**TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO DỰ ÁN CUỐI KÌ**

**Học phần: COMP261 – Trí Tuệ Nhân Tạo**

**NHẬN DIỆN KHUÔN MẶT SỬ DỤNG MÔ HÌNH**

**CNN – CONVOLUTIONAL NEURAL NETWORK**

**Giảng viên: Lê Xuân Hiền**

**Sinh viên: Nguyễn Huy Hải Ninh**

**Mã sinh viên: 735105084**

**Mục lục**

1. Giới thiệu đề tài ………………………………………………
2. Tổng quan giải pháp ………………………………………….
3. Thư viện sử dụng ……………………………………………..
4. Quy trình thực hiện …………………………………………...
   1. Tiền xử lý dữ liệu …………………………………………
   2. Xây dựng và huấn luyện mô hình CNN ………………….
   3. Nhận diện khuôn mặt .…………………………………….
5. Kết quả đạt được ……………………………………………...
6. Hạn chế và hướng phát triển ………………………………….
7. Kết luận ……………………………………………………….
8. **Giới thiệu đề tài**

* Đề tài này xây dựng hệ thống nhận diện khuôn mặt sử dụng mạng nơ-ron tích chập (CNN) và các thư viện Python hiện đại. Ứng dụng giúp tự động nhận diện khuôn mặt từ ảnh, phục vụ các mục đích như điểm danh, kiểm soát an ninh,… và các bài toán thực tiễn khác.

1. **Tổng quan giải pháp**

* Dữ liệu sử dụng:
  + Ảnh khuôn mặt của nhiều người, lưu trong các thư mục dữ liệu của dự án.
* Tiền xử lý dữ liệu:
  + Ảnh được chuyển về kích thước chuẩn, chuyển sang ảnh xám và gán nhãn cho từng người.
* Xây dựng mô hình:
  + Sử dụng mạng nơ-ron tích chập (CNN) để học đặc trưng khuôn mặt từ dữ liệu đã xử lý.
* Huấn luyện và đánh giá:
  + Mô hình được huấn luyện trên tập dữ liệu, sau đó kiểm thử với ảnh mới để đánh giá độ chính xác.

1. **Mô hình và thư viện sử dụng**

* Mô hình học sâu:
  + Sử dụng mạng nơ-ron tích chập (CNN – Convolutional Neural Network) để tự động trích xuất đặc trưng và phân loại khuôn mặt từ ảnh.
* Thư viện chính:
  + OpenCV: xử lý và đọc ảnh khuôn mặt.
  + Numpy: xử lý dữ liệu số và mảng.
  + Tensorflow: Xây dựng và huấn luyện mô hình CNN.
  + Pillow: hỗ trợ xử lý ảnh.
  + Scikit-learn: đánh giá mô hình, xử lý dữ liệu.

1. **Quy trình thực hiện**
2. **Tiền xử lý dữ liệu (Create data.py):**
   1. Khởi tạo bộ phát hiện khuôn mặt

* Sử dụng bộ phân loại Haar Cascade của OpenV để phát hiện khuôn mặt trong ảnh.



* 1. Lựa chọn chế độ để lấy dữ liệu
* Biến id xác định chế độ lấy dữ liệu:
  + Id = 1: xử lý ảnh đã có sẵn trong thư mục **data**.
  + Id = 2: Chụp ảnh trực tiếp từ **webcam**.
  1. Tiền xử lý với ảnh có sẵn (id = 1)
* Tạo thư mục lưu ảnh khuôn mặt đã xử lý tên datasetIM.



* Đọc từng ảnh từ thư mục data theo định dạng tên file.



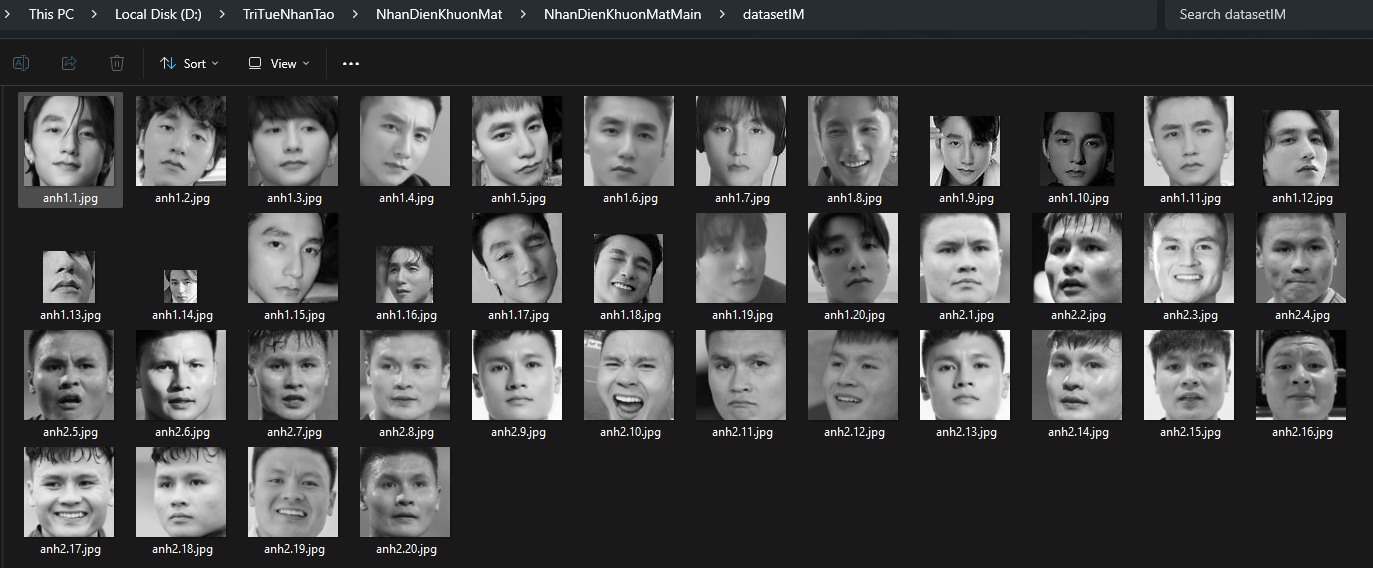
* Ảnh màu được chuyển sang ảnh xám để giảm nhiễu và giảm số chiều dữ liệu.



* Sử dụng detector.detectMultiScale để phát hiện tất cả khuôn mặt trong ảnh.



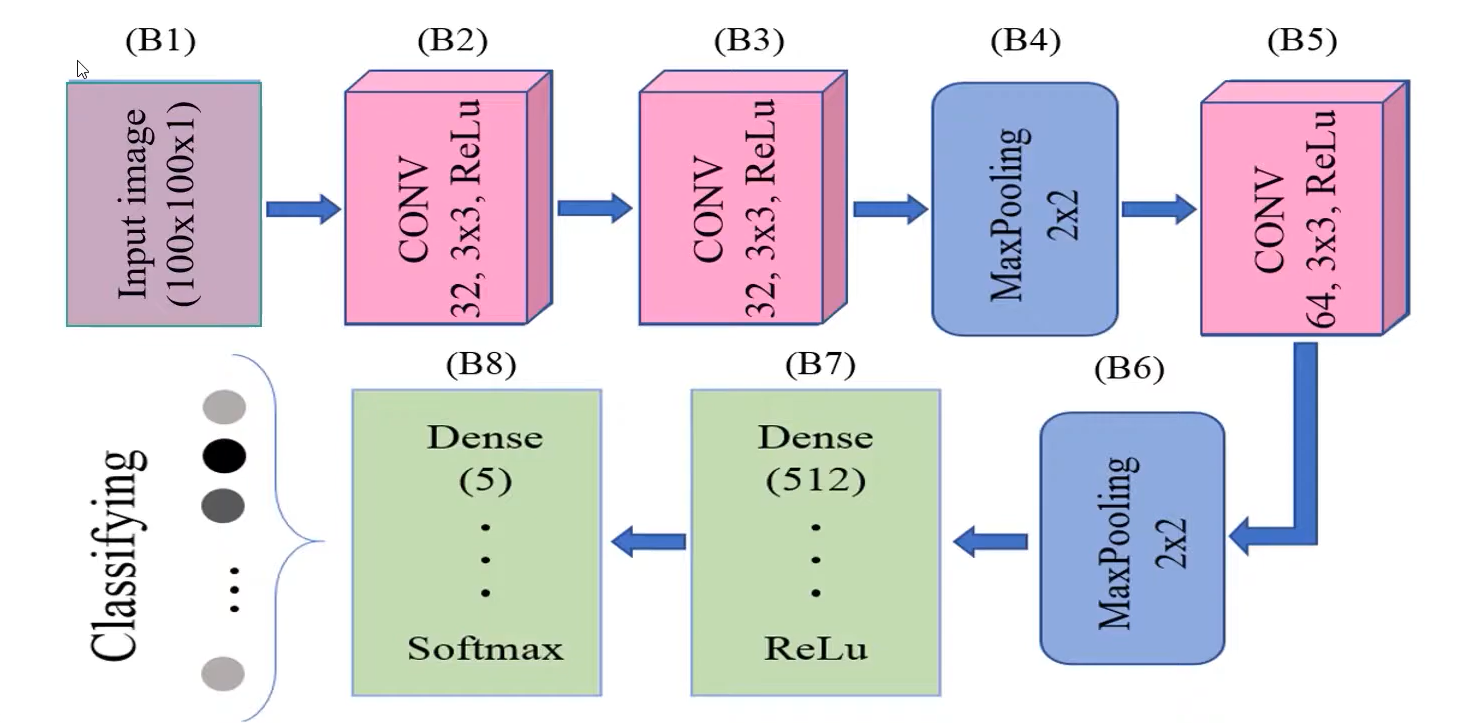
* Cắt và lưu vùng khuôn mặt:
  + Với mỗi khuôn mặt phát hiện được, cắt vùng chứa khuôn mặt từ ảnh xám.
  + Lưu ảnh khuôn mặt đã cắt vào thư mục đầu ra với tên tương ứng là datasetIM.



* 1. Tiền xử lý với ảnh chụp từ webcam (id = 2)
* Tạo thư mục lưu ảnh tên datasetWB.
* Khởi tạo webcam.
* Vòng lặp chụp ảnh:
  + Với mỗi người, chụp đủ số lượng ảnh yêu cầu.
  + Đọc từng khung hình từ webcam, chuyển sang ảnh xám.
  + Phát hiện khuôn mặt trong khung hình.
  + Hiển thị khung hình và vùng nhận diện khuôn mặt.

1. **Xây dựng và huấn luyện mô hình CNN (train.py)**
   1. **Tổng quan mô hình CNN**

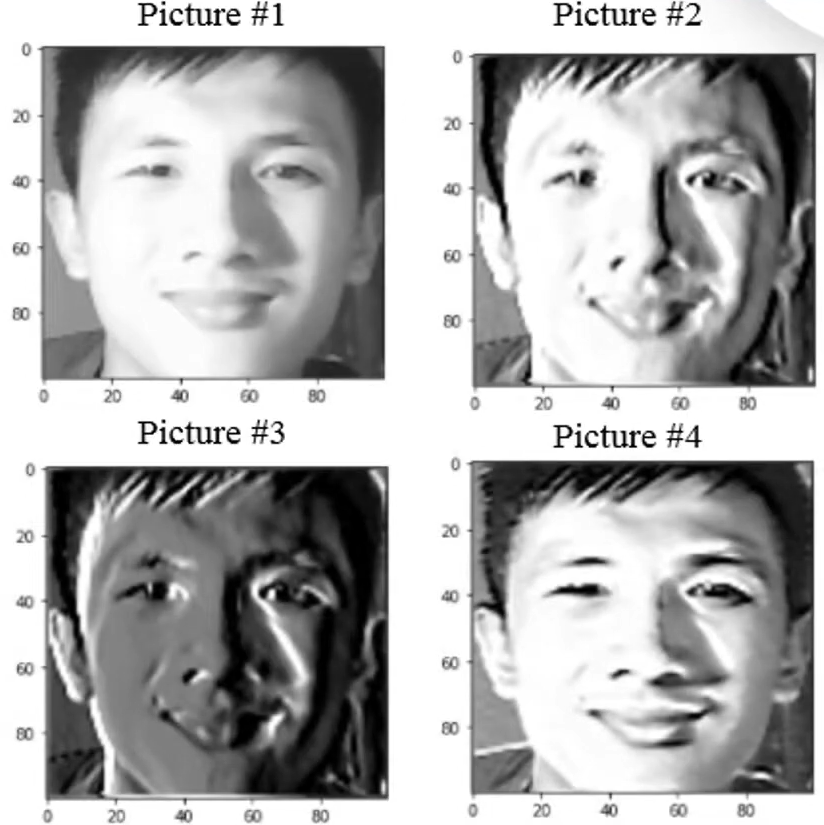
* Mô hình được huấn luyện dựa trên mô hình CNN, mô hình CNN được thiết kế gồm hai phần chức năng:
  + Trích chọn đặc trưng của ảnh khuôn mặt.
  + Phân lớp đối tượng dựa trên đặc trưng đã chọn.
* Mô hình huấn luyện CNN bao gồm:



* + Ảnh đầu vào (B1): kích thước 100x100x1 ảnh xám.
  + Lớp tích chập đầu tiên (B2):
    - Gồm 32 kernel 3x3 và hàm Activation Relu.
    - Dò tìm các đặc trưng cơ bản trong ảnh.
    - Relu được dùng để loại bỏ giá trị âm, giúp học tốt hơn.
  + Lớp tích chập tiếp theo (B3):
    - Gồm 32 kernel 3x3 và hàm Activation Relu.
    - Dò tìm các đặc trưng phức tạp hơn từ feature map ở lớp B2.
    - Relu được dùng để loại bỏ giá trị âm, giúp học tốt hơn.
  + Lớp MaxPooling (B4):
    - Lớp này có kích thước 2x2.
    - Giảm kích thước của feature map giúp giảm số lượng tính toán.
    - Giữ lại thông tin quan trọng nhất trong vùng 2x2.
  + Lớp tích chập thứ ba (B5):
    - Gồm 64 kernel 3x3 và hàm Activation Relu.
    - Dò tìm các đặc trưng cao cấp hơn với 64 kernel.
  + Lớp MaxPooling (B6):
    - Lớp này có kích thước 2x2.
    - Tiếp tục giảm kích thước ảnh đặc trưng.
    - Tăng tốc độ tính toán và giảm độ phức tạp mô hình.
  + Lớp Dense (512):
    - Gồm 512 nơ-ron, và hàm Activation Relu.
    - Là lớp fully connected đầu tiên sau các lớp tích chập và pooling.
    - Input từ feature map sau khi flatten (làm phẳng).
    - Học các đặc trưng phức tạp hơn từ feature map đã rút trích.
  + Lớp Dense (5):
    - Gồm 5 nơ-ron, tương ứng với 5 lớp phân loại.
    - Đây là lớp đầu ra (output layer).
    - Hàm Softmax để chuyển đầu ra thành xác suất ( giá trị từ 0 đến 1).
    - Lớp có xác suất cao nhất là lớp mà ảnh được phân loại vào.

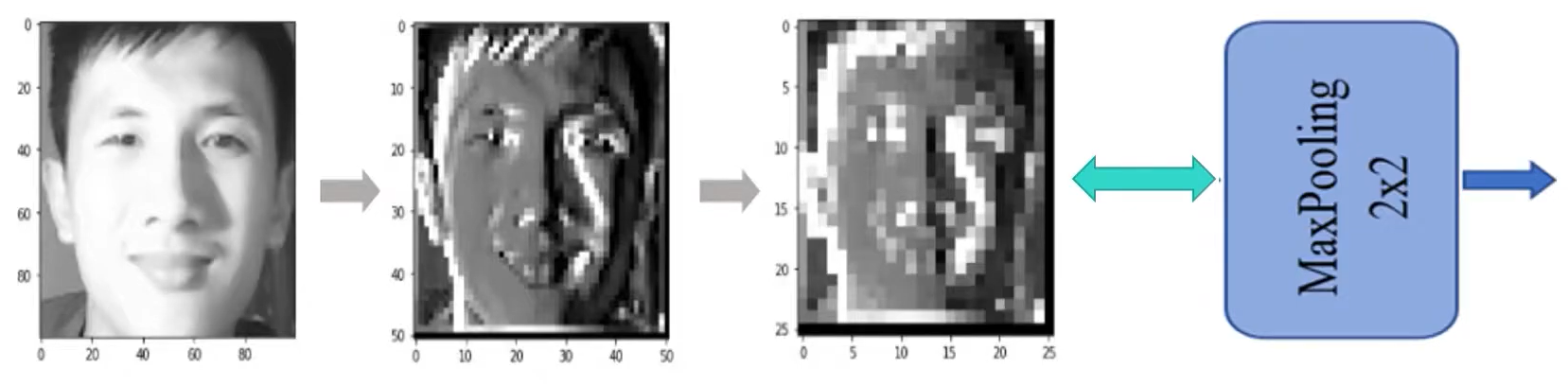
1. **Lớp tích chập (Convolution)**

* Các ảnh sau khi qua các lớp Convolution sẽ lọc ra các đặc trưng của khuôn mặt cần nhận diện.
* Với 32 kernel, thì sẽ cho ra 32 đặc trưng cơ bản của khuôn mặt.
* Với 64 kernel, thì sẽ cho ra 64 đặc trưng cơ bản của khuôn mặt.



1. **Lớp MaxPooling**

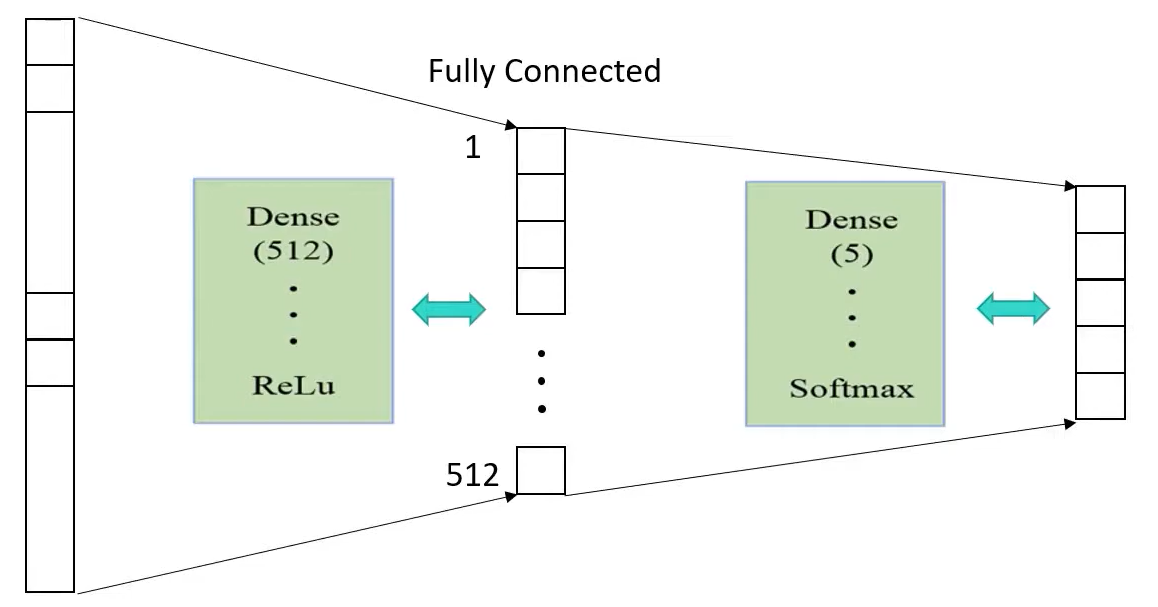
* Các ảnh sau khi qua lớp MaxPooling sẽ giữ lại được các đặc trưng chung nhất của khuôn mặt.



* Ví dụ:
  + Ảnh đầu vào là 100x100, sau khi qua lớp tích chập đầu tiên + hàm Relu và lớp MaxPooling đầu tiên thì sẽ cho ra ảnh có kích thước 50x50.
  + Tiếp theo qua lớp tích chập + hàm Relu và lớp MaxPooling tiếp theo thì sẽ cho ra ảnh có kích thước 25x25.
  + Từ đó ta thấy, ảnh giờ đã mờ hơn và thô hơn, nhưng vẫn giữ được thông tin và cấu trúc chính của khuôn mặt.

1. **Lớp kết nối đầy đủ (Fully Connected)**

* Sử dụng mạng nhiều lớp với hàm kích hoạt Softmax cho lớp nơ-non đầu ra.



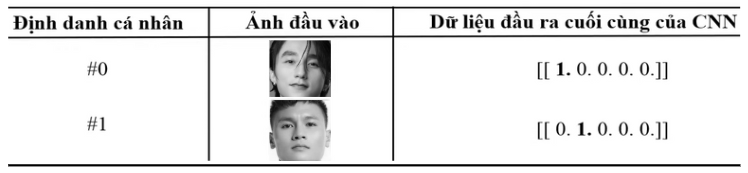
* Sau khi qua lớp MaxPooling cuối cùng, các pixel sẽ được duỗi thẳng ra thành 1 vector 1 chiều, và được kết nối với 512 nơ-ron và thông qua hàm kích hoạt Relu để kết nối với 5 nơ-ron đầu ra tương ứng với 5 khuôn mặt cần nhận dạng (Trong dự án mới chỉ nhận diện cho 2 khuôn mặt, nếu muốn 5 thì chỉ cần điểu chỉnh lại các thông số tăng lên 5).
* 5 nơ-ron này sẽ thông qua hàm Softmax để phân loại cho từng khuôn mặt.

1. **Dữ liệu đầu vào**

* Các ảnh cần huấn luyện sẽ được lấy từ thư mục ảnh (datasetIM hoặc datasetWB) và được resize về 100x100x1.



* Sau đó, tạo nhãn cho từng ảnh:
  + Các ảnh được dán nhãn lần lượt là 0, 1, 2,…. tương ứng với đầu ra của hàm Softmax.
  + Đầu ra nào có xác suất lớn nhất thì đó là nhãn của người được tạo để nhận dạng từng người.

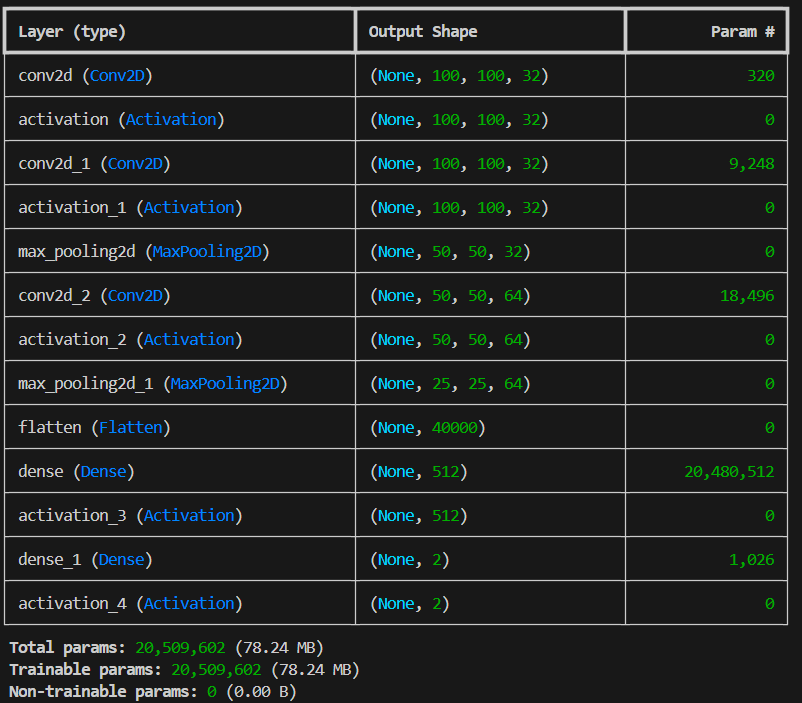


1. **Ứng dụng thư viện Keras để huấn luyện mô hình**

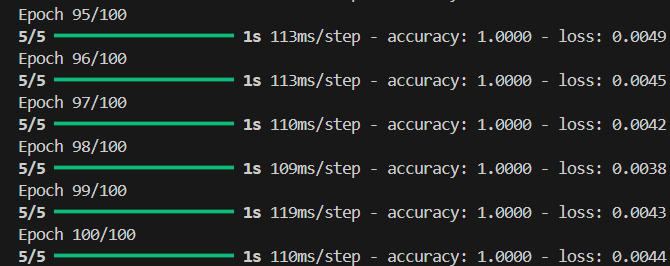


* Khởi tạo mô hình:
  + Sử dụng Sequential() để xây dựng mô hình theo từng lớp nối tiếp nhau.
* Conv2D:
  + Lớp tích chập với 32 và 64 bộ lọc, kích thước kernel 3x3, padding “same”, giúp trích xuất đặc trưng không gian từ ảnh đầu vào.
* Activation (Relu):
  + Sử dụng sau mỗi lớp Conv2D để tăng tính phi tuyến tính cho mô hình.
* MaxPooling2D:
  + Lớp pooling với kích thước 2x2, giúp giảm chiều dữ liệu và giữ lại các đặc trưng quan trọng.
* Flatten:
  + Chuyển dữ liệu từ dạng ma trận sang vector để đưa vào các lớp fully connected.
* Dense:
  + Lớp kết nối đầy đủ với 512 nơ-ron, tiếp theo là lớp đầu ra với số nơ-ron bằng số lớp (số người), sử dụng softmax để phân loại.

1. **Quá trình huấn luyện**



* Qua các lớp tích chập thì số lượng tham số (weights và bias) là 320; 9248; 18496. Đây là số lượng tham số cần huấn luyện ở mỗi lớp.
* Số lượng tham số là các lớp Dense lần lượt là 20480512 và 1026.
* Độ chính xác và giá trị hàm lỗi sau khi huấn luyện 100 epoch:



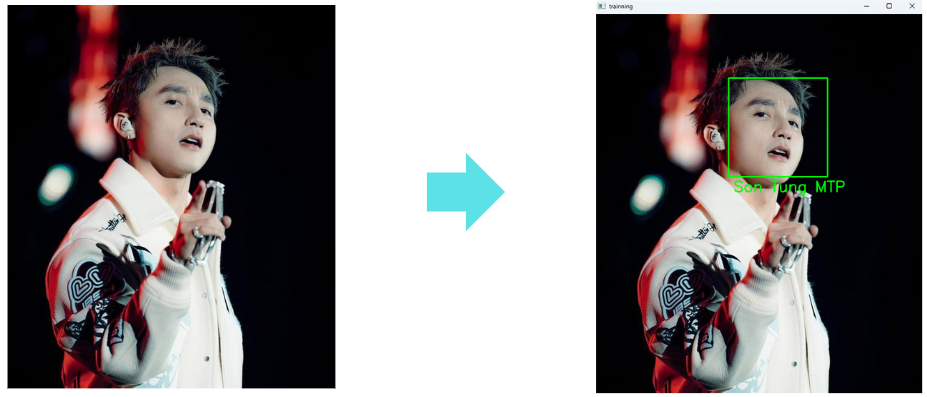
* Giá trị hàm lỗi giảm dần từ epoch 1 đến epoch 100.
* Giá trị độ chính xác tăng dần từ epoch 1 đến epoch 100.
* Kết quả mô hình huấn luyện được lưu vào file khuonmat.h5.

1. **Nhận diện khuôn mặt (Display.py)**

* Tạo tên của người trong ảnh:



* Mỗi khuôn mặt được cắt ra từ ảnh gốc, đưa vào mô hình để dự đoán.
* Kết quả dự đoán là một vector xác suất, chọn giá trị lớn nhất để xác định chỉ số lớp (label) tương ứng.
* Dựa vào chỉ số này, chương trình gán tên cho người trong ảnh. Ví dụ:
  + Nếu kết quả là 0: tên là “Son Tung MTP”.
  + Nếu kết quả là 1: tên là “Quang Hai”.
* Đưa một ảnh mới bất kì vào chương trình và xác định khuôn mặt kèm với tên của khuôn mặt.



1. **Kết quả đạt được**

* Mô hình nhận diện khuôn mặt được huấn luyện tạm thời trên 40 ảnh (2 người, mỗi người 20 ảnh) với độ chính xác trên tập huấn luyện đạt 100% sau epoch, giá trị hàm lỗi rất nhỏ (~0.0044).
* Khi kiểm tra với ảnh mới, chương trình có thể phát hiện và nhận diện chính xác khuôn mặt, đồng thời hiển thị tên người dùng tương ứng lên ảnh.
* Kết quả này cho thấy mô hình hoạt động tốt trên tập dữ liệu nhỏ, có khả năng phân biệt rõ ràng giữa các đối tượng đã được huấn luyện.

1. **Hạn chế và hướng phát triển**
2. Hạn chế

* Số lượng dữ liệu huấn luyện còn ít (chỉ 2 người, mỗi người 20 ảnh), chưa đủ đa dạng về góc chụp, ánh sáng, biểu cảm,…
* Mô hình có thể bị overfitting, nhận diện tốt tập huấn luyện nhưng chưa chắc tổng quát hóa với ảnh thực tế hoặc người mới.

1. Hướng phát triển

* Bổ sung thêm dữ liệu ảnh cho mỗi người để mô hình nhận diện ổn định hơn (Khoảng 100 ảnh/1người).
* Thử nghiệm nhận diện với nhiều ảnh kiểm tra khác nhau để đánh giá khả năng mô hình.
* Có thể mở rộng nhận diện cho thêm 1-2 người mới bằng cách bổ sung dữ liệu và huấn luyện lại mô hình.

1. **Kết luận**

* Mô hình này có 3 lớp nơ-ron tích chập (Convolution) và 2 lớp nơ-ron liên kết đầy đủ (Fully Connected).
* Tổng số tham số là khoảng hơn 20 triệu.
* Mô hình có độ phức tạp ở mức vừa phải, phù hợp với các hệ thống xử lý ở mức trung bình.
* Tiềm năng khả thi trong ứng dụng thực tiễn.

**HẾT**