

Face Recognition App using Streamlit

Ngô Thành Phát¹ - 19521994

Abstract—Nhận dạng khuôn mặt là một bài toán phổ biến đang đặt ra hiện nay. Tồn tại nhiều phương pháp và hướng tiếp cận đối với bài toán nhận dạng khuôn mặt: Tiếp cận theo đặc trưng toàn cục (sử dụng các đặc điểm toàn cục của khuôn mặt) và tiếp cận theo đặc trưng cục bộ (sử dụng các đặc điểm cục bộ của khuôn mặt). Tuy nhiên hiệu quả của các phương pháp nhận dạng này vẫn còn hạn chế và độ chính xác chưa cao khi dữ liệu đầu vào bị ảnh hưởng bởi các yếu tố khách quan của môi trường (độ sáng, hướng nghiêng, kích thước, ...). Do đó, em đề xuất xây dựng ứng dụng nhận dạng khuôn mặt dựa trên thư viện face-recognition đã được xây dựng sẵn API cho các tác vụ phát hiện, trích xuất đặc trưng và nhận dạng khuôn mặt và sử dụng Streamlit làm giao diện điều khiển.

I. GIỚI THIỆU

A. Động lực

Cùng với sự phát triển của xã hội, vấn đề an ninh bảo mật là một điều tất yếu hiện nay. Các hệ thống nhận dạng con người được ra đời với độ tin cậy ngày càng cao. Có thể kể đến như nhận dạng hình dáng, nhận dạng giọng nói, nhận dạng khuôn mặt,... Trong đó, phổ biến và được ứng dụng nhiều hơn cả là bài toán nhận dạng khuôn mặt.

B. Báo cáo vấn đề

Ứng dụng nhận dạng khuôn mặt là phần mềm xác định hoặc xác nhận danh tính của một người qua khuôn mặt của họ. Công nghệ này hoạt động bằng cách xác định và đo lường các đặc điểm khuôn mặt trong hình ảnh. Công nghệ nhận dạng khuôn mặt có thể xác định khuôn mặt người trong hình ảnh hoặc video, xác định xem khuôn mặt xuất hiện trong hai hình ảnh có phải là cùng một người không hoặc tìm kiếm khuôn mặt trong một bộ sưu tập đồ sộ các hình ảnh hiện có. Các hệ thống bảo mật sinh trắc học sử dụng công nghệ nhận dạng khuôn mặt để nhận dạng cá nhân độc nhất trong

lúc triển khai người dùng hoặc đăng nhập, cũng như để tăng cường cho hoạt động xác thực người dùng. Các thiết bị di động và cá nhân cũng thường sử dụng công nghệ phân tích khuôn mặt để bảo mật thiết bị.

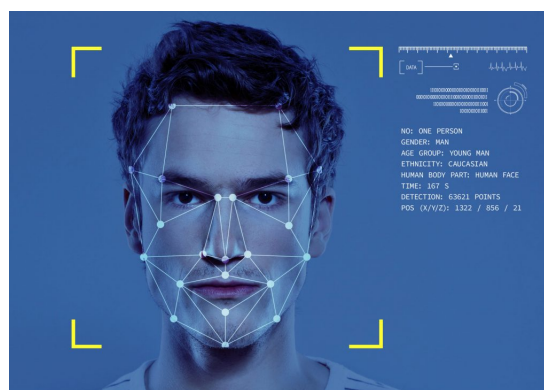


Fig. 1. Face recognition

Trước đây, khi chúng ta xây dựng các hệ thống nhận dạng khuôn mặt thì việc này đòi hỏi chúng ta phải có kiến chuyên sâu về các quy trình xử lý các sub-tasks trong pipeline process của nhận dạng khuôn mặt như phát hiện khuôn mặt, trích xuất đặc trưng,... Nhưng với sự phát triển của các mạng học sâu thì việc này ngày càng trở nên dễ dàng hơn.

Chúng ta có triển khai xây dựng một ứng dụng nhận dạng khuôn mặt mà không đòi hỏi quá nhiều về hiểu quy trình đằng sau nó bằng cách sử dụng các thư viện đã xây dựng sẵn các API. Chúng ta hoàn toàn có thể giải quyết những tác vụ con chỉ vài dòng mã hóa

Trong báo cáo này, chúng tôi sẽ sử dụng thư viện face-recognition cho toàn bộ pipeline process của nhận dạng khuôn mặt và sử dụng Streamlit để deploy thành web/app.

Chính xác hơn, những nội dung của báo cáo này bao gồm:

- Tiếp cận bài toán nhận dạng khuôn mặt

- Giới thiệu thư viện face-reconition và các hỗ trợ từ thư viện cho bài toán
- Triển khai mã hóa và deploy ứng bằng Streamlit
- Hướng cải thiện và phát triển trong tương lai

II. TIẾP CẬN KỸ THUẬT

Trong phần này, đầu tiên chúng ta sẽ xem xét tổng quan về pipeline process của bài toán nhận khuôn mặt bao gồm các sub-tasks phát hiện khuôn mặt, trích xuất đặc trưng và sử dụng các đặc trưng đã được trích xuất để nhận diện khuôn mặt.

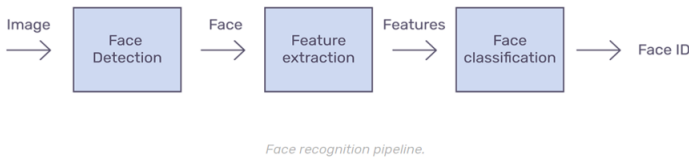


Fig. 2. Pipeline Process

Đầu tiên chúng ta sẽ nhận vào một hình ảnh(hoặc video chứa khuôn mặt) sau đó mô hình sẽ tiến trích xuất các khuôn mặt xuất hiện trong hình ảnh. Các hình ảnh khuôn mặt được trích xuất sẽ chuyển sang một mô hình khác để trích xuất các đặc trưng từ hình ảnh khuôn mặt và cuối cùng chúng sẽ sử dụng các biện so sánh khoảng cách hoặc sự tương đồng để xác định khuôn mặt.

A. Face Detection

Phát hiện khuôn mặt là quá trình tự động định vị khuôn mặt người trong phương tiện trực quan (hình ảnh kỹ thuật số hoặc video). Một khuôn mặt được phát hiện được báo cáo tại một vị trí có kích thước và hướng liên quan



Fig. 3. Example

Chúng ta có một số mô hình phổ biến bài toán phát hiện khuôn mặt như Dlib, MTCNN,.. Các mô hình này chủ sử dụng các biện học sâu huấn luyện mô hình.

B. Features Extraction

Trích xuất đặc trưng là quá trình trích xuất các tính năng quan trọng từ một khuôn mặt để tạo thành một biểu diễn số học (vector) mà máy tính có thể sử dụng để nhận dạng và phân loại các khuôn mặt khác nhau

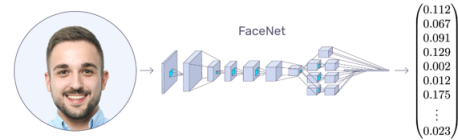


Fig. 4. Example

Mô hình lấy hình ảnh khuôn mặt của người đó làm đầu vào và xuất ra một vector gồm 128 số đại diện cho các đặc điểm quan trọng nhất của khuôn mặt. Trong học máy, vector này được gọi là nhúng

Những con số trong vector nhúng có nghĩa là gì? Kích thước của mắt? Khoảng cách giữa mũi và mắt? Chiều rộng miệng? Có lẽ... những tính năng này có vẻ quan trọng đối với nhận dạng khuôn mặt, nhưng trên thực tế, chúng tôi không thực sự biết những con số này đại diện cho điều gì và thực sự rất khó để diễn giải chúng. Nhìn chung, các vector nhúng sẽ biểu diễn các đặc trưng quan trọng nhất của khuôn mặt. Khi các vector nhúng được thì tạo ra thì nhận dạng khuôn sẽ là vấn đề so sánh các vector với nhau

C. Face Classification

Chúng tôi sẽ dựa trên các vector nhúng đã trích xuất để tiến hành phân loại khuôn mặt. Một số phương pháp phổ biến như Cosine Similarity, Euclidean Distance

Quá trình so sánh vector đặc trưng thường đi kèm với một ngưỡng (threshold) để quyết định xem hai vector có tương đồng đến mức nào. Trong bài toán này chúng ta sẽ sử dụng ngưỡng gọi là Tolerance (ngưỡng chấp nhận). Khi sử dụng Tolerance, hai vector đặc trưng được coi là tương đồng nếu khoảng cách hoặc độ tương đồng giữa chúng nhỏ hơn hoặc bằng giá trị ngưỡng đã được định trước. Nếu khoảng cách hoặc độ tương đồng vượt qua giá trị ngưỡng, hai vector được coi là khác nhau.

III. TRIỂN KHAI

Trong phần này chúng tôi sẽ tiến hành triển khai mã hóa cho ứng dụng nhận dạng khuôn mặt bằng cách sử dụng thư viện face-recognition. Về ý tưởng chúng ta có 3 thành phần, thứ nhất là bộ lưu trữ, thứ hai sẽ là bộ xử lý các sub-tasks của bài toán và cuối cùng giao diện .

A. Bộ lưu trữ

Trong phần này, chúng tôi sử dụng module Pickle để lưu trữ dữ liệu. Module pickle cho phép chúng ta lưu trữ các đối tượng Python (như danh sách, từ điển, đối tượng tùy chỉnh, v.v.) thành các tệp tin nhị phân và sau đó khôi phục lại chúng thành đối tượng Python ban đầu. Quá trình này được gọi là "pickling" (lưu trữ) và "unpickling" (khôi phục).

Chúng ta sẽ lưu trữ các thông tin cần thiết của một đối tượng bao gồm các trường như ID, Name, Image và Vector đặc trưng khuôn mặt của họ. Khi quá trình này được tạo, các dữ liệu sẽ được ghi dưới dạng nhị phân và được định dạng <file>.pkl

Khi chúng ta muốn sử dụng các file này cho các tác vụ như thêm, sửa, xóa và nhận dạng thì chúng ta tiến hành khôi phục trong suốt quá trình sử dụng.

B. Bộ Xử Lý

Như đã đề cập, chúng tôi sử dụng thư viện face-recognition. Đây là thư viện tích hợp sẵn các API để trích xuất tính năng, nhận diện khuôn mặt, đo khoảng cách giữa các tính năng,... Thư viện nhận dạng khuôn mặt đơn giản nhất thế giới Được xây dựng dựa trên một thư viện khác được viết bằng C++ - Dlib

Thư viện face-recognition cung cấp cho chúng ta API để xử lý các tác vụ một cách đơn giản chỉ với vài dòng mã.

```
face_locations = frg.face_locations(image)
face_encodings = frg.face_encodings(image, face_locations)
for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
    matches = frg.compare_faces(known_encoding, face_encoding, tolerance=TOLERANCE)
    distance = frg.face_distance(known_encoding, face_encoding)
```

Fig. 5. APIs Support

Đoạn mã trên xử lý cho 4 tác vụ bao gồm: phát hiện khuôn mặt, trích xuất đặc trưng, so sánh và tính khoảng cách với các khuôn mặt đã encode trước

C. Giao diện

Chúng tôi sẽ dùng Streamlit cho giao diện điều khiển. Streamlit là một thư viện Python mã nguồn mở cho các tác vụ xây dựng và triển khai ứng dụng. Nó tương thích với các thư viện Python chính như scikit-learning, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib,...



Fig. 6. Streamlit

Ở đây, điều quan trọng là deploy mô hình của chúng tôi theo cách mà bất kỳ ai cũng có thể dễ dàng sử dụng. Có các tùy chọn khác nhau như khung Django và Flask trong Python để triển khai các mô hình của chúng tôi. Trong khi sử dụng các khung này, chúng ta cũng cần có kiến thức về xây dựng giao diện người dùng bằng HTML, CSS và JS.

D. Kết quả triển khai

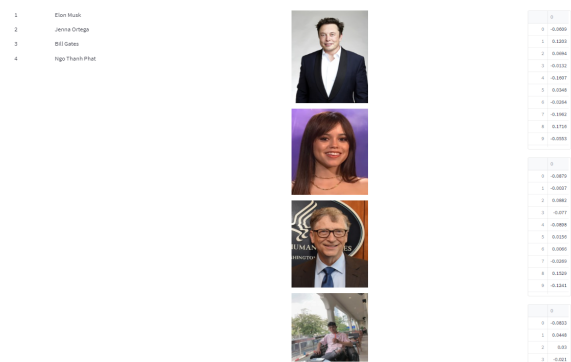


Fig. 7. Bộ lưu trữ

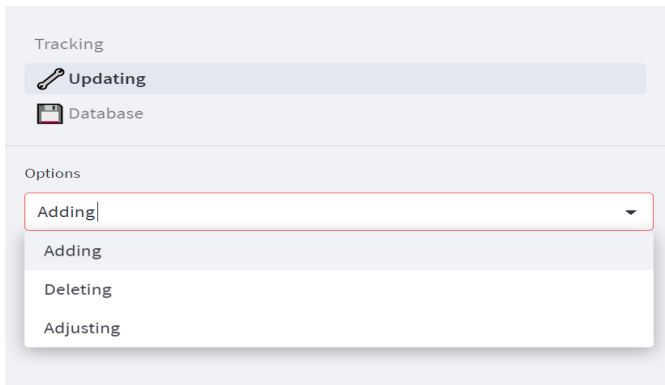


Fig. 8. Thao tác trên bộ lưu trữ

IV. HƯỚNG CẢI THIẾN VÀ PHÁT TRIỂN

Bộ lưu trữ: Để hiệu quả cho việc lưu trữ và xử lý trong các hệ thống lớn hơn chúng ta cần sử dụng các cơ sở dữ liệu hiện đại như MySQL, PostgreSQL,...

Việc chúng ta chỉ sử dụng một khuôn mặt làm đại diện cho một người sẽ tiện ích trong thực tế. Tuy nhiên, trên các ứng dụng có quy mô lớn hơn (số lượng người lớn) thì use cases xấu nhất là việc xảy ra sự nhầm lẫn với nhau. Việc này đến từ đặc trưng các khuôn mặt (giả sử là người Châu Á). Chúng ta nên sử dụng nhiều khuôn mặt hơn để biểu diễn cho một người

Tính năng ứng dụng: Để triển khai vào thực tế, chúng ta cần tổ chức các bảng trong cơ sở dữ liệu chi tiết hơn như có trường dữ liệu Time In, Time Out, Include Time(Day, Week, Month) để dễ dàng quản lý hơn

REFERENCES

- [1] <https://face-recognition.readthedocs.io/en/latest/readme.html>
- [2] <https://github.com/datct00/Face-recognition-app-using-Streamlit>
- [3] <https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/>
- [4] <https://viso.ai/computer-vision/deepface/>

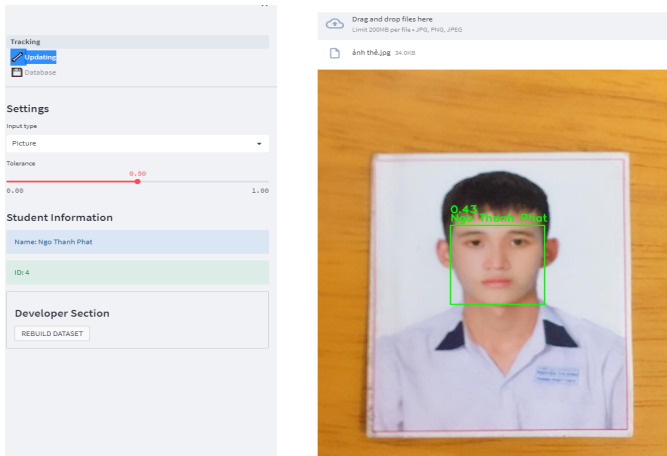


Fig. 9. Nhận dạng - hình ảnh

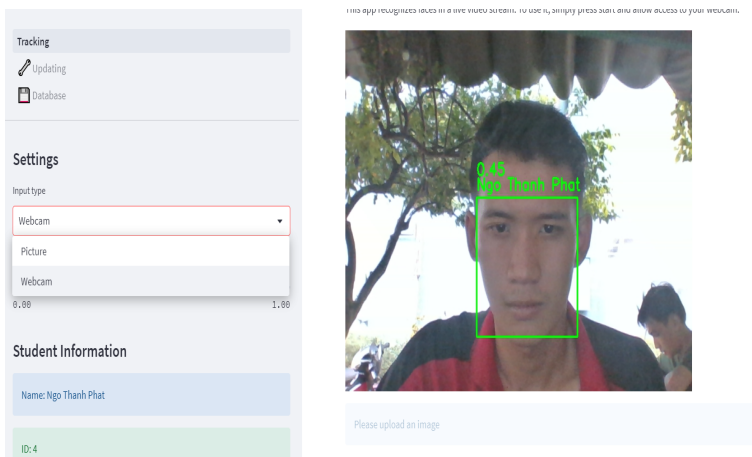


Fig. 10. Nhận dạng - webcam