**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**~~~~~~\*~~~~~~**

A red and white flag with a yellow and red design

Description automatically generated

**BÁO CÁO TIẾN ĐỘ BÀI TẬP LỚN**

|  |  |  |
| --- | --- | --- |
| ***Thành viên nhóm 20*** | **:** | **Hoàng Hải Phong <20210670>**  **Dương Trung Kiên<20215067>**  **Hoàng Kim Trí Thành <20210798>**  **Nghiêm Xuân Diện<20215007>**  **Đinh Văn Luận<20215083>** |
| ***Lớp*** | **:** | **147728 - Nhập môn AI** |
| ***Giáo viên hướng dẫn*** | **:** | **TS. Trần Thế Hùng** |

Contents

[1. Mô tả dự án 2](#_Toc163153081)

[2. Phương pháp 3](#_Toc163153087)

[2.1. Lớp Convolutional 4](#_Toc163153089)

[2.2. Lớp Pooling 4](#_Toc163153090)

[2.3. Lớp Fully Connected 5](#_Toc163153091)

[3. Thu thập và tiền xử lý dữ liệu 7](#_Toc163153092)

[4. Xây dựng mô hình CNN 7](#_Toc163153093)

[5. Training và thực hiện dự đoán 8](#_Toc163153094)

## 

## 1. Mô tả dự án

### Dự án của chúng tôi là xây dựng một ứng dụng Điểm Danh thông minh, sử dụng công nghệ nhận dạng khuôn mặt và trí tuệ nhân tạo (AI). Ứng dụng này được thiết kế để cải thiện quá trình điểm danh trong các cơ sở giáo dục, doanh nghiệp hoặc bất kỳ môi trường nào cần kiểm soát quy trình quản lý ra vào.

### Các tính năng chính của ứng dụng bao gồm:

### Nhận dạng Khuôn Mặt: Bức ảnh chụp sẽ được đưa vào một mô hình AI nhận dạng khuôn mặt, giúp xác định và phân loại đối tượng trong ảnh.

### Phân loại và Ghi Nhận Đối Tượng: Dựa trên kết quả của mô hình nhận dạng, ứng dụng sẽ phân loại và ghi nhận thông tin về những người đã tham gia điểm danh vào hệ thống.

## 2. Phương pháp

### Phương pháp CNN (Convolutional Neural Network) là một trong những phương pháp phổ biến nhất và mạnh mẽ nhất trong lĩnh vực thị giác máy tính và xử lý ảnh. Được thiết kế để mô phỏng cách mà não của con người nhận biết và xử lý hình ảnh, CNN đã giúp cải thiện nhiều ứng dụng như nhận dạng khuôn mặt, phân loại đối tượng, nhận dạng vật thể và nhiều ứng dụng khác.

CNN thường bao gồm các lớp chính sau:

* Lớp Convolutional (Conv layer): Áp dụng các bộ lọc (hay còn gọi là kernel) qua hình ảnh đầu vào để trích xuất các đặc trưng cục bộ.
* Lớp Pooling (Pooling layer): Giảm kích thước không gian của biểu diễn đặc trưng bằng cách chọn giá trị tối đa hoặc trung bình trong một cửa sổ cố định.
* Lớp Fully Connected (FC layer): Lớp này kết nối đầy đủ các đặc trưng đã trích xuất từ lớp trước với các lớp đầu ra để thực hiện phân loại hoặc dự đoán.

### 2.1. Lớp Convolutional

Tích chập là lớp đầu tiên để trích xuất ra các đặc điểm từ hình ảnh ban đầu. Phép tính tích chập một phép tính với hai thành phần đầu vào, là ma trận hình ảnh ban đầu và một ma trận khác được gọi là bộ lọc (filter). Các filter sẽ trượt trên ma trận ảnh đầu vào và thực hiện phép tích chập trên 1 vùng của ma trận đó. Kết quả của các phép tích chập sẽ tạo thành Feature map (bản đồ đặc trưng).A close-up of a grid

Description automatically generated

Sự kết hợp của 1 hình ảnh với các bộ lọc khác nhau có thể thực hiện các hoạt động như phát hiện cạnh, làm mờ và làm sắc nét bằng cách áp dụng các bộ lọc. Ví dụ dưới đây cho thấy kết quả sau khi thực hiện phép tích chập giữa ảnh đầu vào với các bộ lọc khác nhau

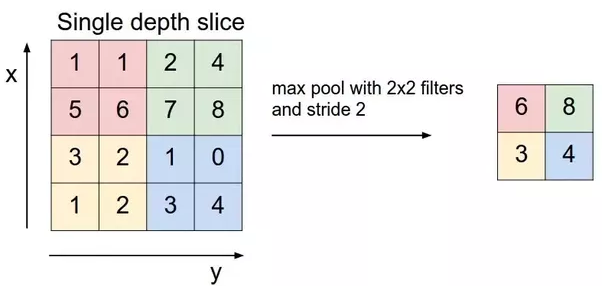
A screenshot of a phone

Description automatically generated

### 2.2. Lớp Pooling

Lớp Pooling thường được sử dụng sau các lớp Convolutional trong một mạng CNN để giảm kích thước không gian của biểu diễn đặc trưng, giúp giảm độ phức tạp tính toán và số lượng tham số của mô hình. Điều này cũng giúp giảm thiểu overfitting và làm tăng tốc độ huấn luyện của mạng.

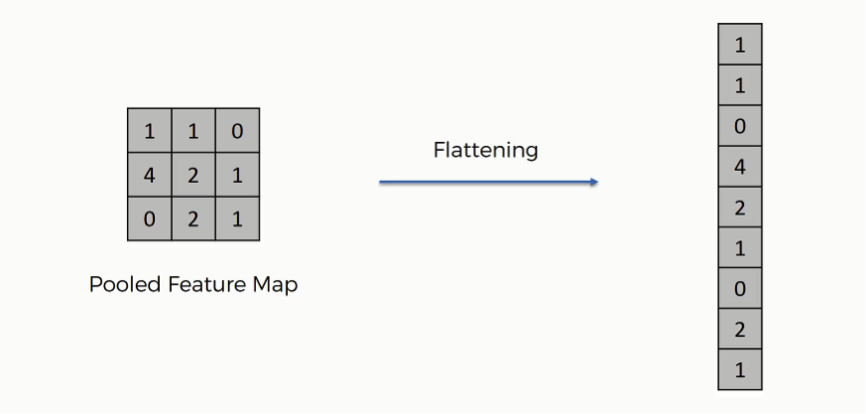
Có nhiều loại lớp Pooling, trong dự án của mình nhóm chúng tôi sẽ sử dụng MaxPooling. Một vùng ma trận sau khi được áp dụng Max Pooling sẽ chỉ giữ lại pixel có giá trị lớn nhất.



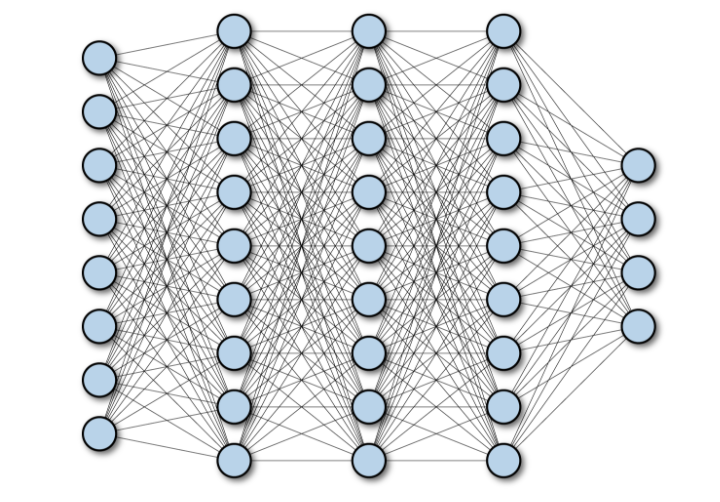
### 2.3. Lớp Fully Connected

Sau khi thu được các feature map từ các lớp tích chập và lớp max pooling, chúng ta thường sẽ phẳng hoá chúng thành một vector 1D trước khi đưa vào lớp Fully Connected. Quá trình phẳng hoá này được gọi là flatten.

Ví dụ, giả sử chúng ta có một feature map sau lớp tích chập có kích thước là 6x6 và một feature map sau lớp max pooling có kích thước là 3x3. Khi phẳng hoá, chúng ta sẽ nối các hàng của các feature map lại với nhau để tạo ra một vector 1D có kích thước là 3x3x1 = 9. Vector này sẽ chứa tất cả các đặc trưng đã trích xuất từ hình ảnh.



Sau đó, vector này sẽ được kết nối với lớp Fully Connected, trong đó mỗi phần tử trong vector tương ứng với một neuron trong lớp Fully Connected. Lớp Fully Connected sẽ thực hiện các phép tính tuyến tính và phi tuyến tính trên vector này để tạo ra các dự đoán hoặc phân loại cho hình ảnh.



## 3. Thu thập và tiền xử lý dữ liệu

Ban đầu nhóm định thu thập dữ liệu từ khuôn mặt của các thành viên trong lớp học để tiến hành dự án xây dựng ứng dụng điểm danh. Tuy nhiên điều này không khả thi do dữ liệu tự quay sẽ thiếu đi sự đa dạng về biểu cảm cũng như khuôn mặt chỉ được ghi lại ở 1 thời điểm nên cuối cùng, nhóm quyết định sử dụng bộ dữ liệu có tên [Celebrity-Face-Recognition-Dataset](https://github.com/prateekmehta59/Celebrity-Face-Recognition-Dataset/tree/master?tab=readme-ov-file) của tác giả Prateekmehta59 từ Github. Đây là bộ dữ liệu vô cùng lớn với khoảng 800.000 hình ảnh của 1100 người nổi tiếng cùng với đó là một lượng nhỏ các ảnh không liên quan để gây nhiễu, tránh việc mô hình bị overfitting (hiện tượng mô hình tìm được quá khớp với dữ liệu training). Tất cả các hình ảnh được thu thập từ Google và không chứa hình ảnh trùng lặp nào. Mỗi lớp người nổi tiếng (thư mục) chứa khoảng 700-800 hình ảnh. Tổng kích thước của Bộ dữ liệu là 172 GB. Chính vì kích thước quá lớn nên nhóm quyết định sẽ cho training trước cho mô hình với 7 khuôn mặt của 7 người (3 nữ, 4 nam).

A screenshot of a computer

Description automatically generated

Do ảnh từ bộ dữ liệu chưa đồng nhất về mặt kích thước nên nhóm sử dụng kĩ thuật cắt khuôn mặt từ ảnh cho trước để đầu vào cho việc training sẽ chỉ đa số là các ảnh khuôn mặt và còn lại là các ảnh nhiễu. Sau đó, các hình ảnh sẽ được xáo trộn và thay đổi kích thước thành 256px \* 256px.

A screen shot of a computer

Description automatically generated

Nhóm sử dụng layers.experimental.preprocessing.RandomFlip và layers.experimental.preprocessing.RandomRotation để thực hiện các phép biến đổi ngẫu nhiên trên dữ liệu đầu vào. Đây là 1 kĩ thuật khá hay, giúp tăng cường dữ liệu huấn luyện và làm cho mô hình học được một cách tổng quát hơn.

A screenshot of a computer code

Description automatically generated

Ngoài ra, để chuẩn bị cho việc training 1 lượng tương đối lớn data, nhóm cũng cần tận dụng tối đa GPU của máy tính thay vì sử dụng CPU truyền thống. Để Tensorflow có thể sử dụng được GPU, nhóm đã cài đặt CUDA và cuDNN, nền tảng và thư viện được phát triển bởi NVIDIA, giúp tăng hiệu suất và tối ưu hóa việc huấn luyện mô hình mạng nơ-ron sâu. Nhóm đã gặp khá nhiều khó khăn trong việc tìm được phiên bản phù hợp cũng như sửa đổi code để phù hợp với thư viện. Cuối cùng thì chúng tôi đã cài đặt phiên bản cuDNN 8.2 và việc xử lí lượng lớn dữ liệu được tối ưu hơn rất nhiều, nhanh gấp 20-30 lần so với khi sử dụng CPU.

A computer screen shot of white text

Description automatically generated

## 4. Xây dựng mô hình CNN

TensorFlow cung cấp một cách tiếp cận linh hoạt và mạnh mẽ để xây dựng và huấn luyện các mạng CNN, với các lớp Conv2D và Max Pooling2D được TensorFlow cung cấp, nhóm chúng tôi đã tiến hành xây dựng mô hình mạng CNN như sau:

A screenshot of a computer program

Description automatically generated

Ma trận ảnh đầu vào sẽ đi qua 6 khối tích chập (Convolution block ) trước khi đến với lớp Fully Connected, mỗi khối tích chập sẽ bao gồm một số lượng các bộ lọc tích chập (64 hoặc 128) kèm với 1 lớp MaxPooling

## 5. Training và thực hiện dự đoán

Nhóm đã tiến hành fit mô hình với bộ dữ liệu đầu vào, sử dụng hàm loss “SparseCategoricalCrossentropy” với metrics là “accuracy”. Dưới đây là kết quả của 6 epochs đầu tiên và cuối cùng:

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

Loss giảm dần qua các epoch, cho thấy mô hình đang học cách giảm thiểu sai số giữa giá trị dự đoán và giá trị thực tế.

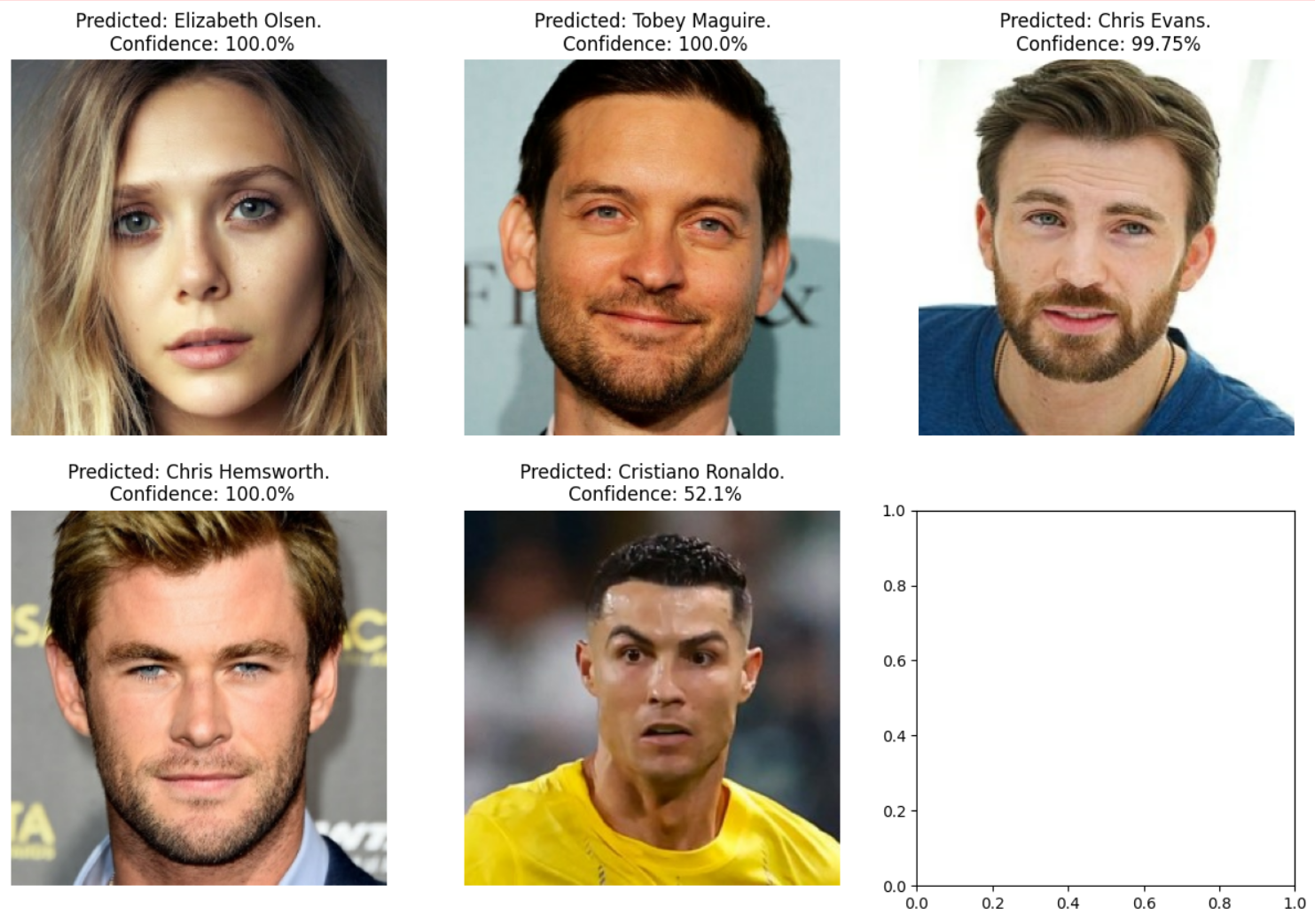
A graph with orange lines

Description automatically generated

Độ chính xác tăng dần qua các epoch, cho thấy mô hình đang cải thiện khả năng phân loại dữ liệu một cách chính xác. Tuy nhiên, độ chính xác (Accuracy) vẫn chưa được quá tốt, đây chính là điểm mà nhóm muốn cải thiện trong những tuần sắp tới.

Kết quả thực hiện dự đoán với ảnh crop từ Google :

A collage of people with text

Description automatically generated

## 6. Điểm cần cải thiện và phương pháp dự kiến

Như đã nói ở trên, độ chính xác (Accuracy) vẫn chưa được quá tốt, và nhóm muốn cải thiện về mặt này.

Phương pháp nhóm dự định bao gồm:

* Cải thiện bộ dữ liệu (bỏ bớt ảnh nhiễu)
* Sử dụng mô hình tốt hơn như VGG
* Thay đổi hàm tính chỉ số loss để đánh giá kết quả tốt hơn

## 7. Bảng phân công nhiệm vụ

|  |  |
| --- | --- |
| **Tên** | **Nhiệm vụ** |
| Hoàng Hải Phong | Thiết kế model |
| Hoàng Kim Trí Thành | Training model |
| Đinh Văn Luận | Xử lí Data |
| Nghiêm Xuân Diện | Xử lí Data |
| Dương Trung Kiên | Thu thập Dataset và bộ test Data |