

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN HẢI BÌNH

**NGHIÊN CỨU ỨNG DỤNG MẠNG NƠI RƠN NHÂN
TẠO VÀO NHẬN DẠNG MẶT NGƯỜI**

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN HẢI BÌNH

**NGHIÊN CỨU ỨNG DỤNG MẠNG NƠI RƠN NHÂN
TẠO VÀO NHẬN DẠNG MẶT NGƯỜI**

CHUYÊN NGÀNH : HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN ĐÌNH HÓA

HÀ NỘI - 2020

LỜI CAM ĐOAN

Tôi xin cam đoan những vấn đề được trình bày trong luận văn “*Nghiên cứu ứng dụng mạng nơ-ron trong việc nhận dạng mặt người*” là do sự tìm hiểu của cá nhân tôi dưới sự hướng dẫn của **TS. Nguyễn Đình Hóa**.

Tất cả những tham khảo từ các nghiên cứu liên quan đều được trích dẫn, nêu rõ nguồn gốc một cách rõ ràng từ danh mục tài liệu tham khảo trong luận văn. Trong luận văn này, tôi cam đoan không sao chép nguyên bản tài liệu, công trình nghiên cứu của người khác mà không chỉ rõ về tài liệu tham khảo.

Hà Nội, ngày tháng năm 2019

Tác giả luận văn

Nguyễn Hải Bình

LỜI CẢM ƠN

Lời đầu tiên tôi xin gửi lời cảm ơn chân thành nhất đến thầy TS. Nguyễn Đình Hóa đã tận tình chỉ bảo, hướng dẫn tôi trong suốt quá trình thực hiện luận văn.

Tôi xin chân thành cảm ơn các thầy cô giáo đã giảng dạy và giúp đỡ tôi trong suốt thời gian học chương trình cao học. Các thầy cô đã trang bị cho tôi những kiến thức quý báu để làm hành trang cho tôi ứng dụng vào công việc hiện tại cũng như tương lai.

Tôi cũng xin gửi lời cảm ơn chân thành đến các bạn đồng môn, gia đình, bạn bè đã luôn ủng hộ, động viên, giúp đỡ và tạo điều kiện tốt cho tôi vượt qua những khó khăn để hoàn thành luận văn này.

MỤC LỤC

LỜI CẢM ƠN	ii
LỜI CAM ĐOAN	i
DANH MỤC CÁC TỪ VIẾT TẮT	v
DANH MỤC CÁC HÌNH VẼ.....	vi
MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN VỀ MẠNG NORON	3
1.1. Giới thiệu mạng noron nhân tạo	3
1.1.1. Lịch sử phát triển của mạng noron	3
1.1.2. Mạng noron sinh học	4
1.1.3. Mạng noron nhân tạo.....	6
1.1.4. Các hàm kích hoạt (hàm truyền)	9
1.2. Cấu trúc mạng noron.....	11
1.3. Các luật học của mạng noron.....	13
1.3.1. Học có giám sát	13
1.3.2. Học không giám sát.....	14
1.3.3. Học bán giám sát	14
1.4. Phân loại mạng noron	14
1.4.1. Phân loại theo số lớp	14
1.4.2. Phân loại theo sự liên kết giữa các lớp.....	16
1.5. Kết luận.....	17
CHƯƠNG II- MẠNG NƠ-RON VÀ ỨNG DỤNG TRONG NHẬN DẠNG MẶT NGƯỜI.....	18

2.1. Tổng quan về mạng noron tích chập.....	18
2.1.1. Lớp tích chập	20
2.1.2. Lớp hàm kích hoạt	23
2.1.3. Lớp Pooling	24
2.1.4. Lớp kết nối đầy đủ.....	24
2.1.5. Nguyên lý hoạt động	25
2.1.6. Overfitting	26
2.1.7. Phương pháp lựa chọn mô hình.....	28
2.1.8. Dropout.....	30
2.2. Tổng quan về bài toán nhận dạng	30
2.2.1. Ứng dụng của nhận dạng khuôn mặt.....	31
2.2.2. Các hướng tiếp cận bài toán nhận dạng khuôn mặt.....	33
2.2.3. Khó khăn.....	39
2.3. Kết luận.....	40
CHƯƠNG III – CÁC KẾT QUẢ THỰC NGHIỆM	41
3.1. Công cụ lập trình.....	41
3.1.1. Ngôn ngữ lập trình Python	41
3.1.2. Các chương trình thực hiện	42
3.2. Bộ dữ liệu.....	45
3.3. Quá trình thực nghiệm	49
3.4. Kết quả thực nghiệm.....	50
KẾT LUẬN	53
TÀI LIỆU THAM KHẢO.....	54

DANH MỤC CÁC TỪ VIẾT TẮT

ANN	Artificial Neural Network	Mạng nơ ron nhân tạo
CNNs	Convolutional Neural Network	Mạng nơ ron tích chập
EBP	Error back propagation	Lan truyền ngược sai số
KNN	K-Nearest Neighbors	K- láng giềng gần nhất
MLP	Multilayer Perceptron	Mạng Perceptron nhiều lớp
NN	Neural Network	Mạng nơ ron
SSE	Sum square error	Tổng bình phương lỗi
SVM	Support Vector Machines	Bộ phận loại dựa trên véc tơ hỗ trợ

DANH MỤC CÁC HÌNH VẼ

Hình 1.1. Mô hình mạng nơron sinh học	5
Hình 1.2. Mô hình mạng nơron	6
Hình 1.4. Đồ thị hàm đồng nhất.....	9
Hình 4. Đồ thị hàm bước nhị phân với $\theta = 1$	10
Hình 1.5. Đồ thị hàm sigmoid	10
Hình 1.6. Đồ thị hàm sigmoid lưỡng cực.....	10
Hình 1.7. Mạng tự kết hợp.	11
Hình 1.8. Kiến trúc mạng truyền thẳng.....	12
Hình 1.9. Kiến trúc mạng phản hồi.	12
Hình 1.10. Học có giám sát.....	13
Hình 1.11. Mô hình mạng nơron 1 lớp.....	15
Hình 1.12 . Mô hình mạng nơron nhiều lớp.....	16
Hình 1. Mô hình mạng hồi quy	17
Hình 2.1. Mô hình CNN.....	19
Hình 2.2. Minh họa tích chập trên ma trận ảnh.....	21
Hình 2.3. Tìm biên cạnh cho ảnh với tích chập	22
Hình 2.4. Làm mờ ảnh gốc với tích chập.....	22
Hình 2.4. Minh họa tích chập 3 chiều	23
Hình 2.5. Max Pooling với lọc 2 x 2.....	24
Hình 2.7. Mô hình chi tiết cấu trúc 1 CNN.....	26
Hình 2.8. Overfitting	27

Hình 2.9. Ví dụ về ảnh một phần khuôn mặt: (a) Ảnh một phần khuôn mặt trong bộ dữ liệu LFW; (b) Ảnh một phần khuôn mặt trong đám đông; (c) Ảnh khuôn mặt bị che bởi mắt kính, áo khoác.....	37
Hình 2.10. Sơ đồ thuật toán Bag of Word.....	38

MỞ ĐẦU

Từ khi máy tính điện tử ra đời, đặc biệt là khi khả năng xử lý tính toán vượt trội của nó được ứng dụng vào các hệ thống trợ giúp con người, con người không ngừng mong muốn có thể tạo ra những cỗ máy có khả năng xử lý và quan sát phân tích hình ảnh giống với hệ thống thị giác con người. Nhận dạng khuôn mặt là một trong những lĩnh vực mới của xử lý ảnh. Đặc biệt sau thảm họa ngày 11/9, nhận dạng được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống như nhận dạng trong lĩnh vực thương mại, hay phát hiện tội phạm trong lĩnh vực an ninh, hay trong lĩnh vực xử lý video, hình ảnh.

Hiện nay có rất nhiều các phương pháp nhận dạng khác nhau được xây dựng để nhận dạng một người cụ thể trong thế giới thực. Tuy nhiên việc nhận dạng được một người trong thế giới thực là vô cùng khó khăn, bởi vì để nhận dạng được ta phải xây dựng được tập cơ sở dữ liệu đủ lớn và việc xử lý dữ liệu lớn này đòi hỏi phải nhanh và chính xác. Nhiệm vụ đặt ra là nghiên cứu và xây dựng một chương trình sử dụng phương pháp nhận dạng có độ chính xác cao mà khối lượng và thời gian tính toán lại ít.

Năm 1998, Daniel Bgraham và Nigel M Allinson đã sử dụng phương pháp được gọi là tạo bản sao không gian đặc trưng để biểu diễn và nhận dạng hướng di chuyển của khuôn mặt.

Năm 2001, Guodong Guo, Stan Z.Li, Kap Luk Chan đã dùng phương pháp SVM để nhận dạng khuôn mặt. Sử dụng chiến lược kết hợp nhiều bộ phân loại nhị phân để xây dựng bộ phân loại SVM đa lớp

Gần đây nhất năm 2017, tại cuộc thi “Sáng tạo trẻ Bách khoa 2017”. Sản phẩm đạt giải Nhất cuộc có tên là “Hệ thống nhận dạng khuôn mặt BKFace”. Hệ thống của 5 sinh viên đến từ 3 trường đại học có khả năng giải quyết 3 vấn đề cốt lõi: ngoài phát hiện khuôn mặt còn có nhận diện và xác thực khuôn mặt. Ba tính năng này giúp Bkface có thể ứng dụng vào nhiều lĩnh vực phức tạp của đời sống: như an ninh, điều tra, truy bắt tội phạm.

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình của mạng nơ-ron nhân tạo tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao. Mạng nơ-ron tích chập được áp dụng khá nhiều trong các bài toán nhận dạng như nhận dạng vật thể trong ảnh, nhận dạng chữ viết tay (chuyển đổi chữ viết trong hình ảnh thành văn bản thô trong máy tính), nhận dạng vật thể 3D, xử lý tiếng nói, xử lý ngôn ngữ tự nhiên, nhận dạng mặt người qua ảnh, video.... Với độ chính xác cao. Trong luận văn cao học này, tác giả đi vào nghiên cứu về mạng nơ-ron, mô hình mạng nơ-ron CNN trong phân lớp ảnh và ứng dụng của mô hình này trong nhận dạng khuôn mặt với đề tài: *“Nghiên cứu ứng dụng mạng nơ-ron trong việc nhận dạng mặt người”*.

Cấu trúc của luận văn được thể hiện qua ba chương chính, với các nội dung tóm lược như sau:

Chương 1: Trình bày tổng quan về mạng nơ-ron nhân tạo bao gồm cấu trúc cơ bản và một số ứng dụng trong thực tế của mạng nơ-ron nhân tạo.

Chương 2: Trình bày chi tiết về mạng nơ-ron tích chập, ý nghĩa cùng ứng dụng của mạng nơ-ron tích chập trên thực tế, cụ thể là trong xử lý ảnh nói chung và nhận dạng mặt người nói riêng.

Chương 3: Trình bày các nội dung về việc ứng dụng mạng nơ-ron tích chập trong nhận dạng khuôn mặt người. Trong chương này giới thiệu cụ thể về bộ dữ liệu được sử dụng trong nghiên cứu của luận văn cũng như các kết quả nhận dạng khuôn mặt sử dụng mạng nơ-ron tích chập.

Luận văn được kết thúc bằng phần kết luận và mục tài liệu tham khảo.

CHƯƠNG 1. TỔNG QUAN VỀ MẠNG NORON

1.1. Giới thiệu mạng noron nhân tạo

1.1.1. Lịch sử phát triển của mạng noron

Mạng noron nhân tạo đã có một lịch sử phát triển lâu dài. Năm 1940 với công trình của Warren McCulloch và Walter Pitts [6]. Họ chỉ ra rằng về nguyên tắc, mạng của các noron nhân tạo có thể tính toán bất kỳ một hàm số học hay logic nào.

Tiếp theo hai người là Donald Hebb, ông đã phát biểu rằng việc thuyết lập luận cổ điển (classical conditioning) là hiện thực bởi do các thuộc tính của từng noron riêng biệt. Ông cũng nêu ra một phương pháp học của các noron nhân tạo. Ứng dụng thực nghiệm đầu tiên của các noron nhân tạo có được vào cuối những năm 50 cùng với phát minh của mạng nhận thức (perceptron network) và luật học tương ứng bởi Frank Rosenblatt. Mạng này có khả năng nhận dạng các mẫu. Điều này đã mở ra rất nhiều hy vọng cho việc nghiên cứu mạng noron. Tuy nhiên nó có hạn chế là chỉ có thể giải quyết một số lớp hữu hạn các bài toán.

Cùng thời gian đó, Bernard Widrow và Ted Hoff đã đưa ra một thuật toán học mới và sử dụng nó để huấn luyện cho các mạng noron tuyến tính thích nghi, mạng có cấu trúc và chức năng tương tự như mạng của Rosenblatt. Luật học Widrow-Hoff vẫn còn được sử dụng cho đến nay.

Tuy nhiên cả Rosenblatt và Widrow-Hoff đều cùng gặp phải một vấn đề, do Marvin Minsky và Seymour Papert đã phát hiện ra, đó là các mạng nhận thức chỉ có khả năng giải quyết các bài toán khả phân tuyến tính. Họ cố gắng cải tiến luật học và mạng để có thể vượt qua được hạn chế này nhưng họ đã không thành công trong việc cải tiến luật học để có thể huấn luyện được các mạng có cấu trúc phức tạp hơn.

Do những kết quả của Minsky-Papert nên việc nghiên cứu về mạng noron gần như bị đình lại trong suốt một thập kỷ do nguyên nhân là không có được các máy tính đủ mạnh để có thể thực hiện.

Mặc dù vậy, cũng có một vài phát kiến quan trọng vào những năm 70. Năm 1972, Teuvo Kohonen và James Anderson độc lập nhau phát triển một loại mạng mới có thể hoạt động như một bộ nhớ. Stephen Grossberg cũng rất tích cực trong việc khảo sát các mạng tự tổ chức (Self organizing networks).

Vào những năm 80, việc nghiên cứu mạng noron phát triển rất mạnh mẽ cùng với sự ra đời của PC. Có hai khái niệm mới liên quan đến sự hồi sinh này, đó là:

- Việc sử dụng các phương pháp thống kê để giải thích hoạt động của một lớp các mạng hồi quy (recurrent networks) có thể được dùng như bộ nhớ liên hợp (associative memory) trong công trình của nhà vật lý học John Hopfield.

- Sự ra đời của thuật toán lan truyền ngược (back-propagation) để luyện các mạng nhiều lớp được một vài nhà nghiên cứu độc lập tìm ra như: David Rumelhart, James McClelland,.... Đó cũng là câu trả lời cho Minsky-Papert.

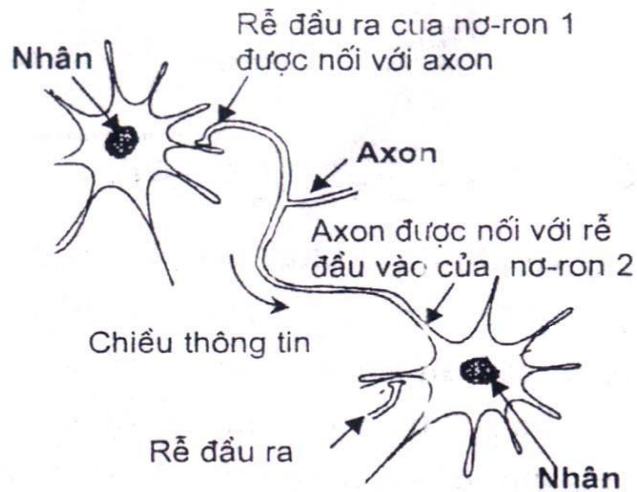
1.1.2. Mạng noron sinh học

Não người là tổ chức vật chất cấp cao, có cấu tạo vô cùng phức tạp, dày đặc các mối liên kết giữa các noron nhưng xử lý thông tin rất linh hoạt trong môi trường bất định.

Trong bộ não người có khoảng $10^{11} - 10^{12}$ tế bào thần kinh được gọi là các noron và mỗi noron có thể liên kết với 104 noron khác thông qua các khớp nối thần kinh (synapse). Dưới con mắt của những người làm tin học cấu tạo của mỗi noron gồm các thành phần cơ bản sau:

- Thân noron được giới hạn trong một màng membran và trong cùng là nhân. Nó là nơi tiếp nhận, tổng hợp và phát ra các xung thần kinh hay các thông tin sau khi đã được tổng hợp. Từ thân noron còn có rất nhiều đường rẽ nhánh tạm gọi là rẽ.

- “Bus” liên kết noron này với các noron khác được gọi là axon, trên axon có các đường rẽ nhánh. Noron còn có thể liên kết với các noron khác qua các rẽ. Chính vì cách liên kết đa dạng như vậy nên mạng noron có độ liên kết rất cao.



Hình 1.1. Mô hình mạng nơron sinh học

Các rễ của nơron được chia làm hai loại: rễ đầu vào nhận thông tin từ các nơron khác qua axon và rễ đầu ra đưa thông tin qua axon tới các nơron khác. Một nơron có thể có nhiều rễ đầu vào, nhưng chỉ có một rễ đầu ra. Bởi vậy nếu coi nơron như một khâu điều khiển thì nó chính là khâu có nhiều đầu vào và một đầu ra.

Quá trình hoạt động của một nơron là một quá trình điện hoá tự nhiên. Khi có tác động từ bên ngoài vào nơron, tức ở đầu vào của nơron xuất hiện một tín hiệu tác động vượt quá ngưỡng cân bằng của nó thì nơron sẽ ở trạng thái kích thích. Trong tế bào nơron xảy ra hàng loạt các phản ứng hoá học tạo thành lực tác động làm nơron bị kích thích hoàn toàn. Thế năng sinh ra khi nơron ở trạng thái bị kích thích hoàn toàn này chỉ tồn tại vài mili giây sau đó nơron lại trở về trạng thái cân bằng cũ. Thế năng này được chuyển vào mạng qua axon và có khả năng kích thích hoặc kìm hãm tự nhiên các nơron khác trong mạng.

Một tính chất rất cơ bản của mạng nơron sinh học là các đáp ứng theo kích thích có khả năng thay đổi theo thời gian. Các đáp ứng có thể tăng lên, giảm đi hoặc hoàn toàn biến mất. Qua các nhánh axon liên kết các tế bào nơron với nhau, sự thay đổi trạng thái của một nơron dẫn theo sự thay đổi trạng thái của những nơron khác và dẫn tới sự thay đổi của toàn bộ mạng nơron. Việc thay đổi trạng thái của mạng nơron có thể thực hiện qua một quá trình “dạy” hoặc do khả năng “học” tự nhiên.

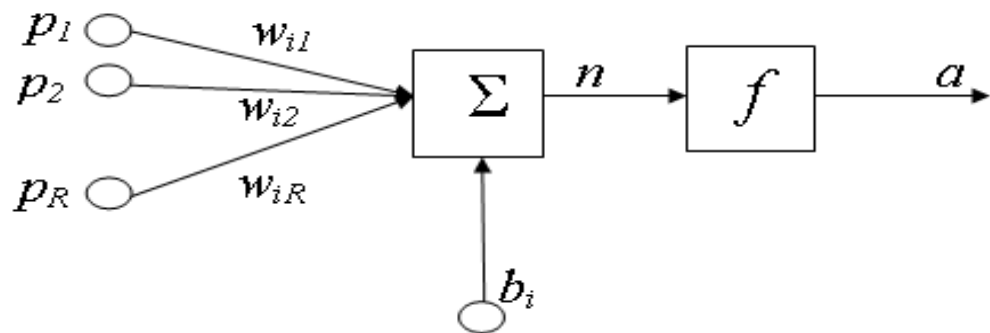
Hoạt động tư duy của con người phần nhiều mang tính chủ quan định tính, dựa trên những bài học, kinh nghiệm học được từ khi còn nhỏ. Càng lớn dần thì những bài học hay kinh nghiệm của con người ngày càng nhiều, giúp con người giải quyết các vấn đề trong tự nhiên, xã hội tốt hơn. Bộ não có khả năng tự thay đổi cấu trúc để thích nghi dần với môi trường, làm cho cấu trúc bộ não ngày càng trở nên phức tạp sau mỗi lần học. Cấu trúc của mạng nơron sinh học cũng luôn luôn được phát triển và thay đổi. Một số cấu trúc của nơron được xác định trước, một số sau này mới được hình thành và một số thì bị huỷ bỏ qua quá trình chọn lọc tự nhiên, học và thích nghi.

Qua quá trình tìm hiểu, nghiên cứu cấu trúc và hoạt động của bộ não người, các nhà khoa học đã và đang xây dựng và phát triển các mô hình xử lý thông tin mô phỏng hoạt động của bộ não người. Đó chính là mô hình mạng nơron nhân tạo.

1.1.3. Mạng nơron nhân tạo

Mạng nơron nhân tạo bao gồm nhiều nơron độc lập liên kết với nhau. Trước khi định nghĩa thế nào là mạng nơron nhân tạo, chúng ta sẽ tìm hiểu mô hình của một nơron nhân tạo.

Một nơron nhân tạo phản ánh các tính chất cơ bản của nơron sinh học. Mỗi nơron nhân tạo là một đơn vị xử lý thông tin làm cơ sở cho hoạt động của một mạng nơron. Nó có chức năng nhận tín hiệu vào, tổng hợp và xử lý các tín hiệu vào để tính tín hiệu ra. Dưới đây là một mô hình của một nơron nhân tạo:



Hình 1.2. Mô hình mạng nơron

Với mục đích tạo ra một mô hình tính toán phỏng theo cách làm việc của noron trong bộ não con người, vào năm 1943, các tác giả McCulloch và Pitts [6] đã đề xuất một mô hình toán cho một noron như hình 1.2.

Trong mô hình này, một noron thứ i sẽ nhận các tín hiệu vào p_j với các trọng số tương ứng là w_{ij} , tổng các thông tin vào có trọng số là: $\sum_{j=1}^n w_{ij} p_j + b_i$.

Thông tin đầu ra ở thời điểm $t+1$ được tính từ các thông tin đầu vào như sau:

$$a(t+1) = f\left(\sum_{j=1}^n w_{ij} p_j + b_i\right) \quad (1)$$

Trong đó f là hàm truyền có dạng là hàm bước nhảy, nó đóng vai trò biến đổi từ thông tin đầu vào thành tín hiệu đầu ra:

$$f(x) = \begin{cases} 0 & \text{nếu } x \geq 0 \\ 1 & \text{nếu } x < 0 \end{cases} \quad (2)$$

Như vậy, $a = 1$ (ứng với việc noron tạo tín hiệu đầu ra) khi tổng các tín hiệu vào lớn hơn độ lệch b_i , còn $a = 0$ (noron không tạo tín hiệu ở đầu ra) khi tổng các tín hiệu vào nhỏ hơn độ lệch b_i .

Trong mô hình noron của McCulloch và Pitts, các trọng số w_{ij} thể hiện ảnh hưởng của khớp nối trong liên kết giữa noron j (noron gửi tín hiệu) và noron i (noron nhận tín hiệu). Trọng số w_{ij} dương ứng với khớp nối kích thích, trọng số âm ứng với khớp nối ức chế còn w_{ij} bằng 0 khi không có liên kết giữa hai noron. Hàm truyền f ngoài dạng hàm bước nhảy còn có thể chọn nhiều dạng khác nhau và sẽ được đề cập ở các phần sau.

Thông qua cách mô hình hoá đơn giản một noron sinh học như trên, McCulloch và Pitts đã đưa ra một mô hình noron nhân tạo có tiềm năng tính toán quan trọng. Nó có thể thực hiện các phép toán logic cơ bản như AND, OR và NOT khi các trọng số và ngưỡng được chọn phù hợp. Sự liên kết giữa các noron nhân tạo với các cách thức khác nhau sẽ tạo nên các loại mạng noron nhân tạo với những tính

chất và khả năng làm việc khác nhau. Một noron nhân tạo được thể hiện thông qua các thành phần cơ bản sau:

- Tập các đầu vào: là tín hiệu vào của mạng noron, các tín hiệu này thường được đưa vào dưới dạng một véc tơ R thành phần, hoặc tín hiệu ra của các noron thuộc lớp đằng trước.

- Trọng số liên kết: là trọng số liên kết giữa tín hiệu vào thứ j cho noron i . Thông thường các trọng số này được khởi tạo ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học.

- Bộ tổng (hàm tổng): thường dùng để tính tổng của các đầu vào đã được nhân với các trọng số liên kết của nó.

- Độ lệch (bias): thường được đưa vào như một thành phần của hàm truyền.

- Hàm kích hoạt (hàm truyền): hàm này dùng để xác định và giới hạn giá trị đầu ra của mỗi noron. Nó nhận đầu vào là kết quả của hàm tổng kết hợp với độ lệch đã cho. Các hàm truyền rất đa dạng có thể là tuyến tính hoặc phi tuyến.

- Đầu ra: là tín hiệu đầu ra của một noron, với mỗi noron sẽ có tối đa một giá trị đầu ra.

Về mặt toán học cấu trúc của một noron i được biểu diễn bằng biểu thức sau:

$$a_i = f\left(\sum_{j=1}^n w_{ij} p_j + b_i\right) \quad (3)$$

Trong đó: p_j : là tín hiệu đầu vào thứ j

w_{ij} : là trọng số liên kết của noron i cho đầu vào j

f : là hàm kích hoạt (hàm truyền)

b_i : là độ lệch

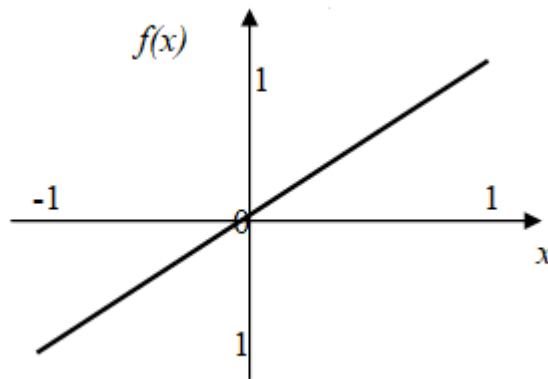
a_i : đầu ra

Như vậy cũng tương tự như noron sinh học, noron nhân tạo cũng nhận các tín hiệu đầu vào, xử lý (nhận các tín hiệu này với trọng số liên kết, tính tổng các tích thu được kết hợp với độ lệch rồi gửi kết quả tới hàm truyền), và cho một tín hiệu đầu ra là kết quả của hàm kích hoạt.

1.1.4. Các hàm kích hoạt (hàm truyền)

a. Hàm đồng nhất (Linear function, Identity function): $f(x) = x$

Nếu coi các đầu vào là một đơn vị thì chúng sẽ sử dụng hàm này. Đôi khi một hằng số được nhân với net-input để tạo ra một hàm đồng nhất.

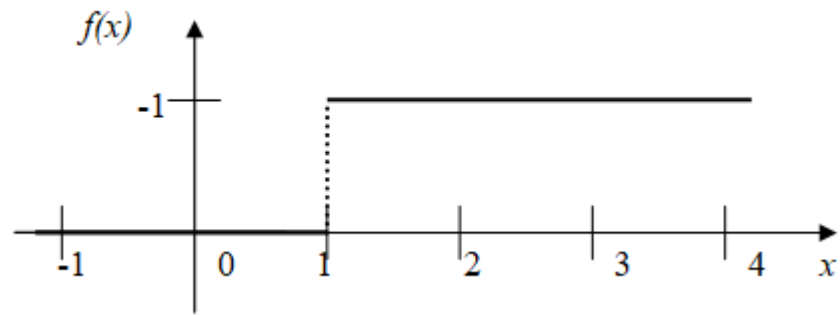


Hình 1.4. Đồ thị hàm đồng nhất

b. Hàm nhị phân (Binary step function, Hard limit function)

Hàm này còn được gọi là *hàm ngưỡng* (Threshold function hay Heaviside function):

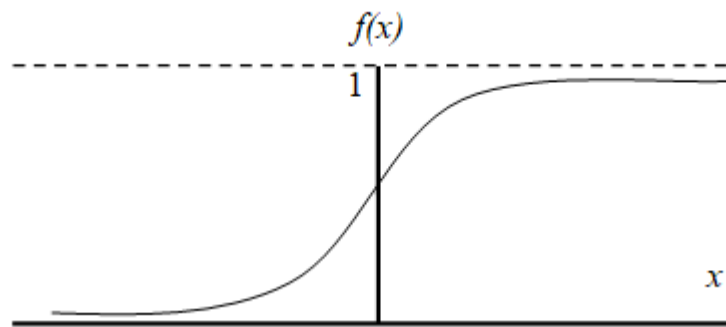
$$f(x) = \begin{cases} 1, & \text{nếu } x \geq \theta \\ 0, & \text{nếu } x < \theta \end{cases} \quad (4)$$



Hình 1.4 a. Đồ thị hàm bước nhị phân với $\theta = 1$

c. Hàm sigmoid (Sigmoid function (logistic))

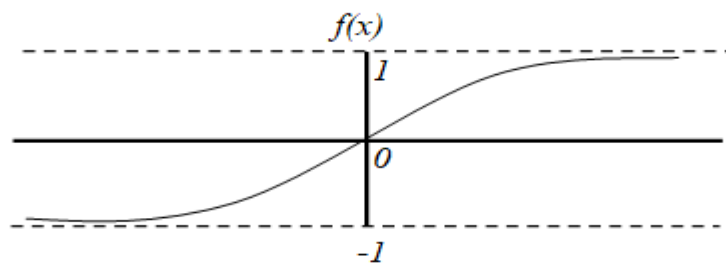
$$f(x) = \frac{1}{1 + e^{-x}} \quad (5)$$



Hình 1.5. Đồ thị hàm sigmoid

d. Hàm sigmoid lưỡng cực (Bipolar sigmoid function (tansig))

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (6)$$



Hình 1.6. Đồ thị hàm Hàm sigmoid lưỡng cực

1.2. Cấu trúc mạng nơron

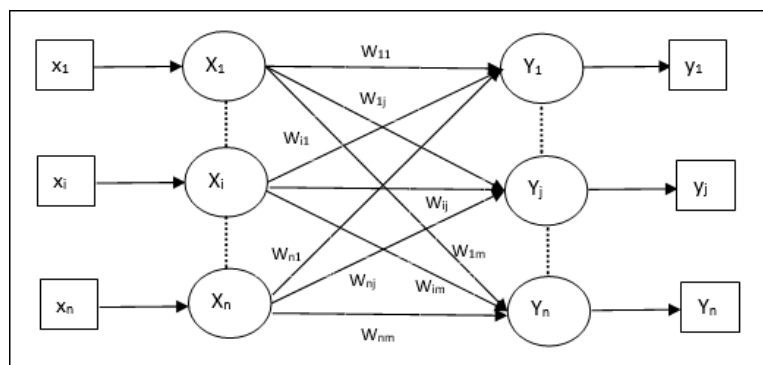
Mặc dù hiểu biết của con người về kiến trúc và hoạt động của não còn chưa đầy đủ, nhưng chúng ta có thể tạo ra được các máy có một số tính năng tương tự như bộ não. Mạng nơron nhân tạo là một công cụ mô phỏng cách bộ não hoạt động nhằm thực hiện các nhiệm vụ nào đó. Một mạng nơron là bộ xử lý song song phân tán lớn, nó giống bộ não người về 2 mặt:

- Tri thức về dữ liệu được nắm bắt bởi các nơron thông qua quá trình học.
- Các trọng số kết nối nơron đóng vai trò khớp nối cất giữ tri thức đã được học của mạng nơron.

Trong phần trước, kiến trúc và cấu tạo của một nơron đơn lẻ đã được giới thiệu sơ qua. Các nơron này sẽ kết nối với nhau theo một quy tắc nào đó để tạo thành một mạng nơron. Tính năng của hệ thống này tùy thuộc vào cấu trúc của hệ, các trọng số liên kết nơron và quá trình tính toán tại các nơron đơn lẻ. Mạng nơron có thể được huấn luyện bằng các dữ liệu mẫu, và có khả năng tổng quát hoá các mô hình dữ liệu dựa trên các mẫu cho trước.

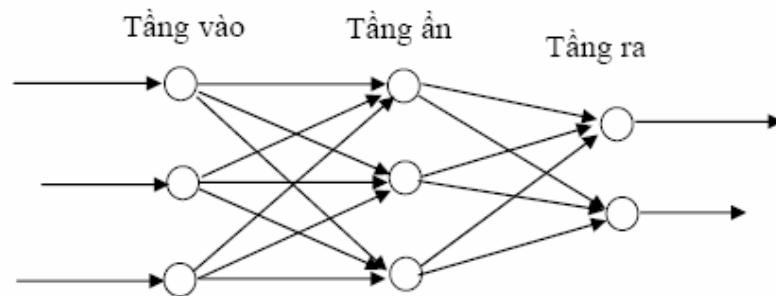
Các loại mạng nơron nhân tạo được xác định bởi cách liên kết giữa các nơron, trọng số của các liên kết đó và hàm truyền tại mỗi nơron. Dưới đây là một số kiến trúc mạng nơron điển hình.

Mạng tự kết hợp - autoassociative: là mạng có các nơron đầu vào cũng là các nơron đầu ra (mạng Hopfield).



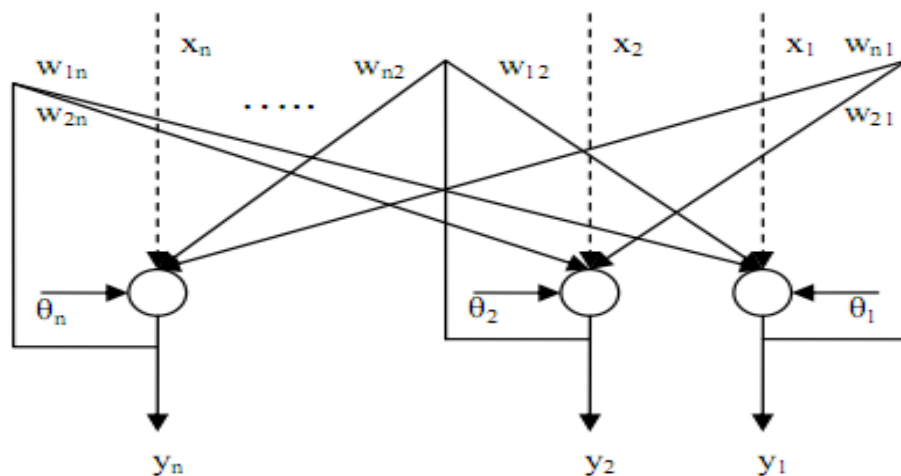
Hình 1.7. Mạng tự kết hợp.

Kiến trúc truyền thẳng - feedforward architecture: là kiểu kiến trúc mạng không có các kết nối ngược trở lại từ các nơ ron đầu ra về các nơ ron đầu vào; mạng không lưu lại các giá trị output trước và các trạng thái kích hoạt của nơ ron. Các mạng nơ ron truyền thẳng cho phép tín hiệu di chuyển theo một đường duy nhất; từ đầu vào tới đầu ra, đầu ra của một tầng bất kì sẽ không ảnh hưởng tới tầng đó. Các mạng kiểu Perceptron là mạng truyền thẳng.



Hình 1.8. Kiến trúc mạng truyền thẳng.

Kiến trúc phản hồi - Feedback architecture: là kiểu kiến trúc mạng có các kết nối từ nơ ron đầu ra tới nơ ron đầu vào. Mạng lưu lại các trạng thái trước đó, và trạng thái tiếp theo không chỉ phụ thuộc vào các tín hiệu đầu vào mà còn phụ thuộc vào các trạng thái trước đó của mạng. Mạng feedback có chu trình khép kín gọi là mạng quy hồi (*recurrent network*).



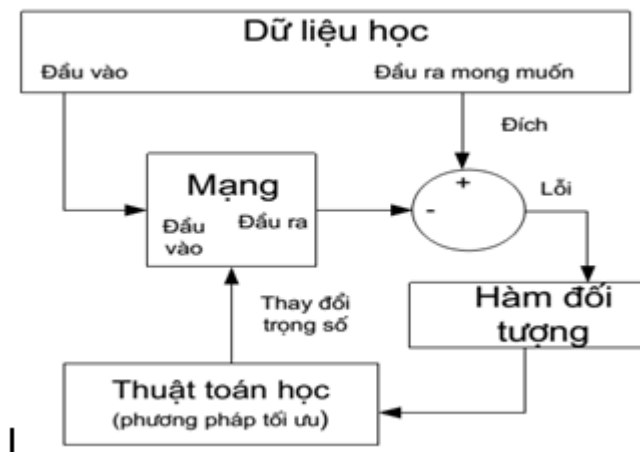
Hình 1.9. Kiến trúc mạng phản hồi.

1.3. Các luật học của mạng noron

1.3.1. Học có giám sát

Mạng được huấn luyện bằng cách cung cấp cho nó các cặp mẫu đầu vào và các đầu ra mong muốn. Mục đích là xây dựng mạng để ứng với mỗi đầu vào trong tập huấn luyện thì kết quả đầu ra của mạng cho đúng bằng đầu ra mong muốn, để làm được điều đó phải điều chỉnh dần mạng do tồn tại sự khác biệt giữa đầu ra thực tế và đầu ra mong muốn. Sự khác biệt này được thuật toán học sử dụng để điều chỉnh các trọng số trong mạng. Việc điều chỉnh các trọng số như vậy thường được mô tả như một bài toán xấp xỉ số - cho dữ liệu huấn luyện bao gồm các cặp (mẫu đầu vào x , và một đích tương ứng t), mục đích là tìm hàm $f(x)$ thỏa mãn tất cả các mẫu huấn luyện đầu vào. Học có giám sát bao gồm:

- Phân loại (classification): là dạng học có giám sát với hàm đích nhận giá trị rời rạc.
- Hồi quy (regression): là học có giám sát với giá trị hàm đích nhận giá trị liên tục. Sơ đồ của học có giám sát được thể hiện trên hình 1.10.



Hình 1.10. Học có giám sát

1.3.2. Học không giám sát

Học không giám sát là cách học không có phản hồi từ môi trường để chỉ ra rằng đầu ra của mạng là đúng như thế nào. Mạng sẽ phải khám phá các đặc trưng, các điều chỉnh, các mối tương quan, hay các lớp trong dữ liệu vào một cách tự động. Trong thực tế, đối với phần lớn các biến thể của học không giám sát, các đích trùng với đầu vào. Nói một cách khác, học không giám sát thực hiện một công việc tương tự như một mạng tự nhiên liên hợp, cô đọng thông tin từ dữ liệu vào. Hai dạng học không giám sát phổ biến là:

- Phân cụm (clustering): là các đối tượng được phân chia thành một số nhóm sao cho mỗi nhóm gồm những đối tượng giống nhau và khác với các đối tượng của nhóm khác.
- Luật học kết hợp (association rule learning): là cách phát hiện những đối tượng hoặc thuộc tính thường xuyên xuất hiện cùng nhau.

1.3.3. Học bán giám sát

Học bán giám sát là một kỹ thuật của ngành học máy để xây dựng một hàm từ dữ liệu huấn luyện sau đó tổng quát hóa mô hình chung cho tất cả các dữ liệu được gắn nhãn và dữ liệu chưa được gắn nhãn. Nhằm tạo ra một kết quả như mong muốn.

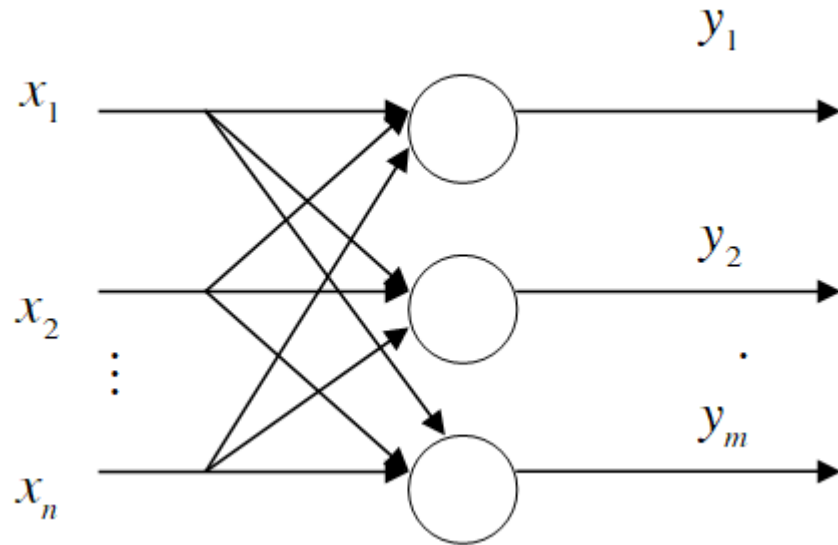
Một số phương pháp sử dụng học bán giám sát: Generative Models, Sime – Supervises Support Vector Machines – S3VM, Self – Training.

1.4. Phân loại mạng nơron

1.4.1. Phân loại theo số lớp

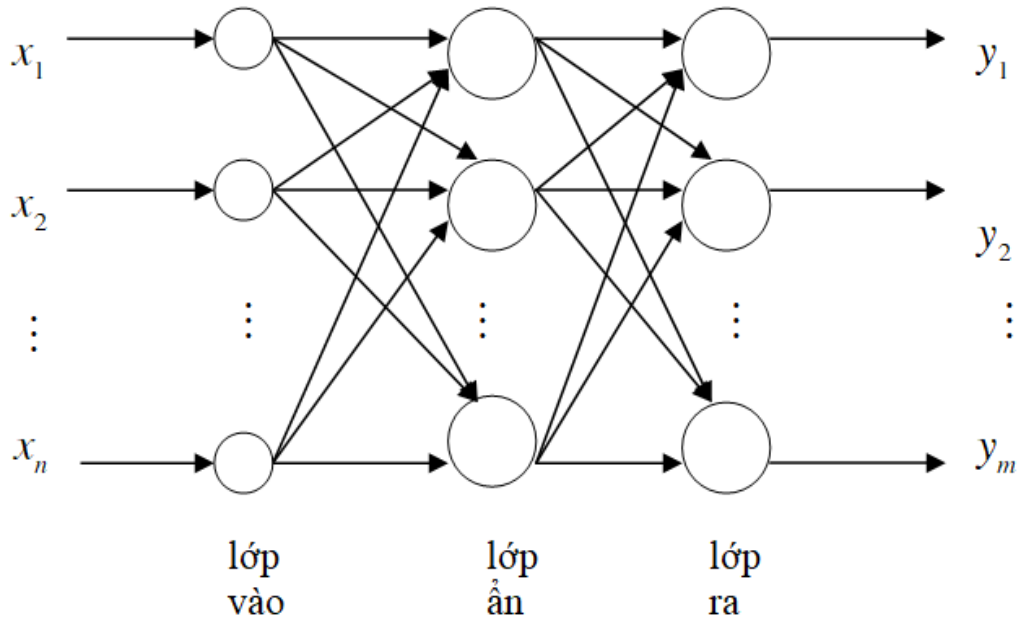
Dựa theo số lớp thì mạng nơron gồm hai loại: mạng một lớp và mạng nhiều lớp.

- Mạng một lớp: Mạng một lớp cấu thành từ một lớp mạng, nó vừa là lớp vào vừa là lớp ra.



Hình 1.11. Mô hình mạng noron 1 lớp

- Mạng noron nhiều lớp: Mạng nhiều lớp được cấu thành từ nhiều lớp liên kết với nhau, bao gồm một lớp vào, lớp ẩn và một lớp ra. Trong đó, lớp nhận tín hiệu đầu vào được gọi là lớp vào. Các tín hiệu đầu ra của mạng được sản sinh bởi lớp ra của mạng. Các lớp nằm giữa lớp vào và lớp ra được gọi là lớp ẩn. Lớp ẩn là thành phần nội tại của mạng, nó không có bất kỳ tiếp xúc nào với môi trường bên ngoài. Số lượng lớp ẩn có thể dao động từ 0 đến một vài lớp. Tuy nhiên thực tế cho thấy chỉ cần một lớp ẩn là mạng đã đủ để giải quyết được một lớp các bài toán phức tạp nào.



Hình 1.12 . Mô hình mạng noron nhiều lớp

1.4.2. Phân loại theo sự liên kết giữa các lớp

Sự liên kết trong mạng nơron tùy thuộc vào nguyên lý tương tác giữa đầu ra của từng nơron riêng biệt với nơron khác và tạo ra cấu trúc mạng nơron. Về nguyên tắc sẽ có rất nhiều kiểu liên kết giữa các nơron, nhưng chỉ có một số cấu trúc hay gặp trong ứng dụng sau:

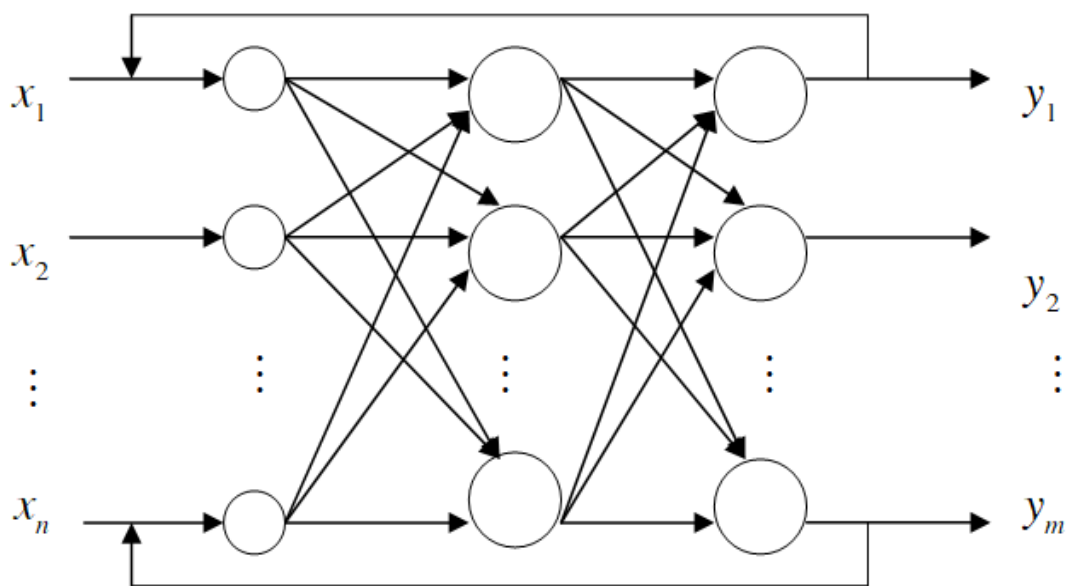
- Mạng truyền thẳng (Feedforward neural networks)

Dòng dữ liệu đầu vào từ các nơron đầu vào đến các nơron đầu ra chỉ được truyền thẳng. Việc xử lý dữ liệu có thể mở rộng ra nhiều lớp, nhưng không có các liên kết ngược. Tức là, không có các liên kết từ các đơn vị đầu ra tới các đơn vị đầu vào trong cùng một lớp hay các lớp trước đó.

Nếu mô hình hoá mạng truyền thẳng bằng một đồ thị, thì nó là một đồ thị có hướng hữu hạn không chu trình. Trong đó, mỗi nơron là một nút, các liên kết giữa các nơron là các cung của đồ thị. Hình 1.12 là một minh họa về mạng truyền thẳng nhiều lớp.

- Mạng hồi quy (mạng nối ngược) (Recurrent neural network)

Khác với mạng truyền thẳng, mạng hồi quy có chứa các liên kết ngược. Mô hình hoá mạng hồi quy bằng một đồ thị thì nó là một đồ thị có hướng hữu hạn có chu trình.



Hình 1.13 Mô hình mạng hồi quy

1.5. Kết luận

Trong chương này, chúng ta tìm hiểu các khái niệm, cấu trúc và các luật học của mạng nơ ron nói chung, đồng thời đi phân loại mạng nơ ron. Trong chương tiếp theo tác giả sẽ trình bày mạng nơ ron CNN trong nhận dạng khuôn mặt người.

CHƯƠNG II- MẠNG NƠ-RON VÀ ỨNG DỤNG TRONG NHẬN DẠNG MẶT NGƯỜI

2.1. Tổng quan về mạng noron tích chập

Mạng nơ-ron tích chập (CNN – Convolutional Neural Network) là một trong những mô hình mạng Học sâu phổ biến nhất hiện nay, có khả năng nhận dạng và phân loại hình ảnh với độ chính xác rất cao, thậm chí còn tốt hơn con người trong nhiều trường hợp. Mô hình này đã và đang được phát triển, ứng dụng vào các hệ thống xử lý ảnh lớn của Facebook, Google hay Amazon... cho các mục đích khác nhau như các thuật toán tagging tự động, tìm kiếm ảnh hoặc gợi ý sản phẩm cho người tiêu dùng.

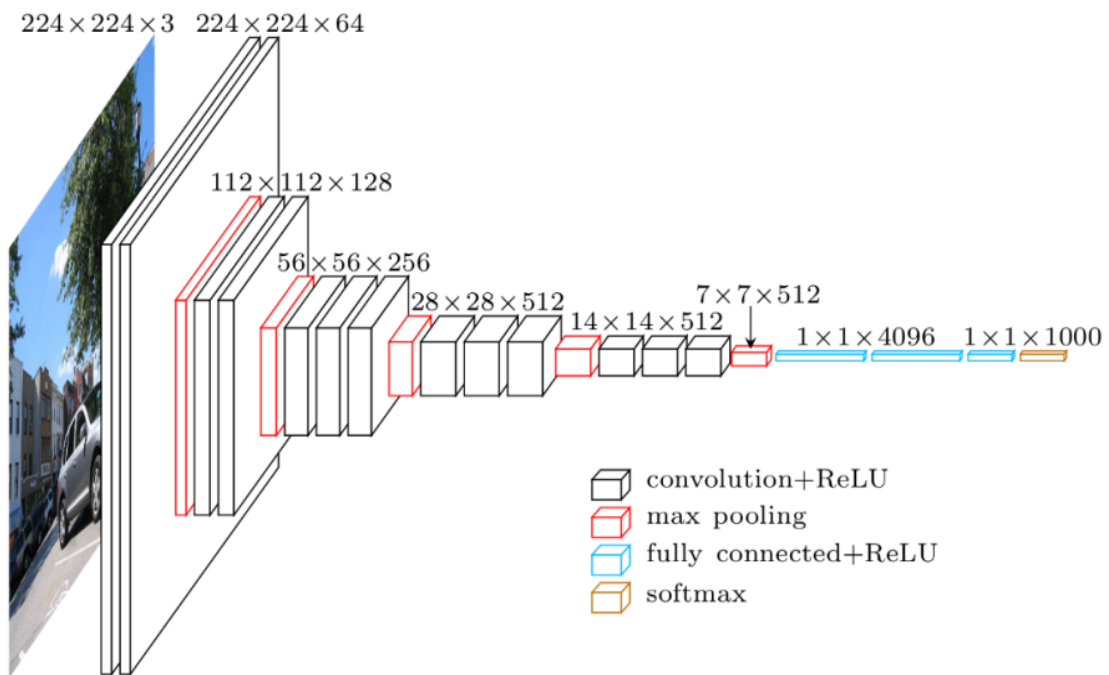
Mô hình mạng neural truyền thẳng ra đời đã được áp dụng nhiều vào các bài toán nhận dạng. Tuy nhiên, đối với dữ liệu hình ảnh, mạng neural truyền thẳng thể hiện không thực sự tốt. Chính sự liên kết quá đầy đủ tạo nên những hạn chế cho mô hình. Dữ liệu ảnh có kích thước khá lớn, một bức ảnh xám 32x32 điểm ảnh sẽ cho ra vector đặc trưng có 1024 chiều, đối với ảnh màu cùng kích thước sẽ có 3072 chiều. Điều này cũng có nghĩa là cần tới 3072 trọng số θ nối giữa lớp đầu vào và một node trong lớp ẩn tiếp theo. Số lượng trọng số sẽ càng nhân rộng hơn nếu số lượng node trong lớp ẩn tăng lên và số lượng lớp ẩn tăng lên. Như vậy chỉ với một bức ảnh nhỏ 32x32 thì cũng cần đến một mô hình mạng neural truyền thẳng khá đồ sộ. điều này khiến cho việc thao tác với các bức ảnh lớn hơn khá khó khăn.

Hơn nữa, việc liên kết đầy đủ các điểm ảnh vào một node trong mạng là dư thừa vì sự phụ thuộc lẫn nhau giữa các điểm ảnh xa nhau là không nhiều, mà chủ yếu là sự phụ thuộc giữa các điểm ảnh lân cận nó. Dựa trên tư tưởng này, mạng neural tích chập ra đời với cấu trúc khác hẳn mạng neural truyền thẳng. Thay vì toàn bộ ảnh nối trực tiếp với một node thì chỉ có một phần cục bộ trong ảnh nối với một node trong lớp tiếp theo. Dữ liệu hình ảnh ban đầu qua các lớp của mô hình mạng neural tích chập sẽ học được ra các đặc trưng để tiến hành phân lớp hiệu quả.

Về cơ bản, CNN bao gồm các lớp sau: lớp tích chập (Convolutional layer), lớp hàm kích hoạt (Activation layer), lớp pooling (Pooling layer) hay còn gọi là lớp subsampling, lớp kết nối đầy đủ (Fully Connected layer).

Trong mô hình CNN (Hình 2.1), các lớp được liên kết với nhau thông qua cơ chế tích chập. Lớp tiếp theo là kết quả của tích chập của lớp trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi neural ở lớp tiếp theo được sinh ra từ các bộ lọc áp đặt lên một vùng ảnh cục bộ của neural ở lớp trước đó.

Mỗi lớp như vậy được áp đặt các bộ lọc khác nhau, thông thường có vài trăm đến vài nghìn bộ lọc như vậy. Một số lớp khác như pooling/subsampling được dùng để chất lọc lại các thông tin hữu ích hơn.



Hình 2.1. Mô hình CNN

Trong suốt quá trình huấn luyện, mạng neural tích chập tự động học được các thông số cho các bộ lọc. Như trong phân loại ảnh, CNN sẽ cố gắng tìm ra thông số tối ưu cho các bộ lọc tương ứng theo thứ tự Điểm ảnh ban đầu → cạnh → hình dạng → mặt → đặc trưng mức cao. Lớp cuối cùng thường được dùng để phân loại ảnh.

2.1.1. Lớp tích chập

2.1.1.1. Tích chập

Tích chập (Convolution) là phép toán thực hiện với hai hàm số, kí hiệu là $*$. Phép tích chập được ứng dụng trong thống kê, thị giác máy tính, xử lý ảnh và xử lý tín hiệu,... Tích chập của hàm số f và g được viết là $f * g$, là một phép biến đổi tích phân đặc biệt

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau = \int_{-\infty}^{\infty} f(t - \tau)g(\tau)d\tau \quad (7)$$

Trong đó: τ : biến giả

t : biến thời gian

Một cách tổng quát, các hàm f và g là hàm số phức trong không gian \mathbb{R}^d , tích chập được định nghĩa là:

$$(f * g)(x) = \int_{\mathbb{R}^d} f(x)g(x - y)dy = \int_{\mathbb{R}^d} f(x - y)g(y)dy \quad (8)$$

Đối với Tích chập rời rạc, với hàm số phức f và g xác định trên tập số nguyên \mathbb{Z} , thì tích chập được định nghĩa

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f(m)g(n - m)dm \quad (9)$$

2.1.2.2. Tích chập trong xử lý ảnh

Tích chập lần đầu tiên được sử dụng trong xử lý tín hiệu số, nhờ vào nguyên lý biến đổi thông tin, các nhà khoa học đã áp dụng kỹ thuật này vào xử lý ảnh và video. Để dễ hình dung, ta có thể xem tích chập như một cửa sổ trượt áp lên một ma trận. Cửa sổ trượt còn được gọi là nhân (kernel).

Hình 2.2 minh họa cách tính tích chập trên ma trận ảnh đen trắng. Ma trận ảnh đen trắng có mỗi ô giá trị là một điểm ảnh 0 là màu đen, 1 là màu trắng. Ta dùng một ma trận 3x3 nhân, nhân từng thành phần tương ứng với ma trận ảnh. Giá

trị đầu ra do tích các thành phần này cộng lại. Kết quả của tích chập là một ma trận sinh ra từ việc trượt ma trận nhân và thực hiện tích chập cùng lúc lên toàn bộ ma trận ảnh gốc.

1	0	0	1	0		1	0	0		5	1	3		
0	1	1	0	1	x	0	1	1		2				
1	0	1	0	1		1	0	1						
1	0	0	1	0										
0	1	1	0	1										
1	0	0	1	0						5	1	3		
0	1	1	0	1		1	0	0		2	3	2		
1	0	1	0	1	x	0	1	1		2	2	4		
1	0	0	1	0		1	0	1						
0	1	1	0	1										

Hình 2.2. Minh họa tích chập trên ma trận ảnh

Sau đây là một số ví dụ của tích chập trên ảnh. Làm mờ ảnh gốc bằng cách lấy giá trị trung bình của các điểm ảnh xung quanh cho vị trí điểm ảnh trung tâm, Hình 2.3. Ngoài ra, còn có thể phát hiện biên cạnh bằng cách tính vi phân giữa các điểm ảnh lân cận, Hình 2.4.

Với đầu vào là một ảnh 3 chiều $W \times H \times 3$, có độ sâu là 3, hay còn được gọi là 3 kênh màu, ta có tích chập 3 chiều bằng cách tính riêng tích chập cho 3 kênh này với một bộ lọc, sau đó lấy trung bình cộng của ba kết quả đầu ra, ta được tích chập 3 chiều.

	0	1	0	
	1	-4	1	
	0	1	0	

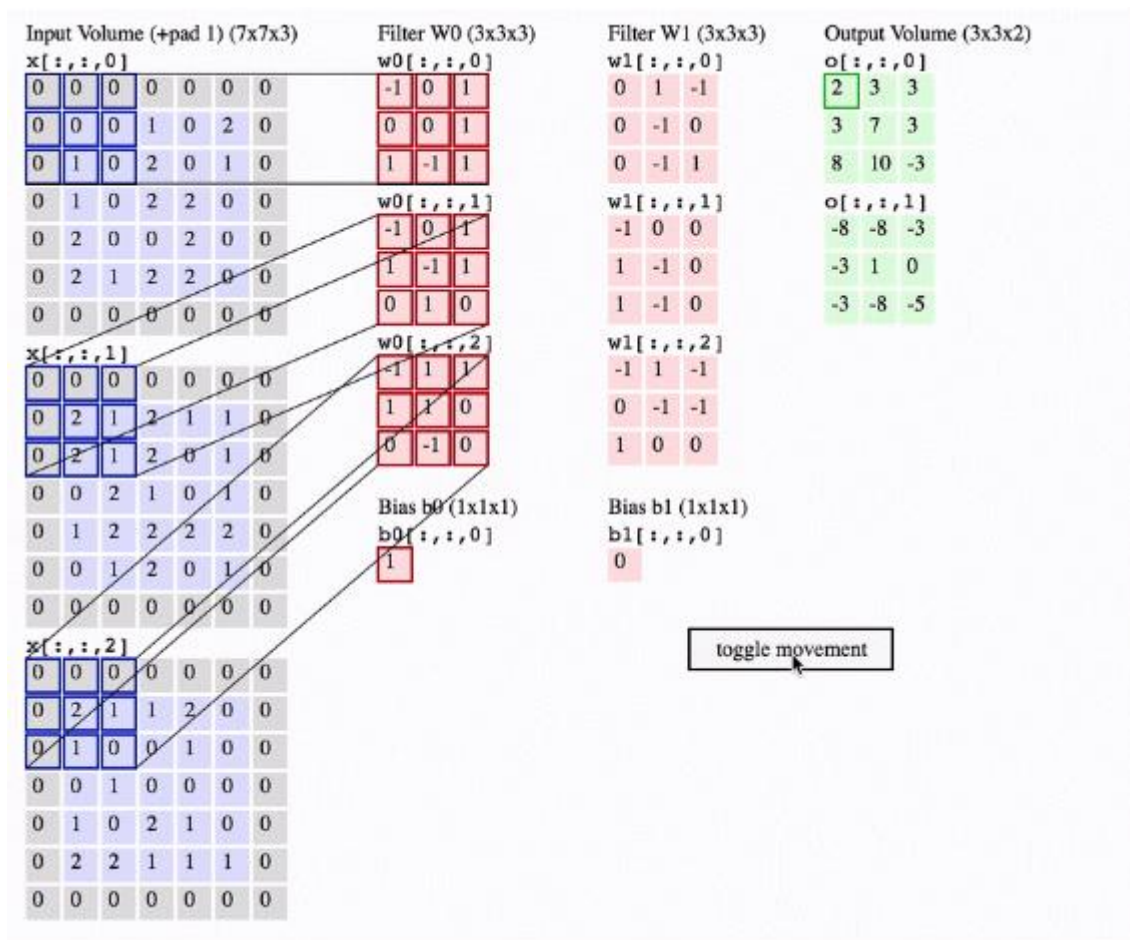


Hình 2.3.a. Tìm biên cạnh cho ảnh với tích chập

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



Hình 2.3.b. Làm mờ ảnh gốc với tích chập



Hình 2.4. Minh họa tích chập 3 chiều

Hình 2.4 là một ví dụ minh họa cho việc tính tích chập 3 chiều. Đầu vào là một ma trận có ba kênh biểu diễn cho một ảnh 3 chiều. Ta sử dụng hai bộ lọc W0 và W1. Cho mỗi kênh của ma trận tính tích chập với hai bộ lọc sau đó lấy trung bình cộng của ba kênh, ta được đầu ra là một ma trận có hai kênh tương ứng với hai bộ lọc.

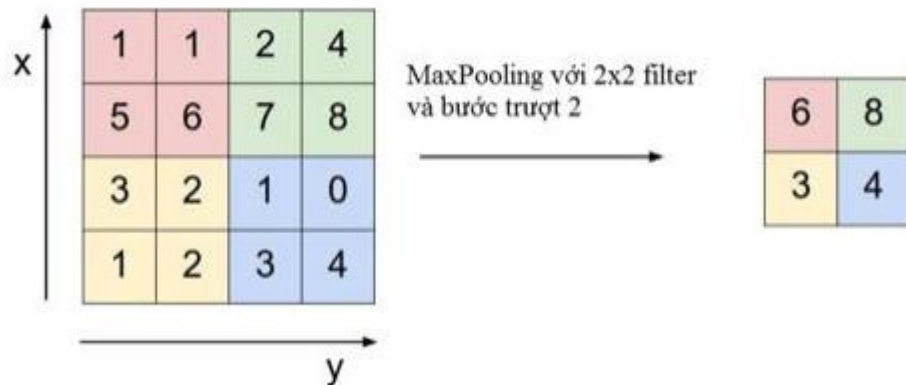
2.1.2. Lớp hàm kích hoạt

Hàm kích hoạt ReLu (Rectified Linear Unit), sau mỗi lớp tích chập thường được đặt một lớp hàm kích hoạt phi tuyến để tạo ra thông tin trừu tượng hơn cho các lớp sau. Hàm kích hoạt thường được sử dụng là hàm ReLU hoặc hàm tanh. Nói một cách đơn giản, lớp này có nhiệm vụ chuyển toàn bộ giá trị âm trong kết quả có được từ lớp tích chập thành giá trị không âm.

2.1.3. Lớp Pooling

Lớp pooling sử dụng một cửa sổ trượt quét qua toàn bộ ảnh, mỗi lần trượt theo các bước trượt cho trước. Lớp pooling không tính tích chập mà tiến hành lấy mẫu con. Khi cửa sổ trượt trên ảnh, chỉ có một giá trị được xem là giá trị đại diện cho thông tin ảnh tại vùng đó được giữ lại, giá trị này gọi là giá trị mẫu. Ta cũng có thể chia ảnh thành các ô hình chữ nhật không chồng lấp nhau để lấy mẫu.

Lớp pooling sẽ cho tính bất biến đối với phép dịch chuyển, phép quay và phép co giãn. Lớp này thường được sử dụng ngay sau lớp hàm kích hoạt, để đơn giản hóa thông tin đầu ra, giảm bớt số lượng neural, giảm bớt lượng tính toán, và do đó cũng giảm bớt hiện tượng quá khớp. Các phương thức lấy phổ biến trong lớp pooling là max pooling (lấy giá trị lớn nhất, Hình 2.5), min pooling (lấy giá trị nhỏ nhất) và average pooling (lấy giá trị trung bình), trong đó max pooling là phổ biến nhất.



Hình 2.5. Max Pooling với lọc 2 x 2

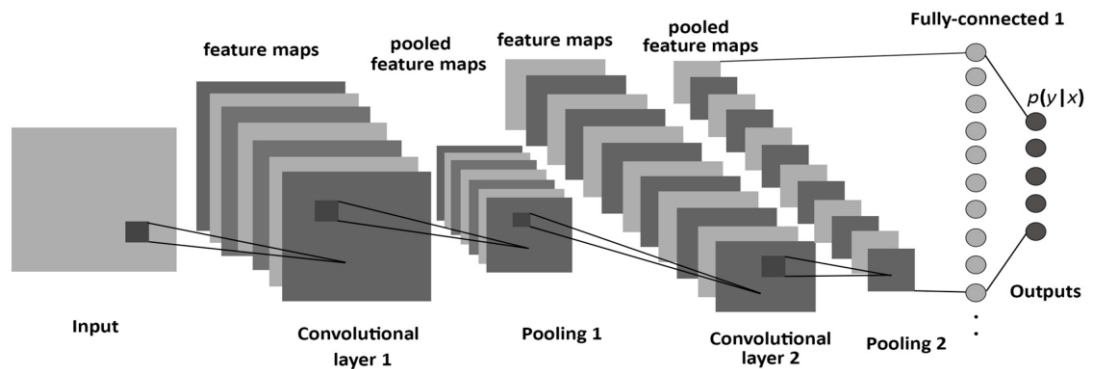
2.1.4. Lớp kết nối đầy đủ

Lớp Kết nối đầy đủ (Fully Connected - FC) tương tự với lớp trong mạng neural truyền thẳng, các giá trị ảnh được liên kết đầy đủ vào các node trong lớp tiếp theo. Sau khi ảnh được xử lý và rút trích đặc trưng từ các lớp trước đó, dữ liệu sẽ không còn quá lớn so với mô hình truyền thẳng để tiến hành nhận dạng. Lớp kết nối đầy đủ đóng vai trò phân lớp dữ liệu dựa trên dữ liệu đã được xử lý trước đó. Ở lớp

này ta thường dùng hàm softmax để phân loại dữ liệu tương tự như trong mạng neural truyền thẳng

2.1.5. Nguyên lý hoạt động

CNN được hình thành bằng cách ghép các lớp đã nêu ở trên lại. CNN bắt đầu với lớp tích chập. Theo sau lớp tích chập luôn là một lớp hàm kích hoạt, đôi khi ta có thể ghép cả hai lớp lại thành một lớp. Các lớp tiếp theo có thể là lớp pooling hoặc lớp tích chập tùy theo kiến trúc mà ta muốn xây dựng. Cuối cùng sẽ là lớp FC, để tiến hành phân lớp dữ liệu. Hình 2.6 là ví dụ về cấu trúc của CNN.

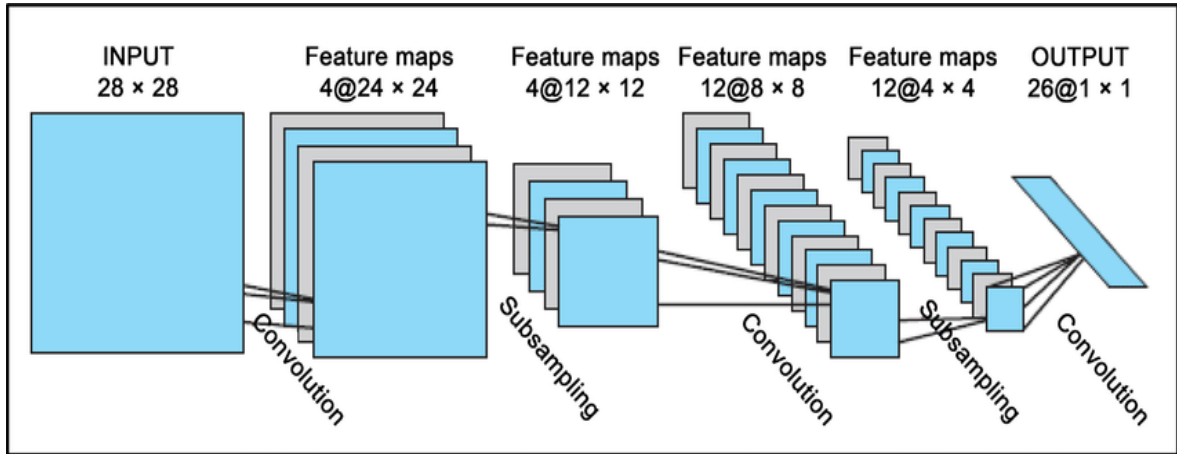


Hình 2.6. Ví dụ về cấu trúc CNN

Đối với bài toán nhận dạng ảnh, ta thường gọi ma trận ảnh lớp đầu vào là ảnh xạ đặc trưng (feature map). Như vậy với một ảnh 2 chiều đầu vào qua một bộ lọc sẽ cho ta một ảnh xạ đặc trưng tiếp theo. Có nghĩa là, mỗi bộ lọc cho ta một ảnh xạ đặc trưng tương ứng với ảnh đầu vào. Trong nhận dạng ảnh, ta cần nhiều hơn một ảnh xạ đặc trưng cho mỗi lớp.

Ví dụ ở hình 2.7, với ảnh đầu vào kích thước 32×32 , bộ lọc có kích thước 5×5 . Ta thấy mỗi ảnh xạ đặc trưng cần $5 \times 5 = 25$ trọng số và 1 hướng, như vậy mỗi ảnh xạ đặc trưng cần 26 tham số. Nếu có 10 ảnh xạ đặc trưng thì ta cần $26 \times 10 = 260$ tham số. Ta xét lại từ đầu, nếu lớp đầu tiên có FC nghĩa là ta có $32 \times 32 = 1024$ neural đầu vào, như vậy chỉ có 30 neural ẩn. Ta cần $32 \times 32 \times 30 = 30720$ trọng số và 30 hướng. Tổng tham số là $32 \times 32 \times 30 + 30 = 30750$ tham số lớn hơn rất nhiều so với tham số trong CNN. Ví dụ vừa rồi chỉ mô tả để thấy được sự ước lượng số

tham số, chứ chúng ta không so sánh được trực tiếp vì hai mô hình khác nhau. Nhưng nếu mô hình có số lượng tham số ít hơn thì sẽ chạy nhanh hơn.



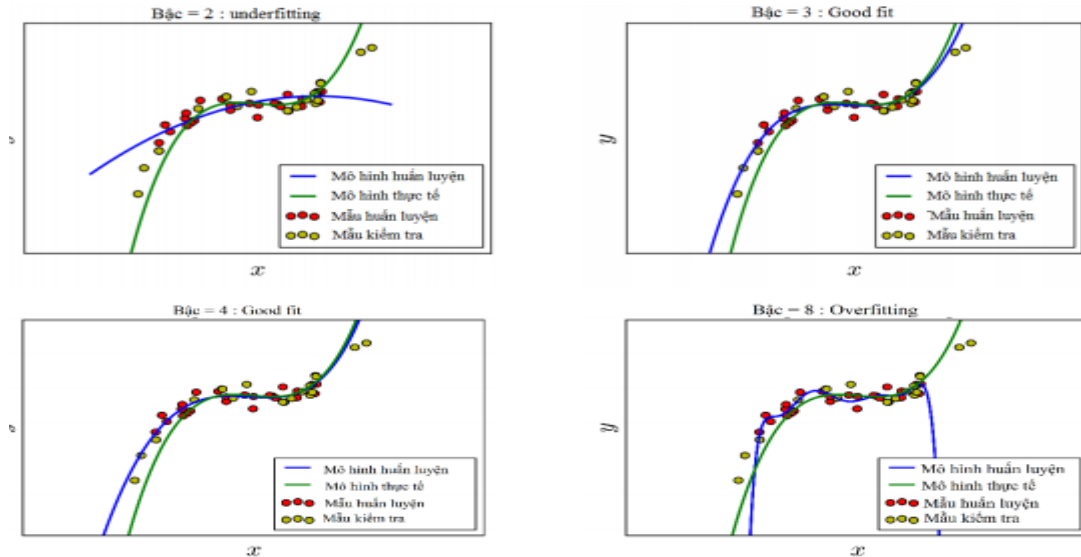
Hình 2.7. Mô hình chi tiết cấu trúc 1 CNN

2.1.6. Overfitting

Overfitting là hiện tượng mô hình tìm được quá khớp với dữ liệu huấn luyện, do đó quá khớp có thể dẫn đến việc dự đoán nhầm nhiều, và chất lượng mô hình không còn tốt trên dữ liệu kiểm tra nữa. Dữ liệu kiểm tra được giả sử là không được biết trước và không được sử dụng để xây dựng mô hình. Về cơ bản, quá khớp xảy ra khi mô hình quá phức tạp để mô phỏng dữ liệu huấn luyện. Điều này đặc biệt xảy ra khi lượng dữ liệu huấn luyện quá nhỏ trong khi độ phức tạp mô hình quá cao. Trong perceptron đa lớp, độ phức tạp của mô hình có thể được coi là số lượng các lớp ẩn và số lượng các node trong các lớp ẩn.

Ở Hình 2.8, ta có 50 điểm dữ liệu được tạo bởi một đa thức bậc ba cộng thêm nhiễu. Tập dữ liệu được chia làm hai. 30 điểm dữ liệu màu đỏ cho huấn luyện dữ liệu, 20 điểm dữ liệu màu vàng cho việc kiểm tra mô hình. Đồ thị của đa thức bậc ba này được biểu diễn bởi đường màu xanh lá cây, còn đường màu xanh dương biểu diễn cho mô hình dự đoán đa thức. Ta nhận thấy, với mô hình dự đoán có bậc hai, mô hình dự đoán chưa khớp với mô hình thực tế. Mô hình dự đoán có bậc bằng ba và bốn khớp với dữ liệu huấn luyện và khớp với cả mô hình thực tế. Mô hình dự

đoán có bậc bằng tám quá khớp với dữ liệu huấn luyện, dẫn tới không khớp với mô hình thực tế.



Hình 2.8. Overfitting

Một bài toán trên thực tế thường rất phức tạp và sẽ có nhiều nhiễu, việc mô hình quá khớp với dữ liệu huấn luyện sẽ dẫn đến dự đoán sai nhiều, dẫn đến kết quả tính toán thấp. Vì vậy, khi xây dựng một mô hình, quá khớp là điều ta luôn cần tránh. Sau đây là hai đại lượng đơn giản để đánh giá chất lượng của mô hình trên dữ liệu huấn luyện và dữ liệu kiểm tra. Giả sử, y là nhãn của dữ liệu và \hat{y} là đầu ra dự đoán của mô hình, khi đó:

- Sai số huấn luyện (train error) thường là hàm lỗi áp dụng lên dữ liệu huấn luyện. Hàm lỗi này tính tổn thất trung bình trên mỗi điểm dữ liệu huấn luyện. Với học có giám sát hồi quy, đại lượng này thường được định nghĩa

$$\text{sai số huấn luyện} = \frac{1}{N_{\text{huấn luyện}}} \sum_{\text{tập huấn luyện}} \|y - \hat{y}\|_p^2$$

- Sai số kiểm tra (test error) tương tự như sai số huấn luyện, nhưng việc tính toán được áp dụng trên dữ liệu kiểm tra. Lưu ý, dữ liệu kiểm tra chỉ để đánh giá mô hình đã được xây dựng bằng dữ liệu huấn luyện, không được dùng trong quá trình xây dựng mô hình.

Vì lượng dữ liệu trong hai tập huấn luyện và kiểm tra có sự chênh lệch rất lớn, nên việc lấy trung bình là rất quan trọng. Một mô hình được coi là tốt hay khớp nếu cả hai đại lượng sai số huấn luyện và sai số kiểm tra đều thấp. Nếu sai số huấn luyện thấp nhưng sai số kiểm tra cao, ta nói mô hình bị quá khớp. Nếu sai số huấn luyện cao và sai số kiểm tra cao, ta nói mô hình bị dưới khớp (underfitting). Có rất nhiều kỹ thuật được sử dụng để tránh quá khớp, điển hình là duyệt chéo (Cross-Validation) và điều chỉnh (Regularization). Trong kỹ thuật điều chỉnh có hai kỹ thuật thường dùng là giảm trọng số (Weight Decay) và loại bỏ (Dropout).

2.1.7. Phương pháp lựa chọn mô hình

2.1.7.1. Phương pháp tập duyệt

Theo như bình thường ta sẽ chia tập dữ liệu thành hai phần là tập dữ liệu huấn luyện và tập dữ liệu kiểm tra. Khi xây dựng mô hình ta không được dùng dữ liệu trong tập dữ liệu kiểm tra. Để đánh giá được chất lượng của mô hình đối với dữ liệu chưa thấy bao giờ, ta có phương pháp đơn giản nhất là trích từ tập dữ liệu huấn luyện ra một tập con nhỏ và ta thực hiện việc đánh giá mô hình trên tập con nhỏ này. Tập con nhỏ được trích ra từ tập dữ liệu huấn luyện gọi là tập duyệt (validation set). Lúc này, tập dữ liệu huấn luyện mới chính là phần còn lại của tập dữ liệu huấn luyện ban đầu sau khi trích ra tập duyệt. Sai số huấn luyện sẽ được tính trên tập huấn luyện mới này. Ta có một khái niệm mới là sai số duyệt (Validation Error). Sai số duyệt được tính tương tự như sai số kiểm tra, nhưng thay vì tính trên tập kiểm tra thì sai số này được tính trên tập duyệt. Với khái niệm mới là tập duyệt, ta sẽ cố gắng tìm một mô hình sao cho cả sai số huấn luyện và sai số duyệt đều nhỏ. Qua đó có thể dự đoán Sai số kiểm tra cũng nhỏ. Phương pháp thường sử dụng là sử dụng nhiều mô hình khác nhau, nếu mô hình nào cho sai số duyệt nhỏ nhất sẽ là mô hình tốt. Khi mô hình càng phức tạp thì sai số huấn luyện có xu hướng càng nhỏ đi. Thông thường ta bắt đầu bằng mô hình đơn giản, sau đó tăng dần độ phức tạp của mô hình. Cho tới khi Sai số duyệt có xu hướng tăng lên thì ta dừng và chọn mô hình ngay trước nó.

2.1.7.2. Phương pháp kiểm duyệt chéo

Trong nhiều trường hợp, dữ liệu để xây dựng mô hình bị hạn chế. Nếu ta lấy quá nhiều dữ liệu để làm tập xác nhận thì phần còn lại của tập huấn luyện sẽ không đủ để xây dựng mô hình. Khi đó, ta phải lấy lượng dữ liệu để làm tập xác nhận thật nhỏ để giữ cho lượng dữ liệu trong tập huấn luyện còn lại đủ lớn. Tuy nhiên, một vấn đề mới lại phát sinh. Lượng dữ liệu trong tập xác nhận quá nhỏ sẽ dẫn đến hiện tượng quá khớp xảy ra trên tập huấn luyện còn lại. Để giải quyết vấn đề này, một cải tiến của kỹ thuật xác nhận được đề xuất là kỹ thuật duyệt chéo (Cross - Validation).

Trong kỹ thuật duyệt chéo, ta sẽ đánh giá chất lượng mô hình trên nhiều tập duyệt có lượng dữ liệu nhỏ khác nhau. Có nghĩa là, ta chia tập huấn luyện thành k tập con, không có phần tử chung, số lượng phần tử trong mỗi tập nhỏ, các tập con này có kích thước gần bằng nhau. Tại mỗi lần kiểm thử hay được gọi là chạy, ta sẽ lấy ra một tập con để làm tập duyệt, và phần còn lại là tập huấn luyện mới. Qua nhiều lần kiểm thử, ta xác định mô hình dựa trên trung bình của các sai số huấn luyện và sai số duyệt. Kỹ thuật này còn được gọi là duyệt chéo k -gấp (k -fold cross validation). Trường hợp đặc biệt, khi k bằng với số lượng phần tử trong tập huấn luyện, tức là mỗi tập con chỉ có một phần tử, thì ta gọi trường hợp này là kỹ thuật loại bỏ một phần tử (leave-oneout).

2.1.7.8. Phương pháp điều chỉnh

Kỹ thuật duyệt chéo có một nhược điểm lớn là số lượng các lần chạy trong quá trình huấn luyện tỉ lệ thuận với giá trị k , có nghĩa là số các tập con càng lớn thì số lần chạy xây dựng mô hình càng nhiều. Trong khi các bài toán của Máy học có lượng tham số thường rất lớn, khoảng giá trị của mỗi tham số rộng, tham số có thể có giá trị thực. Như vậy, việc chỉ xây dựng một mô hình thôi cũng đã rất phức tạp. Có một cách làm giảm đi số mô hình cần huấn luyện, thậm chí chỉ còn một mô hình. Cách này có tên gọi chung là kỹ thuật điều chỉnh (Regularization)[17].

Kỹ thuật điều chỉnh được hiểu một cách cơ bản là thay đổi mô hình một chút để tránh sự quá khớp trong khi vẫn giữ được tính tổng quát của tập duyệt. Tính tổng quát này là tính mô tả được nhiều dữ liệu, trong cả tập huấn luyện và tập kiểm tra. Một cách cụ thể hơn, ta sẽ cố gắng di chuyển nghiệm của bài toán tối ưu hàm lỗi tới một điểm gần nó hơn. Hướng di chuyển sẽ là hướng làm cho mô hình ít phức tạp hơn mặc dù giá trị của hàm lỗi có tăng lên một chút.

2.1.8. Dropout

Loại bỏ (Dropout) là một kỹ thuật điều chỉnh (regularization) nhằm giảm sự quá khớp trong mạng neural bằng cách ngăn chặn sự đồng biến phức tạp trên tập dữ liệu huấn luyện. Thuật ngữ loại bỏ đề cập đến việc loại bỏ các node (cả node ẩn và node nhìn thấy được) trong một mạng neural. Trong mạng neural sâu chứa nhiều lớp ẩn không tuyến tính. Điều này thể hiện mô hình có thể học được các mối quan hệ rất phức tạp giữa đầu ra và đầu vào. Tuy nhiên, với dữ liệu huấn luyện hạn chế, nhiều mối quan hệ phức tạp này có thể dẫn đến kết quả của việc lấy mẫu bị nhiễu. Vì vậy, các mẫu nhiễu này sẽ xuất hiện trong tập huấn luyện mà không xuất hiện trong tập kiểm tra. Điều này dẫn đến hiện tượng quá khớp. Việc loại bỏ này tức là cho các node giá trị bằng 0, rồi tính toán truyền thẳng và truyền ngược như bình thường. Ta nên kết hợp sử dụng kỹ thuật loại bỏ này với kỹ thuật khác, chẳng hạn như kỹ thuật suy giảm trọng số. Thông thường, một số mô hình học sâu sử dụng kỹ thuật loại bỏ trong lớp kết nối đầy đủ. Nhưng ta cũng có thể sử dụng kỹ thuật loại bỏ sau các lớp pooling, tạo ra một số ảnh tăng thêm nhiễu.

2.2. Tổng quan về bài toán nhận dạng

Nhận dạng mặt người là một trong những ứng dụng quan trọng của thị giác máy tính nói riêng cũng như khoa học máy tính nói chung. Bài toán nhận dạng khuôn mặt người vốn được nghiên cứu từ những năm 1970 và cho đến nay, rất nhiều nghiên cứu lần ứng dụng cho bài toán này đã ra đời. Bài toán nhận dạng mặt người có thể áp dụng rộng rãi trong nhiều lĩnh vực khác nhau. Các ứng dụng liên quan đến nhận dạng mặt người có thể kể như: hệ thống phát hiện tội phạm, hệ thống

theo dõi nhân sự trong một đơn vị, hệ thống tìm kiếm thông tin trên ảnh, video dựa trên nội dung,.... Để giải quyết bài toán này, cần xác định hai vấn đề chính. Thứ nhất, dùng thông tin nào để nhận dạng, chẳng hạn như mắt, mũi, miệng, chân mày,... hay kết hợp tất cả các thông tin trên. Thứ hai, dùng phương pháp nào để nhận dạng nguồn thông tin đó [3].

Hệ thống nhận dạng mặt người với tập dữ liệu đầu vào là bộ ảnh về mặt người đã thông qua tiền xử lý, như các kỹ thuật xử lý ảnh (cân bằng sáng, tách ngưỡng, histogram ...), cũng như giai đoạn phát hiện mặt người (Face Detection). Từ tập dữ liệu này hệ thống sẽ tiến hành trích chọn các đặc trưng. Quá trình rút trích đặc trưng nhằm giảm số chiều của không gian dữ liệu bằng cách loại bỏ bớt những thành phần (chiều) dư thừa trong dữ liệu sao cho lượng thông tin sau khi rút trích vẫn đảm bảo các đặc trưng của dữ liệu ban đầu. Từ các đặc trưng được trích chọn, sẽ tiến hành so khớp, phân lớp để nhận dạng. Đây Là giai đoạn cuối cùng trong bài toán nhận dạng mặt người. So sánh đặc trưng ảnh với đặc trưng của toàn bộ ảnh trong cơ sở dữ liệu nhận dạng. Dựa vào khoảng cách so sánh, hệ thống xác định thông tin của người được nhận dạng. Thường dữ liệu đặc trưng được thể hiện bằng một vector nên có thể dùng khoảng cách Euclid giữa hai vector để phân lớp, ngoài ra người ta cũng có thể dùng kết hợp PCA, LDA và độ đo Mean-KNN để tính khoảng cách kết hợp, từ đó cho ra kết quả.

2.2.1. Ứng dụng của nhận dạng khuôn mặt

Có nhiều ứng dụng đã được xây dựng, tôi xin trình bày một số ứng dụng chính:

Hệ thống tương tác giữa người và máy: giúp những người bị tật hoặc khiếm khuyết có thể trao đổi. Những người dùng ngôn ngữ tay có thể giao tiếp với những người bình thường. Những người bị bại liệt thông qua một số ký hiệu nháy mắt có thể biểu lộ những gì họ muốn, Đó là các bài toán điều bộ của bàn tay (hand gesture), điều bộ, khuôn mặt.

Nhận dạng người A có phải là tội phạm truy nã hay không? Giúp cơ quan an ninh quản lý tốt con người. Công việc nhận dạng có thể ở trong môi trường bình thường cũng như trong bóng tối (sử dụng camera hồng ngoại).

Hệ thống quan sát, theo dõi và bảo vệ. Các hệ thống camera sẽ xác định đâu là con người và theo dõi con người đó xem họ có vi phạm gì không, ví dụ xâm phạm khu vực không được vào,

Lưu trữ (rút tiền ATM, để biết ai rút tiền vào thời điểm đó), hiện nay có tình trạng những người bị người khác lấy mất thẻ ATM hay mất mã số PIN và những người ăn cắp này đi rút tiền, hoặc những người chủ thẻ đi rút tiền nhưng lại báo cho ngân hàng là mất thẻ và mất tiền. Các ngân hàng có nhu cầu khi có giao dịch tiền sẽ kiểm tra hay lưu trữ khuôn mặt người rút tiền để sau đó đối chứng và xử lý, thẻ căn cước, chứng minh nhân dân (Face Identification).

Điều khiển vào ra: văn phòng, công ty, trụ sở, máy tính, Palm, Kết hợp thêm vân tay và móng mắt. Cho phép nhân viên được ra vào nơi cần thiết, hay mỗi người sẽ đăng nhập máy tính cá nhân của mình mà không cần nhớ tên đăng nhập cũng như mật khẩu mà chỉ cần xác định thông qua khuôn mặt.

An ninh sân bay, xuất nhập cảnh (hiện nay cơ quan xuất nhập cảnh Mỹ đã áp dụng). Dùng để xác thực người xuất nhập cảnh và kiểm tra có phải là nhân vật khủng bố không.

Tương lai sẽ phát triển các loại thẻ thông minh có tích hợp sẵn đặc trưng của người dùng trên đó, khi bất cứ người dùng khác dùng để truy cập hay xử lý tại các hệ thống sẽ được yêu cầu kiểm tra các đặc trưng khuôn mặt so với thẻ để biết nay có phải là chủ thẻ hay không.

Tìm kiếm và tổ chức dữ liệu liên quan đến con người thông qua khuôn mặt người trên nhiều hệ cơ sở dữ liệu lưu trữ thật lớn, như internet, các hãng truyền hình, Ví dụ: tìm các đoạn video có tổng thống Bush phát biểu, tìm các phim có diễn viên Lý Liên Kiệt đóng, tìm các trận đá banh có Ronaldo đá.

Hiện nay có nhiều hướng tiếp cận để xác định một ảnh có phải là ảnh khóa thân hay không? Khuôn mặt người được xem như một yếu tố để xác định cho một hướng tiếp cận mà được dùng gần đây.

Kiểm tra trạng thái người lái xe có ngủ gật, mất tập trung hay không, và hỗ trợ thông báo khi cần thiết.

Phân tích cảm xúc trên khuôn mặt.

2.2.2. Các hướng tiếp cận bài toán nhận dạng khuôn mặt

2.2.2.1. Hướng tiếp cận đặc trưng

Có hai hướng tiếp cận chính làm hạt nhân của các kỹ thuật phân tích đặc trưng mặt người: hướng tiếp cận hình học và hướng tiếp cận hình ảnh [1].

Hướng tiếp cận hình học: sử dụng việc ánh xạ không gian các đặc trưng mặt người. Mặt người được phân loại theo khoảng cách hình học, theo đường bao và theo các góc giữa các điểm.

Hướng tiếp cận hình ảnh: bao gồm việc xây dựng các mẫu từ những đặc trưng mặt người. Mẫu của các đặc trưng nổi bật, hoặc thậm chí là toàn khuôn mặt được thiết lập, việc nhận dạng được thực hiện bằng cách duyệt các khuôn mặt rồi tìm mặt nào khớp nhất với mẫu.

Hiện nay hệ thống nhận dạng mặt người vẫn đang tiếp tục được phát triển. Dưới đây tôi trình bày một số phương pháp trích chọn đặc trưng:

- Phương pháp phân tích thành phần chính (Principal Component Analysis - PCA)
- Phương pháp phân tích biệt số tuyến tính (Linear Discriminant Analysis - LDA)
- Phương pháp biến đổi hình thái
- Phương pháp trích đặc trưng tự động (Deep Neural Networks - DNN)
- a. Phương pháp phân tích thành phần chính

Ý tưởng chính của phương pháp phân tích thành phần chính (Principal Component Analysis - PCA) là tìm các vector biểu diễn những sự phân bố đặc trưng nhất của các ảnh trong toàn bộ không gian ảnh. Những vector này hình thành nên một không gian con của các ảnh được gọi là “không gian mặt người”[17]. Mỗi ảnh là một ma trận $M \times N$ các điểm ảnh, ma trận này được chuyển thành một vector bằng cách xếp các cột điểm ảnh liên tiếp nhau. Như vậy, mỗi ảnh được coi là một điểm trong không gian MN chiều. Các vector của tập ảnh huấn luyện được xếp thành một ma trận. Từ ma trận này, ta sẽ tính được ma trận hiệp phương sai của dữ liệu, là ma trận trong đó mỗi phần tử (i, j) là giá trị hiệp phương sai giữa chiều thứ i và chiều thứ j . Công việc tiếp theo là tính các giá trị riêng và vector riêng của ma trận hiệp phương sai. Các vector riêng chính là các thành phần đặc trưng cho sự phân bố của tập dữ liệu. Cuối cùng, ma trận biến đổi của phương pháp PCA là ma trận hình thành từ các vector riêng ứng với các giá trị riêng lớn nhất vừa tính được, nghĩa là các thành phần đặc trưng nhất, còn các thành phần kém quan trọng hơn sẽ được bỏ qua để giảm số chiều của dữ liệu [4]. Mục tiêu của PCA là tìm một không gian mới (với số chiều nhỏ hơn không gian cũ). Các trục tọa độ trong không gian mới được xây dựng sao cho trên mỗi trục, độ biến thiên của dữ liệu trên đó là lớn nhất có thể.

Phương pháp phân tích thành phần chính cho phép xác định các mẫu trong một tập dữ liệu và biểu diễn dữ liệu theo cách có thể làm nổi bật được sự giống nhau và khác nhau giữa các mẫu. Vì các mẫu có thể rất khó tìm ra trong dữ liệu nhiều chiều (do không thể biểu diễn trực quan bằng đồ họa), nên PCA là một công cụ hữu hiệu để phân tích dữ liệu [1].

b. Phương pháp phân tích biệt số tuyến tính

Phương pháp PCA ở trên còn có nhược điểm là chỉ làm nổi bật lên các đặc trưng của từng ảnh, mà chưa quan tâm đến các ảnh đó là của cùng một người hay của những người khác nhau. Phương pháp phân tích biệt số tuyến tính (Linear Discriminant Analysis - LDA) có thể khắc phục được những nhược điểm đó [4].

Nhiệm vụ chính của phương pháp là tính sự biến thiên giữa các ảnh của những người khác nhau và tính sự biến thiên giữa các ảnh của cùng một người, sau đó tìm một phép biến đổi để làm cực đại tỉ số của hai sự biến thiên trên. Nghĩa là, tập ảnh huấn luyện sẽ được biến đổi sang một không gian mới sao cho sự khác nhau giữa các 9 ảnh của những người khác nhau được tăng lên tối đa, còn sự giống nhau giữa các ảnh của cùng một người được làm cực tiểu [1].

LDA cho phép đánh giá khách quan về tầm quan trọng của thông tin hình ảnh trong các đặc điểm khác nhau của khuôn mặt để xác định khuôn mặt người. LDA cũng cho chúng ta một tập hợp nhỏ các tính năng mà thực hiện phù hợp nhất thông tin cho các mục đích phân loại. Phương pháp LDA khắc phục những hạn chế của phương pháp phân tích thành phần chính bằng cách áp dụng các tiêu chuẩn phân biệt tuyến tính. Mục tiêu của LDA là thực hiện giảm chiều trong khi vẫn giữ càng nhiều các thông tin phân biệt giữa các lớp càng tốt. Tìm hướng dọc theo đó các lớp được phân cách tốt nhất. Thực hiện bằng cách xem xét các tán xạ trong các lớp và giữa các lớp. Nó có khả năng thay đổi hình ảnh từ những biến đổi do các nguồn khác nhau như ánh sáng.

c. Phương pháp biến đổi hình thái

Hai phương pháp PCA và LDA cho phép phân tích dựa trên các đặc điểm mang tính thống kê của ảnh. Tuy nhiên, các phương pháp này chưa xét đến những đặc điểm về hình thái của mặt người. Đó chính là những đặc trưng về đường nét của các bộ phận trên khuôn mặt như mắt, mũi, miệng,.... Phương pháp biến đổi hình thái sẽ bổ sung cho những thiếu sót này của hai phương pháp trên [9].

Xử lý ảnh về hình thái là một phép xử lý trong đó dạng không gian hoặc cấu trúc của các đối tượng trong ảnh được chỉnh sửa. Phép dẫn ảnh và phép co ảnh là hai thao tác xử lý hình thái cơ bản. Với phép dẫn ảnh, một đối tượng sẽ nổi đều lên trong không gian, còn với phép co ảnh đối tượng sẽ co đều xuống [16],[15].

d. Phương pháp trích xuất đặc trưng tự động

Phương pháp phân tích thành phần chính PCA là một kỹ thuật được tạo ra nhằm khử tương quan và giảm chiều dữ liệu. PCA là một hình thức cơ bản của học đặc trưng với khả năng tự động phát hiện các biểu diễn nhỏ gọn và có ý nghĩa của dữ liệu thô mà không dựa vào tri thức chuyên gia hay tri thức về kiểu ứng dụng cụ thể. Các véc-tơ riêng của hiệp phương sai cho một tập mẫu (tương ứng với giá trị riêng lớn nhất) được sử dụng để mở rộng thành một không gian phụ có số chiều thấp hơn, tập trung vào phương sai của dữ liệu gốc. Việc chiếu dữ liệu gốc lên không gian con với phương sai cực đại được sử dụng làm biểu diễn đặc trưng.

Mặc dù PCA có thể học đặc trưng theo cách không giám sát nhưng nó chỉ là kết hợp tuyến tính của các đặc trưng chưa xử lý. Do đó, PCA không đủ khả năng mô hình hóa cho các phụ thuộc phi tuyến phức tạp. Plotz và cộng sự đã đề xuất sử dụng mạng nơ-ron sâu (DNN) để tách được các đặc trưng có ý nghĩa hơn. Một trong những khác biệt chính giữa mạng nơ-ron truyền thống và DNN là DNN có thể có nhiều lớp, trong khi mạng nơ-ron truyền thống thường sử dụng nhiều nhất 3 lớp. Một lợi thế quan trọng khác nữa của DNN là khả năng biểu diễn các dữ liệu đầu vào. DNN có thể mô hình hóa nhiều hoạt động đa dạng mà không cần nhiều dữ liệu huấn luyện. Nó có thể chia sẻ các phần không gian đầu vào tương tự chỉ với một vài nút ẩn, trong khi giữ cho các nút khác “cảm nhận” một tập con các đặc trưng đầu vào quan trọng cho việc nhận dạng.

CNN bao gồm một hoặc nhiều các cặp thuộc lớp convolution và pooling. Các bộ phận nhỏ cục bộ của đầu vào được bắt giữ bởi các lớp convolution với một tập các bộ lọc cục bộ. Lớp pooling có thể lưu giữ được các đặc trưng bất biến. Cuối cùng, lớp kết nối đầy đủ đứng đầu sẽ kết hợp các đầu vào từ tất cả các đặc trưng để thực hiện phân loại toàn bộ đầu vào.

2.2.2.2. Hướng tiếp cận nhận dạng

a. Không cần canh chỉnh

Ảnh trích xuất từ các thiết bị thu ảnh (như camera giám sát hay camera du lịch) thường xuất hiện ảnh chỉ chụp một phần mặt người, Hình 3.1. Những phương

pháp nhận dạng khuôn mặt theo kiểu toàn cục (PCA và LDA) hay địa phương (Gabor, L P) thường canh chỉnh và biểu diễn các ảnh khuôn mặt theo cùng một kích cỡ, sau đó nối dài các giá trị điểm ảnh hoặc trích xuất các vector đặc trưng theo số chiều nhất định. Tuy nhiên, với ảnh một phần khuôn mặt thì cách làm này không khả thi do mặt không đủ chi tiết cũng như sẽ gặp khó khăn khi canh chỉnh mặt, với không có gì đảm bảo rằng những điểm mốc phổ biến của khuôn mặt sẽ xuất hiện ở ảnh một phần mặt. Do đó nhóm tác giả Liao và cộng sự [18] đề xuất thuật toán nhận dạng một phần mặt người mà không cần dùng tọa độ 2 mắt (hay bất kỳ điểm nào khác) để canh chỉnh mặt.



Hình 2.9. Ví dụ về ảnh một phần khuôn mặt: (a) Ảnh một phần khuôn mặt trong bộ dữ liệu LFW; (b) Ảnh một phần khuôn mặt trong đám đông; (c) Ảnh khuôn mặt bị che bởi mắt kính, áo khoác.

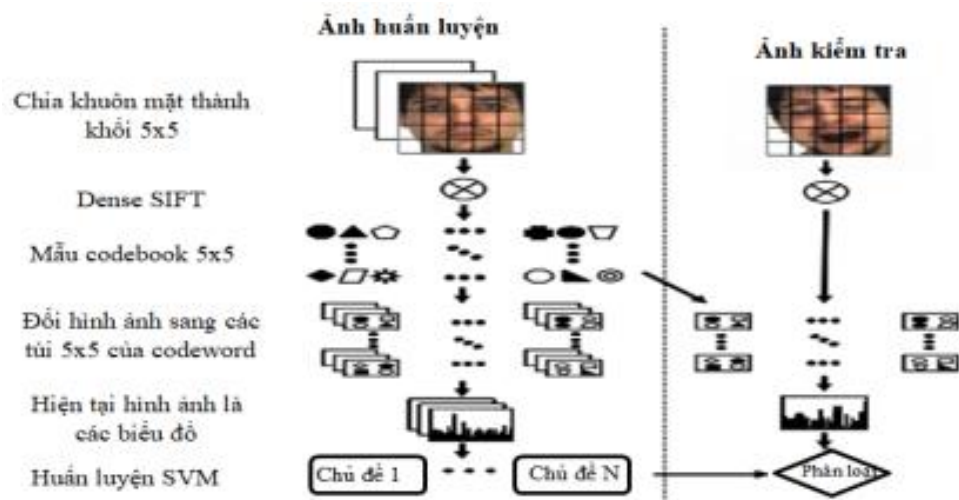
Nhóm của Liao đã đề xuất một thuật toán biểu diễn khuôn mặt không cần canh chỉnh dựa trên phép Mô tả đa điểm chính (Multi Keypoint Descriptor - MKD), trong đó kích thước mô tả khuôn mặt được xác định bằng thành phần có trong ảnh. Làm theo cách này, ta có thể dùng một tập lớn các mô tả để biểu diễn bất kỳ ảnh khuôn mặt kiểm tra nào, dù là một phần hay toàn phần. Nhóm tác giả đã phát triển một phép mô tả điểm chính mới gọi là mẫu tam phân Gabor (Gabor Ternary Patter -

GTP) nhằm giúp nhận dạng khuôn mặt dễ dàng hơn. Thuật toán này sử dụng hiệu quả với ảnh khuôn mặt bị vật thể khá che khuất, ảnh không chính diện, ảnh có đeo phụ kiện, ảnh bị giới hạn góc nhìn, ảnh phơi sáng [9].

b. Bag of Word (Túi từ)

Nhóm tác giả Li và cộng sự [17] đề xuất một thuật toán túi từ (Bag of Word) để nhận dạng khuôn mặt bằng cách chia khuôn mặt thành nhiều khối đặc trưng SIFT, từ đó tính toán và lượng tử hóa vector thành các codeword khác nhau. Cuối cùng, ở mỗi khối ta tính tần số phân phối của mỗi codeword, sau đó nối dài các tần số từ các khối để biểu diễn khuôn mặt.

Nhóm tác giả Li đánh giá rằng các ảnh khuôn mặt đều cùng một loại vật thể, cho nên nếu ta trích xuất đặc trưng khuôn mặt bằng cách thành tập các phần nhỏ thì điều này không đảm bảo rõ thông tin của khuôn mặt. Do đó, nhóm đề xuất thuật toán rút trích đặc trưng khuôn mặt như Hình 2.10 [9].



Hình 2.10. Sơ đồ thuật toán Bag of Word

c. Thuật toán FaceNet

Nhóm tác giả Schroff và cộng sự [12] từ Google đề xuất một thuật toán có tên là FaceNet sẽ học cách ánh xạ từ ảnh khuôn mặt vào không gian Euclide đầy đủ với khoảng cách đo được tương ứng với độ tương đồng của khuôn mặt. Thuật toán

này có thể tạo ra vector đặc trưng và nhúng vào bài toán nhận dạng khuôn mặt, kiểm tra khuôn mặt và phân cụm khuôn mặt. Nhóm tác giả sử dụng Mạng tích chập sâu (Deep Convolution Network - DNN) được huấn luyện để tự tối ưu hóa bài toán. Mạng được huấn luyện sao cho khoảng cách L2 bình phương trong không gian nhúng tương ứng với mức độ tương đồng của khuôn mặt: Mặt cùng người sẽ có khoảng cách nhỏ, mặt khác người sẽ có khoảng cách lớn. FaceNet huấn luyện output thành nhúng đầy đủ 128 chiều sử dụng hàm bộ ba sai số dựa trên LMNN [13], mẫu bộ ba này gồm 2 ảnh cùng loại và 1 ảnh khác loại và hàm lỗi có nhiệm vụ tách ảnh đúng ra khỏi ảnh sai dựa vào biên khoảng cách. Nhóm tác giả sử dụng 2 kiến trúc DNN, một mạng dựa theo mô hình của Zeiler và Fergus, mạng còn lại sử dụng mô hình Inception từ GoogLeNet. Tính đến thời điểm FaceNet ra đời, thuật toán này đã lập nên kỷ lục mới trong nhận dạng khuôn mặt dưới nhiều điều kiện ảnh khác nhau. Tuy nhiên, FaceNet huấn luyện với một số lượng lớn hình ảnh (hơn 200 triệu ảnh của 8 triệu đối tượng), lớn gấp 3 lần so với các bộ dữ liệu hiện có. Để xây dựng bộ dữ liệu lớn như vậy rất khó thực hiện trong các phòng thí nghiệm, học thuật do đòi hỏi kiến trúc máy lớn [19].

2.2.3. Khó khăn

Bài toán nhận dạng mặt người gặp khó khăn lớn khi ảnh mặt người bị tác động từ môi trường xung quanh, ví dụ như góc chụp, mức độ sáng, tối khi lấy ảnh. Chất lượng ảnh đôi khi không được sắc nét do ảnh hưởng bởi nhiễu, phơi sáng. Ngoài ra, một người theo thời gian thì khuôn mặt sẽ biến đổi, do đó phần nào ảnh hưởng đến thông tin trên mặt như ria mép, râu, cũng như một số ảnh mặt người có đeo thêm phụ kiện như mắt kính, đôi khi vị trí của mắt kính vẫn khác nhau, như lệch xuống không đúng vị trí. Bên cạnh đó trạng thái của khuôn mặt cũng ảnh hưởng lớn đến vấn đề nhận dạng như vui, buồn, cười, há miệng, ngậm môi, nhắm mắt hay mở mắt to,... Với sự phát triển của Mạng neural đã hỗ trợ rất lớn cho bài toán nhận dạng mặt người với độ chính xác cao, nhưng bù lại đòi hỏi quá trình huấn luyện cần nhiều dữ liệu, đôi lúc không thể huấn luyện bằng những máy tính học thuật thông thường.

2.3. Kết luận

Trong chương 2, tác giả đã tập chung nghiên cứu cấu trúc mạng CNN, ứng dụng mạng CNN trong nhận dạng khuôn mặt, đưa ra cách nhìn tổng quan về phương pháp nhận dạng khuôn mặt. Trong chương tiếp theo, chúng ta sẽ tìm hiểu về công cụ lập trình được sử dụng, bộ dữ liệu và đưa ra kết quả thực nghiệm.

CHƯƠNG III – CÁC KẾT QUẢ THỰC NGHIỆM

3.1. Công cụ lập trình

Trong quá trình tìm hiểu và xây dựng thực nghiệm mô hình, luận văn lựa chọn ngôn ngữ lập trình Python để xây dựng

3.1.1. Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ngôn ngữ lập trình Python là một trong những ngôn ngữ lập trình được sử dụng nhiều nhất trong việc phát triển trí tuệ nhân tạo. Python được nhiều người lựa chọn để lập trình trí tuệ nhân tạo vì tính phổ biến, dễ học và khả năng tương tích với công nghệ 4.0. Nó có thể được sử dụng một cách trơn tru với các cấu trúc dữ liệu và các thuật toán AI được sử dụng thường xuyên khác.

Ngoài những ứng dụng thực tế nổi bật trên, Python còn được sử dụng để:

- Làm game.
- Máy học với Theano, tensorflow, scikit-learn.
- Khoa học máy tính: Python Opencv, numpys, panda, scipy.
- Lập trình cho bo mạch: Arduino, raspberry pi.

Những tiện ích này đã làm cho Python trở thành ngôn ngữ được ưa chuộng để sáng tạo nên những công nghệ đỉnh cao.

3.1.2. Các chương trình thực hiện

Phần cuối của mạng là một lớp Fully Connected với 128 phần tử và một lớp L2 normalization cho phép thu được một embedding vector. Chính vì vậy, nhờ việc so sánh khoảng cách Euclide của các embedding vector của các khuôn mặt thì có thể xác định các khuôn mặt đó giống hay khác, cụ thể hai khuôn mặt giống nhau có khoảng cách Euclide nhỏ và hai khuôn mặt khác biệt thì có khoảng cách Euclide lớn.

Các chương trình thực hiện như sau:

- Chương trình kiểm tra và cắt khuôn mặt là ảnh người
- Chương trình xoay ảnh
- Chương trình model của mạng CNN
- Chương trình train model để trích xuất ra vector đặc trưng
- Chương trình nhận diện khuôn mặt

Dưới đây là mô tả cấu hình hệ thống và yêu cầu cài đặt:

Môi trường thử nghiệm:

- Processor: Intel(R) Core i3 – 3110M CPU @ 2.40GHz
- Memory (Ram): 12.00 GB
- GPU: NVIDIA GeForce GT 620M
- System type: 64-bit Operating System, x64-based processor
- Windows 10 Pro -2018

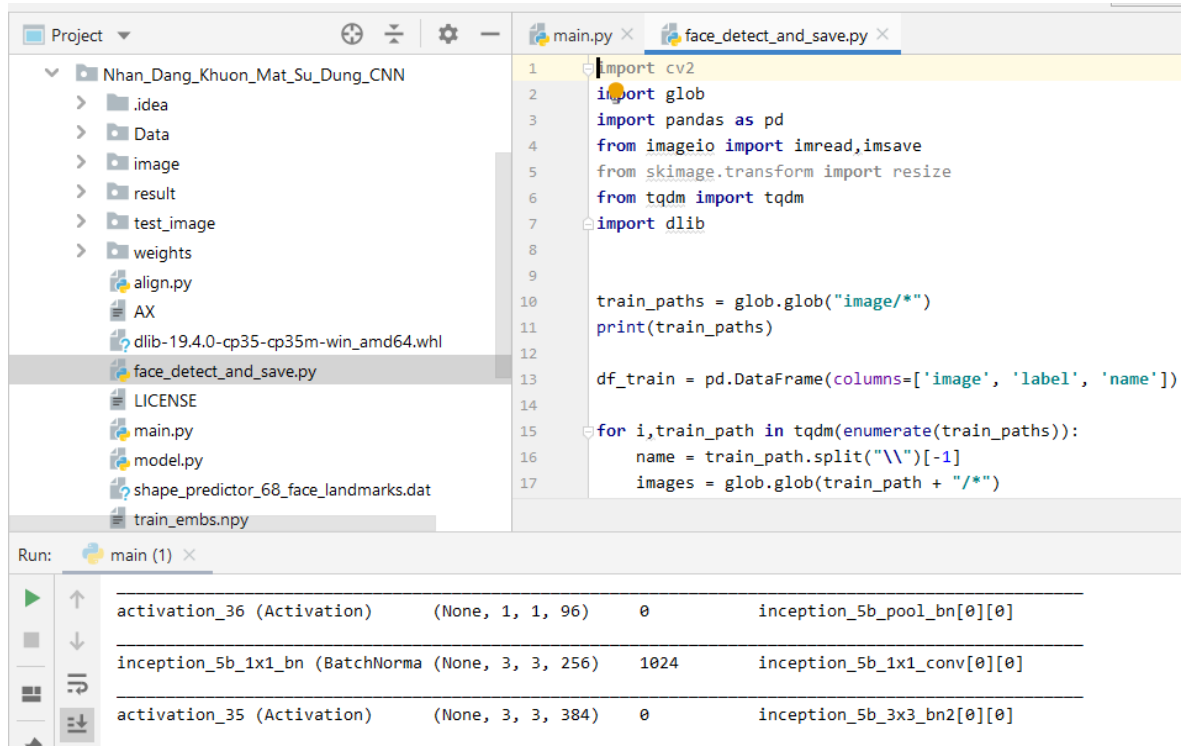
Ngôn ngữ lập trình: Python 3.6.0

Công cụ lập trình: JetBrains PyCharm Community Edition 2018

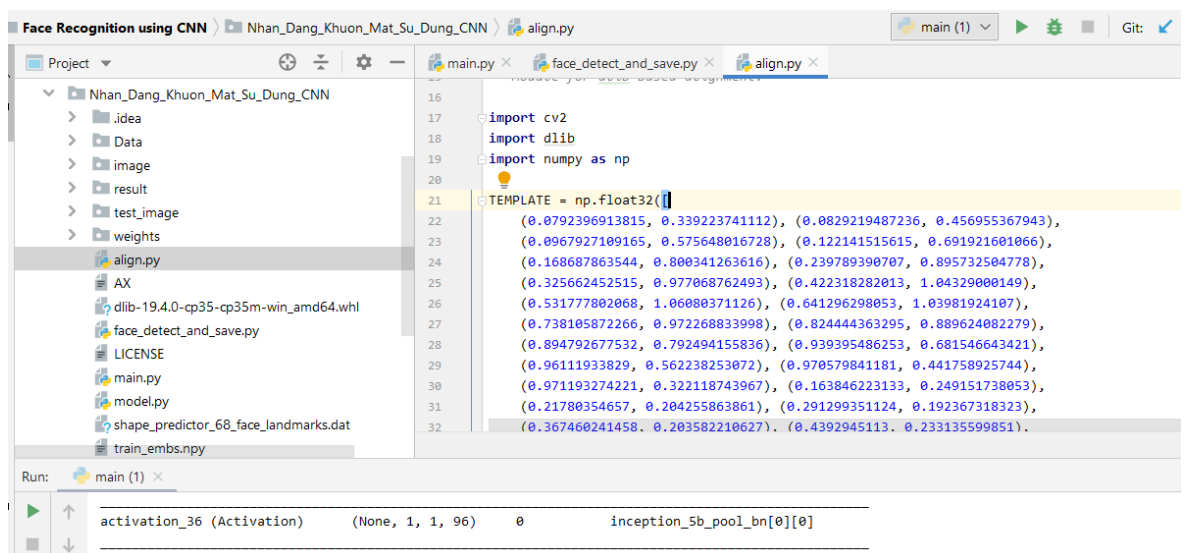
Thư viện và các gói cài đặt chính: Python 3.6, Tensorflow, Dlib, OpenCV2, Opencv 3.4.5, keras 2.2.4, tqdm 4.31.1, pandas 0.23.4, scipy 1.2.0, Model mạng CNN nn4.small2, Model shape_predictor_68_face_landmarks, Sparqlwrapper ...

Một số chương trình:

- Chương trình nhận dạng và cắt khuôn mặt trong hình



- Chương trình xoay ảnh và tính đặc trưng ảnh



- Chương trình tính khoảng cách Euclide

```
label2idx = []

for i in tqdm(range(len(train_paths))):
    label2idx.append(np.asarray(df_train[df_train.label == i].index))

match_distances = []
for i in range(nb_classes):
    ids = label2idx[i]
    distances = []
    for j in range(len(ids) - 1):
        for k in range(j + 1, len(ids)):
            distances.append(distance.euclidean(train_embs[ids[j]].reshape(-1), train_embs[ids[k]].reshape(-1)))
    match_distances.extend(distances)

unmatch_distances = []
for i in range(nb_classes):
    ids = label2idx[i]
    distances = []
    for j in range(10):
        idx = np.random.randint(train_embs.shape[0])
        while idx in label2idx[i]:
            idx = np.random.randint(train_embs.shape[0])
        distances.append(distance.euclidean(train_embs[ids[np.random.randint(len(ids))]].reshape(-1), train_embs[idx].reshape(-1)))
    unmatch_distances.extend(distances)
```

- Chương trình test ảnh

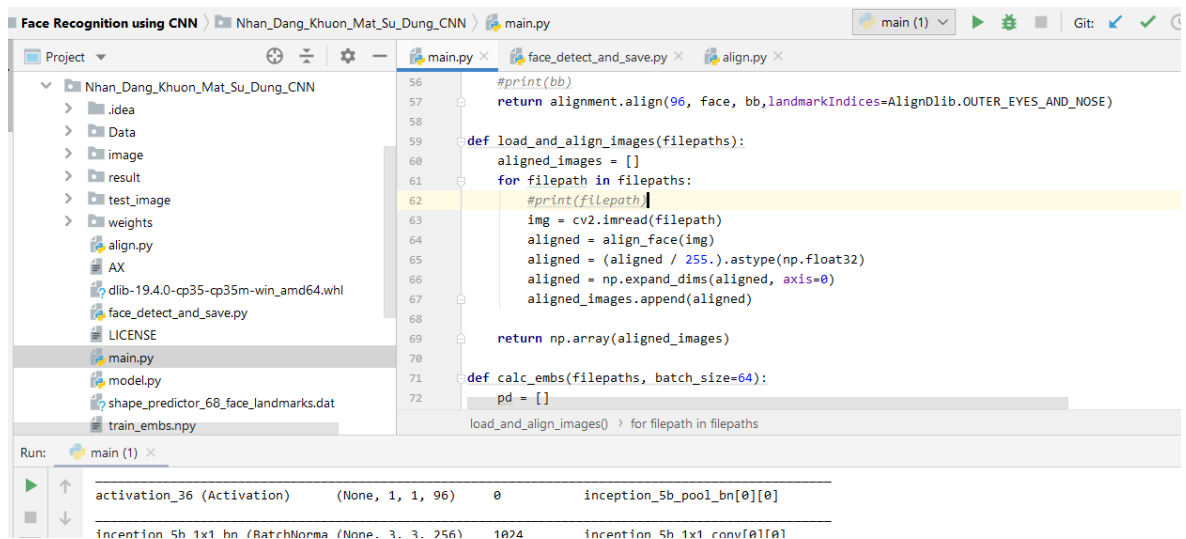
```
test_paths = glob.glob("test_image/*.jpg")
for path in test_paths:
    test_image = cv2.imread(path)
    show_image = test_image.copy()

    hogFaceDetector = dlib.get_frontal_face_detector()
    faceRects = hogFaceDetector(test_image, 0)

    faces = []
    for faceRect in faceRects:
        x1 = faceRect.left()
        y1 = faceRect.top()
        x2 = faceRect.right()
        y2 = faceRect.bottom()
        face = test_image[y1:y2, x1:x2]

        faces.append(face)
```

- Chương trình chính



3.2. Bộ dữ liệu

Dữ liệu huấn luyện là tập dữ liệu được thu thập theo các hình thức: Thu thập từ nguồn ảnh trên mạng bằng công cụ tìm kiếm ảnh của google, và được thu thập bằng thư viện Sparqlwrapper hoặc được sử dụng trong bộ dữ liệu được sử dụng thành công trong một cuộc thi về nhận diện người nổi tiếng do AIVIVN tổ chức hồi tháng 3/2019.

Dữ liệu thô ban đầu được thu thập về sẽ được chức năng nhận diện và cắt sát khuôn mặt trong ảnh để làm đầu vào cho quá trình trích chọn đặc trưng. Quá trình nhận diện khuôn mặt được sử dụng bởi mạng CNN cụ thể trong nội dung luận văn sử dụng mạng nn4.small2 đây là một mạng mà có số lượng tham số lớn với vector nhúng 128 chiều[19] trong mô hình mạng được huấn luyện trước bằng các tập dữ liệu lớn như Facescrub [20] và CASIA-Webface [20].

Cấu tạo dữ liệu khi huấn luyện mô hình như sau:

```

|——image
|  |——Phuong Thuy
|  |  |——1.jpg
  
```

```

| | | |——2.jpg
| | | |——3.jpg
| | | |——4.jpg
| | | |——5.jpg
| |——Tien Dung
| | | |——1.jpg
| | | |——2.jpg
| | | |——3.jpg
| | | |——4.jpg
| | | |——5.jpg
| |——Cu Tit
| | | |——1.jpg
| | | |——2.jpg
| | | |——3.jpg
| | | |——4.jpg
| | | |——5.jpg

```

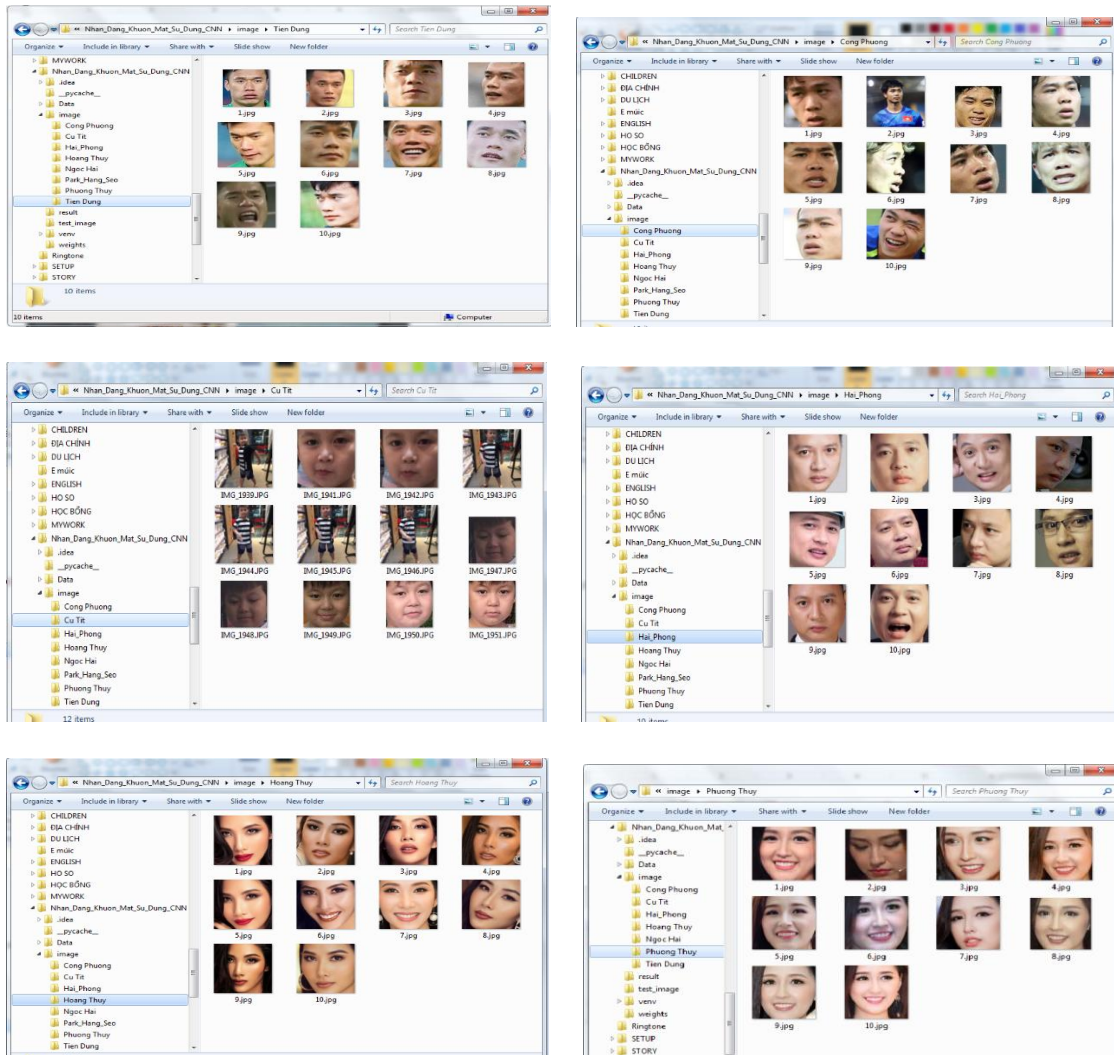
Dữ liệu ảnh thô được lựa trong thư mục Image, và được sắp xếp các ảnh theo thư mục là nhãn của chúng, sau khi thực hiện quá trình xử lý ảnh thô thì ảnh sau xử lý sẽ được lưu đè vào thư mục này với số thứ tự và nhãn như ban đầu.

Do mô hình nn4.small được huấn luyện dựa trên các đầu vào là khuôn mặt thẳng, trong khi thực tế dữ liệu thực tế lại ở mọi chiều xoay do đó để xoay ảnh về dạng thẳng và có kích thước 96x96 thì tôi sử dụng một chương trình xoay và tính đặc trưng của ảnh sử dụng model `shape_predictor_68_face_landmarks` được định nghĩa trong đoạn chương trình `align.py`.

Sau khi quá trình train được diễn ra để tìm ra embedding vector của các khuôn mặt, thì các vector này được lưu dưới dạng file train_embs.py để sử dụng trong quá trình nhận dạng sau này.

Bộ dữ liệu huấn luyện

Bộ dữ liệu huấn luyện bao gồm 72 ảnh với 7 nhãn dữ liệu tương ứng với 7 đối tượng người khác nhau. Cụ thể, có 10 ảnh của Bùi Tiến Dũng, 10 ảnh của Nguyễn Công Phượng, 10 ảnh của Nguyễn Hải Phong, 10 ảnh của Quế Ngọc Hải, 10 ảnh của Hoàng Thùy, 10 ảnh của Mai Phương Thúy và 12 ảnh của cậu bé Cu Tít. Hình 3.1 minh họa các tập ảnh huấn luyện được sử dụng trong luận văn này.

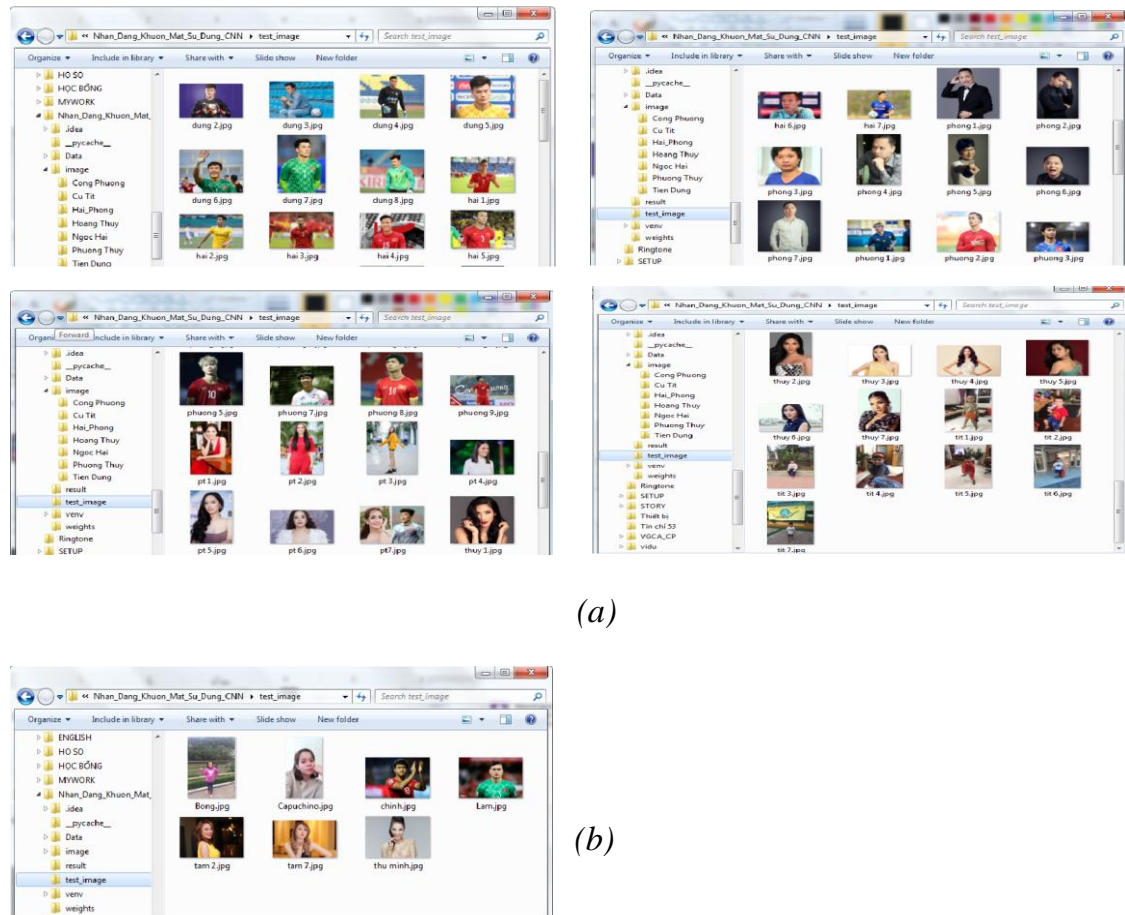


Hình 3.1. Dữ liệu huấn luyện

Bộ dữ liệu kiểm tra

Để đánh giá chất lượng hoạt động của chương trình, luận văn có sử dụng bộ dữ liệu kiểm tra bao gồm 49 ảnh có nhãn giống với dữ liệu huấn luyện, trong đó có 7 ảnh được lấy từ cùng đối tượng Bùi Tiến Dũng, nhưng với các góc chụp và bối cảnh chụp khác so với ảnh huấn luyện. Tương tự, có 7 được lấy từ đối tượng Nguyễn Công Phượng, có 7 được lấy từ đối tượng Nguyễn Hải Phong, 7 ảnh của Quế Ngọc Hải, 7 ảnh của Hoàng Thùy, 7 ảnh của Mai Phương Thúy và 7 ảnh của cậu bé Cu Tít

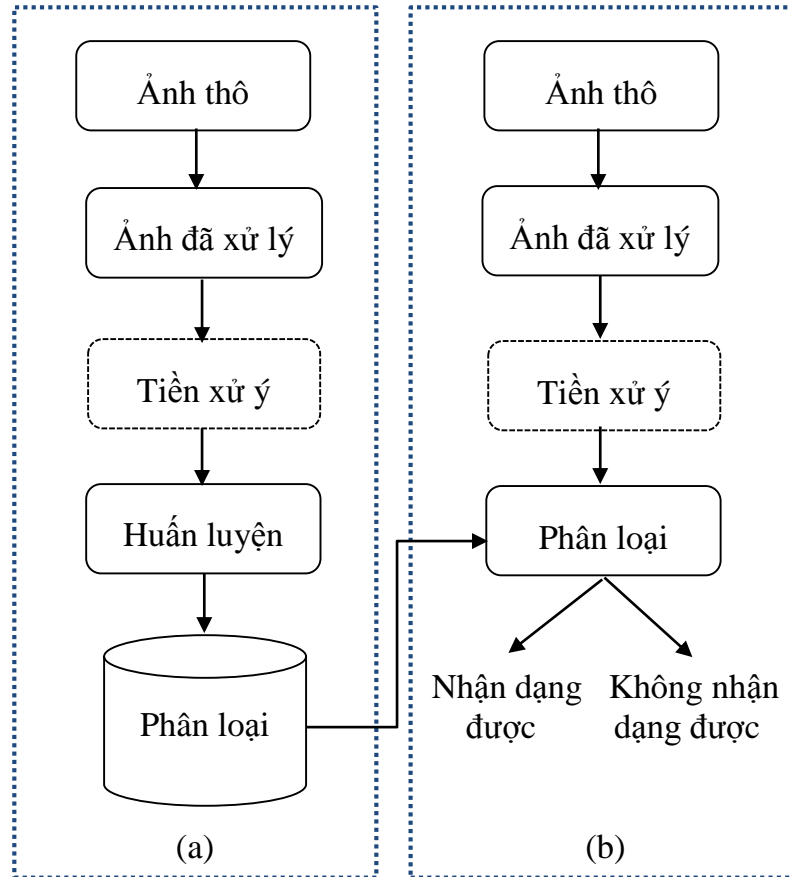
Ngoài các ảnh kiểm tra lấy từ cùng các đối tượng của bộ dữ liệu huấn luyện, luận văn cũng lấy thêm các ảnh của các đối tượng khác với đối tượng của dữ liệu huấn luyện. Cụ thể, có 7 ảnh của các đối tượng khác, được gán nhãn là “unknown”. Tổng cộng, 56 ảnh có nhãn khác nhau đã được sử dụng trong bộ dữ liệu kiểm tra. Minh hoạ về bộ dữ liệu kiểm tra được trình bày trong Hình 3.2.



Hình 3.2. Bộ dữ liệu kiểm tra: (a) Dữ liệu kiểm tra có nhãn giống bộ dữ liệu huấn luyện, (b) Dữ liệu kiểm tra có nhãn “unknown”

3.3. Quá trình thực nghiệm

Mô hình nhận dạng khuôn mặt sử dụng mạng CNN



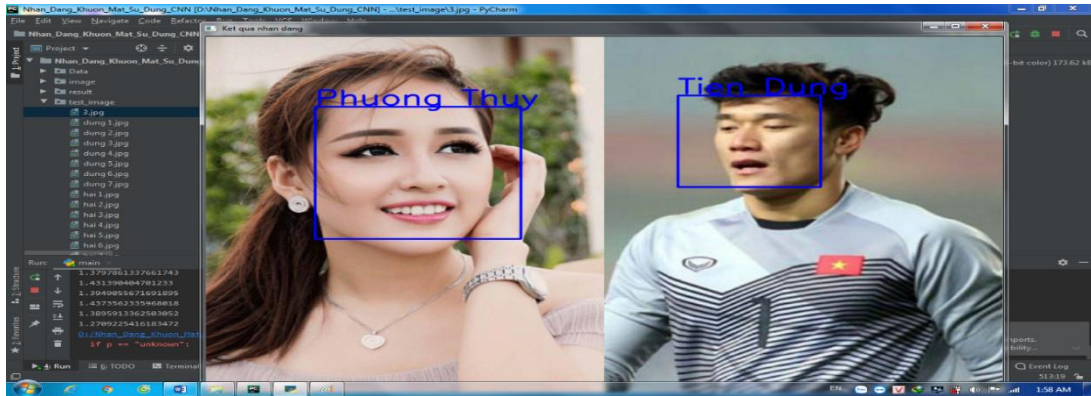
Hình 3.3. Mô tả mô hình nhận dạng ảnh bao gồm giai đoạn (a) huấn luyện và giai đoạn (b) nhận dạng

Hình 3.3 mô tả mô hình nhận dạng khuôn mặt người dựa trên mô hình CNN. Mô hình này được thực hiện thành 2 giai đoạn: (a) giai đoạn huấn luyện và (b) giải đoạn nhận dạng. Trong giai đoạn huấn luyện, ảnh thô đầu vào được xử lý để đưa về định dạng chuẩn, sau đó ảnh này được đưa qua mô hình tính đặc trưng, sau đó được huấn luyện và đưa ra bộ phân loại là các vector 128 chiều tương ứng với mỗi ảnh. Trong giai đoạn nhận dạng, những ảnh cần nhận dạng ban đầu được xử lý thô và qua quá trình tiền xử lý và đến khâu phân loại sử dụng bộ phân loại từ giai đoạn huấn luyện để xác định đúng ảnh của đối tượng. Tại khâu phân loại này thực chất là

so sánh khoảng cách Euclide của ảnh đã xử lý từ giai đoạn huấn luyện với ảnh cần kiểm tra trong giai đoạn nhận dạng.

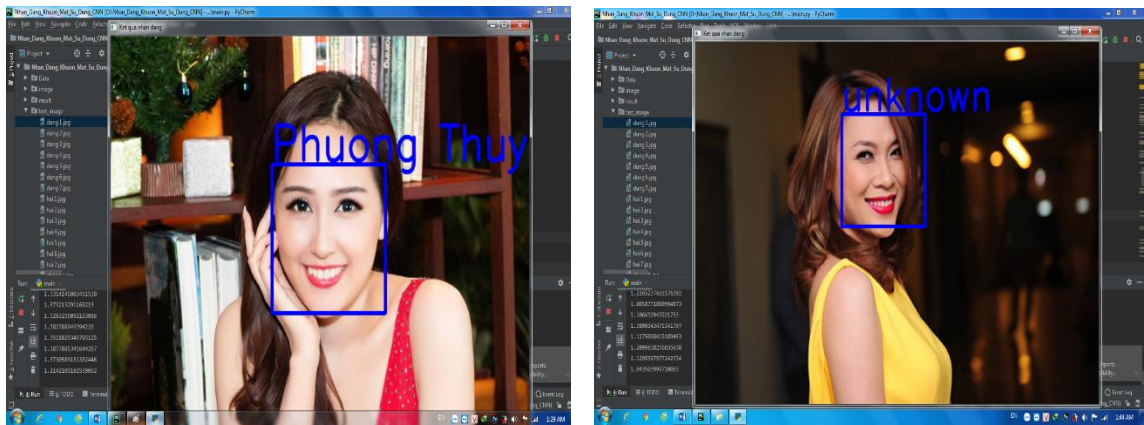
3.4. Kết quả thực nghiệm

Nhận dạng được hai khuôn mặt có trong dữ liệu huấn luyện.



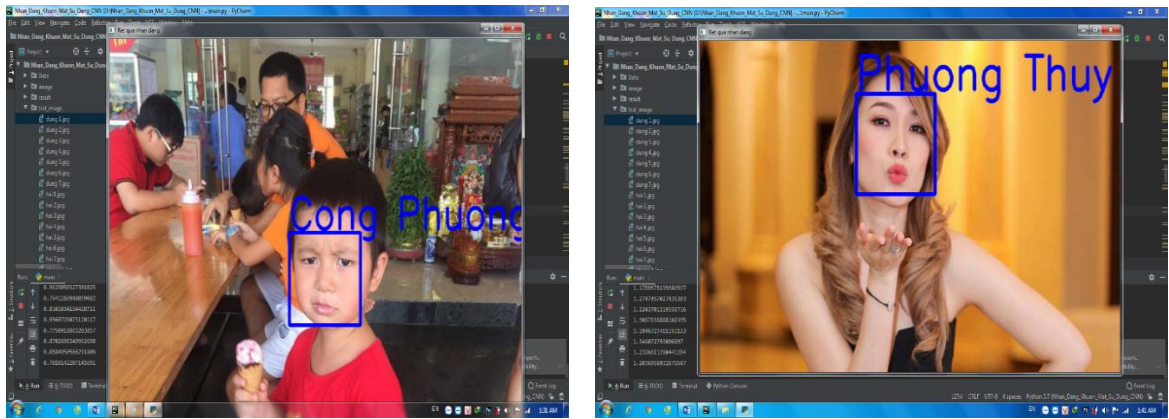
Hình 3.4. Kết quả nhận dạng đúng hai khuôn mặt có trong dữ liệu train

Nhận dạng được một khuôn mặt có trong dữ liệu huấn luyện, một khuôn mặt báo unknown do không có trong dữ liệu huấn luyện.



Hình 3.5. Kết quả sau khi chạy chương trình nhận dạng ảnh cần nhận dạng trong thư mục Test Image

Nhận dạng nhầm một khuôn mặt có trong dữ liệu huấn luyện, 1 khuôn mặt không có trong dữ liệu huấn luyện.



Hình 3.6. Kết quả nhận dạng nhảm một khuôn mặt có trong dữ liệu huấn luyện, một khuôn mặt không có trong dữ liệu huấn luyện

Bảng 3.1 biểu diễn ma trận nhầm lẫn của kết quả chạy hệ thống nhận dạng khuôn mặt bằng dữ liệu kiểm tra. Trong ma trận này, mỗi hàng biểu diễn số ảnh với nhãn đầu vào cần phải kiểm tra (gán nhãn thật), mỗi cột biểu diễn số ảnh mà hệ thống đã gán nhãn cho các dữ liệu đầu vào đó (nhãn do hệ thống gán cho dữ liệu).

Bảng 3.1. Ma trận nhầm lẫn biểu diễn kết quả nhận diện khuôn mặt từ dữ liệu kiểm tra

	Tiến Dũng	Ngọc Hải	Công Phượng	Hải Phong	Hoàng Thùy	Phương Thúy	Cu Tít	Unknown
Tiến Dũng	7	0	0	0	0	0	0	0
Ngọc Hải	1	5	0	0	0	0	0	1
Công Phượng	1	0	6	0	0	0	0	0
Hải Phong	0	0	0	6	0	0	0	1
Hoàng Thùy	0	0	0	0	6	1	0	0
Phương Thúy	0	0	0	0	0	6	0	1
Cu Tít	0	0	1	0	0	0	6	0
Unknown	0	0	0	0	0	1	1	5

Nhận xét:

Từ kết quả trong Bảng 3.1. *Ma trận nhầm lẫn biểu diễn kết quả nhận diện khuôn mặt từ dữ liệu kiểm tra* cho thấy việc nhận dạng khuôn mặt được thực hiện rất tốt, với tỷ lệ đúng tương đối cao lên đến 83.92%. Tuy nhiên việc thay đổi ngưỡng trong quá trình nhận dạng cũng làm thay đổi đáng kể tới kết quả của bài toán. Trong khi nhận dạng vẫn có trường hợp nhận dạng nhầm diễn hình như Hình 3.6.

Trong quá trình nhận dạng khuôn mặt, tốc độ của máy tính chưa đủ đáp ứng trong quá trình huấn luyện. Dẫn đến thời gian lâu và chưa hiệu quả.

KẾT LUẬN

Luận văn đã trình bày một hướng tiếp cận đang là xu thế của thế giới và được ứng dụng rất nhiều trong cuộc sống và xã hội.

1. Luận văn đã thực hiện được kết quả sau:

Luận văn đã trình bày một cách tổng quát về mô hình mạng nơron và ứng dụng mạng nơron tích chập trong bài toán nhận dạng khuôn mặt. Trình bày có hệ thống các loại mạng nơron và các giải pháp học trong mô hình mạng nơron. Bên cạnh đó, luận văn còn nghiên cứu và phân tích việc sử dụng mạng nơron tích chập để giải quyết bài toán nhận dạng khuôn mặt. Nghiên cứu cấu trúc hoạt động của bộ công cụ lập trình Python, thư viện Tensorflow, Google colab để giải quyết bài toán nhận dạng khuôn mặt. Luận văn cũng đã xây dựng bài toán nhận dạng khuôn mặt bằng bộ dữ liệu là những người nổi tiếng tại Việt Nam, bộ dữ liệu được lấy trực tiếp từ công cụ Google download Images.

2. Những tồn tại và hướng phát triển

Những kết quả khả quan từ việc ứng dụng mạng nơron tích chập trong bài toán nhận dạng khuôn mặt đã chứng tỏ rằng đây là một mô hình có thể ứng dụng hiệu quả đối với các bài toán này. Tuy nhiên bài toán vẫn còn nhận dạng chưa đúng đối tượng và còn nhận dạng nhầm đối tượng. Việc phân bổ ngưỡng trong bài toán cũng ảnh hưởng tới kết quả của luận văn.

Việc cải tiến mô hình CNN và ứng dụng mô hình vào thực tế vẫn đang được các nhà nghiên cứu quan tâm và xây dựng. Luận văn này tuy đạt được một số kết quả nêu trên, nhưng luận văn còn nhiều hạn chế do điều kiện về mặt thời gian và phạm vi nghiên cứu của đề tài. Vì vậy, hướng nghiên cứu tiếp theo của học viên là:

Nghiên cứu tăng độ chính xác của cho việc nhận dạng khuôn mặt qua các kỹ thuật xử lý ảnh cao hơn trước khi đưa vào huấn luyện mô hình CNN. Bên cạnh đó, nghiên cứu thêm về mô hình CNN để có thể tăng độ chính xác cho việc nhận dạng ảnh từ đó ứng dụng trong thực tế. Có thể phát triển ứng dụng trên bộ dữ liệu đầy đủ và chi tiết trong nhiều lĩnh vực hơn.

TÀI LIỆU THAM KHẢO

- [1] Đỗ Duy Cốp, "Nghiên cứu phương pháp nhận dạng ảnh mặt người và Ứng dụng", Luận văn thạc sỹ khoa học máy tính, Đại học Thái Nguyên, 2014.
- [2] Giáo trình Mạng neural, Tác giả: Phan Văn Hiền – Trường Đại học Bách khoa Đà Nẵng, 2013.
- [3] Phạm Thế Bảo và Trịnh Tấn Đạt, "Dùng đặc trưng gabor kết hợp adaboost và kmeans trong bài toán nhận dạng mặt người", Tạp chí Khoa học ĐHSP Thành phố Hồ Chí Minh , Số 43, 2013.
- [4] Võ Phúc Nguyên, "Nhận dạng ảnh mặt người sử dụng mạng nơron", Tạp chí Khoa học & Công nghệ, Số 64, trang 52-57, 2014
- [5] Võ Hoàng Trọng, Nguyễn Thanh Thủy, "Nhận dạng mặt người dựa trên thông tin mặt người không đầy đủ", Luận văn tốt nghiệp đại học, Đại học Khoa học Tự nhiên, Thành phố Hồ Chí Minh, 2017.
- [6]. B. D. Ripley. Pattern recognition and neural networks. Cambridge university prees, 1996.
- [7]. Breiman. Random forests. Machine Learning 45(1):5–32, 2001.
- [8]. Chellappa, R., Wilson, C.L., and Sirohey, S. (1995), "Human and Machine Recognition of Faces: A Survey", In Proc. of IEEE Conf., 83, 705-740.
- [9] F. Schroff, D. Kalenichenko and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815-823, 2015.
- [10]. G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linearregions of deep neural networks. In NIPS, 2014.
- [11]. J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
- [12] Kresimir Delac, Mislav Grgic, " Face Recognition ", I-tech Education and Publishing chapters, ISBN 978-3-902613-03-5, Austria, 2007.
- [13] O. M. Parkhi, A. Vedaldi and A. Zisserman, "Deep face recognition," British Machine Vision Conference, vol. 1, no. 3, 2015.

- [14]. Samal, A., Iyengar, P.A. (1992), "Automatic Recognition and Analysis of Human Faces and Facial Expressions", Pattern Recognition, 25, pp.65-77
- [15] S. Liao, A. K. Jain and S. Z. Li, "Partial face recognition: Alignment-free approach," IEEE Transactions on Pattern Analysis and Machine Intelligence 35.5 , pp. 1193-1205, 2013.
- [16] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," IEEE transactions on image processing, vol. 19(6), pp. 1635-1650, 2010.
- [17] W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld, "Face Recognition - A Literature Survey", ACM Computing Surveys, Vol. 35 (No. 4), 2003.
- [18] Z. Li, J.-i. Imai and M. Kaneko, "Robust face recognition using block-based bag of words.," Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, pp. 1285-1288, 2010.
- [19] <https://cmusatyalab.github.io/openface/models-and-accuracies/#running-the-lfw-experiment>.
- [20] <http://vintage.winklerbros.net/facescrub.html>.
- [21] <https://techinsight.com.vn/nhan-dien-khuon-mat-va-ung-dung-thuc-te/>
- [22] Nguyễn Thành An và Nguyễn Phát Tài (ĐH Khoa học tự nhiên - ĐH Quốc gia TP.HCM) giành giải nhất lĩnh vực công nghệ thông tin Giải thưởng sinh viên nghiên cứu khoa học Eureka 2017 với đề tài "Tổng hợp và tìm kiếm trên video dựa vào phát hiện và nhận biết mặt người"