KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG - UIT

BÁO CÁO THỰC HÀNH

Lab 6

CÔNG NGHỆ INTERNET OF THINGS HIỆN ĐẠI

HOÀN THIỆN GIẢI PHÁP IOT

GVHD: **Phan Trung Phát**

Lớp: **NT532.O21**

Họ và tên	MSSV
Nguyễn Thành Đăng	21520683
Nguyễn Trần Bảo Quốc	21520421

ĐÁNH GIÁ KHÁC (*):

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình (1)	1,5 tuần
Link Video thực hiện (2) *(nếu có)*	Link tổng hợp
Ý kiến (3) *(nếu có)* + Khó khăn + Đề xuất ...	Phòng lab đóng cửa do
Điểm tự đánh giá (4)	10/10

(*): phần (1) và (4) bắt buộc thực hiện.

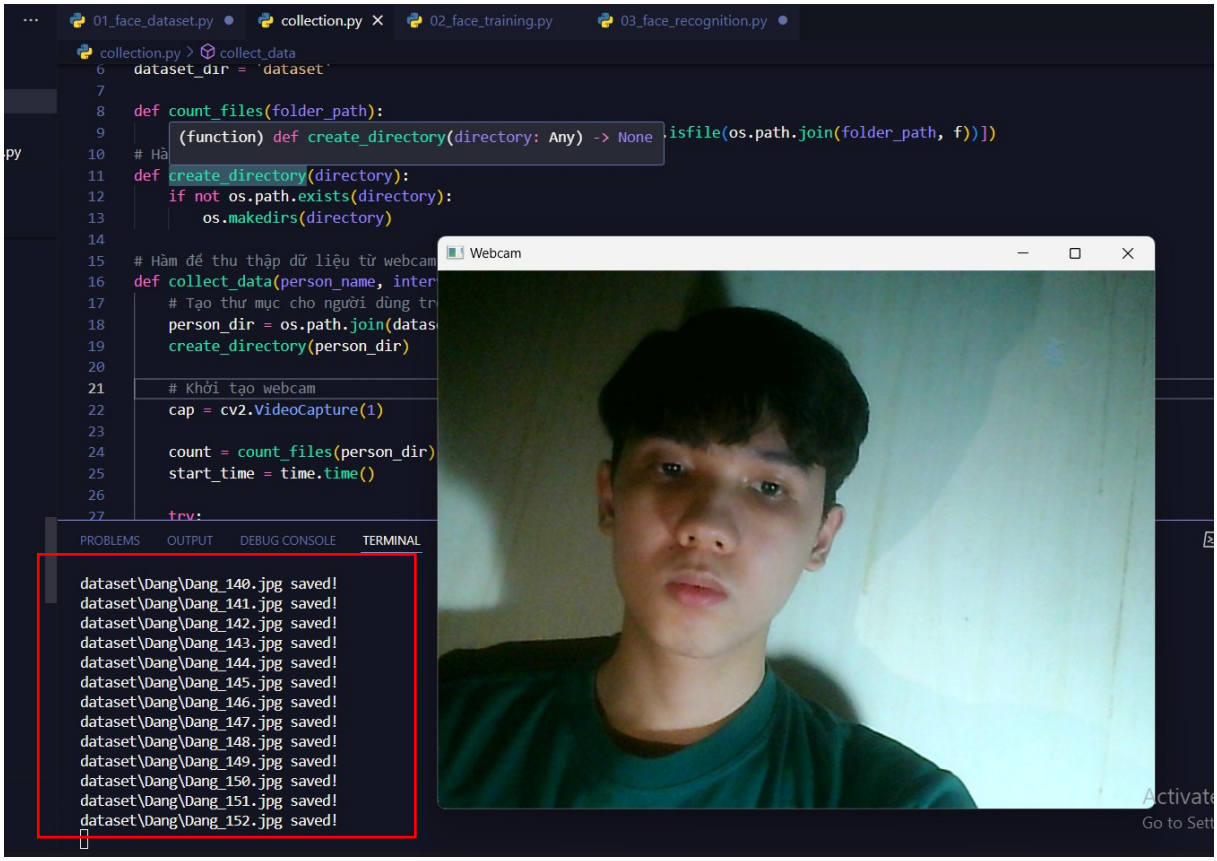
Phần bên dưới là báo cáo chi tiết của nhóm/cá nhân thực hiện.

WARNING LƯU HÀNH NỘI BỘ

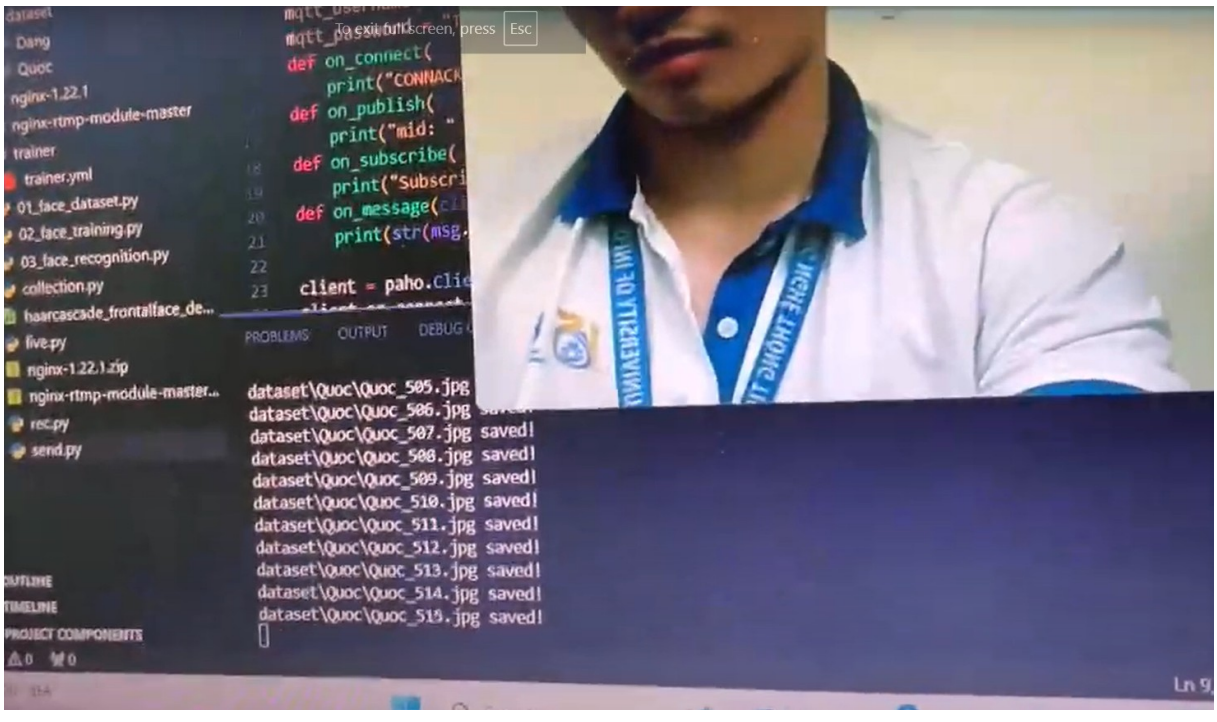
Câu hỏi 1: Thu thập hình ảnh các thành viên trong nhóm, sau đó huấn luyện lại mô hình Facenet với độ chính xác trên 60%.

1. Minh chứng

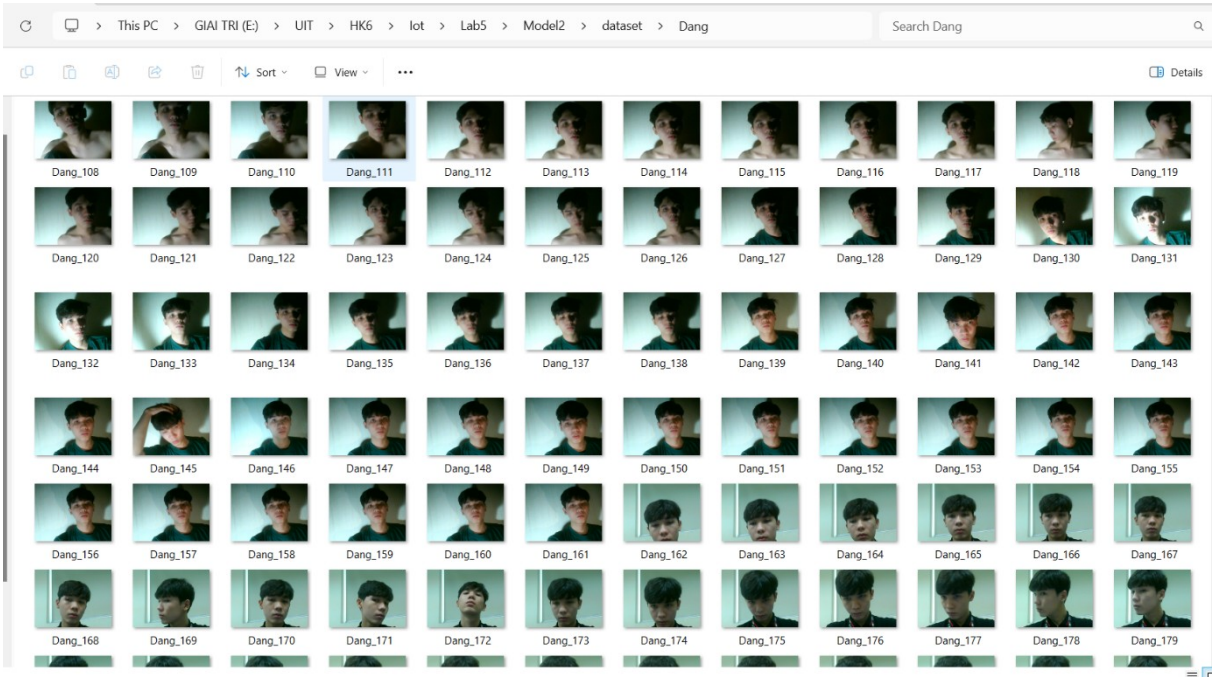
Thu thập dữ liệu



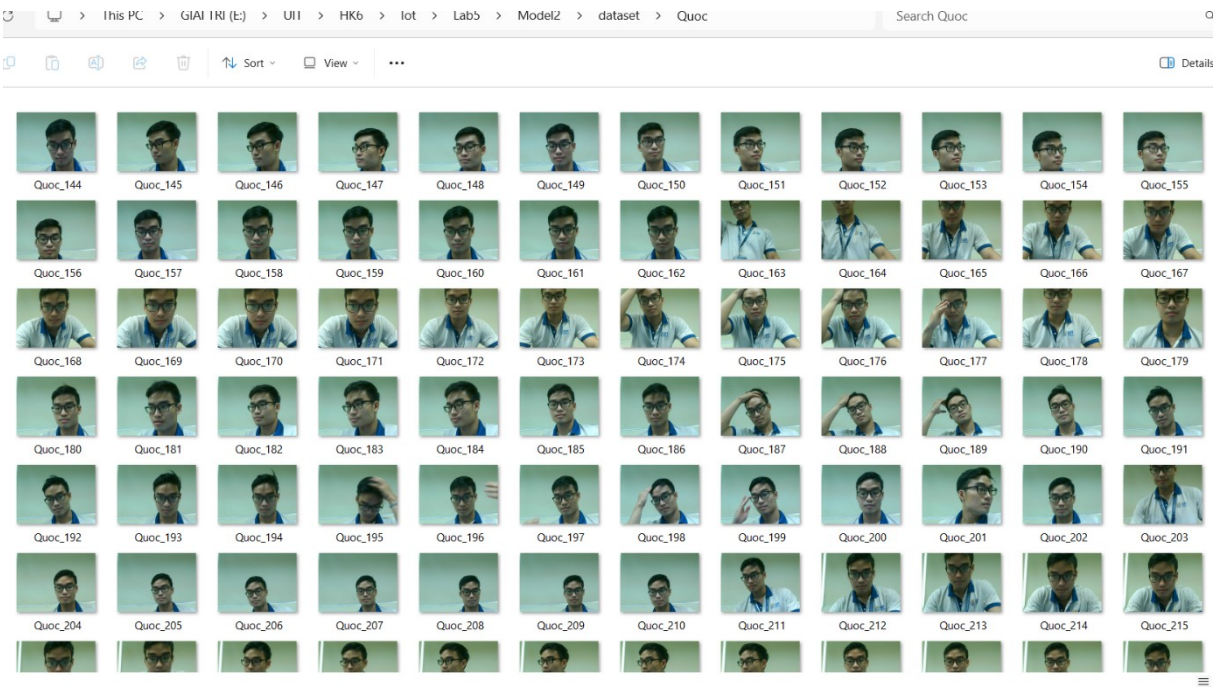
Chương trình tự động lấy dữ liệu (Đăng)



Chương trình tự động lấy dữ liệu (Quốc)



Dataset (Đăng)



Dataset (Quốc)

Train model nhận diện khuôn mặt

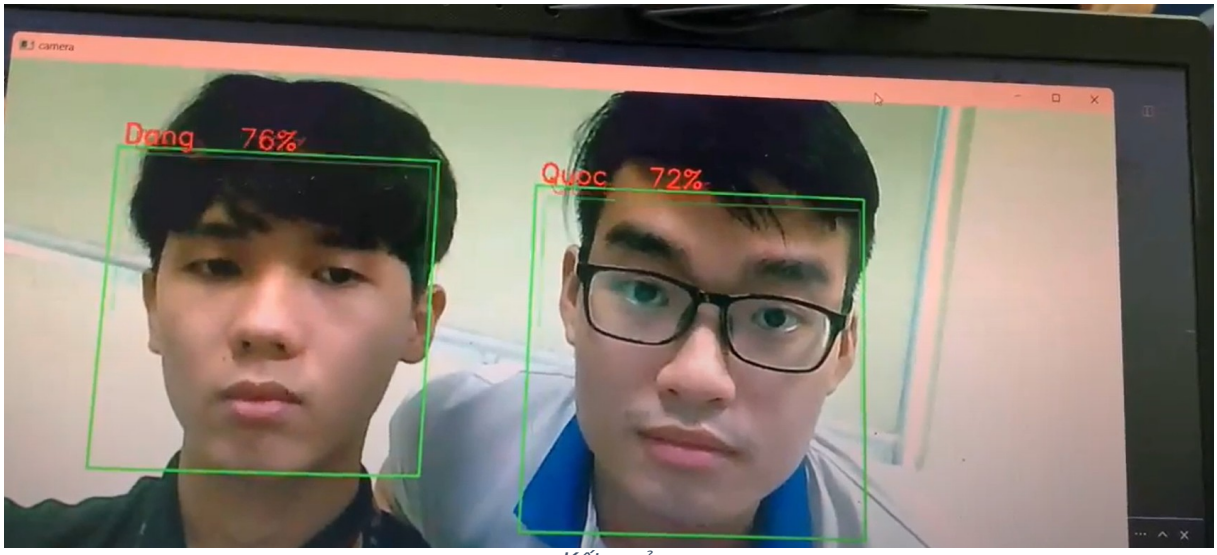
```python
def getImagesAndLabels(main_path):
    faceSamples = []
    ids = []
    folder_index = 0

    for user_folder in os.listdir(main_path):
        user_folder_path = os.path.join(main_path, user_folder)
        if os.path.isdir(user_folder_path):  # Check if it's a directory
            for image_name in os.listdir(user_folder_path):
                image_path = os.path.join(user_folder_path, image_name)
                if os.path.isfile(image_path):  # Check if it's a file
                    PIL_img = Image.open(image_path).convert('L')  # Convert it to grayscale
                    img_numpy = np.array(PIL_img, 'uint8')
                    id = folder_index  # Use the folder index as the ID
                    faces = detector.detectMultiScale(img_numpy)
                    for (x, y, w, h) in faces:
                        faceSamples.append(img_numpy[y:y + h, x:x + w])
                        ids.append(id)
            folder_index += 1

    return faceSamples, ids
```

Hàm training

Kết quả nhận diện khuôn mặt, với độ chính xác lần lượt là 76% và 72%.



Kết quả

2. Giải thích

a) Thu thập dữ liệu tự động

```python
collection.py > collect_data
import cv2
import os
import time

# Thư mục gốc lưu dataset
dataset_dir = 'dataset'
```

Khai báo thư viện và đường dẫn parent của dữ liệu sẽ train

```python
def count_files(folder_path):
    return len([f for f in os.listdir(folder_path) if os.path.isfile(os.path.join(folder_path, f))])
```

Hàm đếm số lượng file có trong folder

Hàm **count_files** này có tác dụng đếm số lượng file có trong **<Person>** để từ đó có thể thu thập data kế tiếp thay vì ghi đè các data đã có.

```
def create_directory(directory):
    if not os.path.exists(directory):
        os.makedirs(directory)
```

Hàm tạo thư mục nếu nó chưa tồn tại

```
person_name = input('Enter the name of the person: ')
interval = int(input('Enter the interval between captures (in seconds): '))
collect_data(person_name, interval)
```

Nhập dữ liệu đối tượng đang lấy dữ liệu

Ở đây, **person_name** sẽ là folder của người đang thu thập dữ liệu (nơi chứa các ảnh được tự động chụp vào), **interval** là số giây trên 1 bức hình được chụp.

```
def collect_data(person_name, interval=0.5):
    person_dir = os.path.join(dataset_dir, person_name)
    create_directory(person_dir)
    cap = cv2.VideoCapture(1)
    count = count_files(person_dir)
    start_time = time.time()
    try:
        while True:
            ret, frame = cap.read()
            if not ret:
                break
            flipped_frame = cv2.flip(frame, 1)
            cv2.imshow('Webcam', flipped_frame)
            if time.time() - start_time >= interval:
                img_name = os.path.join(person_dir, f'{person_name}_{count}.jpg')
                cv2.imwrite(img_name, frame)
                print(f'{img_name} saved!')
                count += 1
                start_time = time.time()

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    finally:
        cap.release()
        cv2.destroyAllWindows()
```

Hình 1. Hàm chụp tự động

Ở đây sẽ có tham số của người đang thu thập dữ liệu và khoảng thời gian chụp mỗi bức hình (được nhập vào).

Tiến hành chạy vòng lặp để chụp lần lượt các bức hình và đặt tên theo **person_nam**e cùng với số count (là số chỉ index của hình).

Nhấn phím ‘q’ để kết thúc chương trình.

b) Huấn luyện model

```
import cv2
import numpy as np
from PIL import Image
import os

pathData = 'dataset'
```

Khai báo thư viện và đường dẫn tập tin huấn luyện

pathData là đường dẫn của dataset sẽ huấn luyện.

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
```

Bộ nhận dạng khuôn mặt của opencv

- recognizer = cv2.face.LBPHFaceRecognizer_create() là một bộ nhận dạng khuôn mặt dựa trên thuật toán LBPH.
- detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml") là một bộ phát hiện khuôn mặt dựa trên Haar Cascade.

```
def getImagesAndLabels(main_path):
    faceSamples = []
    ids = []
    folder_index = 0

    for user_folder in os.listdir(main_path):
        user_folder_path = os.path.join(main_path, user_folder)
        if os.path.isdir(user_folder_path):  # Check if it's a directory
            for image_name in os.listdir(user_folder_path):
                image_path = os.path.join(user_folder_path, image_name)
                if os.path.isfile(image_path):  # Check if it's a file
                    PIL_img = Image.open(image_path).convert('L')  # Convert it to grayscale
                    img_numpy = np.array(PIL_img, 'uint8')
                    id = folder_index  # Use the folder index as the ID
                    faces = detector.detectMultiScale(img_numpy)
                    for (x, y, w, h) in faces:
                        faceSamples.append(img_numpy[y:y + h, x:x + w])
                        ids.append(id)
            folder_index += 1

    return faceSamples, ids
```

Chương trình xử lý, huấn luyện dữ liệu

1. Khởi tạo hai danh sách faceSamples và ids để lưu trữ các mẫu khuôn mặt và nhãn tương ứng. *(Line 12 – 14)*
2. Lặp qua các thư mục con trong thư mục main_path. *(Line 16 - 29)*

- Với mỗi thư mục con, coi đó là một người dùng và gán ID tương ứng.
- Lặp qua các tập tin ảnh trong thư mục con.
  - Mở từng tập tin ảnh, chuyển đổi sang ảnh xám.
  - Sử dụng detector.detectMultiScale() để phát hiện các khuôn mặt trong ảnh.
  - Với mỗi khuôn mặt được phát hiện, thêm nó vào faceSamples và thêm ID tương ứng vào ids.

3. Cuối cùng, trả về faceSamples và ids. *(Line 31)*

Hàm này có thể được sử dụng để chuẩn bị dữ liệu cho việc huấn luyện một bộ nhận dạng khuôn mặt.

```
faces,ids = getImagesAndLabels(pathData)
recognizer.train(faces, np.array(ids))

recognizer.write('trainer/trainer.yml') # recognizer.save() worked on Mac, but not on Pi

print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))
```

Lưu lại model

Câu hỏi 2: Sử dụng Apache Kafka làm kênh giao tiếp giữa Client và Edge / Cloud Server.

1. Minh chứng

Link cập nhật minh chứng và demo được update version (*Em sẽ trình bày bằng video thay vì text do nó khá giống lab5*) tại đây.

Câu 3. Kết quả nhận diện lưu ở trên IoT Platform

Link cập nhật minh chứng và demo được update version (*Em sẽ trình bày bằng video thay vì text do nó khá giống lab5*) tại đây.