

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

MÁY HỌC

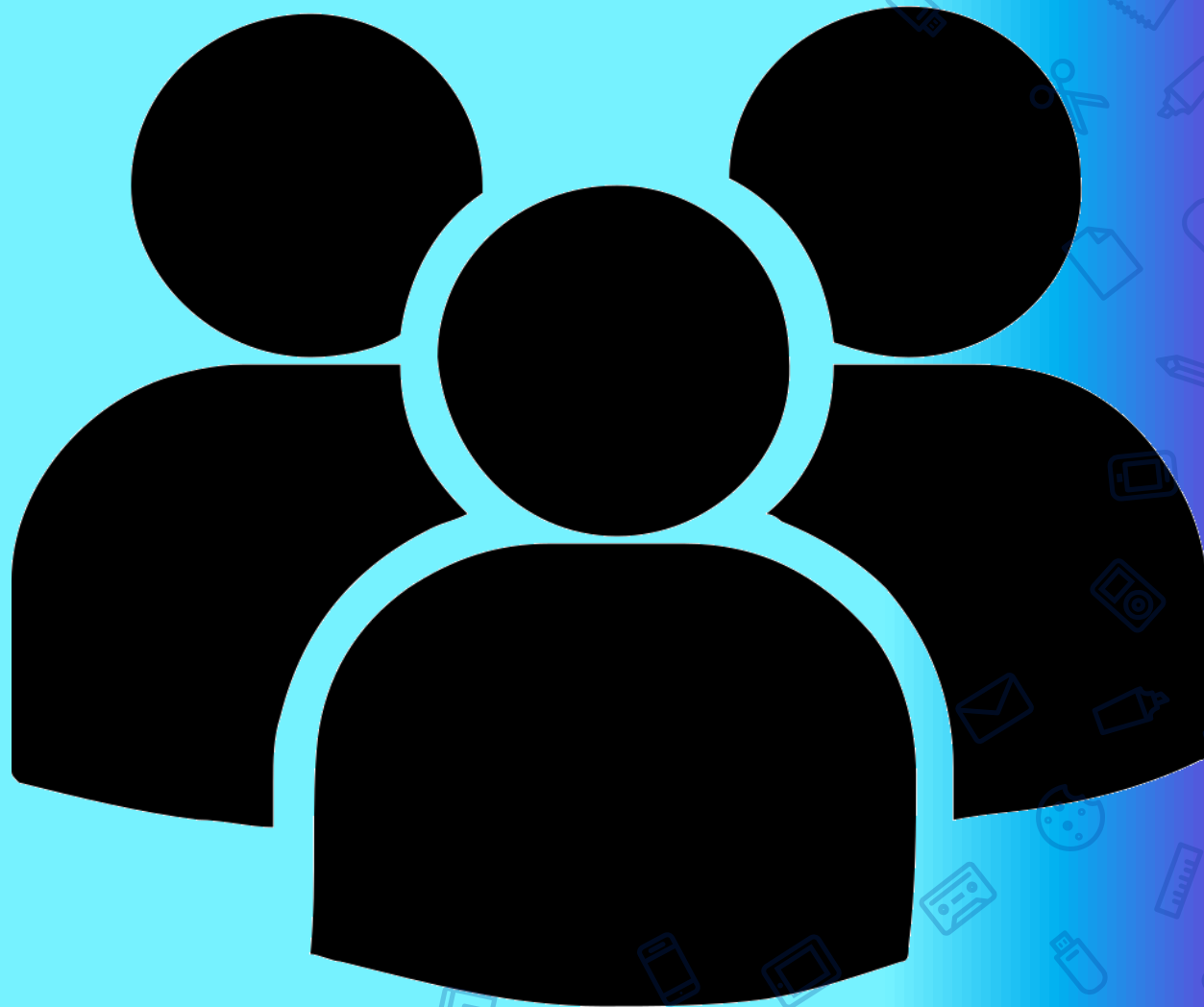
Đề tài: NHẬN DIỆN KHUÔN MẶT SỬ
DỤNG CẤU TRÚC MẠNG MULTITASK
CONVOLUTIONAL NEURAL NETWORK

GVHD: Lê Đình Duy
Phạm Nguyễn Trường An
Lớp: CS114.K21



THÀNH VIÊN

THÀNH VIÊN	MSSV
Hồ Đặng Thanh Hồ	18520760
Đoàn Ngọc Tài	16521045
Lê Hà Minh Trung	18521451



NỘI DUNG

I

GIỚI THIỆU

II

PHƯƠNG PHÁP

III

THỰC NGHIỆM

IV

TỔNG KẾT



MỤC ĐÍCH

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.

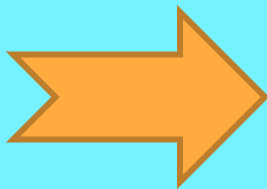
GIỚI THIỆU

1.1 MÔ TẢ BÀI TOÁN

Đây là bài toán phân tích hồi quy (Regression) giúp phát hiện và nhận diện khuôn mặt, cho một ảnh khuôn mặt chính diện (không quá nghiêng), dự đoán danh tính của khuôn mặt đó.



INPUT



OUTPUT

GIỚI THIỆU

1.2 TẦM QUAN TRỌNG BÀI TOÁN

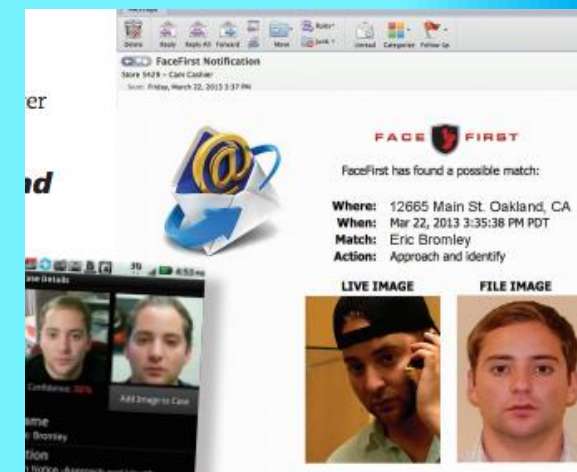
Nhận diện khuôn mặt đã được sử dụng rộng rãi, có nhiều ứng dụng như:



Tự động tag trên facebook



Nhận diện, điểm danh sinh viên, nhân viên, người lạ vào lớp, công ty



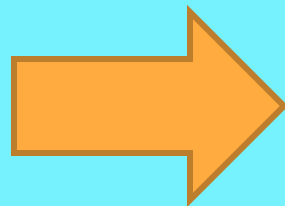
Bắt tội phạm

Và một số ứng dụng khác như: thanh toán online, mở khóa thiết bị điện tử, tìm người mất tích và thú đi lạc.....

PHƯƠNG PHÁP

2.2 GIỚI THIỆU MULTI – TASK CNN

MTCNN Là 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 task. Đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi, miệng...

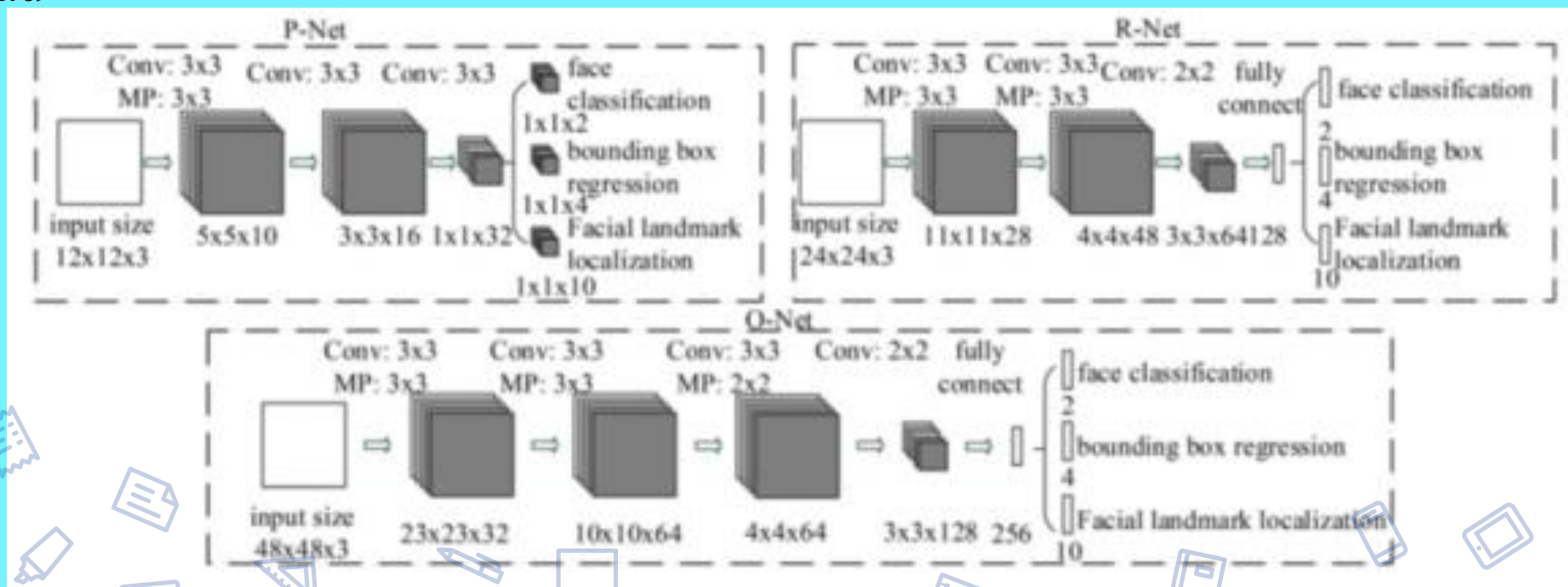


PHƯƠNG PHÁP

2.3 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



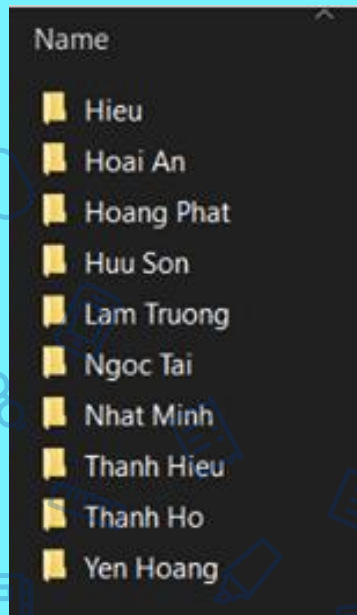
THỰC NGHIỆM

3.1 TIỀN XỬ LÝ DỮ LIỆU

Nhóm sử dụng 1000 ảnh của 10 sinh viên nam UIT.

Mỗi người 100 ảnh.

Ảnh này được cắt ra từ video của mỗi người, mỗi video tầm 15-20s
bộ dữ liệu được chia thành 2 phần với 70% dùng để train cho
model và 30% để test và đánh giá model đó



THỰC NGHIỆM

3.1 CÀI ĐẶT CHƯƠNG TRÌNH

Ngôn ngữ sử dụng: Ngôn ngữ Python 3.6.8

Công cụ thực hiện: Visual studio code 1.47.3, GoogleColab 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, login, pickle.



DEMO

THỰC NGHIỆM

3.2 KẾT QUẢ THỰC NGHIỆM

- 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_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

Sử dụng Deep learning để trích xuất đặc trưng

THỰC NGHIỆM

Sử dụng Deep learning để trích xuất đặc trưng

```
MI 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

Sử dụng Machine learning để trích xuất đặc trưng

```
MI 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

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

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 nguồn 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`

TỔNG KẾT

2.1 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.

2.2 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 ít bị vấn đề như overfitting, high loss, ...

TỔNG KẾT

2.3 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 trường hợp khó hơn như xoay ngược xoay ngang
Áp dụng ứng dụng vào thực tế như điểm danh danh sinh viên, quản lý sinh viên lạ vào lớp....



Thanks for watching