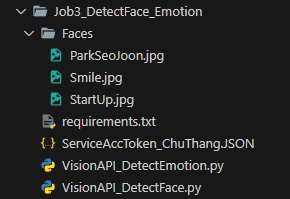
**GIẢI THÍCH MÃ PYTHON SỬ DỤNG GOOGLE CLOUD VISION API ĐỂ PHÁT HIỆN KHUÔN MẶT VÀ VẼ HÌNH VUÔNG XUNG QUANH CHÚNG**

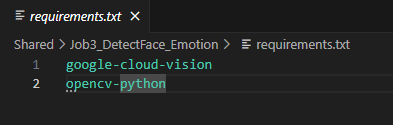
1. **Mục đích của đoạn mã:**

Đoạn mã Python này sử dụng Google Cloud Vision API để phân tích một hình ảnh và phát hiện các khuôn mặt trong hình ảnh đó. Sau đó, nó đánh giá cảm xúc trên mỗi khuôn mặt, như cảm giác tức giận (angry), vui vẻ (joy), buồn (sorrow), ngạc nhiên (surprise), và việc đội mũ (headwear). Kết quả cuối cùng là việc hiển thị thông tin về cảm xúc trên mỗi khuôn mặt trong hình ảnh.

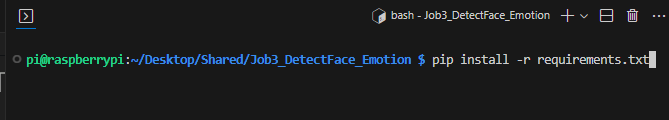
1. **Cấu trúc thư mục**

****

1. **Cài đặt các gói cần thiết**

Chúng ta cần cài đặt các gói cần thiết như sau. Trước hết ta sẽ để những gói cần thiết vào 1 file mang tên **requirements.txt**. Trong tệp cần có tên các gói như hình dưới  


Sau đó sẽ gõ lệnh sau đây trên terminal: **pip install -r requirements.txt**



1. **Thư viện được sử dụng:**

* **os** : Thư viện này được sử dụng để làm việc với hệ thống tệp và thư mục.
* **io** : Thư viện này được sử dụng để đọc nội dung của tệp hình ảnh.
* **google.cloud.vision\_v1**: Thư viện này chứa các công cụ cho việc gửi hình ảnh đến Google Cloud Vision API và nhận kết quả.

1. **Quy trình chính:**

1. Đoạn mã bắt đầu bằng việc thiết lập đường dẫn đến tệp JSON chứa thông tin xác thực dịch vụ của Google Cloud Vision API. Đường dẫn này được lưu trong biến `key\_path`, sau đó được đặt trong biến môi trường `GOOGLE\_APPLICATION\_CREDENTIALS` để xác thực với API.

|  |
| --- |
| key\_path = '/home/pi/Desktop/Shared/GoogleVisionAPI/ServiceAccToken\_ChuThang.JSON'  os.environ['GOOGLE\_APPLICATION\_CREDENTIALS'] = key\_path |

2. Sau đó, đoạn mã tạo một phiên bản của `ImageAnnotatorClient` để gửi hình ảnh đến Google Cloud Vision API và nhận kết quả.

|  |
| --- |
| client = vision\_v1.ImageAnnotatorClient() |

3**. Hàm `detectEmotion(FILE\_PATH)`** được định nghĩa để thực hiện công việc phát hiện khuôn mặt và vẽ hình vuông xung quanh chúng. Hàm này nhận đối số `FILE\_PATH`, đó là đường dẫn đến tệp hình ảnh cần xử lý.

|  |
| --- |
| def detectFace(FILE\_PATH): |

* 1. **Đọc nội dung của hình ảnh**: Hàm bắt đầu bằng việc mở tệp hình ảnh từ đường dẫn **FILE\_PATH** và đọc nội dung của nó bằng cách sử dụng **io.open** trong chế độ đọc nhị phân **('rb').** Nội dung hình ảnh được lưu trong biến content.

|  |
| --- |
| with io.open(FILE\_PATH, 'rb') as image\_file:  content = image\_file.read() |

* 1. **Tạo đối tượng hình ảnh**: Sử dụng **vision\_v1.Image** để tạo một đối tượng hình ảnh từ nội dung của tấm hình. Điều này chuẩn bị dữ liệu hình ảnh để được gửi đến Google Cloud Vision API để phân tích

|  |
| --- |
| image = vision\_v1.Image(content=content) |

* 1. **Gửi hình ảnh đến Google Cloud Vision API**: Hàm sử dụng **client.face\_detection(image=image)** để gửi đối tượng hình ảnh đã chuẩn bị đến Google Cloud Vision API và yêu cầu phân tích khuôn mặt trong hình ảnh. Kết quả phân tích được lưu trong biến response.

|  |
| --- |
| response = client.face\_detection(image=image) |

* 1. **Trích xuất thông tin khuôn mặt**: Từ kết quả trả về (**response**), hàm trích xuất danh sách các khuôn mặt được phát hiện và lưu chúng trong biến **faces**. Mỗi khuôn mặt được biểu diễn bằng một đối tượng **face\_annotations**.

|  |
| --- |
| faces = response.face\_annotations |

* 1. **Khởi tạo danh sách các cảm xúc**: Hàm khởi tạo một danh sách chứa các nhãn cảm xúc từ 'UNKNOWN' đến 'VERY LIKELY'. Biến **likehood** chứa danh sách này.

|  |
| --- |
| likehood = ('UNKNOWN', 'VERY UNLIKELY', 'UNLIKELY', 'POSSIBLY', 'LIKELY', 'VERY LIKELY') |

* 1. **In thông tin kết quả**: Hàm bắt đầu bằng việc in ra số lượng khuôn mặt được phát hiện trong hình ảnh bằng cách sử dụng **len(faceAnnotation).** Sau đó, nó duyệt qua từng khuôn mặt trong biến **faceAnnotation** và in ra thông tin về cảm xúc trên từng khuôn mặt.
* **Detection** **confidence**: Đây là tỷ lệ xác định khuôn mặt, được làm tròn và hiển thị dưới dạng phần trăm (ví dụ: 95.0%).
* **Angry**, **Joy**, **Sorrow**, **Surprise**, **Headwear**: Đây là các cảm xúc hoặc sự xuất hiện của đội mũ trên khuôn mặt, được hiển thị dựa trên nhãn từ danh sách likehood.

|  |
| --- |
| print(f"Found {len(faceAnnotation)} face.")  for face in faceAnnotation:  print(f"Faces:")  print(f' Detection confidence : {round(face.detection\_confidence \* 100, 2)}')  print(f' Angry : {likehood[face.anger\_likelihood]}')  print(f' Joy : {likehood[face.joy\_likelihood]}')  print(f' Sorrow : {likehood[face.sorrow\_likelihood]}')  print(f' Sup : {likehood[face.surprise\_likelihood]}')  print(f' Headwear : {likehood[face.headwear\_likelihood]}\n') |

1. **Sử dụng mã:**

* Để sử dụng đoạn mã này, bạn cần cung cấp đường dẫn đến một tệp hình ảnh muốn xử lý bằng cách đặt đường dẫn này vào biến `**file\_path**`.

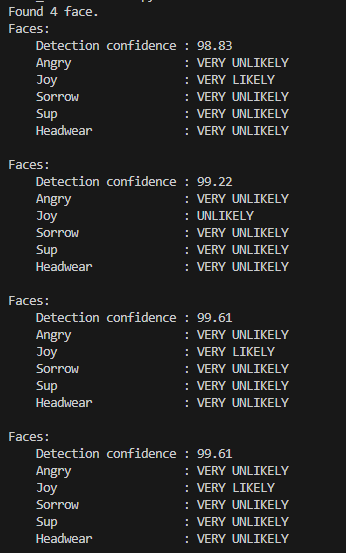
|  |
| --- |
| img\_name = 'StartUp.jpg'  file\_path = f'./Faces/{img\_name}' |

* Sau đó, gọi hàm ` **detectEmotion(file\_path)** để thực hiện phát hiện khuôn mặt và vẽ hình vuông.

|  |
| --- |
| detectEmotion(file\_path) |

* Kết quả sau cùng sẽ như sau:





**CHI TIẾT CODE**

|  |
| --- |
| # Import các thư viện cần thiết  import os, io  from google.cloud import vision\_v1  # Thiết lập đường dẫn đến tệp JSON chứa thông tin xác thực dịch vụ của Google Cloud Vision API  key\_path = '/home/pi/Desktop/Shared/GoogleVisionAPI/ServiceAccToken\_ChuThang.JSON'  # Đặt biến môi trường để xác thực với Google Cloud Vision API  os.environ['GOOGLE\_APPLICATION\_CREDENTIALS'] = key\_path  # Tạo một phiên bản của ImageAnnotatorClient để gửi hình ảnh đến Google Cloud Vision API  client = vision\_v1.ImageAnnotatorClient()  # Định nghĩa hàm để phân tích cảm xúc trong khuôn mặt  def detectEmotion(FILE\_PATH):  with io.open(FILE\_PATH, 'rb') as image\_file:  content = image\_file.read()  image = vision\_v1.Image(content=content)  response = client.face\_detection(image=image)  faceAnnotation = response.face\_annotations    likehood = ('UNKNOWN', 'VERY UNLIKELY', 'UNLIKELY', 'POSSIBLY', 'LIKELY', 'VERY LIKELY')  print(f"Found {len(faceAnnotation)} face.")  for face in faceAnnotation:  print(f"Faces:")  print(f' Detection confidence : {round(face.detection\_confidence \* 100, 2)}')  print(f' Angry : {likehood[face.anger\_likelihood]}')  print(f' Joy : {likehood[face.joy\_likelihood]}')  print(f' Sorrow : {likehood[face.sorrow\_likelihood]}')  print(f' Sup : {likehood[face.surprise\_likelihood]}')  print(f' Headwear : {likehood[face.headwear\_likelihood]}')  print()  # Tên hình ảnh cần xử lý  img\_name = 'StartUp.jpg'  # Đường dẫn đến tệp hình ảnh  file\_path = f'./Faces/{img\_name}'  # Gọi hàm để phân tích cảm xúc trong khuôn mặt  detectEmotion(file\_path) |