

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

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2019

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **TS. NGUYỄN ĐÌNH HÓA**

Phản biện 1:

Phản biện 2:

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: giờ ngày tháng năm

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

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 nơron, mô hình mạng nơron 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 nơron nhân tạo bao gồm cấu trúc cơ bản và một số ứng dụng trong thực tế của mạng nơron nhân tạo.

Chương 2: Trình bày chi tiết về mạng nơron tích chập, ý nghĩa cùng ứng dụng của mạng nơron 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 nơron 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 nơron 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 [3]. 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.

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.

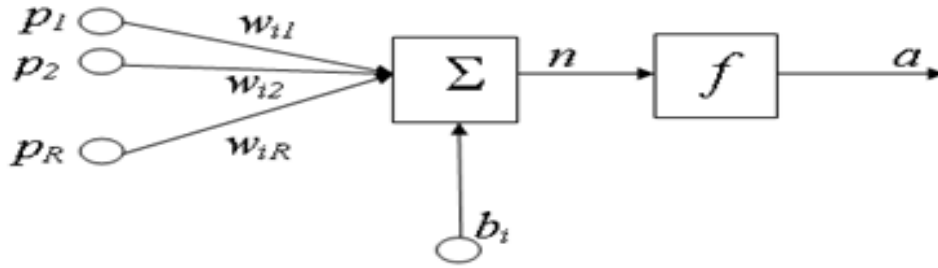
Các rẽ của noron được chia làm hai loại: rẽ đầu vào nhận thông tin từ các noron khác qua axon và rẽ đầu ra đưa thông tin qua axon tới các noron khác. Một noron có thể có nhiều rẽ đầu vào, nhưng chỉ có một rẽ đầu ra. Bởi vậy nếu coi noron như một khâu điều khiển thì nó chính là khâu có nhiều đầu vào và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

1.1.3. Mạng noron nhân tạo

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

Một noron nhân tạo phản ánh các tính chất cơ bản của noron sinh học. Mỗi noron nhân tạo là một đơn vị xử lý thông tin làm cơ sở cho hoạt động của một mạng noron. 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 noron nhân tạo:



Hình 1.2. Mô hình mạng noron

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 [3] 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

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):
- b. Hàm nhị phân (Binary step function, Hard limit function)
- c. Hàm sigmoid (Sigmoid function (logistic))
- d. Hàm sigmoid lưỡng cực (Bipolar sigmoid function (tansig))

1.2. Cấu trúc mạng noron

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 noron nhân tạo là một công cụ mô phỏng cách bộ não hoạt động nhằm thực hiện các nhiệm vụ nào đó.

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

- ✓ Mạng tự kết hợp - autoassociative
- ✓ Kiến trúc truyền thẳng - feedforward architecture
- ✓ Kiến trúc phản hồi - Feedback architecture

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. Học có giám sát bao gồm:

- Phân loại (classification)
- Hồi quy (regression)

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. Hai dạng học không giám sát phổ biến là:

- Phân cụm (clustering)
- Luật học kết hợp (association rule learning)

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.

- Mạng nơron 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.

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

1.5. Kết luận

Trong chương này, chúng ta tìm hiểu các khái niệm, cấu trúc và các luật học của mạng nơron nói chung, đồng thời đi phân loại mạng nơron. Trong chương tiếp theo tác giả sẽ trình bày mạng nơron 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 nơron 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.

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 \rightarrow cạnh \rightarrow hình dạng \rightarrow mặt \rightarrow 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

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

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:

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

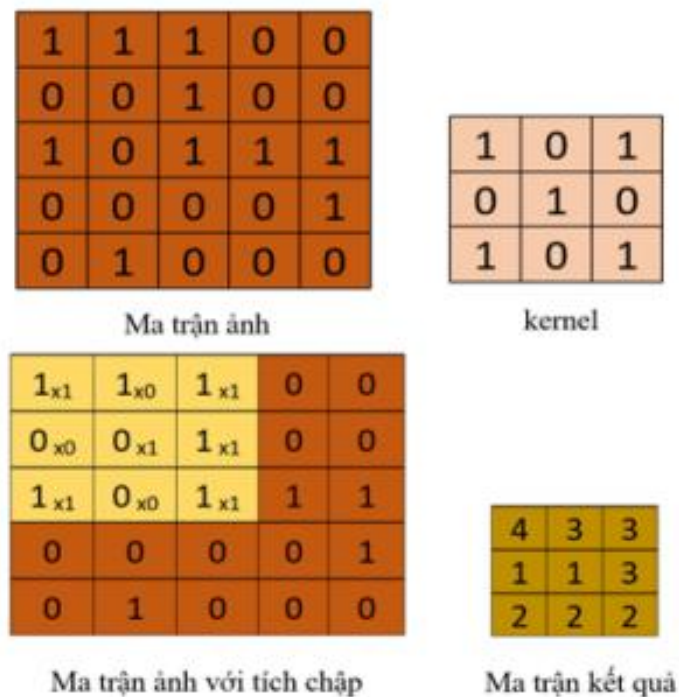
Đối với Tích chập rời rạc, với hàm số phức f và g xác định trên tập số nguyên Z , thì tích chập được định nghĩa

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

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.



Hình 2.2. Minh họa tích chập trên ma trận ảnh

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.

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.

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 lẫn, 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.

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 lầm, 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.

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.

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.

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)

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.

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

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.

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,

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 [9].

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)

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 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.

b. Bag of Word (Túi từ)

Nhóm tác giả Li và cộng sự 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.

c. Thuật toán FaceNet

Nhóm tác giả Schroff và cộng sự 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, 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 thiết bị, học thuật do đòi hỏi kiến trúc máy lớn.

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, nheo 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

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 bộ 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

```

1 import cv2
2 import glob
3 import pandas as pd
4 from imageio import imread, imsave
5 from skimage.transform import resize
6 from tqdm import tqdm
7 import dlib
8
9
10 train_paths = glob.glob("image/*")
11 print(train_paths)
12
13 df_train = pd.DataFrame(columns=['image', 'label', 'name'])
14
15 for i, train_path in tqdm(enumerate(train_paths)):
16     name = train_path.split("\\")[-1]
17     images = glob.glob(train_path + "/*")

```

Run:	main (1) ×
activation_36 (Activation)	(None, 1, 1, 96) 0 inception_5b_pool_bn[0][0]
inception_5b_1x1_bn (BatchNorma	(None, 3, 3, 256) 1024 inception_5b_1x1_conv[0][0]
activation_35 (Activation)	(None, 3, 3, 384) 0 inception_5b_3x3_bn2[0][0]

- Chương trình xoay ảnh và tính đặc trưng ảnh

```

16
17 import cv2
18 import dlib
19 import numpy as np
20
21 TEMPLATE = np.float32([
22     (0.0792396913815, 0.339223741112), (0.0829219487236, 0.456955367943),
23     (0.0967927109165, 0.575648016728), (0.122141515615, 0.691921601066),
24     (0.168687863544, 0.800341263616), (0.239789390707, 0.895732504778),
25     (0.325662452515, 0.977068762493), (0.422318282013, 1.04329000149),
26     (0.531777802068, 1.06080371126), (0.641296298053, 1.03981924107),
27     (0.738105872266, 0.972268833998), (0.824444363295, 0.889624082279),
28     (0.894792677532, 0.792494155836), (0.939395486253, 0.681546643421),
29     (0.96111933829, 0.562238253072), (0.970579841181, 0.441758925744),
30     (0.971193274221, 0.322118743967), (0.163846223133, 0.249151738053),
31     (0.21780354657, 0.204255863861), (0.291299351124, 0.192367318323),
32     (0.367460241458, 0.203582210627), (0.4392945113, 0.233135599851),

```

Run:	main (1) ×
activation_36 (Activation)	(None, 1, 1, 96) 0 inception_5b_pool_bn[0][0]
inception_5b_1x1_bn (BatchNorma	(None, 3, 3, 256) 1024 inception_5b_1x1_conv[0][0]
activation_35 (Activation)	(None, 3, 3, 384) 0 inception_5b_3x3_bn2[0][0]

- 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), tr
            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(i
        unmatch_distances.extend(distances)

```

- Chương trình test ảnh

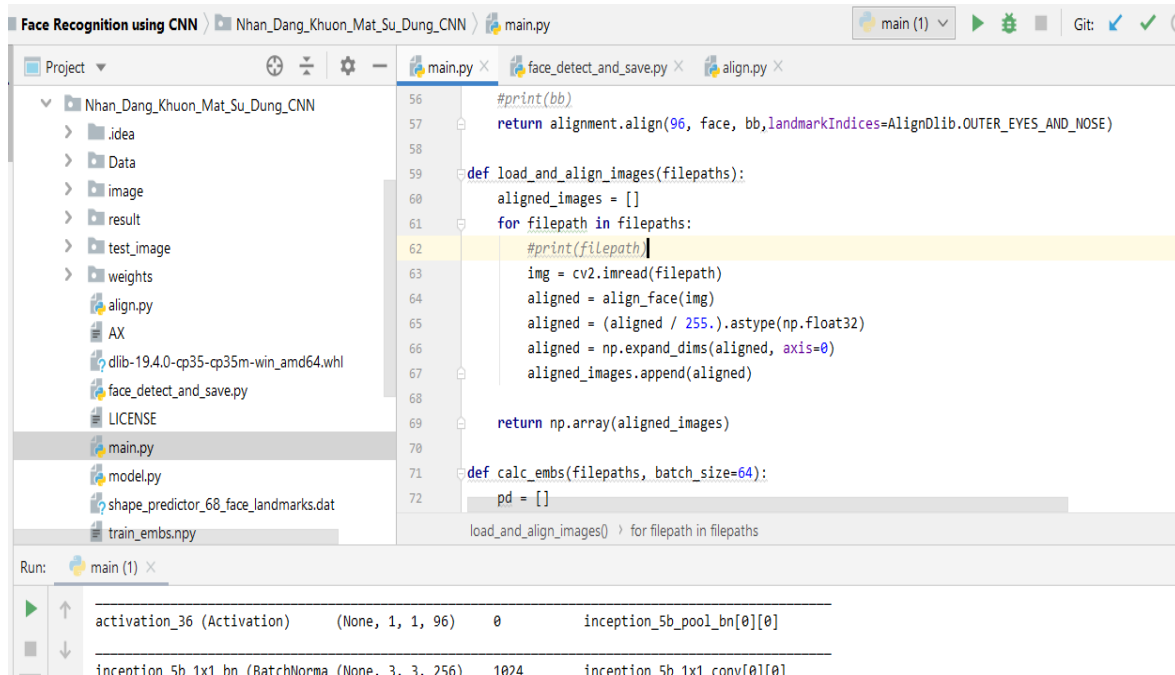
```
test_paths = glob.glob("test_image/*.jpg")
for path in test_paths:
    test_image = cv2.imread(path)
    show_image = test_image.copy()

    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



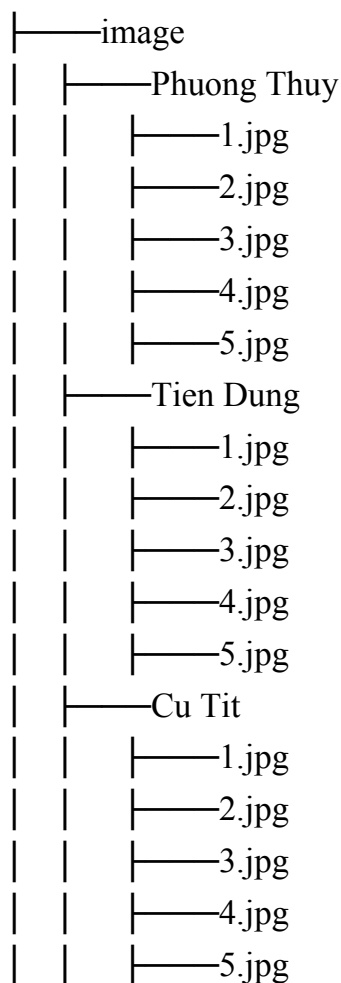
3.2. Bộ dữ liệu

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[20] trong mô hình

mạng được huấn luyện trước bằng các tập dữ liệu lớn như Facescrub [21] và CASIA-Webface [21].

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



Dữ liệu ảnh thô được lựa trong thư mục Image, và được sắp xếp các ảnh theo thư mục là nhân 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 nhân như ban đầu.

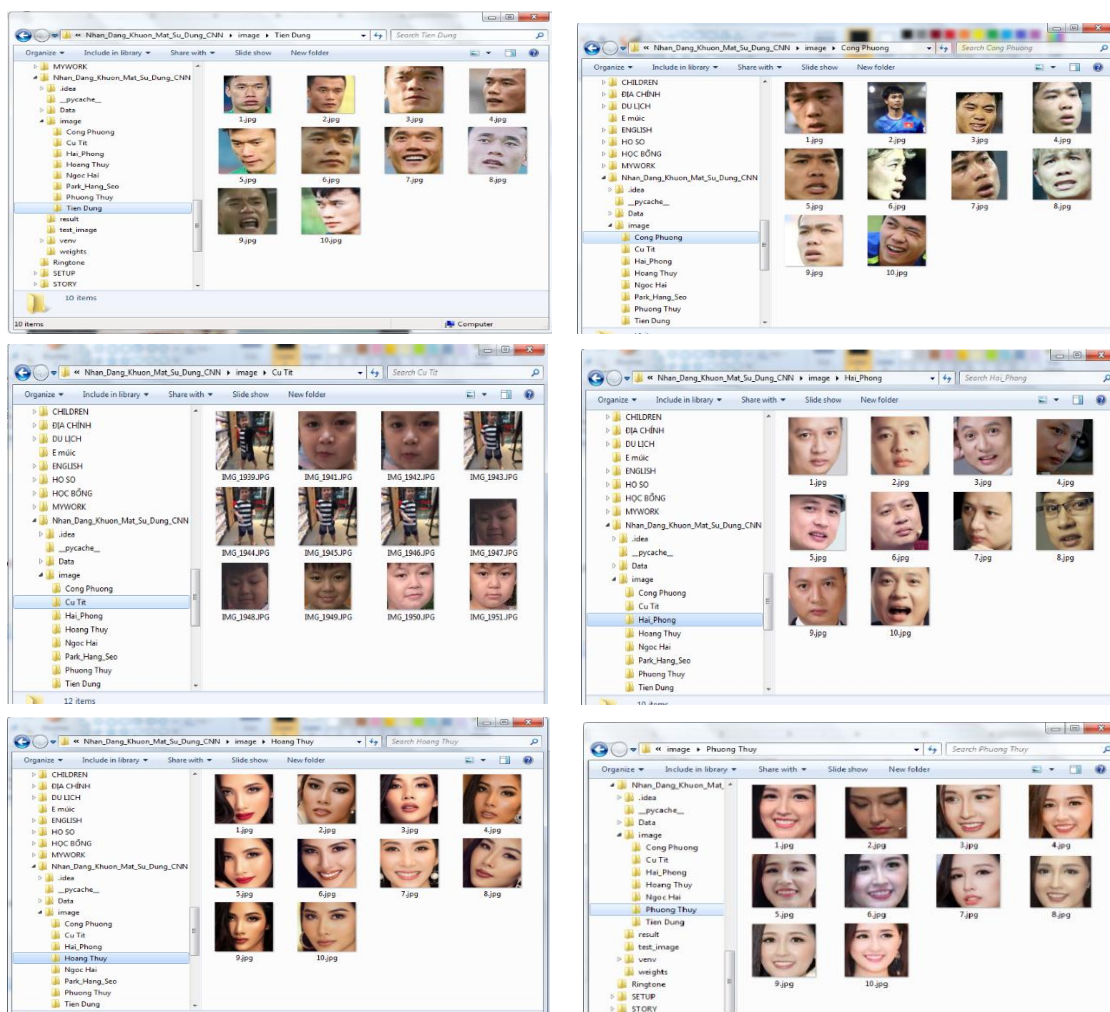
Do mô hình nn4.small được huấn luyện dựa trên các đầu vào là khuôn mặt thẳng, trong khi thực tế dữ liệu thực tế lại ở mọi chiều xoay do đó để xoay ảnh về dạng thẳng và có kích thước 96x96 thì tôi sử dụng một chương trình xoay và tính đặc trưng của ảnh sử dụng model `shape_predictor_68_face_landmarks` 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 nhân 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 họa về bộ dữ liệu kiểm tra được trình bày trong Hình 3.2.

```

graph TD
    subgraph (a)
        A1[Ảnh thô] --> A2[Ảnh đã xử lý]
        A2 --> A3[Tiền xử lý]
        A3 --> A4[Huân luyện]
        A4 --> A5[(Phân loại)]
    end
    subgraph (b)
        B1[Ảnh ban đầu] --> B2[Ảnh đã xử lý]
        B2 --> B3[Tiền xử lý]
        B3 --> B4[Phân loại]
        B4 --> B5[Nhận dạng được]
        B4 --> B6[Không nhận dạng được]
    end
    A5 --> B4

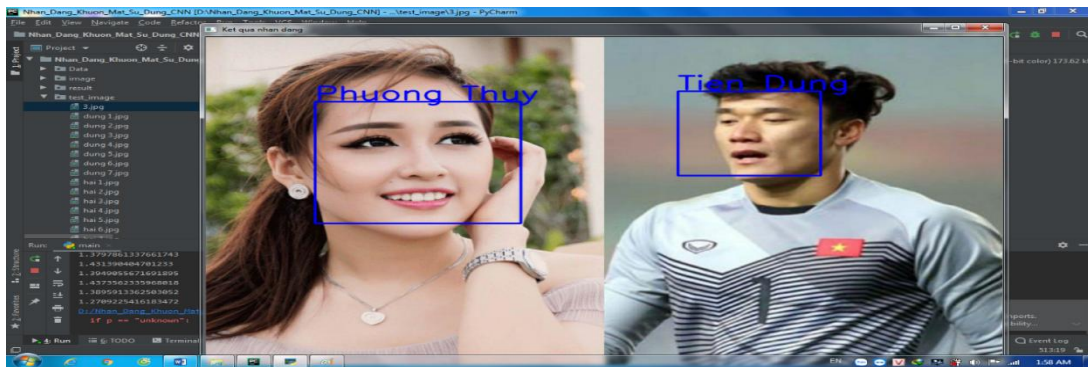
```

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) giai 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.

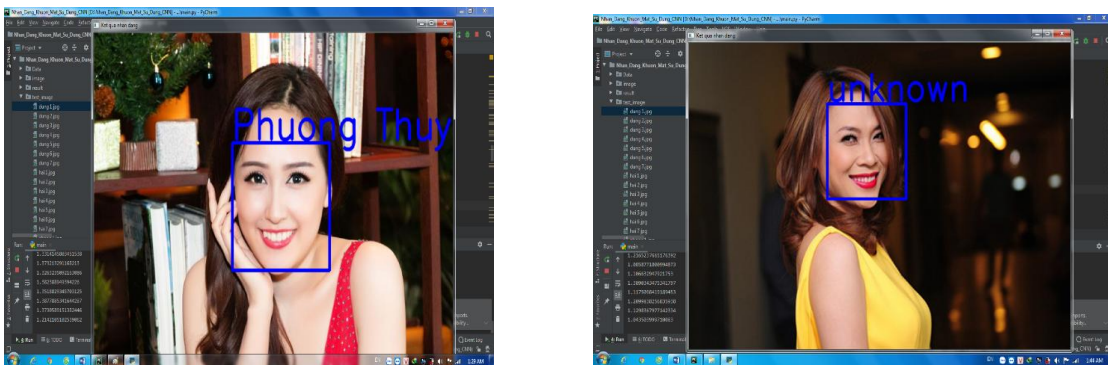
3.4. Kết quả thực nghiệm

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



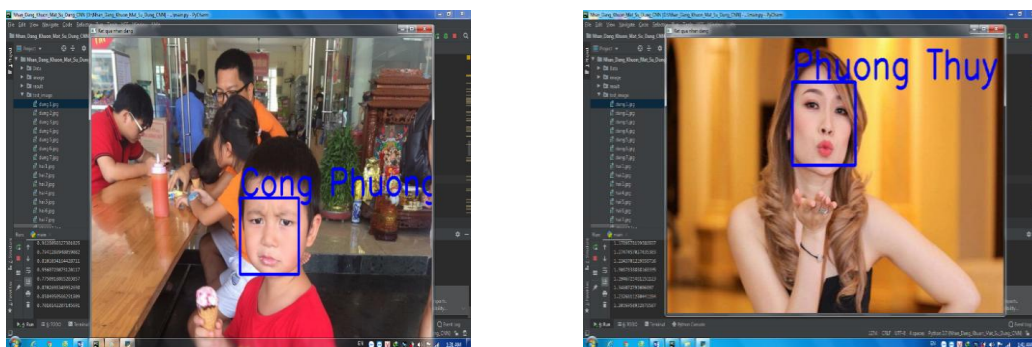
Hình 3.4. Kết quả nhận dạng đúng hai khuôn mặt có trong dữ liệu train

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 nhằm 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 Phương	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 Phương	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 lẫn 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.