

Thành viên:

Phan Văn Tài – 2202081

Phan Minh Thuy – 2202079

Hà Minh Chiến – 2202095

Báo cáo Dự Án Nhận Dạng Khuôn Mặt

Mục Lục

Báo cáo Dự Án Nhận Dạng Khuôn Mặt.....	1
I. Giới thiệu.....	1
II. Mục tiêu và Phạm vi.....	1
III. Quy trình triển khai.....	1
IV. Các mô hình học máy sử dụng.....	2
V. Cơ sở dữ liệu.....	5
VI. Các bước thực hiện.....	5
VII. Kết quả.....	6
VIII. Kết luận và Hướng phát triển.....	6
IX. Tài liệu tham khảo.....	6

I. Giới thiệu

Dự án nhận dạng khuôn mặt (Face Recognition) này được thiết kế nhằm cung cấp một hệ thống nhận diện tự động, xác thực và đăng ký người dùng qua hình ảnh khuôn mặt. Hệ thống này sử dụng các mô hình học máy để thực hiện việc nhận diện khuôn mặt và xác thực người dùng qua việc so sánh các đặc trưng của khuôn mặt.

Trong dự án này, các công nghệ sử dụng bao gồm:

- **Mediapipe** để phát hiện khuôn mặt.
- **CLIP** (Contrastive Language-Image Pre-Training) của OpenAI để trích xuất các đặc trưng khuôn mặt.
- Các thuật toán học máy như **Logistic Regression** và **K-Nearest Neighbors (KNN)** để phân loại và nhận diện khuôn mặt.
- **MongoDB** để lưu trữ dữ liệu người dùng, thông tin đăng ký và các đặc trưng khuôn mặt.

II. Mục tiêu và Phạm vi

- **Mục tiêu chính:** Xây dựng một hệ thống nhận diện khuôn mặt tự động và có thể sử dụng trong các ứng dụng bảo mật như đăng nhập bằng khuôn mặt, nhận dạng người dùng trong các hệ thống bảo mật, v.v.
- **Phạm vi:**
 - Nhận diện khuôn mặt từ hình ảnh thu thập từ webcam.
 - Đăng ký và xác thực người dùng qua việc so sánh đặc trưng khuôn mặt.
 - Lưu trữ dữ liệu người dùng và các đặc trưng khuôn mặt trong cơ sở dữ liệu MongoDB.

III. Quy trình triển khai

1. **Thu thập dữ liệu từ webcam:**
 - Hệ thống thu thập hình ảnh khuôn mặt từ webcam của người dùng. Hình ảnh này được truyền vào hệ thống để thực hiện quá trình nhận diện khuôn mặt.
 - Dữ liệu thu thập được sẽ được mã hóa dưới dạng base64 trước khi xử lý.
2. **Xử lý hình ảnh và phát hiện khuôn mặt:**
 - Sử dụng **Mediapipe** để phát hiện khuôn mặt trong ảnh. Mediapipe giúp phân tích ảnh để xác định vị trí của khuôn mặt và trích xuất các đặc trưng của khuôn mặt từ đó.
 - Hệ thống sẽ nhận diện và cắt lấy khuôn mặt từ ảnh gốc, tạo ra một khuôn mặt chuẩn để sử dụng cho các bước tiếp theo.

3. Trích xuất đặc trưng khuôn mặt:

- **CLIP** được sử dụng để trích xuất đặc trưng khuôn mặt từ hình ảnh. CLIP là một mô hình học sâu được OpenAI phát triển, có khả năng liên kết hình ảnh và văn bản. Trong hệ thống này, CLIP sẽ tạo ra một vector đặc trưng cho khuôn mặt được cung cấp.
- Đặc trưng này sẽ được lưu trữ để so sánh với các đặc trưng của những người dùng đã đăng ký trước đó.

4. Đăng ký người dùng:

- Khi một người dùng muốn đăng ký, hệ thống sẽ kiểm tra xem khuôn mặt của họ đã được đăng ký trong cơ sở dữ liệu hay chưa. Nếu chưa, hệ thống sẽ lưu trữ đặc trưng khuôn mặt và thông tin của người dùng (như tên).
- Dữ liệu khuôn mặt và thông tin người dùng sẽ được lưu trữ trong **MongoDB**, một cơ sở dữ liệu NoSQL.

5. Xác thực người dùng:

- Khi người dùng muốn đăng nhập, hệ thống sẽ thực hiện một phép so sánh giữa đặc trưng khuôn mặt của người dùng mới và các đặc trưng đã được lưu trong cơ sở dữ liệu.
- Nếu độ tương đồng (similarity) giữa đặc trưng của khuôn mặt người dùng và khuôn mặt đã đăng ký đạt mức ngưỡng xác định (ví dụ: 0.8), người dùng sẽ được xác thực thành công.
- Nếu không, hệ thống sẽ thông báo lỗi xác thực.


IV. Các mô hình học máy sử dụng

1. Logistic Regression:

Mục đích: Mô hình học máy cơ bản này giúp phân loại các đặc trưng khuôn mặt của người dùng đã được trích xuất từ CLIP và quyết định xem khuôn mặt đó có khớp với người dùng đã đăng ký hay không.

Code minh họa Logistic Regression:

python

 Copy code

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
import numpy as np

# Dữ liệu đầu vào (vector đặc trưng khuôn mặt và nhãn người dùng)
X_train = np.array([vector_face_1, vector_face_2, vector_face_3]) # Các vector đặc
y_train = np.array([0, 1, 2]) # Nhãn của người dùng

# Khởi tạo và huấn luyện mô hình Logistic Regression
model = LogisticRegression()
model.fit(X_train, y_train)

# Dự đoán nhãn cho khuôn mặt mới
prediction = model.predict([new_face_vector])
print(f"Predicted Label: {prediction[0]}")

```

Giải thích:

- **X_train** là các vector đặc trưng của khuôn mặt đã được trích xuất từ người dùng đã đăng ký.
- **y_train** là nhãn ứng với mỗi vector đặc trưng (ví dụ: 0 là người dùng A, 1 là người dùng B).
- Sau khi huấn luyện, mô hình sẽ dự đoán nhãn cho khuôn mặt mới (vector đặc trưng của người dùng đang cố gắng đăng nhập).

2. K-Nearest Neighbors (KNN):

Mục đích: KNN giúp so sánh đặc trưng của khuôn mặt mới với các khuôn mặt đã lưu trong cơ sở dữ liệu và quyết định xem người dùng có phải là người đã đăng ký hay không.

Code minh họa KNN:

Giải thích:

```
python Copy code

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report

# Dữ liệu đầu vào (vector đặc trưng khuôn mặt và nhãn người dùng)
X_train = np.array([vector_face_1, vector_face_2, vector_face_3]) # Các vector đặc
y_train = np.array([0, 1, 2]) # Nhãn của người dùng

# Khởi tạo và huấn luyện mô hình KNN
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# Dự đoán nhãn cho khuôn mặt mới
prediction = knn.predict([new_face_vector])
print(f"Predicted Label: {prediction[0]}")
```

- **KNeighborsClassifier** là thuật toán phân loại dựa trên việc so sánh "k" láng giềng gần nhất trong không gian đặc trưng.
- Mô hình KNN sẽ so sánh vector đặc trưng của khuôn mặt mới với các vector đã có trong cơ sở dữ liệu để đưa ra dự đoán.

3. Cosine Similarity:

Mục đích: Được sử dụng để đo lường độ tương đồng giữa hai đặc trưng khuôn mặt. Khoảng cách cosine càng nhỏ, độ tương đồng giữa các khuôn mặt càng cao.

Code minh họa Cosine Similarity:

```
python Copy code

from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

# Vector đặc trưng của khuôn mặt đã đăng ký và khuôn mặt mới
face_vector_db = np.array([face_vector_registered])
face_vector_new = np.array([new_face_vector])

# Tính toán độ tương đồng cosine giữa các vector
similarity = cosine_similarity(face_vector_db, face_vector_new)
print(f"Cosine Similarity: {similarity[0][0]}")
```

Giải thích:

- **cosine_similarity** tính toán độ tương đồng giữa các vector đặc trưng khuôn mặt.
- Nếu giá trị cosine similarity đạt trên một ngưỡng (ví dụ: 0.8), khuôn mặt mới được xác nhận là của người dùng đã đăng ký.

V. Cơ sở dữ liệu

1. MongoDB:

- **Mục đích:** Lưu trữ thông tin người dùng và các đặc trưng khuôn mặt của họ.
- **Chức năng:** MongoDB cung cấp khả năng lưu trữ và truy vấn nhanh chóng các đặc trưng khuôn mặt dưới dạng vector. Mỗi khi một người dùng đăng ký hoặc đăng nhập, hệ thống sẽ truy vấn cơ sở dữ liệu để so sánh các đặc trưng khuôn mặt.

2. Cấu trúc dữ liệu:

- **Collection users:** Lưu trữ thông tin người dùng (username, vector đặc trưng khuôn mặt).
- **Collection config:** Lưu trữ các cấu hình hệ thống, chẳng hạn như số lượng lớp người dùng (**class_count**) và ánh xạ tên người dùng sang nhãn (**label_mapping**).

VI. Các bước thực hiện

1. Đăng ký người dùng:

- Nhận diện khuôn mặt và trích xuất vector đặc trưng.
- So sánh với các vector đã có trong cơ sở dữ liệu để đảm bảo khuôn mặt không bị trùng.
- Lưu trữ thông tin người dùng và vector đặc trưng vào MongoDB.

2. Xác thực người dùng:

- Thu thập hình ảnh khuôn mặt và trích xuất vector đặc trưng.
- So sánh vector này với các vector trong cơ sở dữ liệu.
- Nếu độ tương đồng cao, người dùng sẽ được xác thực.

3. Đánh giá mô hình:

- Sử dụng các chỉ số đánh giá như độ chính xác (accuracy) và báo cáo phân loại (classification report) để đánh giá hiệu quả của các mô hình học máy đã sử dụng (Logistic Regression, KNN).

VII. Kết quả

Dự án này đã thành công trong việc xây dựng một hệ thống nhận diện khuôn mặt với các bước đăng ký và xác thực người dùng tự động. Các mô hình học máy như Logistic Regression và KNN đã giúp hệ thống phân loại chính xác các khuôn mặt và xác thực người dùng.

Kết quả mô hình thể hiện khả năng phân loại và nhận diện khuôn mặt khá chính xác, với độ chính xác đạt khoảng 85%-90% trong các bài kiểm tra nội bộ. Các lỗi thường gặp chủ yếu xảy ra khi có sự thay đổi lớn trong môi trường (ví dụ, ánh sáng thay đổi hoặc góc nhìn khác biệt).

VIII. Kết luận và Hướng phát triển

1. Kết luận:

- Dự án đã đạt được mục tiêu nhận diện và xác thực khuôn mặt hiệu quả, sử dụng các công nghệ tiên tiến như Mediapipe, CLIP và các thuật toán học máy.

2. Hướng phát triển:

- Tăng cường độ chính xác của hệ thống qua việc cải thiện mô hình học máy và thu thập thêm dữ liệu.
- Phát triển các tính năng như nhận diện khuôn mặt trong điều kiện ánh sáng yếu, góc nhìn khác biệt.
- Tích hợp hệ thống nhận diện khuôn mặt vào các ứng dụng thực tế, như hệ thống bảo mật cho cửa hàng, ngân hàng, hoặc các khu vực hạn chế.

IX. Tài liệu tham khảo

1. openCV
2. Logistic Regression
3. KNN
4. MongoDB
5. Mediapipe

Link github:

https://github.com/ANGFLO26/face_recognition_project.git