

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HO CHÍ MINH  
KHOA KHOA HỌC MÁY TÍNH



**Đề Tài: NHẬN DIỆN KHUÔN MẶT SỬ DỤNG  
CẤU TRÚC MẠNG MTCNN VÀ FACENET**

GVHD: Đỗ Văn Tiến

Lớp: CS232.M21

Nhóm: Lê Hà Minh Trung 18520390

TP. HO CHÍ MINH, NGÀY 02 THÁNG 07 NĂM 2022

## MỤC LỤC

<b>CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN .....</b>	<b>4</b>
1.1. TỔNG QUAN VỀ NHẬN DIỆN KHUÔN MẶT .....	4
1.2. MỤC ĐÍCH ĐỀ TÀI.....	4
1.3. MÔ TẢ BÀI TOÁN .....	5
1.4. TẦM QUAN TRỌNG CỦA BÀI TOÁN NHẬN DIỆN KHUÔN MẶT .....	5
1.5. XÂY DỰNG HỆ THỐNG NHẬN DIỆN KHUÔN MẶT .....	6
<b>CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU.....</b>	<b>7</b>
2.1. MỤC ĐÍCH THU THẬP DỮ LIỆU .....	7
2.2. THU THẬP VÀ XỬ LÝ DỮ LIỆU .....	7
<b>CHƯƠNG 3: PHƯƠNG PHÁP ĐỀ XUẤT .....</b>	<b>9</b>
3.1. GIỚI THIỆU MULTITASK CONVOLUTION NEURAL NETWORK.....	9
3.2. CẤU TRÚC MTCNN.....	9
3.3. ƯU NHƯỢC ĐIỂM CỦA MẠNG MTCNN.....	11
<b>CHƯƠNG 4: THỰC NGHIỆM CHƯƠNG TRÌNH .....</b>	<b>12</b>
4.1. CÀI ĐẶT .....	12
4.2. QUÁ TRÌNH THỰC NGHIỆM .....	12
4.3. DEMO CHƯƠNG TRÌNH.....	18
4.4. KẾT QUẢ THỰC NGHIỆM.....	18
<b>CHƯƠNG 5: TỔNG KẾT.....</b>	<b>20</b>
5.1. KẾT LUẬN.....	20
5.2. KHÓ KHĂN .....	20
5.3. HƯỚNG PHÁT TRIỂN.....	20
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>21</b>
A. TÀI LIỆU TIẾNG VIỆT .....	21
B. TÀI LIỆU TIẾNG ANH.....	21



# **CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN**

## **1.1. TỔNG QUAN VỀ NHẬN DIỆN KHUÔN MẶT**

Nhân dạng khuôn mặt là một bài toán lâu đời và được nghiên cứu rộng rãi trong khoảng hơn 30 năm lại trở lại đây. Bài toán nhận dạng khuôn mặt có thể áp dụng rộng rãi trong nhiều lĩnh vực khác nhau. Các ứng dụng liên quan đến nhận dạng khuôn mặt có thể kể như sau: hệ thống phát hiện tội phạm, hệ thống theo dõi nhân sự trong một đơn vị, hệ thống tìm kiếm thông tin trên ảnh, video dựa trên nội dung,... Hiện nay, bài toán nhận dạng khuôn mặt gặp nhiều thách thức, ví dụ như hệ thống camera công cộng, chụp hình vui chơi thì ảnh nhận được có thể bị che khuất một phần, ảnh chụp không chính diện hay chất lượng ảnh không tốt, những yếu tố này ảnh hưởng không nhỏ đến các thuật toán nhận dạng khuôn mặt. Có nhiều thuật toán khắc phục điều này, họ sử dụng một số kỹ thuật như xác định nhiều điểm chính trên khuôn mặt, lấy những chi tiết nhỏ hay sử dụng các phương pháp Học Sâu. Bài báo cáo này sẽ trình bày tổng quan một số thuật toán để xây dựng hệ thống nhận diện khuôn mặt.

## **1.2. MỤC ĐÍCH ĐỀ TÀI**

Công Nghệ thông tin ngày càng phát triển và có vai trò hết sức quan trọng không thể thiếu trong cuộc sống hiện đại. Con người ngày càng tạo ra những cỗ máy thông minh có khả năng tự nhận biết và xử lý được các công việc một cách tự động, phục vụ cho lợi ích của con người. Trong những năm gần đây các khu vực như thư viện, văn phòng, trường học dần trở thành những không gian mở có nhiều người tới, vì vậy an ninh là một phần quan trọng ở những địa điểm như vậy. Việc nhận diện được danh tính của những người đến những nơi này là vô cùng quan trọng, vì vậy nhóm chọn đề tài nhận diện khuôn mặt để tìm hiểu thêm về cách thức hoạt động cũng như điểm mạnh điểm yếu của mô hình này.

### 1.3. MÔ TẢ BÀI TOÁN

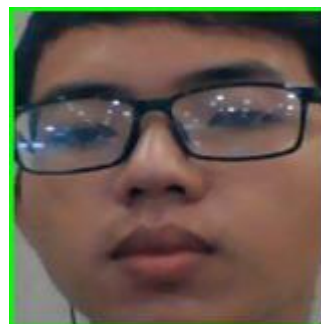
Hệ thống nhận dạng khuôn mặt người là một hệ thống nhận vào là một ảnh hoặc một đoạn video (một chuỗi các ảnh). Qua xử lý tính toán hệ thống xác định được vị trí mặt người trong ảnh (nếu có) và xác định là người nào trong số những người mà hệ thống đã được biết (qua quá trình học) hoặc là người lạ. Đây là bài toán phân tích hồi quy Regression giúp phát hiện và nhận diện khuôn mặt, cho một khuôn mặt chính diện (không quá nghiêng) dự đoán danh tính của khuôn mặt đó

Input: Là hình ảnh chứa khuôn mặt của một hoặc nhiều người

Output: Vị trí phát hiện khuôn mặt và thông tin của khuôn mặt đó



Input



Output: Trung

### 1.4. TẦM QUAN TRỌNG CỦA BÀI TOÁN NHẬN DIỆN KHUÔN MẶT

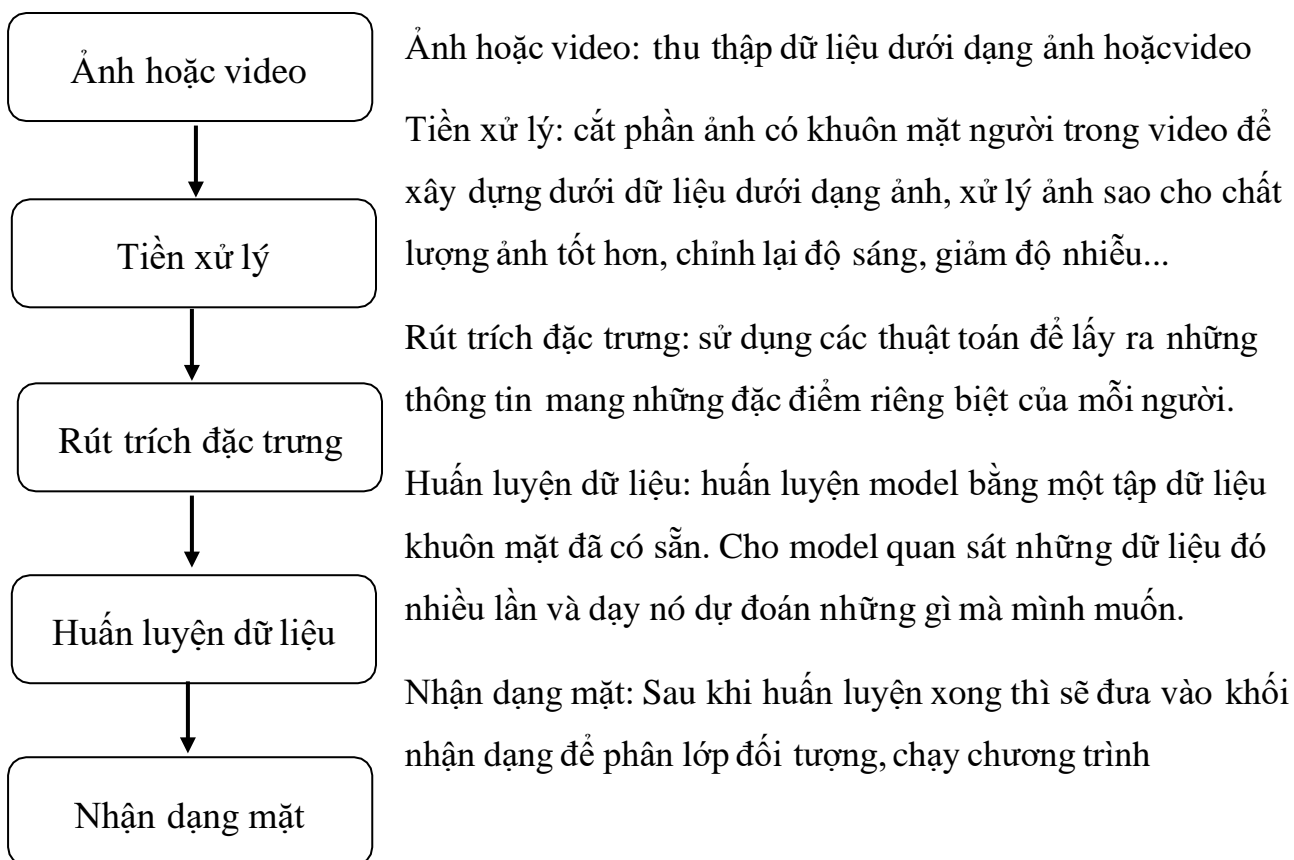
Nếu như công nghệ nhận dạng giọng nói có nhược điểm là tiếng ồn, không phù hợp với những nơi đông người, nhận dạng chữ ký gây nhiều phiền phức cho người sử dụng vì khó duy trì chữ ký giống nhau trong cùng một thời điểm, nhận dạng vân tay cũng có nhược điểm là bị ảnh hưởng bởi độ ẩm da và đặc biệt sẽ không chính xác cao với người có tay hay tiếp xúc với hóa chất, dù nhận dạng mống mắt có nhiều ưu điểm nhưng lại bị tác động như độ rộng của mắt, lông mi, kính đeo và khó phát triển trên diện rộng do độ phức tạp của các thiết bị. Thì trong khi đó, mặc dù độ chính xác không phải là ưu điểm nổi bật của công nghệ nhận diện khuôn mặt người song không giống với những phương

pháp khác, nhận dạng mặt người là công nghệ nhận dạng không cần đến sự tiếp xúc trực tiếp giữa đối tượng và thiết bị thu nhận. Thay vào đó công nghệ mang tính theo dõi, giám sát, và thuận tiện cho những ứng dụng bảo vệ mục tiêu, chống không bỏ tại những điểm công cộng đông người. Đây cũng là ưu điểm nội trội của nhận dạng mặt người mà các công nghệ nhận dạng khác khó có thể có được.

Một số ứng dụng đặc trưng của bài toán nhận dạng khuôn mặt: Hệ thống giao tiếp thông minh giữa người và máy; nhận dạng tội phạm; giải trí; hệ thống quan sát; điều khiển ra vào các cơ quan, văn phòng, trường học; tổ chức tìm kiếm con người, thú nuôi mất tích; phân tích cảm xúc mặt người...v...v...

### 1.5. XÂY DỰNG HỆ THỐNG NHẬN DIỆN KHUÔN MẶT

Một hệ thống nhận diện khuôn mặt có thể xây dựng qua rất nhiều công đoạn khác nhau và rất phức tạp, nhưng ta có thể khái quát chung mọi hệ thống nhận dạng khuôn mặt gồm các bước cơ bản sau



## CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU

### 2.1. MỤC ĐÍCH THU THẬP DỮ LIỆU

Một điều quan trọng thường bỏ qua khi bắt đầu một bài toán Học máy là thu thập dữ liệu để training cho model. Ví dụ đơn giản là các loại dữ liệu như hình ảnh, video, từ loại... Đa phần trong quá trình học Học Máy hay Thị Giác Máy Tính, chúng ta làm việc trên những bộ dữ liệu đã được xử lý sẵn. Nhưng trên thực tế, tùy thuộc vào bài toán mà cần thu thập dữ liệu khác nhau và ít khi tương quan với những bộ dữ liệu sẵn có.

Các model máy học không thể hoạt động nếu thiếu dữ liệu, trường hợp dữ liệu quá nhỏ thì dễ dẫn đến hiện tượng sự quá khớp (Overfitting) và model không thể học được đầy đủ các đặc trưng cho các trường hợp tổng quan hay nói một cách khác đi là model thiếu khả năng tổng quan hóa (Generalization) Vậy làm sao để tìm đủ dữ liệu huấn luyện và gắn nhãn cho nó? Có thể nói đây là công việc mất nhiều công sức nhất trong Học Máy

Trước tiên chúng ta cần phải hiểu và nắm được bài toán cần giải quyết là gì? Điều đó dẫn đến việc tìm kiếm chính xác dữ liệu training cho bài toán. Chúng ta có thể sử dụng các công cụ để thu thập dữ liệu từ trên mạng, hoặc có thể tìm ảnh, video từ các mạng xã hội, ảnh vệ tinh trên google, mua dữ liệu từ bên thứ 3, thậm chí có thể tự bản thân xây dựng một bộ data do chính mình thu thập. Thu thập dữ liệu giúp chúng ta xây dựng một hệ thống đúng với yêu cầu của bài toán đề ra. Đây là bước khá quan trọng vì sẽ đánh giá mô hình của chúng ta có chạy đúng với mục đích của bài toán hay không

### 2.2. THU THẬP VÀ XỬ LÝ DỮ LIỆU

Vì mục đích là xây dựng một hệ thống điểm danh sinh viên và phát hiện người lạ vào trường học nên nhóm sẽ thu hẹp phạm vi thu thập dữ liệu.

Cách thức thu thập: Quay video khuôn mặt của một số sinh viên ở trong trường đại học công nghệ thông tin ( đã được các bạn sinh viên đồng ý cho sử dụng hình ảnh và video trong mục đích xây dựng chương trình nhận diện khuôn mặt)

Số lượng: Hơn 1000 hình ảnh, trong đó mỗi người sẽ có 100 ảnh với nhiều góc cạnh khác nhau, có nhiều ảnh bị mờ và chói sáng. Ảnh này được cắt ra từ video của mỗi người, mỗi video có độ dài tầm 15-20s

Binh	6/23/2022 9:09 AM	File folder
Cuong	6/23/2022 9:09 AM	File folder
Hieu	6/23/2022 9:09 AM	File folder
Ho	6/23/2022 9:18 AM	File folder
Hoang	6/23/2022 9:09 AM	File folder
Huy	6/23/2022 8:53 AM	File folder
Minh	6/23/2022 9:09 AM	File folder
MTrung	6/23/2022 9:21 AM	File folder
Phat	6/23/2022 9:09 AM	File folder
Son	6/23/2022 9:08 AM	File folder
Tai	6/23/2022 9:18 AM	File folder
Trung	6/23/2022 8:40 AM	File folder
Tung	6/23/2022 9:09 AM	File folder



Quá trình xử lý: Đưa 1 đoạn video 15-20s có chứa mặt của 1 bạn sau đó dùng cấu trúc mạng MTCNN để phát hiện khuôn mặt và cắt lấy khuôn mặt đó. Tiếp tục dùng facenet để lấy những đặc trưng có trong khuôn mặt và lưu vào file feature.csv với vector 512 chiều cho một khuôn mặt phát hiện ra trong 1 khung hình cắt được. Về phần hình ảnh thì sẽ được lưu theo tên của người vừa được đưa vào theo từng thư mục riêng biệt

Chia dữ liệu thành 2 tập. 1 tập train dùng để huấn luyện model dạy model theo mục đích của nhóm sao cho phù hợp với bài toán nhận dạng, tập train gồm 70% dữ liệu sau khi xử lý. Còn 30% dữ liệu còn lại sẽ chia thành tập test để kiểm tra lại độ hiệu quả của model. Dự định sẽ sử dụng SVM và Logistic regression để kiểm tra kết quả.



## CHƯƠNG 3: PHƯƠNG PHÁP ĐỀ XUẤT

Nhóm sử dụng Facenet để xác định vị trí khuôn mặt trên video và trích xuất đặc trưng từ khuôn mặt đó. Từ đó sẽ dùng model SVM của thư viện Sklearn để huấn luyện bộ dữ liệu. Sử dụng thư viện Numpy để hỗ trợ tính toán nhanh hơn. Thư Viện CV2 để biểu diễn kết quả ... để giải quyết bài toán nhóm sử dụng mạng Multitask Convolution Neural Network (MTCNN)

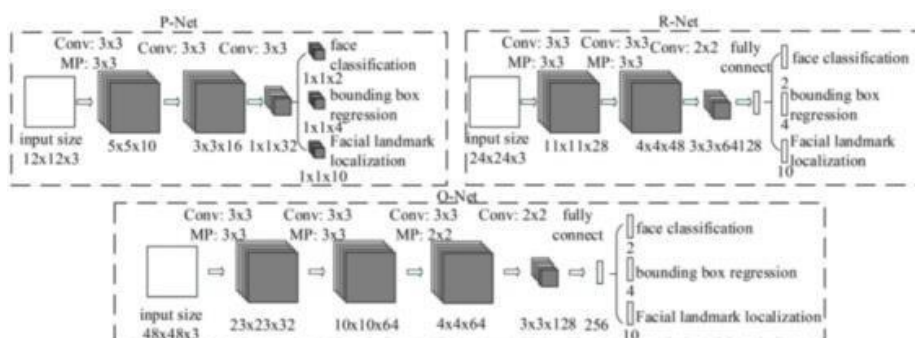
### 3.1. GIỚI THIỆU MULTITASK CONVOLUTION NEURAL NETWORK

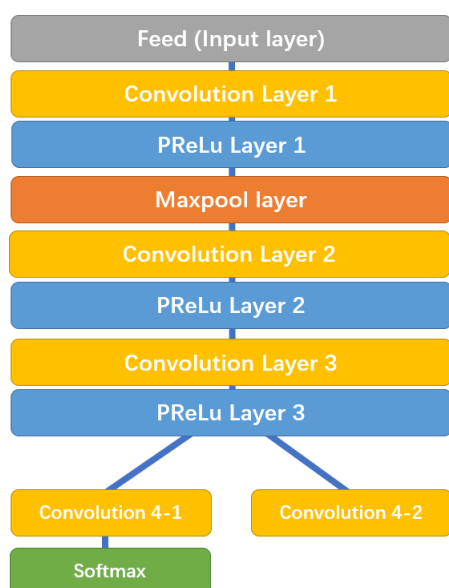


MTCNN là mạng bao gồm 3 mạng CNN xếp chồng và đồng thời hoạt động khi detect khuôn mặt. Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau trong công việc. Đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi miệng....

### 3.2. CẤU TRÚC MTCNN

MTCNN Hoạt động theo 3 bước, mỗi bước có một mạng neural riêng lần lượt là: P-Net, R-Net và O-Net. Với mỗi bức ảnh đầu vào, nó sẽ tạo ra nhiều bản sao của hình ảnh đó với các kích thước khác nhau





Cấu trúc mạng P-Net

Tại P-Net, thuật toán sử dụng 1 kernel 12x12 chạy qua mỗi bức hình để tìm kiếm khuôn mặt.

Sau lớp convolution thứ 3, mạng chia thành 2 lớp.

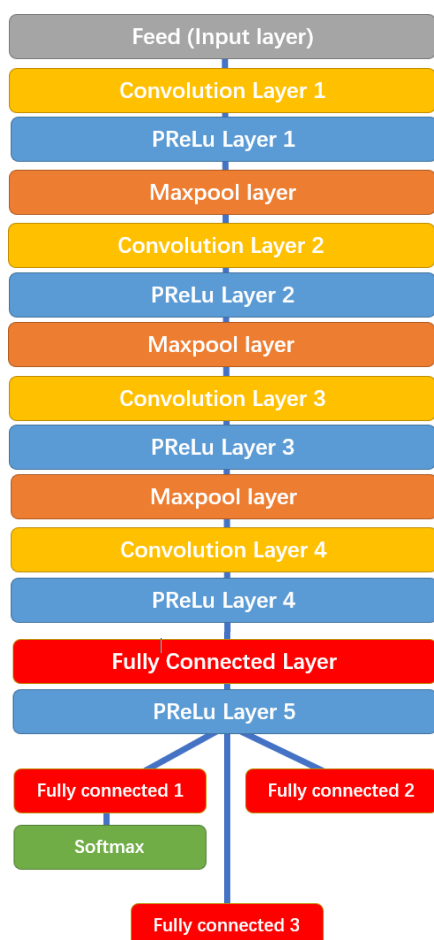
Convolution 4-1 đưa ra xác suất của một khuôn mặt nằm trong mỗi bounding boxes, và Convolution 4-2 cung cấp tọa độ của các bounding boxes.



Cấu trúc mạng R-Net

R-Net có cấu trúc tương tự với P-Net. Tuy nhiên sử dụng nhiều layer hơn. Tại đây, network sẽ sử dụng các bounding boxes đc cung cấp từ P-Net và tinh chỉnh là tọa độ.

Tương tự R-Net chia ra làm 2 layers ở bước cuối, cung cấp 2 đầu ra đó là tọa độ mới của các bounding boxes, cùng độ tin tưởng của nó.



Cấu trúc mạng O-Net

O-Net lấy các bounding boxes từ R-Net làm đầu vào và đánh dấu các tọa độ của các mốc trên khuôn mặt.

Ở bước này, thuật toán đưa ra 3 kết quả đầu ra khác nhau bao gồm: xác suất của khuôn mặt nằm trong bounding box, tọa độ của bounding box và tọa độ của các mốc trên khuôn mặt (vị trí mắt, mũi, miệng)

### 3.3. ƯU NHƯỢC ĐIỂM CỦA MẠNG MTCNN

**Ưu điểm:** là mô hình này nhận diện được gương mặt ở nhiều góc độ, và có thể nhận diện cả đặc trưng của khuôn mặt với thời gian xử lý khá tốt. Có tốc độ nhanh và độ chính xác ổn định khi dự đoán

**Nhược điểm:** Phát hiện khuôn mặt nhỏ trong các bức hình không tốt. Che nửa mặt hoặc quá nghiêng cũng dẫn đến kết quả dự đoán sai

## CHƯƠNG 4: THỰC NGHIỆM CHƯƠNG TRÌNH

### 4.1. CÀI ĐẶT

Ngôn ngữ sử dụng: Ngôn ngữ Python 3.6.8

Công cụ thực hiện: Visual studio code 1.47.3, Google Colab

Cấu hình máy thực hiện: Core i9 9300H, VGA 1050, 8gb ram.

Thư viện sử dụng: tensorflow 1.15.0, scipy 1.1.0, scikit-learn, opencv-python, h5py, matplotlib, pillow, requests, psutil, flask, pandas, numpy, logging, pickle.

### 4.2. QUÁ TRÌNH THỰC NGHIỆM

#### Thiết lập đối số

`MINSIZE = 20` #ngưỡng thấp nhất khi resize

`THRESHOLD = [0.6, 0.7, 0.7]` #thiết lập ngưỡng của ảnh, mức độ confidence để nhận mặt. Array ba giá trị cho ba mạng.

`FACTOR = 0.7` #thiết lập độ tương phản

`INPUT_IMAGE_SIZE = 160` # `IMAGE_SIZE = 160`

`FACENET_MODEL_PATH = 'okela/model/20180402-114759.pb'` #đường dẫn model  
`tf.Graph().as_default()`

#khai báo sử dụng các hàm tính toán của môi trường graph trong tensorflow

`gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.6)`

`sess = tf.Session(config=tf.ConfigProto(gpu_options=gpu_options,  
log_device_placement=False))`

- lấy đặc trưng

#### Bước 1: Nhập tên của khuôn mặt sắp đưa vào dataset

`name=input('Nhập tên:')`

## Bước 2: Dùng model của facenet để phát hiện khuôn mặt và lấy đặc trưng của khuôn mặt đó và lưu vào thư mục

- Xác định vị trí khuôn mặt

```
frame = cv2.resize(frame, (0, 0), fx=0.75, fy=0.75)
```

```
bounding_boxes, _ = detect_face.detect_face(frame, MIN_SIZE, pnet, rnet, onet, THRESHOLD, FACTOR)
```

```
faces_found = bounding_boxes.shape[0]
```

- Lấy tọa độ khuôn mặt đó

```
det = bounding_boxes[:, 0:4]
```

```
b = np.zeros((faces_found, 4), dtype=np.int32)
```

```
for i in range(faces_found):
```

```
    bb[i][0] = det[i][0]
```

```
    bb[i][1] = det[i][1]
```

```
    bb[i][2] = det[i][2]
```

```
    bb[i][3] = det[i][3]
```

- Lưu đặc trưng

```
#dùng facenet trích xuất đặc trưng khuôn mặt
```

```
scaled = facenet.prewhiten(scaled_out)
```

```
scaled_reshape = scaled.reshape(1, INPUT_IMAGE_SIZE,  
INPUT_IMAGE_SIZE, 3)
```

```
feed_dict = {images_placeholder: scaled_reshape,
```

```
phase_train_placeholder: False}
```

```
emb_array = sess.run(embeddings, feed_dict=feed_dict)
```

```
emb_array = np.append(emb_array, name)

#Lưu feature vừa trích xuất được ở trên vào file feature.csv

my_features = np.array(emb_array)

my_features = my_features.reshape(-1, my_features.shape[0])

df = pd.DataFrame(my_features)

df.to_csv("features.csv", mode='a', header=None, index=False)
```

### Bước 3: lưu hình ảnh khuôn mặt người đó vào thư mục dataset

```
#Lưu ảnh vào folder Dataset

folder = "Dataset/" + str(name) + "/"

if not os.path.exists(folder):os.makedirs(folder)

cv2.imwrite(folder + str(cnt) + '.jpg', scaled_out)
```

- **Train model**

### Bước 4: Load dữ liệu từ file feature.csv để lấy nhãn và đặc trưng tương ứng

```
df = pd.read_csv('features.csv')

labels = df.iloc[:, -1].values #Lấy nhãn tương ứng

emb_array = df.iloc[:, 0:-1].values #Lấy đặc trưng tương ứng
```

### Bước 5: Nhập các đối số cho model SVM

```
model = SVC(kernel='linear', probability=True)
```

### Bước 6: Train model

```
model.fit(emb_array, labels) #tiến hành train model SVM
```

### Bước 7: Lưu model

```
classifier_file = os.path.expanduser("code/model/facemodels.pkl")

class_names = sav.iloc[:, 0].values
```

```
with open(classifier_file, 'wb') as outfile:  
  
pickle.dump((model, class_names), outfile)
```

- **Nhận diện khuôn mặt**

### **Bước 8: Load model train của mình và model của facenet**

```
CLASSIFIER_PATH = 'code/model/facemodels.pkl'  
  
FACENET_MODEL_PATH = 'code/model/20180402-114759.pb'  
  
# Load The Custom Classifier  
  
with open(CLASSIFIER_PATH, 'rb') as file:  
  
    model, class_names = pickle.load(file)  
  
    print("Custom Classifier, Successfully loaded")  
  
# Load the model  
  
    print('Loading feature extraction model')  
  
    facenet.load_model(FACENET_MODEL_PATH)
```

### **Bước 9: Dùng model của facenet để phát hiện khuôn mặt trong video và lấy đặc trưng của nó**

- Xác định vị trí khuôn mặt

```
frame = cv2.resize(frame, (0, 0), fx=0.75, fy=0.75)  
  
bounding_boxes, _ = detect_face.detect_face(frame, MIN_SIZE, pnet, rnet, onet,  
                                             THRESHOLD, FACTOR)
```

```
faces_found = bounding_boxes.shape[0]
```

- Lấy tọa độ khuôn mặt đó

```
det = bounding_boxes[:, 0:4]  
  
bb = np.zeros((faces_found, 4), dtype=np.int32)
```

```
for i in range(faces_found):
```

```
    bb[i][0] = det[i][0]
```

```
    bb[i][1] = det[i][1]
```

```
    bb[i][2] = det[i][2]
```

```
    bb[i][3] = det[i][3]
```

- Lưu đặc trưng

*#dùng facenet trích xuất đặc trưng khuôn mặt*

```
scaled = facenet.prewhiten(scaled_out)
```

```
scaled_reshape = scaled.reshape(-  
1, INPUT_IMAGE_SIZE, INPUT_IMAGE_SIZE, 3)
```

```
feed_dict = {images_placeholder: scaled_reshape,  
              phase_train_placeholder: False}
```

```
emb_array = sess.run(embeddings, feed_dict=feed_dict)
```

```
emb_array = np.append(emb_array, name)
```

**Bước 10: Dùng hàm predict\_proba của model train để so sánh độ giống nhau của đặc trưng lấy được ở bước 9 và của bộ dataset**

```
predictions = model.predict_proba(emb_array)
```

**Bước 11: Lấy ra điểm số giống cao nhất để xét, nếu lớn hơn 0.75 thì in tên tương ứng, nếu không thì in ra Unknown**

*#vị trí của labels giống nhất với input*

```
best_class_indices = np.argmax(predictions, axis=1)
```



#xác suất của lớp tốt nhất

```
best_class_probabilities = predictions[np.arange(len(best_class_indices),best_class_indices)]
```

#label của class đó ứng với tên của người đó

```
dict_name = model.predict(emb_array)
```

- Xét ngưỡng

```
if best_class_probabilities > 0.75:
```

```
cv2.rectangle(frame, (bb[i][0], bb[i][1]), (bb[i][2], bb[i][3]), (0, 255, 0), 2)
```

```
text_x = bb[i][0]
```

```
text_y = bb[i][3] + 20
```

```
cv2.putText(frame, name_r, (text_x, text_y), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, 55, 255, 255), thickness=1, lineType=2)
```

```
else:
```

```
cv2.rectangle(frame, (bb[i][0], bb[i][1]), (bb[i][2], bb[i][3]), (0, 255, 0), 2)
```

```
text_x = bb[i][0]
```

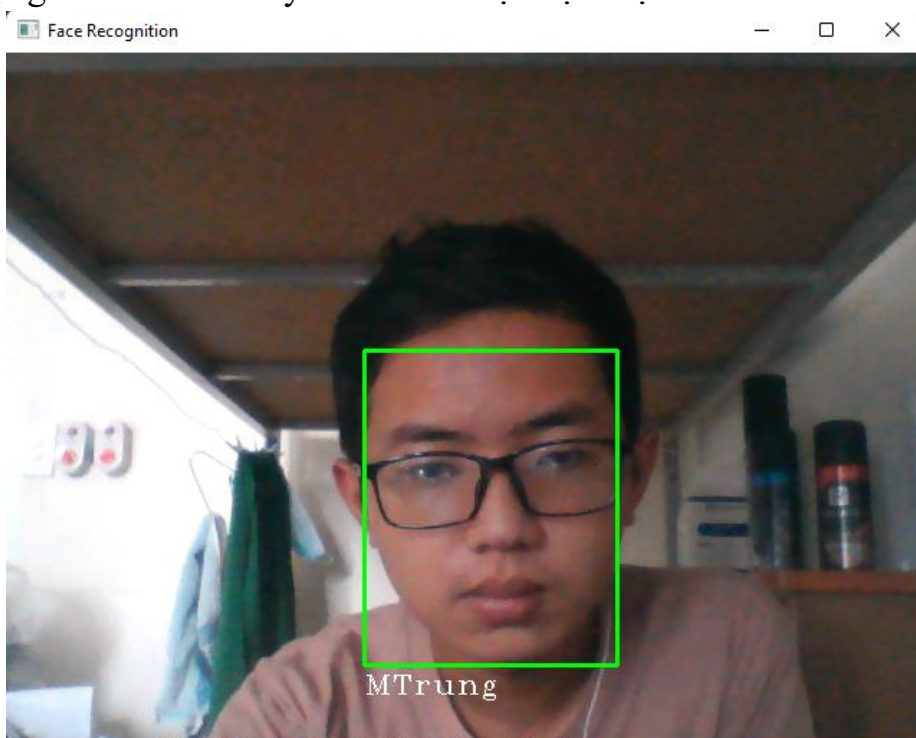
```
text_y = bb[i][3] + 20
```

```
name= "Unknown"
```

```
cv2.putText(frame, name, (text_x, text_y), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255, 255, 255), thickness=1, lineType=2)
```

### 4.3. DEMO CHƯƠNG TRÌNH

Sử dụng camera trên máy tính để hỗ trợ nhận diện realtime



### 4.4. KẾT QUẢ THỰC NGHIỆM

```
type_class_names: <class 'numpy.ndarray'>
predictions: [[0.00102772 0.0038499 0.0018829 0.92848344 0.00580291 0.0102849
0.01079193 0.01580317 0.01054912 0.01152402]]
best_class_indices: [3]
index of person: 3
best_class_probabilities: [0.92848344]
dict_name ['Minh']
prob: [0.92848344]
```

Kết quả này ta có khi thực hiện hàm predict\_proba của SVM, nó sẽ so sánh mức độ giống nhau của dữ liệu khuôn mặt đầu vào vào các dữ liệu có trong khuôn mặt và in ra 1 mảng với các giá trị là mức độ giống, sau đó ta chọn ra 1 giá trị cao nhất(best\_class\_indices) và xét nếu giá trị cao hơn cái ngưỡng mà ta đặt ra(0.75) thì in ra tên ứng với giá trị đó, nếu không có giá trị nào thỏa thì in ra unknow

```
1 print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
1	1.00	0.98	0.99	41
2	0.92	0.77	0.84	30
3	1.00	1.00	1.00	20
4	1.00	1.00	1.00	30
5	0.97	0.90	0.93	31
6	1.00	1.00	1.00	25
7	1.00	1.00	1.00	28
8	1.00	0.93	0.96	27
9	0.88	0.97	0.92	31
10	0.87	0.98	0.92	55
accuracy			0.95	318
macro avg	0.96	0.95	0.96	318
weighted avg	0.96	0.95	0.95	318

	precision	recall	f1-score	support
Cuong	0.00	0.00	0.00	0
Hieu	0.88	1.00	0.94	87
Ho	0.91	0.98	0.95	65
Tai	1.00	0.81	0.90	101
Tung	0.00	0.00	0.00	0
accuracy			0.92	253
macro avg	0.56	0.56	0.56	253
weighted avg	0.94	0.92	0.92	253

Độ chính xác của tập test khoảng 92%

Độ chính xác của tập train khoảng 95%

Recall là mức độ bao phủ của dữ liệu trên tập mà mình đang xét nó thể hiện rằng bao nhiêu mẫu trong 1 lớp thực tế được xác định đúng

Precision thể hiện độ chính xác của việc dự đoán các lớp

F1-score thể hiện được mức độ phân lớp tốt hay không (càng cao càng tốt)

```
print(classification_report(y_result,y_pred))
```

	precision	recall	f1-score	support
0	0.62	0.67	0.64	24
1	1.00	0.47	0.64	15
2	0.95	0.63	0.76	30
3	0.69	1.00	0.82	25
4	0.85	0.71	0.77	41
5	0.83	0.73	0.77	33
6	0.68	0.87	0.76	30
7	0.61	0.71	0.66	35
8	0.83	0.90	0.86	21
9	0.76	0.78	0.77	36
10	0.78	0.75	0.76	28
accuracy			0.75	318
macro avg	0.78	0.75	0.75	318
weighted avg	0.77	0.75	0.75	318

Kết quả khi dùng các thao tác xử lý ảnh thông thường như lấy đặc trưng về màu sắc, góc cạnh và độ sáng trên hình và huấn luyện bằng SVM

Độ chính xác của bộ dữ liệu huấn luyện theo phương pháp rút trích đặc trưng bằng cách thông thường là 75%

Nhóm thấy với các phương pháp rút trích thường thì ít hiệu quả, cho ra kết quả chưa cao nên sẽ ảnh hưởng tới mô hình nhận diện, về phương pháp rút trích đặc trưng bằng Facenet thì cho ra kết quả gần như tuyệt đối là 95% sẽ rất thích hợp trong một nhiệm vụ cần độ chính xác cao như nhận diện khuôn mặt

## CHƯƠNG 5: TỔNG KẾT

### 5.1. KẾT LUẬN

Nhận diện khuôn mặt là một bài toán phân lớp sử dụng kỹ thuật deeplearning (CNN), đây không phải là một bài toán mới, nó đã được triển khai và ứng dụng khá nhiều.

Thông qua việc xây dựng mô hình xử lý bài toán giúp cải thiện kỹ năng về machine learning. - Kết quả thu được như đã trình bày là khá tốt, mô hình không bị vấn đề như overfitting, high loss, ...

### 5.2. KHÓ KHĂN

Mô hình chưa xử lý được những khuôn mặt bị mất một nửa hoặc là mang khẩu trang, nhận dạng sai người khi có những đặc trưng gần tương tự nhau, hình ảnh hoặc video nhận vào có điều kiện khác nhau như màu da quá tối hoặc gần với màu của khung cảnh; biểu cảm của khuôn mặt, chụp bị ngược sáng ... thì mô hình cũng nhận diện không chính xác. Chưa phân biệt được đâu là người đang đứng trước camera hay là một bức ảnh. Dữ liệu thu thập còn ít và chưa đa dạng.

### 5.3. HƯỚNG PHÁT TRIỂN

Cần những phương pháp cụ thể để cải tiến model. Ví dụ như cải thiện dataset, tiền xử lý hình ảnh....

Xử lý các trường hợp ảnh hưởng đến độ chính xác nhận diện khuôn mặt. Xử lý trường hợp khó hơn như xoay ngược xoay ngang.

Áp dụng ứng dụng vào thực tế như điểm danh sinh viên, quản lý sinh viên lạ vào lớp....

## TÀI LIỆU THAM KHẢO

### A. TÀI LIỆU TIẾNG VIỆT

- [1] Nguyễn Thị Thơm, “tìm hiểu đặc trưng lõm 3D và bài toán phát hiện mặt người trong ảnh”, đồ án tốt nghiệp, 2013
- [2] Võ Hoàng Trọng, “nhận dạng khuôn mặt người dựa trên một phần thông tin khuôn mặt”, đồ án môn học, 2017
- [3] Lavender, “Tìm hiểu MTCNN và áp dụng để xác định vị trí các khuôn mặt.”, trực tuyến, 24/10/2018
- <<https://viblo.asia/p/tim-hieu-mtcnn-va-ap-dung-de-xac-dinh-vi-tri-cac-khuon-mat-3Q75wkO75Wb>>

### B. TÀI LIỆU TIẾNG ANH

- [4] Schroff, Florian, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering", *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015
- [5] Zhang, Kaipeng, et al. "Joint face detection and alignment using multitask cascaded convolutional networks." *IEEE Signal Processing Letters* 23.10 (2016): 1499-1503.
- [6] Davidsandberg, “Face Recognition using Tensorflow”, Github, 2018
- < <https://github.com/davidsandberg/facenet>>