp.net/blog/2018/11/tong-quan-don-gian-ve-mang-no-ron-tich-chap-convolutional-neural-networks/>
- [17] https://en.wikipedia.org/wiki/Convolutional_neural_network
- [18] [MNIST](http://yann.lecun.com/exdb/mnist/), <http://yann.lecun.com/exdb/mnist/>
- [19] <http://udl.stanford.edu/wiki/resources/mnistHelper.zip>
- [20] <https://www.kaggle.com/fournierp/captcha-version-2-images>