A white rectangular frame with red and yellow text

Description automatically generated**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO BÀI TẬP LỚN**

**MÔN NHẬP MÔN TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI ĐIỂM DANH BẰNG NHẬN DIỆN KHUÔN MẶT**

Vũ Thành Long-B21DCAT012

Giảng viên hướng dẫn : Nguyễn Thị Mai Trang

**Hà Nội 2024**

**Mục lục**

[**1.GIỚI THIỆU TỔNG QUAN** 4](#_Toc166509659)

[**1.1. Tổng quan về nhận diện khuôn mặt** 4](#_Toc166509660)

[**1.2. Phân tích bài toán** 4](#_Toc166509661)

[**1.3. Phương pháp giải quyết** 5](#_Toc166509662)

[**2. Cơ sở lý thuyết** 6](#_Toc166509663)

[**2.1. CNN (Convolutional neural network)** 6](#_Toc166509664)

[**2.1.1. Giới thiệu** 6](#_Toc166509665)

[**2.1.2. Một số khái niệm** 7](#_Toc166509666)

[**2.1.3.Xác nhận khuôn mặt và nhận diện khuôn mặt** 13](#_Toc166509667)

[**2.1.4. Áp dụng mô hình CNN cho bài toán** 15](#_Toc166509668)

[**2.2. Keras** 15](#_Toc166509669)

[**2.2.1. Keras là gì?** 15](#_Toc166509670)

[**2.2.2. Cấu trúc của Keras** 15](#_Toc166509671)

[**2.2.3. Ưu điểm và nhược điểm của Keras** 17](#_Toc166509672)

[**2.3. Tiêu chí đánh giá mô hình** 18](#_Toc166509673)

[**3. Thực nghiệm và kết quả** 19](#_Toc166509674)

[**3.1. Cài đặt môi trường** 19](#_Toc166509675)

[**3.2. Thực nghiệm** 24](#_Toc166509676)

[**3.2.1. Dữ liệu chuẩn bị** 24](#_Toc166509677)

[**3.2.2. Giao diện của sản phẩm** 26](#_Toc166509678)

[**3.2.3. Model nhận diện và các chức năng của sản phẩm** 28](#_Toc166509679)

[**3.3. Đánh giá và kết quả** 32](#_Toc166509680)

[**4. Kết luận** 36](#_Toc166509681)

[**4.1. Đánh giá kết quả đạt được** 36](#_Toc166509682)

[**4.2. Hướng phát triển đề tài** 37](#_Toc166509683)

[**5. Tài liệu tham khảo** 37](#_Toc166509684)

**MỞ ĐẦU**

Khuôn mặt đóng vai trò quan trọng trong quá trình giao tiếp giữa người với người, và cũng mang một lượng thông tin giàu có, chẳng hạn có thể xác định giới tính, tuổi tác, trạng thái cảm xúc của người đó,… hơn nữa khảo sát chuyển động của các đường nét trên khuôn mặt có thể biết được người đó đang muốn nói gì.

Trong hệ thống nhận dạng người thì quá trình nhận dạng khuôn mặt được đánh giá là bước khó khăn và quan trọng nhất so với các bước còn lại của hệ thống. Do đó, nhận dạng khuôn mặt là điều quan trọng và cần thiết.

Nhận dạng khuôn mặt người là một công nghệ được ứng dụng rộng rãi trong đời sống hằng ngày của con người như các hệ thống giám sát, quản lí vào ra, tìm kiếm thông tin người nổi tiếng,… có rất nhiều phương pháp nhận dạng khuôn mặt để nâng cao hiệu suất tuy nhiên dù ít hay nhiều những phương pháp này đang vấp phải những thử thách về độ sáng, hướng nghiêng, kích thước ảnh, hay ảnh hưởng của tham số môi trường.

Cùng với sự phát triển của xã hội, vấn đề an ninh, bảo mật đang được yêu cầu khắt khe tại mọi quốc gia trên thế giới. Các hệ thống nhận dạng con người, đồ vật… được ra đời và phát triển với độ tin cậy ngày càng cao. Với cách tiếp cận đối tượng nhận dạng theo phương pháp này, chúng ta có thể thu nhập được nhiều thông tin từ đối tượng hơn, mà không cần tác động nhiều đến đối tượng cũng vấn đảm bảo tính chính xác, an toàn, thuận tiện.

Trong phạm vi bài báo cáo này chúng em xin được trình bày quá trình thực hiện điểm danh qua nhận diện khuôn mặt bằng mô hình CNN. Cuối cùng, mặc dù đã cố gắng rất nhiều nhưng do thời gian có hạn, khả năng dịch và hiểu tài liệu nên nội dung báo cáo này không thể tránh khỏi những thiếu sót, rất mong được sự chỉ bảo, góp ý của các thầy cô và các bạn.

# **1.GIỚI THIỆU TỔNG QUAN**

## **1.1. Tổng quan về nhận diện khuôn mặt**

Hơn một thập kỉ qua có rất nhiều công trình nghiên cứu về bài toán xác định khuôn mặt người từ ảnh đen trắng, xám đến ảnh màu như ngày hôm nay. Các nhà nghiên cứu đi từ bài toán đơn giản, mỗi ảnh chỉ có một mặt người nhìn thẳng vào thiết bị thu hình và đầu ở tư thế thẳng đứng trong ảnh đen trắng. Cho đến ngày hôm nay bài toán mở rộng cho ảnh màu, có nhiều khuôn mặt trong cùng một ảnh, có nhiều tư thế thay đổi trong ảnh. Không những thế mà còn mở rộng cả phạm vi từ môi trường xung quanh khá đơn giản cho đến môi trường xung quanh rất phức tạp nhằm đáp ứng nhu cầu của con người.

Đặc biệt bài toán nhận diện khuôn mặt là một trong những bài toán quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Nó được áp dụng rộng rãi từ nhận dạng khuôn mặt cho mục đích an ninh, tự động đăng nhập, nhận dạng người trong ảnh hoặc video, đến các ứng dụng giải trí như thêm hiệu ứng vào khuôn mặt trong video

## **1.2. Phân tích bài toán**

Bài toán Nhận Diện Khuôn mặt

Một hệ thống nhận diện mặt người thông thường bao gồm bốn bước xử lý sau:

1. Phát hiện khuôn mặt (Face Detection).

2. Phân đoạn khuôn mặt (Face Alignment hay Segmentation).

3. Trích chọn đặc trưng (Feature Extraction).

4. Nhận diện (Recognition) hay Phân lớp khuôn mặt (Face Classification).

Bên cạnh những bước chính nêu trên, ta còn có thể áp dụng thêm một số bước khác như tiền xử lý, hậu xử lý nhằm làm tăng độ chính xác cho hệ thống. Sau bước phát hiện khuôn mặt, ta có thể thực hiện bước tiền xử lý(Preprocessing) bao gồm các bước căn chỉnh ảnh (face image alignment) và chuẩn hóa ánh sáng (illumination normalization). Do một vài thông số như: tư thế khuôn mặt, độ sáng, điều kiện ánh sáng,… phát hiện khuôn mặt được đánh giá là bước khó và quan trọng nhất so với các bước còn lại của hệ thống.

## **1.3. Phương pháp giải quyết**

Những vấn đề cần giải quyết:

* Đầu tiên, nhìn vào hình và tìm tất cả các khuôn mặt có trong đó.
* Thứ 2, tập trung vào một khuôn mặt của một người và nhận diện ngay cả khi khuôn mặt quay đi hướng khác, hoặc trong môi trường thiếu ánh sáng.
* Thứ 3, chọn ra những đặc điểm đặc trưng của khuôn mặt mà bạn sử

dụng để phân biệt với khuôn mặt của người khác. Vd: mắt lớn bao nhiêu, mặt dài bao nhiêu,mũi cao hay không, …

* Cuối cùng, đối chiếu những đặc điểm đặc trưng đó với những người bạn đã biết và xác định được địa chỉ id hoặc tên của người đó.

Trong bài toán này, tôi đã đề xuất mô hình dựa trên mạng nơron tích chập (CNN) để nhận diện khuôn mặt con người. Đựa vào cơ sở đó tôi phát triển tiếp và xây dựng ứng dụng cho bài toán thực tiễn như hệ thống điểm danh khuôn mặt các sinh viên trong lớp học.

# **2. Cơ sở lý thuyết**

## **2.1. CNN (Convolutional neural network)**

### **2.1.1. Giới thiệu**

Trong mạng neural, mô hình mạng neural tích chập (CNN) là 1 trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định đối tượng và nhận dạng khuôn mặt là 1 trong số những lĩnh vực mà CNN được sử dụng rộng rãi.

CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, …). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy H x W x D (H: Chiều cao, W: Chiều rộng, D: Độ dày).

A close-up of several squares

Description automatically generated

*Ví dụ: Hình ảnh là mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).*

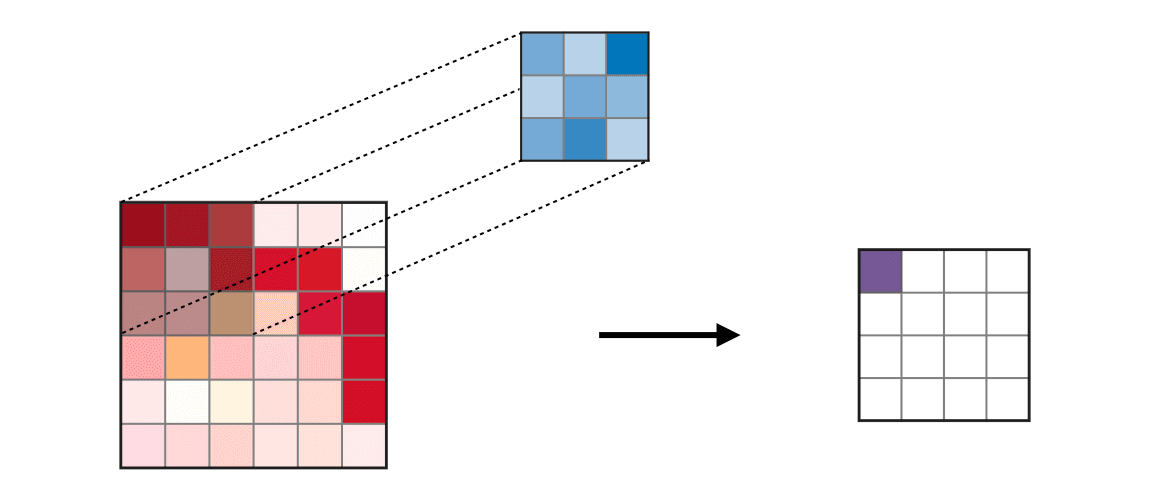
Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernals), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới là toàn bộ luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.A diagram of a diagram of a model of a model of a model of a model of a model of a model of a model of a model of a model of a model of a model of

Description automatically generated

### **2.1.2. Một số khái niệm**

**2.1.2.1. Các kiểu tầng**

**Tầng tích chập (CONV):** Tầng tích chập (CONV) sử dụng các bộ lọc để thực hiện phép tích chập khi đưa chúng đi qua đầu vào I theo các chiều của nó. Các siêu tham số của các bộ lọc này bao gồm kích thước bộ lọc 𝐹 và độ trượt (stride) S. Kết quả đầu ra O được gọi là feature map hay activation map.



**Pooling (POOL):** Tầng pooling (POOL) là một phép downsampling, thường được sử dụng sau tầng tích chập, giúp tăng tính bất biến không gian. Cụ thể, max pooling và average pooling là những dạng pooling đặc biệt, mà tương ứng là trong đó giá trị lớn nhất và giá trị trung bình được lấy ra.

Có 2 kiểu pooling:

* Max pooling: Từng phép pooling chọn giá trị lớn nhất trong khu vực mà nó đang được áp dụng
* Average pooling: Từng phép pooling tính trung bình các giá trị trong khu vực mà nó đang được áp dụng

**Fully Connected (FC):** Tầng kết nối đầy đủ (FC) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNNs, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.

A group of blue dots on a black background

Description automatically generated

**2.1.2.2.Các siêu tham số của bộ lọc**

Tầng tích chập chứa các bộ lọc mà rất quan trọng cho ta khi biết ý nghĩa đằng sau các siêu tham số của chúng.

**Các chiều của một bộ lọc:** Một bộ lọc kích thước 𝐹×F áp dụng lên đầu vào chứa C kênh (channels) thì có kích thước tổng kể là 𝐹×𝐹×C thực hiện phép tích chập trên đầu vào kích thước 𝐼×𝐼×C và cho ra một feature map (hay còn gọi là activation map) có kích thước 𝑂×𝑂×1

A blue cube with squares

Description automatically generated

**Stride:** Đối với phép tích chập hoặc phép pooling, độ trượt 𝑆 ký hiệu số pixel mà cửa sổ sẽ di chuyển sau mỗi lần thực hiện phép tính.

A blue squares with white background

Description automatically generated.

**Zero-padding**Zero-padding là tên gọi của quá trình thêm 𝑃 số không vào các biên của đầu vào.

**2.1.2.3.Điều chỉnh siêu tham số**

**Tính tương thích của tham số trong tầng tích chập**

Bằng cách ký hiệu 𝐼 là độ dài kích thước đầu vào, 𝐹 là độ dài của bộ lọc, 𝑃 là số lượng zero padding, 𝑆 là độ trượt, ta có thể tính được độ dài 𝑂 của feature map theo một chiều bằng công thức:

A black and white text

Description automatically generated

A teddy bear reading a book

Description automatically generated

**Hiểu về độ phức tạp của mô hình:** Để đánh giá độ phức tạp của một mô hình, cách hữu hiệu là xác định số tham số mà mô hình đó sẽ có. Trong một tầng của mạng neural tích chập, nó sẽ được tính toán như sau:

*Bảng độ phức tạp của các thành phần trong CNN*

|  |  |  |  |
| --- | --- | --- | --- |
|  | CONV | POOL | FC |
| Minh họa | A diagram of a square with blue squares  Description automatically generated | A square with arrows and letters  Description automatically generated | A diagram of a network  Description automatically generated |
| Kích thước đầu vào | I×I×C | I×I×C | Nin |
| Kích thước đầu ra | O×O×K | O×O×C | Nout |
| Số lượng tham số | (F×F×C+1)⋅K | 0 | (Nin​+1)×Nout |
| Lưu ý | • Một tham số bias với mỗi bộ lọc • Trong đa số trường hợp, 𝑆<F • Một lựa chọn phổ biến cho 𝐾 là 2C | • Phép pooling được áp dụng lên từng kênh (channel-wise) • Trong đa số trường hợp, 𝑆=F | • Đầu vào được làm phẳng • Mỗi neuron có một tham số bias • Số neuron trong một tầng FC phụ thuộc vào ràng buộc kết cấu |

**Trường thụ cảm:**Trường thụ cảm (receptive field) tại tầng 𝑘 là vùng được ký hiệu 𝑅k​×Rk​ của đầu vào mà những pixel của activation map thứ 𝑘kcó thể "nhìn thấy". Bằng cách gọi 𝐹j​ là kích thước bộ lọc của tầng 𝑗 và 𝑆i là giá trị độ trượt của tầng i và để thuận tiện, ta mặc định 𝑆0​=1, trường thụ cảm của tầng 𝑘 được tính toán bằng công thức:

A black and white math equation

Description automatically generated

**2.1.2.4.Các hàm kích hoạt thường gặp**

**Rectified Linear Unit:** Tầng rectified linear unit (ReLU) là một hàm kích hoạt 𝑔g được sử dụng trên tất cả các thành phần. Mục đích của nó là tăng tính phi tuyến tính cho mạng. Những biến thể khác của ReLU được tổng hợp ở bảng dưới:

|  |  |  |
| --- | --- | --- |
| Relu | Leaky Relu | ELU |
| g(z)=max(0,z) | g(z)=max(ϵz,z) với 𝜖≪1 | g(z)=max(α(ez−1),z) với 𝛼≪1 |
| A yellow line with black arrows  Description automatically generated with medium confidence | A graph of a function  Description automatically generated | A graph of a function  Description automatically generated |
| • Độ phức tạp phi tuyến tính có thể thông dịch được về mặt sinh học | • Gán vấn đề ReLU chết cho những giá trị âm | • Khả vi tại mọi nơi |

**Softmax:** Bước softmax có thể được coi là một hàm logistic tổng quát lấy đầu vào là một vector chứa các giá trị 𝑥∈R n và cho ra là một vector gồm các xác suất 𝑝∈Rn thông qua một hàm softmax ở cuối kiến trúc. Nó được định nghĩa như sau:

A group of math equations

Description automatically generated with medium confidence

### **2.1.3.Xác nhận khuôn mặt và nhận diện khuôn mặt**

**Các kiểu mô hình:** Hai kiểu mô hình chính được tổng hợp trong bảng dưới:

|  |  |
| --- | --- |
| Xác nhận khuôn mặt | Nhận diện khuôn mặt |
| • Có đúng người không? • Tra cứu một-một | • Đây có phải là 1 trong K người trong cơ sở dữ liệu không? • Tra cứu một với tất cả |
|  | A group of stuffed animals  Description automatically generated |

**One Shot Learning:** One Shot Learning là một thuật toán xác minh khuôn mặt sử dụng một tập huấn luyện hạn chế để học một hàm similarity nhằm ước lượng sự khác nhau giữa hai tấm hình. Hàm này được áp dụng cho hai tấm ảnh thường được ký hiệu 𝑑(image 1,image 2).

**Siamese Network:** Siamese Networks hướng tới việc học cách mã hóa tấm ảnh để rồi định lượng sự khác nhau giữa hai tấm ảnh. Với một tấm ảnh đầu vào 𝑥(i), đầu ra được mã hóa thường được ký hiệu là 𝑓(𝑥(i)).

**Triplet loss:** Triplet loss ℓ là một hàm mất mát được tính toán dựa trên biểu diễn nhúng của bộ ba hình ảnh 𝐴 (mỏ neo), 𝑃 (dương tính) và 𝑁(âm tính). Ảnh mỏ neo và ảnh dương tính đều thuộc một lớp, trong khi đó ảnh âm tính thuộc về một lớp khác. Bằng các gọi 𝛼∈R+ là tham số margin, hàm mất mát này được định nghĩa như sau:

A black and white text

Description automatically generated

A diagram of a circle with circles and lines

Description automatically generated

**2.1.4. Áp dụng mô hình CNN cho bài toán**

**A diagram of a diagram

Description automatically generated**

## **2.2. Keras**

### **2.2.1. Keras là gì?**

Keras là một thư viện Deep Learning mã nguồn mở được phát triển bởi François Chollet. Nó cung cấp một giao diện dễ sử dụng và linh hoạt cho việc xây dựng và huấn luyện mạng nơ-ron. Keras cho phép người dùng xây dựng mô hình từ các lớp nơ-ron, tùy chỉnh mô hình, biên soạn mô hình, và huấn luyện mô hình một cách dễ dàng.

### **2.2.2. Cấu trúc của Keras**

Keras được thiết kế theo nguyên tắc modular, với mỗi lớp của mạng nơ-ron được coi là một thành phần độc lập. Các lớp này có thể được xếp chồng lên nhau để tạo thành một mô hình nơ-ron.

**2.2.2.1. Lớp (Layer)**

• Lớp là thành phần cơ bản của một mạng nơ-ron trong Keras. Mỗi lớp nhận đầu vào từ các lớp trước đó, thực hiện một loại biến đổi và truyền đầu ra cho các lớp tiếp theo.

• Keras cung cấp nhiều loại lớp khác nhau như Dense (lớp nơ-ron kết nối đầy đủ), Conv2D (lớp tích chập 2D), MaxPooling2D (lớp gộp tối đa 2D), LSTM (lớp LSTM), và nhiều lớp khác.

**2.2.2.2. Mô hình (Model)**

• Mô hình trong Keras là cấu trúc tổng quan của một mạng nơ-ron, bao gồm một hoặc nhiều lớp được kết hợp với nhau theo một cách cụ thể.

• Có hai cách chính để xây dựng mô hình trong Keras:

• Sử dụng lớp Sequential: Xây dựng một chuỗi tuần tự các lớp theo thứ tự tuyến tính.

• Sử dụng API Functional: Cho phép xây dựng mô hình có cấu trúc phức tạp hơn bằng cách kết hợp các lớp một cách tự do và tạo ra nhiều đầu ra hoặc đầu vào.

**2.2.2.3. Cấu trúc tuần tự (Sequential)**

• Trong cấu trúc tuần tự, các lớp được xếp chồng lên nhau một cách tuần tự. Mỗi lớp chỉ nhận đầu vào từ lớp trước đó và truyền đầu ra cho lớp tiếp theo.

• Đây là phương pháp đơn giản và phổ biến nhất trong việc xây dựng mô hình trong Keras, thích hợp cho các mạng nơ-ron có cấu trúc đơn giản.

**2.2.2.4. API Functional**

• API Functional của Keras cho phép xây dựng các mô hình nhiều phần, có cấu trúc phức tạp hơn so với cấu trúc tuần tự.

• Người dùng có thể tạo ra nhiều đầu vào và đầu ra, kết hợp các lớp một cách tự do, và tạo ra các dòng dữ liệu (data flows) phức tạp.

**2.2.2.5. Biên soạn mô hình (Compiling the Model)**

• Sau khi xây dựng mô hình, người dùng cần biên soạn mô hình trước khi huấn luyện bằng cách chỉ định các tham số quan trọng như hàm mất mát (loss function), thuật toán tối ưu hóa (optimizer), và các chỉ số đánh giá (metrics).

• Phương thức compile() của mô hình được sử dụng cho công việc này.

**2.2.2.6. Huấn luyện và Đánh giá**

• Sau khi mô hình đã được biên soạn, nó có thể được huấn luyện bằng cách sử dụng dữ liệu đào tạo thông qua phương thức fit().

• Khi huấn luyện kết thúc, mô hình có thể được đánh giá trên dữ liệu thử nghiệm để đánh giá hiệu suất sử dụng phương thức evaluate().

**2.2.2.7. Dự đoán**

• Mô hình đã được huấn luyện có thể được sử dụng để dự đoán đầu ra cho các dữ liệu mới thông qua phương thức predict().

### **2.2.3. Ưu điểm và nhược điểm của Keras**

Ưu điểm của Keras:

1. Giao diện dễ sử dụng: Keras được thiết kế để có giao diện sử dụng đơn giản và trực quan. Việc xây dựng mô hình được thực hiện thông qua việc thêm các lớp vào mô hình Sequential hoặc sử dụng API Functional của Keras.

2. Linh hoạt và mở rộng: Keras cho phép người dùng tạo ra các mô hình nơ-ron phức tạp bằng cách kết hợp các lớp và tùy chỉnh mô hình theo nhu cầu cụ thể của họ. Người dùng có thể dễ dàng thêm, xóa và sửa đổi các lớp trong mô hình để thí nghiệm và tinh chỉnh mô hình của mình.

3. Hỗ trợ mạnh mẽ: Keras có một cộng đồng lớn và nhiều tài liệu hỗ trợ, bao gồm ví dụ, tài liệu hướng dẫn, và các diễn đàn thảo luận. Sự phổ biến của Keras đồng nghĩa với việc có sẵn nhiều tài nguyên để hỗ trợ người dùng trong quá trình học và phát triển mô hình.

Nhược điểm của Keras

1. Giới hạn tính linh hoạt: Việc tùy chỉnh mô hình ở mức độ thấp hơn có thể gặp khó khăn, đặc biệt là đối với các mô hình phức tạp hoặc yêu cầu kiểm soát chặt chẽ hơn về phần cứng.

2. Hiệu suất không tốt như các framework khác: Trong một số trường hợp, hiệu suất của Keras có thể không tốt bằng các framework Deep Learning khác như TensorFlow hoặc PyTorch, đặc biệt là khi xử lý các mô hình lớn và phức tạp.

3. Độ ổn định:Mặc dù Keras được cập nhật và phát triển liên tục, nhưng đôi khi việc sử dụng các tính năng mới có thể gây ra vấn đề về độ ổn định.

## **2.3. Tiêu chí đánh giá mô hình**

- Ảnh âm tính (Negative Image): Đây là loại ảnh mà mô hình phân loại là không chứa đối tượng hoặc tính chất cụ thể mà mô hình đang tìm kiếm.

Ví dụ, nếu một mô hình được huấn luyện để phát hiện xe hơi trong ảnh, thì bất kỳ hình ảnh nào không chứa xe hơi sẽ được coi là ảnh âm tính.

- Ảnh dương tính (Positive Image): Đây là loại ảnh mà mô hình phân loại là chứa đối tượng hoặc tính chất cụ thể mà mô hình đang tìm kiếm.

Tiếp tục ví dụ về mô hình phát hiện xe hơi, bất kỳ hình ảnh nào chứa ít nhất một chiếc xe hơi sẽ được coi là ảnh dương tính.

Mô hình nhận diện khuôn mặt triển khai sẽ được đánh giá theo 4 tham số:

**Accuracy**: tính tỉ lệ số ảnh dự đoán đúng và trong tổng số tập dữ liệu kiểm thử. Độ chính xác được tính theo công thức:

* TP- True Positive: số ảnh dương tính phân loại là dương tính
* FN - False Negative: số ảnh dương tính phân loại là âm tính
* TN- True Negative: số ảnh âm tính phân loại là âm tính
* FP - False Positive: số ảnh âm tính phân loại là dương tính

**Precision:** Tỉ lệ giữa số lượng các ảnh được phân loại là dương tính mà thực sự là dương tính và tổng số lượng các ảnh được phân loại là dương tính. Công thức tính:

**Recall:** Tỉ lệ giữa số lượng các ảnh dương tính thực sự được phát hiện đúng và tổng số lượng các ảnh dương tính trong tập dữ liệu. Công thức tính:

**F1-score:** Là một số tổ hợp của độ nhạy và độ chính xác, là trung bình điều hòa của Precision và Recall. Công thức tính:

# **3. Thực nghiệm và kết quả**

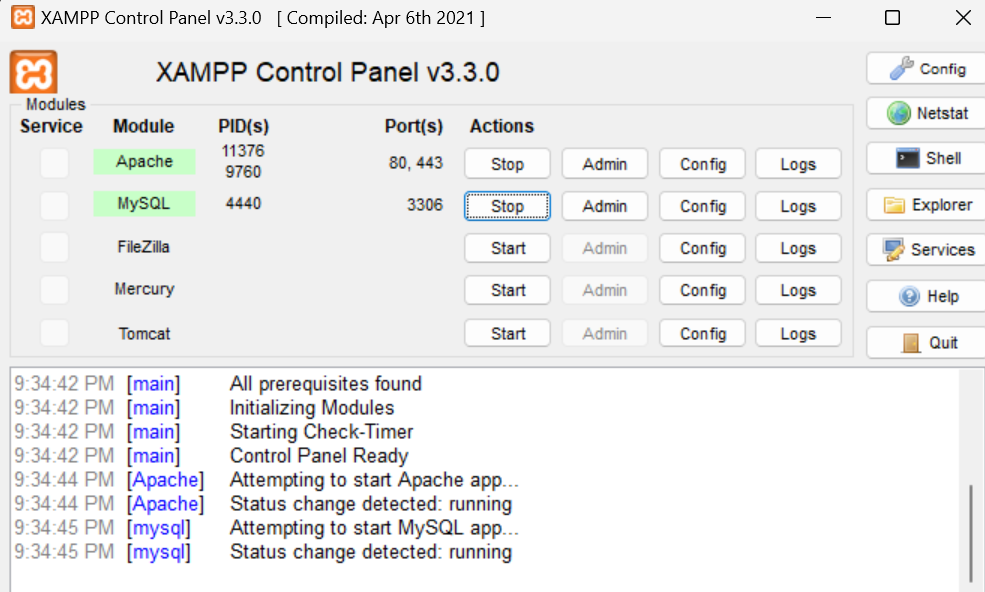
## **3.1. Cài đặt môi trường**

- Để có thể thực hiện bài toán cần cài đặt một số công cụ:

* Xampp
* QTDesigner
* PyCharm
* Một số thư viện hỗ trợ như: opencv, numpy, pyqt6,…

- Xampp:

* XAMPP là một phần mềm nguồn mở và miễn phí dùng để tạo web server trên máy tính cá nhân (Localhost), XAMPP tương thích với các hệ điều hành phổ biến như : Linux, MacOS, Windows,..
* Ưu điểm lớn nhất của XAMPP là mã nguồn mở và tính dễ sử dụng, tương đối đơn giản, gọn nhẹ nên được sử dụng ngày càng phổ biến hiện nay.
* XAMPP được ứng dụng rộng rãi từ người dùng phổ thông đến lập trình viên, nhằm để vận hành cũng như phát triển các website dùng ngôn ngữ lập trình PHP như: WordPress, Joomla!, Magento, Drupal, OpenCart, phpBB,..
* Một số cài đặt đã được tôi triển khai



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

- QTDesigner

* Là công cụ để nhanh chóng xây dựng giao diện người dùng đồ họa với các widget từ khung Qt GUI . Nó cung cấp một giao diện kéo và thả đơn giản để bố trí các thành phần như nút, trường văn bản, hộp tổ hợp và hơn thế nữa.
* Qt Designer tạo ra .ui các tệp. Đây là một định dạng dựa trên XML đặc biệt để lưu trữ các widget của bạn dưới dạng cây. Ta có thể dịch chúng sang ngôn ngữ lập trình như C ++ hoặc Python.
* Một số cài đặt đã được tôi triển khai

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

- PyCharm: Đây sẽ là môi trường tôi thực hiện tạo ra các chức năng cũng như tương tác với cơ sở dữ liệu và giao diện

* PyCharm là môi trường phát triển tích hợp được sử dụng để lập trình bằng Python. Nó cung cấp phân tích mã, trình gỡ lỗi đồ họa, trình kiểm tra đơn vị tích hợp, tích hợp với hệ thống kiểm soát phiên bản và hỗ trợ phát triển web với Django. PyCharm được phát triển bởi công ty JetBrains của Séc
* Các thư viện được thực hiện cài đặt bên trong PyCharm
* OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.
* NumPy là một thư viện dành cho ngôn ngữ lập trình Python, bổ sung hỗ trợ cho các mảng và ma trận lớn, đa chiều, cùng với một bộ sưu tập lớn các hàm toán học cấp cao để hoạt động trên các mảng này.
* Một số cài đặt đã được tôi triển khai

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**3.2. Thực nghiệm**

**3.2.1. Dữ liệu chuẩn bị**

- Một số ảnh ví dụ để dùng cho việc thực hiện huấn luyện model

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

- Cơ sở dữ liệu để mô tả cho các đối tượng (vì có khá nhiều table nên tôi sẽ biểu diễn một số đại diện)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**3.2.2. Giao diện của sản phẩm**

- Một số hình ảnh của sản phẩm

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**3.2.3. Model nhận diện và các chức năng của sản phẩm**

- Như đã nêu ở trên, tôi sử dụng CNN để giải quyết vấn đề nhận diện khuôn mặt

- Để xây dựng model tôi chia làm 2 phần

* Đầu tiên thực hiện tiền xử lý dữ liệu: Xử lý dữ liệu đầu vào thành dữ liệu huấn luyện phù hợp với mô hình.

A computer screen shot of a program code

Description automatically generated

A computer screen with white and yellow text

Description automatically generated

A computer screen shot of a code

Description automatically generated

* Thực hiện dựng model sau khi đã có dữ liệu chuẩn bị để huấn luyện và lưu model sau khi đã huyến luyện xong

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

- Giải quyết xong vấn đề nhận diện khuôn mặt, ta cần thực hiện tạo ra các chức năng quản lí như mục tiêu bài toán đã đặt ra ban đầu

- Chức năng của sản phẩm gồm có:

* Đăng nhập, đăng xuất
* Quản lí sinh viên
* Thêm, sửa, xóa thông tin sinh viên
* Tìm kiếm sinh viên
* Lấy ảnh sinh viên
* Train model
* Thêm, sửa, xóa lớp học
* Tìm kiếm lớp học
* Nhận diện
* Cho phép điểm danh sinh viên (đi, muộn, vắng)
* Tìm kiếm lớp sinh viên đang học theo id giảng viên
* Quản lí bản điểm danh
* Thêm, sửa, xóa bản điểm danh
* Tìm kiếm bản điểm danh
* Quản lí môn học
* Thêm, sửa, xóa môn học
* Tìm kiếm môn học
* Tìm kiếm giảng viên theo id môn học
* Quản lí giảng viên
* Thêm, sửa, xóa thông tin giảng viên
* Thống kê
* Thống kê số sinh viên, số bản điểm danh, số lần sinh viên đi muộn, vắng
* Tìm kiếm và xem các bản điểm danh đi muộn
* Tìm kiếm và xem các bản điểm danh vắng
* Xem ảnh
* Liên kết tới phần mục ảnh train và test để có thể xem và điều chỉnh sao cho hợp lí

**3.3. Đánh giá và kết quả**

- **Link project**: https://ptiteduvn-my.sharepoint.com/:f:/g/personal/longvt\_b21at012\_stu\_ptit\_edu\_vn/Ep4HSog7EY9MiNtyAWjjYswBqwI6SIsRMBLY9NMuRcfJKw?e=ME999F

- Đánh giá

Tổng hợp bộ dữ liệu (Bộ dữ liệu nhỏ)

90 ảnh dương tính

90 ảnh âm tính

**Bảng . Kết quả thực nghiệm của mô hình CNN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N. of layer | N. of parameter | Evaluation of image | | | |
| **Acc** | **Pre** | **Rec** | **F1** |
| 2 | 57.877.162 | 0.4 | 0.5 | 0.4 | 0.44 |
| 3 | 25.439.018 | 0.7 | 0.67 | 0.8 | 0.73 |
| 4 | 5.106.602 | **0.5** | 0.6 | 0.6 | 0.6 |

Với số lượng lớp khác nhau sẽ có số lượng tham số khác nhau và theo như mô hình đã triển khai thì không phải mô hình phức tạp thì sẽ cho hiệu suất cao hơn so với mô hình đơn giản mà sẽ cần phải điều chỉnh mô hình phù hợp với bài toán

- Kết quả nhận diện khi điểm danh và một số chức năng  
A screenshot of a computer

Description automatically generated

*Nhận diệm điểm danh*

*A screenshot of a computer

Description automatically generated*

*Thống kê bản điểm danh*

*A screenshot of a computer

Description automatically generated*

*Thông tin sinh viên*

*A screenshot of a computer

Description automatically generated*

*Thông tin giảng viên*

*A screenshot of a computer

Description automatically generated*

*Quản lí môn học*

*A screenshot of a computer

Description automatically generated*

*Quản lí các bản điểm danh*

**4. Kết luận**

**4.1. Đánh giá kết quả đạt được**

Khi thực hiện đề tài tôi thu được kết quả như sau:

- Trình bày được báo cáo một cách khoa học và có hệ thống những kiến thức hiểu biết của bản thân, có tham khảo các tài liệu về các vấn đề có liên quan đến nội dung tìm hiểu, nghiên cứu.

- Tăng khả năng tư duy logic, có thể nghiên cứu độc lập một vấn đề mà trước đây tôi chưa thực sự quan tâm.

- Trau dồi những kinh nghiệm quý giá trong quá trình thiết kế, làm quen và sử dụng các mô hình mạng.

- Tạo được một ứng dụng cơ bản về nhận diện khuôn mặt

**4.2. Hướng phát triển đề tài**

Đây là một bài toán có nhiều tiềm năng trong quá trình hội nhập, để phát triển thành một hệ thống hoàn chỉnh và có thể đưa ứng dụng vào thực tế một cách rộng rãi chương trình cần:

• Nâng cấp hệ thống để có thể áp dụng trên mạng diện rộng trên nhiều hệ điều hành khác nhau.

• Thiết kế giao diện chương trình mang tính chuyên nghiệp hơn.

• Phần mềm ứng dụng được áp dụng cho hầu hết các mô hình lớn chứ không chỉ là mô hình nhỏ.

**5. Tài liệu tham khảo**

- CNN: https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-convolutional-neural-networks

- Xampp: https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/

- PyQt: https://realpython.com/python-pyqt-gui-calculator/