

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ



MÔN HỌC: NHẬP MÔN AI

ĐỀ TÀI:

NHẬN DIỆN KHUÔN MẶT
(Cảm xúc và Giới tính)

HÀ NỘI – 2023

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ



MÔN HỌC: NHẬP MÔN AI

ĐỀ TÀI:
NHẬN DIỆN KHUÔN MẶT
(Cảm xúc và Giới tính)

Giảng viên: Thân Quang Khoát

Thành viên: Nguyễn Thế Phong - 20002150

Dương Xuân Đức - 20002114

Phùng Phúc Hậu - 20002123

Trần Anh Minh - 20002144

Đinh Khả Vy - 20002183

HÀ NỘI - 2023

Lời cảm ơn

Trong quá trình nghiên cứu môn học nhập môn trí tuệ nhân tạo, em đã nhận được sự giúp đỡ nhiệt tình của Thầy Thân Quang Khoát cùng với Thầy Phạm Cường. Thầy đã cung cấp cho em những kiến thức cơ bản quan trọng để phục vụ trong quá trình nghiên cứu. Em rất trân trọng khoảng thời gian cùng Thầy nghiên cứu môn học.

Do những hạn chế về kiến thức, dự án này của chúng em cũng không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự đóng góp, phê bình của các thầy để đề tài của chúng em được hoàn thiện hơn. Cuối cùng em xin kính chúc Thầy nhiều sức khỏe, thành công và hạnh phúc,...

Chúng em xin chân thành cảm ơn!

MỤC LỤC

<i>Lời mở đầu</i>	1
A. TỔNG QUAN VỀ NHẬN DIỆN KHUÔN MẶT	2
1. Nhận diện khuôn mặt là gì ?	2
2. Công nghệ nhận diện khuôn mặt mang lại những lợi ích gì ?	2
2.1. Bảo mật hiệu quả	2
2.2. Cải thiện độ chính xác	2
2.3. Tích hợp dễ dàng	3
3. Hệ thống nhận diện khuôn mặt được sử dụng trong những trường hợp nào ?	3
3.1. Phát hiện gian lận	3
3.2. An ninh mạng	3
B. HOẠT ĐỘNG	3
1. Phát hiện	3
2. Phân tích	4
3. Nhận diện	5
C. THUẬT TOÁN VÀ TRAINING MODEL	5
1. Thuật toán	5
2. Cài đặt model và training	6
D. DEMO	14
1. Cài đặt môi trường Virtual Environment	14
2. Tạo môi trường ảo (virtual enviroment)	14
3. Kích hoạt môi trường	15

Lời mở đầu

AI (Artificial Intelligence) hay còn gọi là trí tuệ nhân tạo là là một lĩnh vực trong khoa học máy tính và công nghệ thông tin mô phỏng các quá trình suy nghĩ và học tập của con người dựa trên máy móc, đặc biệt là máy tính. Cụ thể, AI tập trung vào việc phát triển các thuật toán và công nghệ để máy tính có thể học hỏi, suy luận và tự động hóa các quá trình phức tạp mà trước đây chỉ có thể do con người thực hiện.

AI đang đóng vai trò ngày càng quan trọng trong đời sống hiện nay. Nó được ứng dụng rộng rãi trong các lĩnh vực như y tế, tài chính, truyền thông, giáo dục, sản xuất và các ngành công nghiệp khác. Ngoài ra, trí tuệ nhân tạo cũng đóng vai trò quan trọng trong việc cải thiện chất lượng cuộc sống của con người thông qua các ứng dụng y tế, giáo dục và các dịch vụ trực tuyến. AI đang trở thành một trong những xu hướng công nghệ phát triển mạnh mẽ nhất trong thế kỷ 21, với tiềm năng lớn để đóng góp vào sự phát triển và cải thiện cuộc sống của con người.

Trí tuệ Nhân tạo đang mở ra một thế giới mới, nơi mà khả năng của chúng ta không còn bị giới hạn bởi khả năng của chính mình. Trong tương lai không xa, AI có thể trở thành người bạn đồng hành đáng tin cậy, nguồn sức mạnh sáng tạo và nguồn cảm hứng không tưởng.

Hãy cùng nhau khám phá và đi vào cuộc hành trình này, nơi mà AI và con người hợp tác và cùng nhau khám phá những giới hạn mới, mở ra một tương lai tươi sáng đầy tiềm năng.

A. TỔNG QUAN VỀ NHẬN DIỆN KHUÔN MẶT

1. Nhận diện khuôn mặt là gì ?

Trình phân tích khuôn mặt là phần mềm xác định hoặc xác nhận danh tính của một người qua khuôn mặt của họ. Công nghệ này hoạt động bằng cách xác định và đo lường các đặc điểm khuôn mặt trong hình ảnh. Công nghệ nhận diện khuôn mặt có thể xác định khuôn mặt người trong hình ảnh hoặc video, xác định xem khuôn mặt xuất hiện trong hai hình ảnh có phải là cùng một người không hoặc tìm kiếm khuôn mặt trong một bộ sưu tập đồ sộ các hình ảnh hiện có. Các hệ thống bảo mật sinh trắc học sử dụng công nghệ nhận diện khuôn mặt để nhận dạng cá nhân độc nhất trong lúc triển khai người dùng hoặc đăng nhập, cũng như để tăng cường cho hoạt động xác thực người dùng. Các thiết bị di động và cá nhân cũng thường sử dụng công nghệ phân tích khuôn mặt để bảo mật thiết bị.

2. Công nghệ nhận diện khuôn mặt mang lại những lợi ích gì ?

2.1. Bảo mật hiệu quả

Nhận diện khuôn mặt là hệ thống xác minh nhanh chóng và hiệu quả. Đây là công nghệ nhanh và tiện lợi hơn so với những công nghệ sinh trắc học khác như quét vân tay hoặc võng mạc. Số lượng điểm tiếp xúc trong nhận diện khuôn mặt cũng ít hơn so với việc nhập mật khẩu hoặc PIN. Công nghệ này hỗ trợ xác thực nhiều yếu tố để xác minh bảo mật bổ sung.

2.2. Cải thiện độ chính xác

Nhận diện khuôn mặt là phương thức xác định cá nhân chính xác hơn so với đơn giản chỉ sử dụng số điện thoại, địa chỉ email, địa chỉ gửi thư hoặc địa chỉ IP. Ví dụ: hiện nay, hầu hết các dịch vụ giao dịch từ cổ phiếu cho tới tiền điện tử đều tin nhiệm nhận diện khuôn mặt để bảo vệ khách hàng và tài sản của họ.

2.3. Tích hợp dễ dàng

Công nghệ nhận diện khuôn mặt tương thích và tích hợp dễ dàng với hầu hết các phần mềm bảo mật. Ví dụ: điện thoại thông minh có camera trước được tích hợp hỗ trợ các thuật toán hoặc mã phần mềm nhận diện khuôn mặt.

3. Hệ thống nhận diện khuôn mặt được sử dụng trong những trường hợp nào ?

3.1. Phát hiện gian lận

Các công ty sử dụng nhận diện khuôn mặt để xác định người dùng độc nhất đang tạo một tài khoản mới trên nền tảng trực tuyến. Sau khi hoàn tất, nhận diện khuôn mặt có thể được sử dụng để xác minh danh tính của người thực sử dụng tài khoản phòng trường hợp có hoạt động tài khoản đáng ngờ hoặc có khả năng gây rủi ro

3.2. An ninh mạng

Các công ty sử dụng công nghệ nhận diện khuôn mặt thay cho mật khẩu để củng cố các biện pháp an ninh mạng. Rất khó để truy cập trái phép vào hệ thống nhận diện khuôn mặt vì khuôn mặt của bạn không thể thay đổi được. Phần mềm nhận diện khuôn mặt cũng là một công cụ bảo mật tiện lợi và có độ chính xác cao để mở khóa điện thoại thông minh và những thiết bị cá nhân khác.

B. HOẠT ĐỘNG

Nhận diện khuôn mặt hoạt động theo ba bước: phát hiện, phân tích và nhận diện.

1. Phát hiện

Phát hiện là quá trình tìm khuôn mặt trong hình ảnh. Nhận diện khuôn mặt được thị giác máy tính hỗ trợ có thể phát hiện và xác định khuôn mặt của cá

nhân từ hình ảnh có một hoặc nhiều khuôn mặt người. Công nghệ này có thể phát hiện dữ liệu khuôn mặt ở cả nét mặt nhìn từ phía trước và từ phía bên.

- Thị giác máy tính

Thiết bị máy sử dụng *thị giác máy tính* để xác định người, địa điểm và mọi thứ trong hình ảnh với độ chính xác bằng hoặc cao hơn mức của con người và với hiệu quả và tốc độ cao hơn nhiều. Nhờ sử dụng công nghệ trí tuệ nhân tạo (AI) phức tạp, thị giác máy tính tự động hóa hoạt động trích xuất, phân tích, phân loại và nắm bắt thông tin hữu ích từ dữ liệu hình ảnh. Dữ liệu hình ảnh tồn tại ở nhiều dạng, ví dụ:

- Hình ảnh đơn lẻ
- Cảnh video
- Góc nhìn từ nhiều camera
- Dữ liệu ba chiều

2. Phân tích

Sau đó, hệ thống nhận diện khuôn mặt phân tích hình ảnh khuôn mặt. Hệ thống này ánh xạ và đọc hình diện khuôn mặt cũng như biểu cảm trên khuôn mặt. Hệ thống xác định các điểm quan trọng trên khuôn mặt, những điểm này đóng vai trò là chìa khóa để phân biệt khuôn mặt với những vật thể khác.

Công nghệ nhận diện khuôn mặt thường tìm kiếm những đặc điểm sau:

- Khoảng cách giữa hai mắt
- Khoảng cách từ trán đến cằm
- Khoảng cách giữa mũi và miệng
- Độ sâu của hốc mắt
- Hình dạng của gò má
- Đường viền môi, tai và cằm

Sau đó, hệ thống sẽ chuyển đổi dữ liệu nhận diện khuôn mặt thành một dãy số hoặc điểm được gọi là dấu khuôn mặt (faceprint). Mỗi người sẽ có một dấu khuôn mặt độc nhất, tương tự như dấu vân tay. Thông tin được nhận diện

khuôn mặt sử dụng cũng có thể được sử dụng ngược lại để tái cấu trúc khuôn mặt của một người thành phiên bản kỹ thuật số.

3. Nhận diện

Nhận diện khuôn mặt có thể xác định một người bằng cách so sánh các khuôn mặt trong hai hoặc nhiều hình ảnh và đánh giá khả năng khớp khuôn mặt. Ví dụ: nhận diện khuôn mặt có thể xác minh khuôn mặt có trong ảnh tự chụp bằng camera của di động khớp với khuôn mặt trong hình ảnh giấy tờ tùy thân do chính phủ cấp như giấy phép lái xe hoặc hộ chiếu, cũng như xác minh khuôn mặt có trong ảnh tự chụp không khớp với khuôn mặt trong tập hợp các khuôn mặt được chụp trước đó.

C. THUẬT TOÁN VÀ TRAINING MODEL

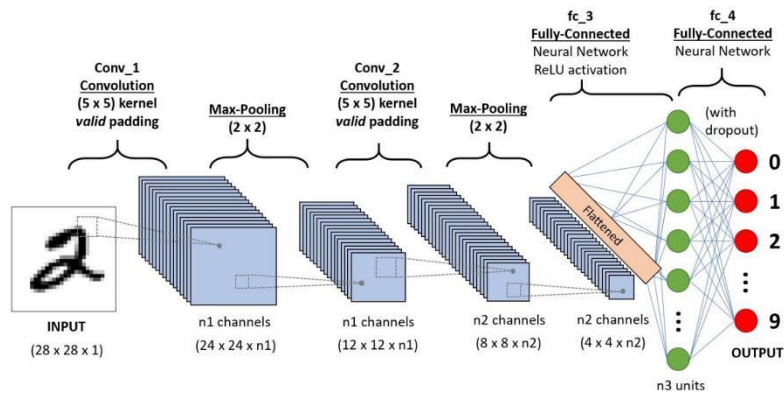
1. Thuật toán

Thuật toán trong CNN (Convolutional Neural Network) là một phương pháp học sâu được sử dụng trong việc nhận diện dựa trên hình ảnh chụp khuôn mặt của con người dựa trên nhiều góc độ.

CNN là một kiến trúc mạng nơ-ron nhân tạo được thiết kế để nhận dữ liệu hình ảnh và phân tích chúng.

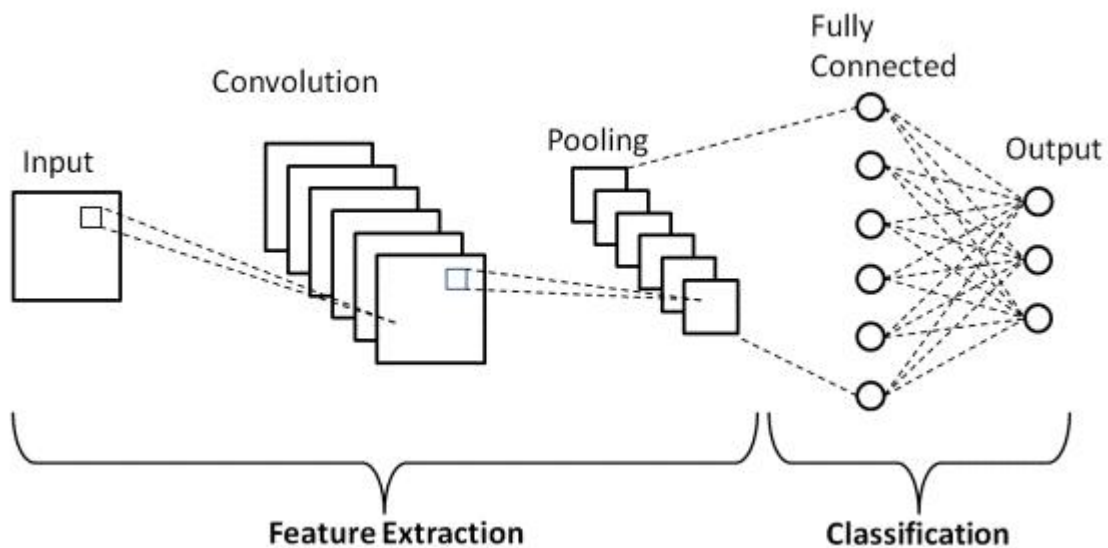
CNN hoạt động bằng cách sử dụng các bộ lọc (filters) để thực hiện phép tích chập (convolution) trên hình ảnh đầu vào. Bộ lọc này sẽ di chuyển trên toàn bộ hình ảnh, tìm kiếm các đặc trưng của hình ảnh bằng cách so sánh các khu vực nhỏ của hình ảnh với một mẫu được học trước. Các đặc trưng được tìm thấy sau đó sẽ được đưa vào một hoặc nhiều lớp ẩn để xử lý và trích xuất thông tin từ các đặc trưng này.

Sau khi thông tin được trích xuất, CNN sử dụng một số lớp Fully Connected (FC) để xử lý dữ liệu và cuối cùng đưa ra dự đoán khuôn mặt nhận diện được.



2. Cài đặt model và training

Trong bài toán này nhóm chúng tôi sử dụng 2 bộ data : fer2013 trong việc training về nhận diện cảm xúc và imdb datasets trong việc training giới tính



Xây dựng model simple_CNN trong việc training detect về nhận diện giới tính:

```
def simple_CNN(input_shape, num_classes):
    model = Sequential()
    model.add(Convolution2D(filters=16, kernel_size=(7, 7), padding='same',
                             name='image_array', input_shape=input_shape))
    model.add(BatchNormalization())
    model.add(Convolution2D(filters=16, kernel_size=(7, 7), padding='same'))
    model.add(BatchNormalization())
```

```

model.add(Activation('relu'))
model.add(AveragePooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.5))

model.add(Convolution2D(filters=32, kernel_size=(5, 5), padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(filters=32, kernel_size=(5, 5), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(AveragePooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.5))

```

```

model.add(Convolution2D(filters=64, kernel_size=(3, 3), padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(filters=64, kernel_size=(3, 3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(AveragePooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.5))

```

```

model.add(Convolution2D(filters=128, kernel_size=(3, 3), padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(filters=128, kernel_size=(3, 3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(AveragePooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.5))

```

```

model.add(Convolution2D(filters=256, kernel_size=(3, 3), padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(
    filters=num_classes, kernel_size=(3, 3), padding='same'))

```

```

model.add(GlobalAveragePooling2D())

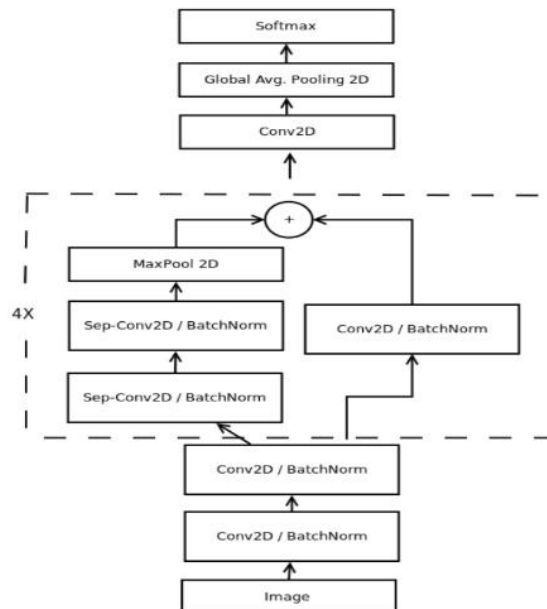
model.add(Activation('softmax', name='predictions'))

return model

```

Xây dựng model mini_XCEPTION trong việc detect các cảm xúc của con người

mini-Xception



```

def mini_XCEPTION(input_shape, num_classes, l2_regularization=0.01):
    regularization = l2(l2_regularization)

    # base
    img_input = Input(input_shape)
    x = Conv2D(8, (3, 3), strides=(1, 1), kernel_regularizer=regularization,
              use_bias=False)(img_input)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Conv2D(8, (3, 3), strides=(1, 1), kernel_regularizer=regularization,
              use_bias=False)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

```

```
# module 1

residual = Conv2D(16, (1, 1), strides=(2, 2),
                  padding='same', use_bias=False)(x)
residual = BatchNormalization()(residual)
```

```
x = SeparableConv2D(16, (3, 3), padding='same',
                    kernel_regularizer=regularization,
                    use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(16, (3, 3), padding='same',
                    kernel_regularizer=regularization,
                    use_bias=False)(x)
x = BatchNormalization()(x)
```

```
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
```

```
# module 2

residual = Conv2D(32, (1, 1), strides=(2, 2),
                  padding='same', use_bias=False)(x)
residual = BatchNormalization()(residual)
```

```
x = SeparableConv2D(32, (3, 3), padding='same',
                    kernel_regularizer=regularization,
                    use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(32, (3, 3), padding='same',
                    kernel_regularizer=regularization,
                    use_bias=False)(x)
x = BatchNormalization()(x)
```

```
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
```

```
# module 3
residual = Conv2D(64, (1, 1), strides=(2, 2),
                  padding='same', use_bias=False)(x)
residual = BatchNormalization()(residual)
```

```
x = SeparableConv2D(64, (3, 3), padding='same',
                    kernel_regularizer=regularization,
                    use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = SeparableConv2D(64, (3, 3), padding='same',
                    kernel_regularizer=regularization,
                    use_bias=False)(x)
x = BatchNormalization()(x)
```

```
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
```

```
# module 4
residual = Conv2D(128, (1, 1), strides=(2, 2),
                  padding='same', use_bias=False)(x)
residual = BatchNormalization()(residual)
```

```
x = SeparableConv2D(128, (3, 3), padding='same',
                    kernel_regularizer=regularization,
                    use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
```

```

x = SeparableConv2D(128, (3, 3), padding='same',
                    kernel_regularizer=regularization,
                    use_bias=False)(x)
x = BatchNormalization()(x)

```

```

x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])

```

```

x = Conv2D(num_classes, (3, 3),
           # kernel_regularizer=regularization,
           padding='same')(x)
x = GlobalAveragePooling2D()(x)
output = Activation('softmax', name='predictions')(x)

```

```

model = Model(img_input, output)

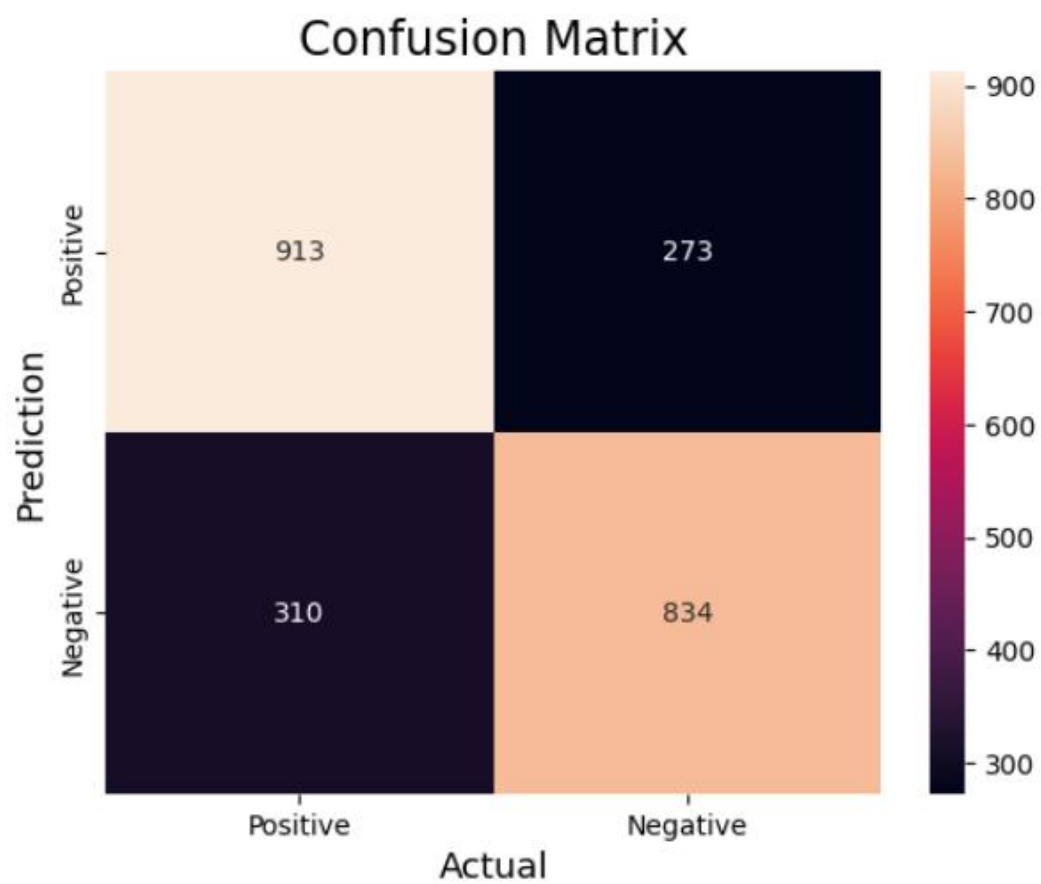
return model

```

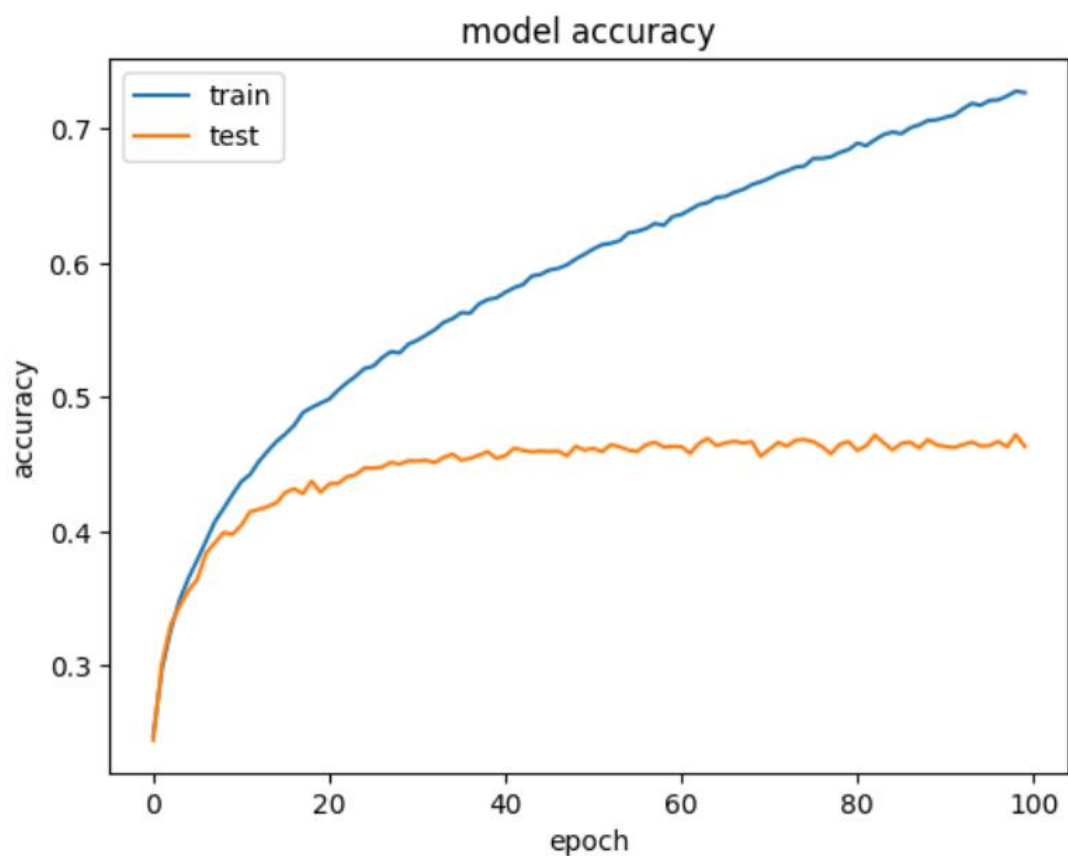
Đánh giá 2 model:

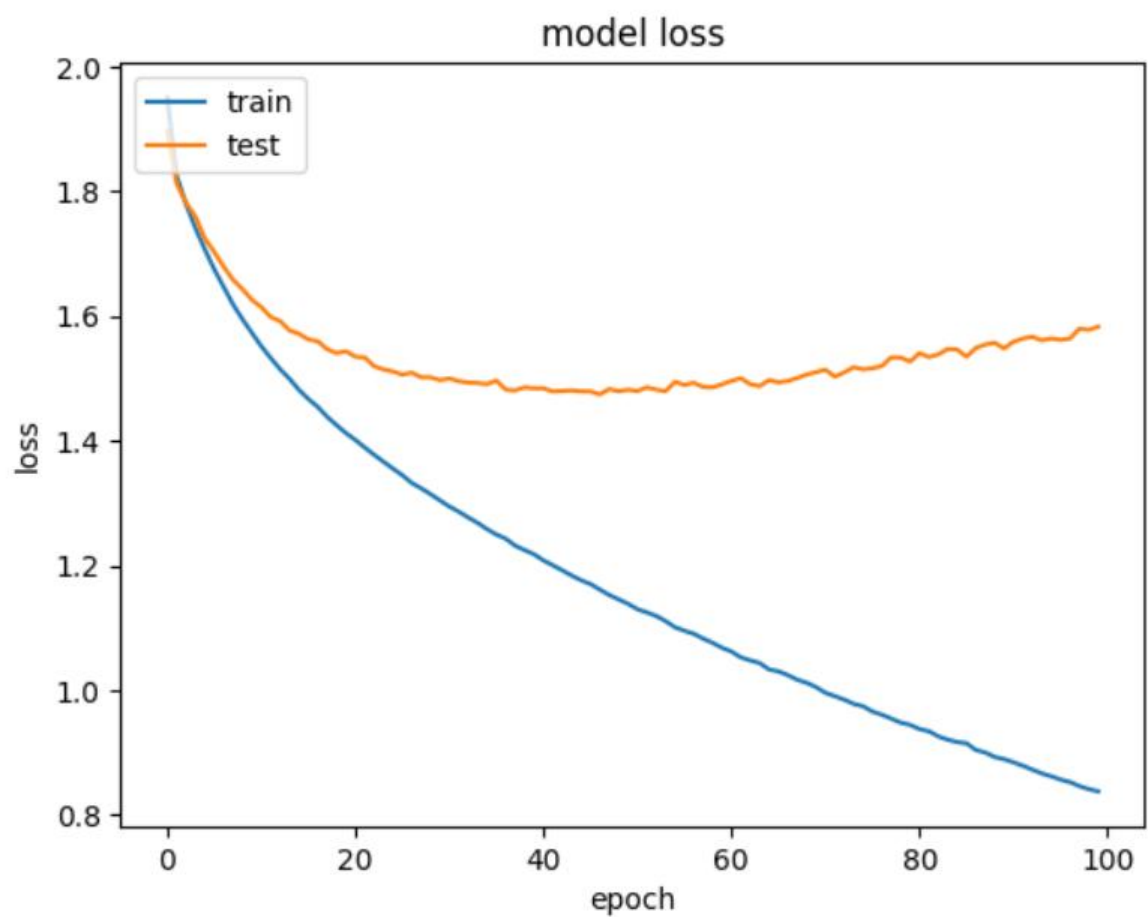
+ Gender

	precision	recall	f1-score	support
0	0.75	0.77	0.76	1186
1	0.75	0.73	0.74	1144
accuracy			0.75	2330
macro avg	0.75	0.75	0.75	2330
weighted avg	0.75	0.75	0.75	2330

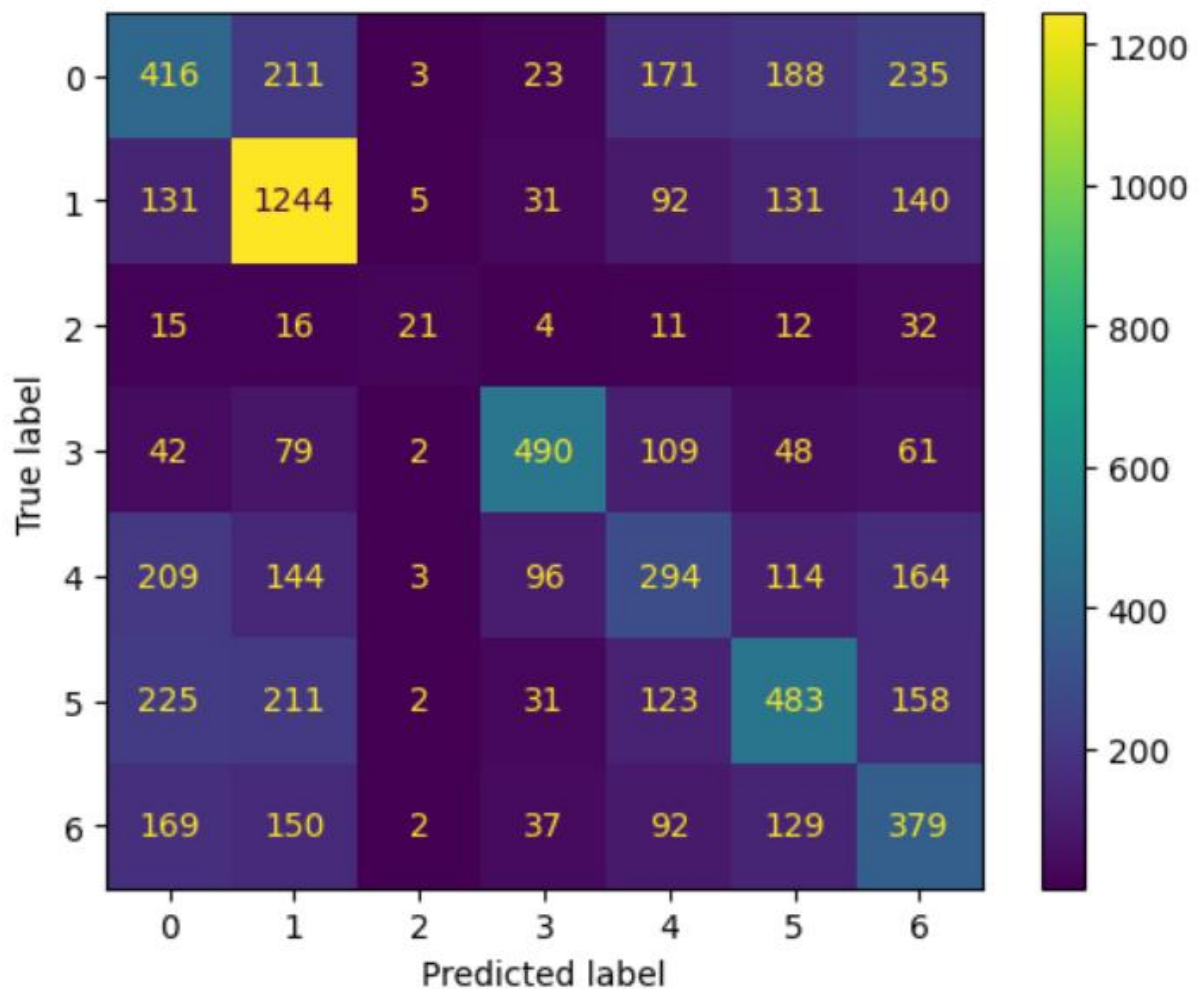


+ Emotion





	precision	recall	f1-score	support
0	0.34	0.33	0.34	1247
1	0.61	0.70	0.65	1774
2	0.55	0.19	0.28	111
3	0.69	0.59	0.64	831
4	0.33	0.29	0.31	1024
5	0.44	0.39	0.41	1233
6	0.32	0.40	0.36	958
accuracy			0.46	7178
macro avg	0.47	0.41	0.43	7178
weighted avg	0.46	0.46	0.46	7178



D. DEMO

1. Cài đặt môi trường Virtual Environment

pip install virtualenv

- Với Debian/Ubuntu

apt-get install python3-venv

2. Tạo môi trường ảo (virtual enviroment)

- Sử dụng câu lệnh: `py -m env (env_name)` để tạo ra môi trường ảo
- Tải các package: sử dụng `pip install -m requirements.txt` (file requirements.txt chứa tên và version của các package cần sử dụng trong quá trình chạy mô hình)

3. Kích hoạt môi trường

- Khi chạy xong câu lệnh tạo môi trường, trong folder chứa model sẽ xuất hiện thư mục là môi trường vừa được tạo

- **Kích hoạt môi trường ảo:**

- **(env_name)/Scripts/activate**

- **Kết quả:**

 env	6/5/2023 17:24	File folder
 src	6/5/2023 17:24	File folder
 trained_models	6/5/2023 17:24	File folder

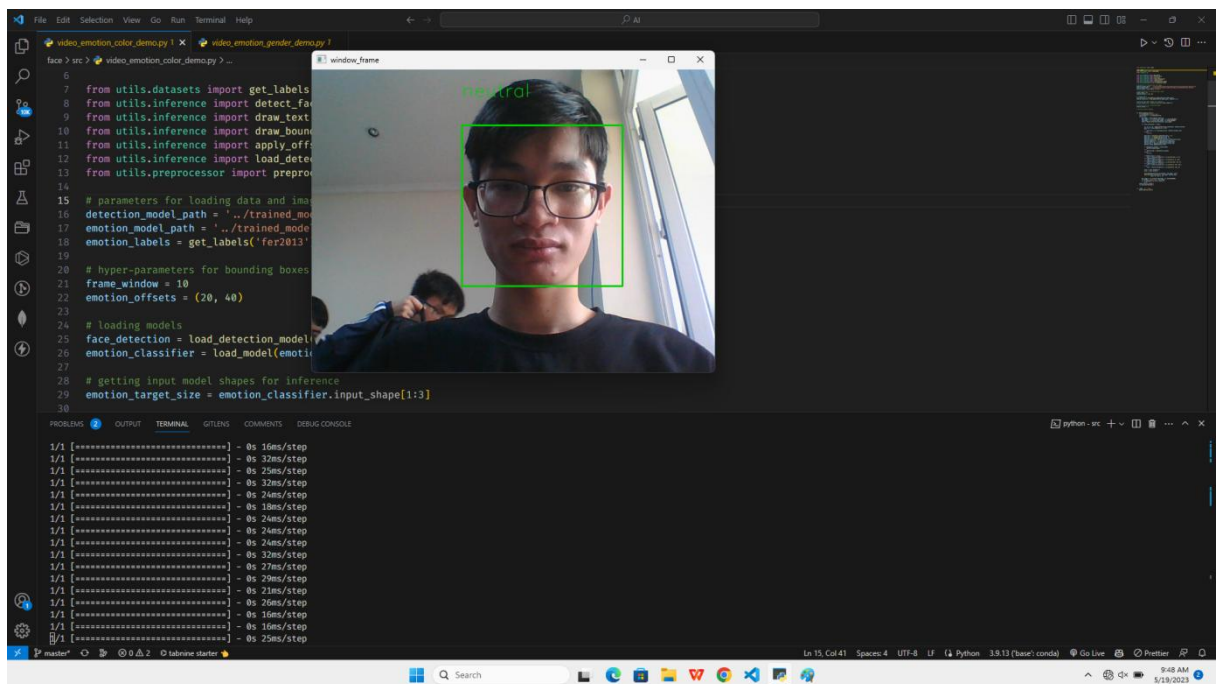
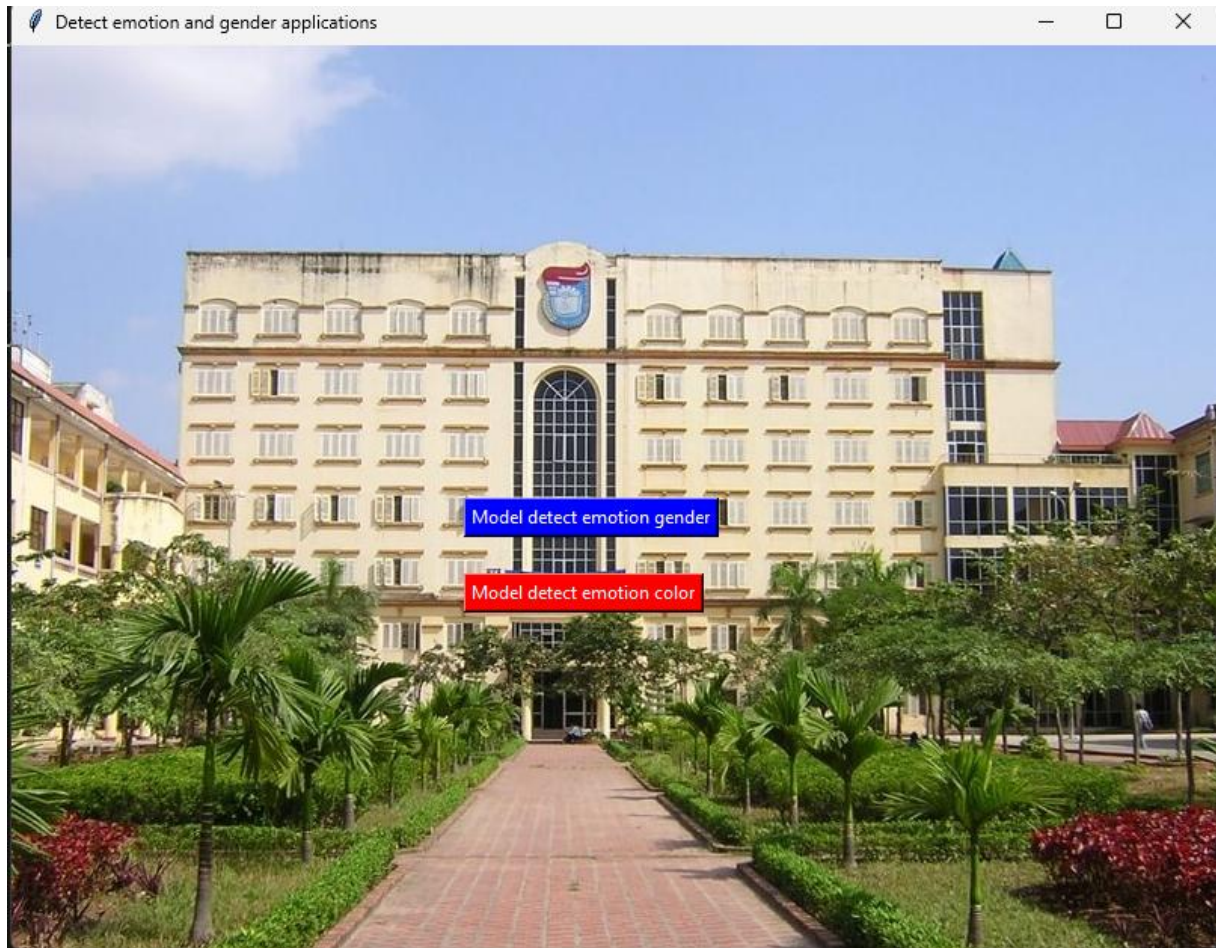
```
D:\BAITAP\CodeAI\face\face>env\Scripts\activate  
(env) D:\BAITAP\CodeAI\face\face>
```

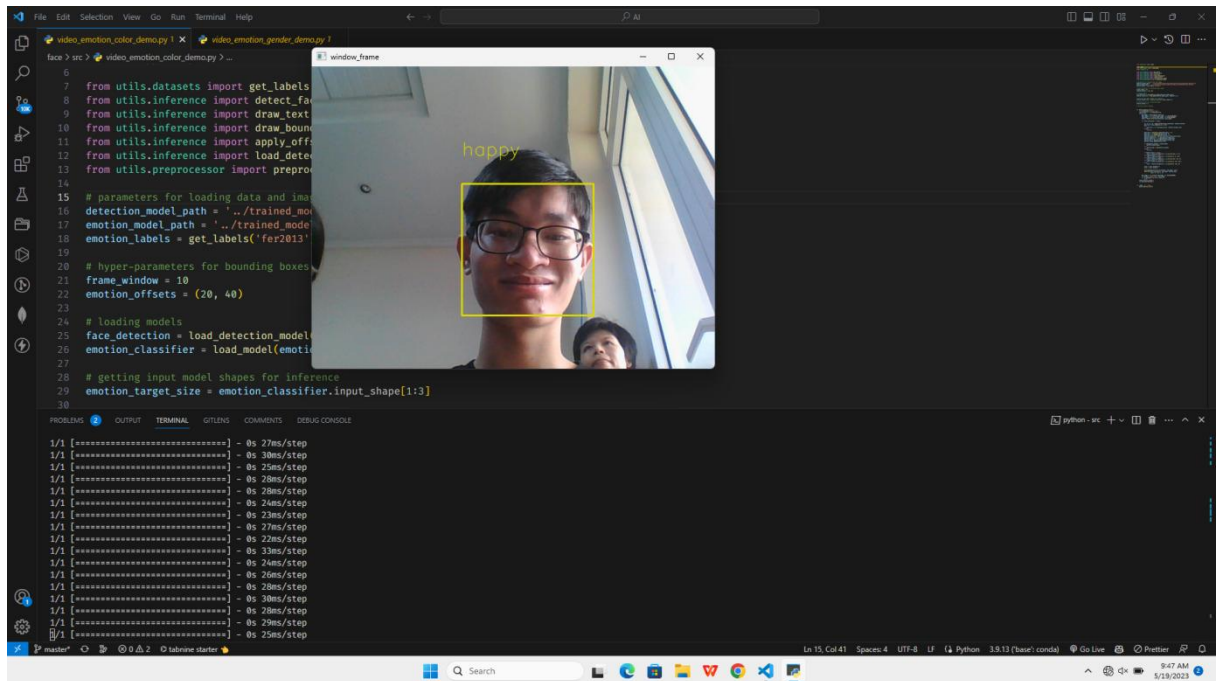
+ **video_emotion_color_demo.py : detect cảm xúc**

- Describe: Mỗi hình ảnh detect được các cảm xúc khác nhau sẽ có các màu khác nhau

- run: **python video_emotion_color_demo.py**

+ Kết quả:



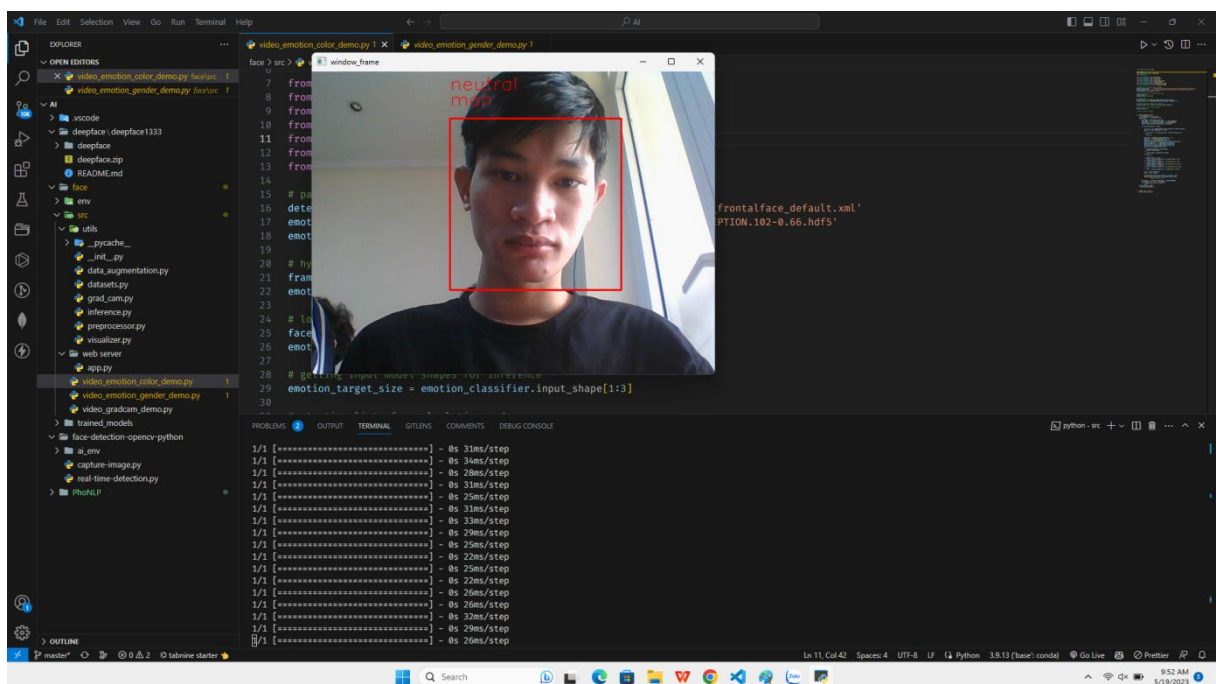


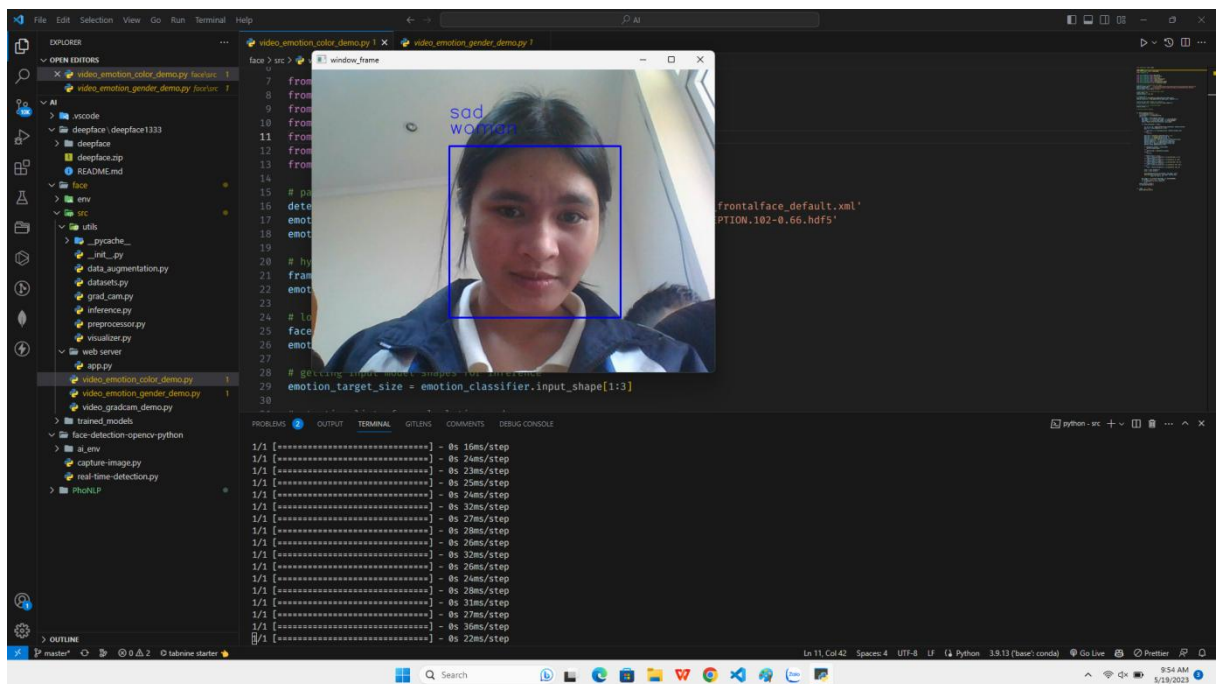
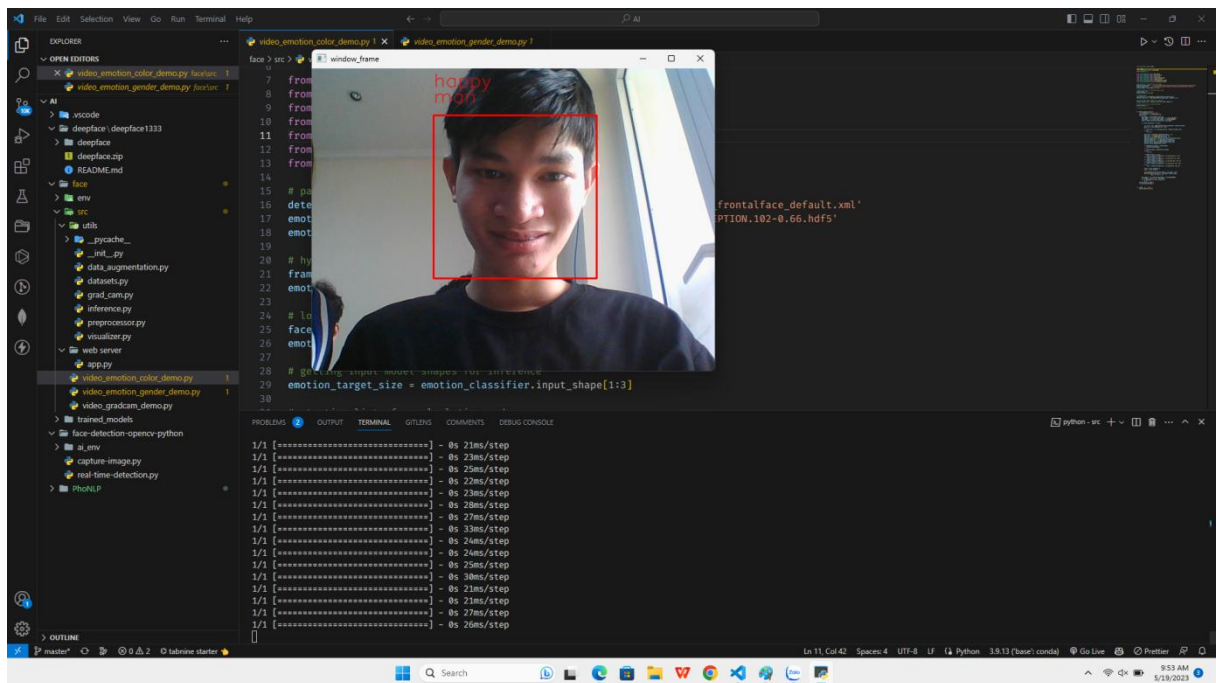
+ **video_emotion_gender_demo.py** : detect giới tính và cảm xúc

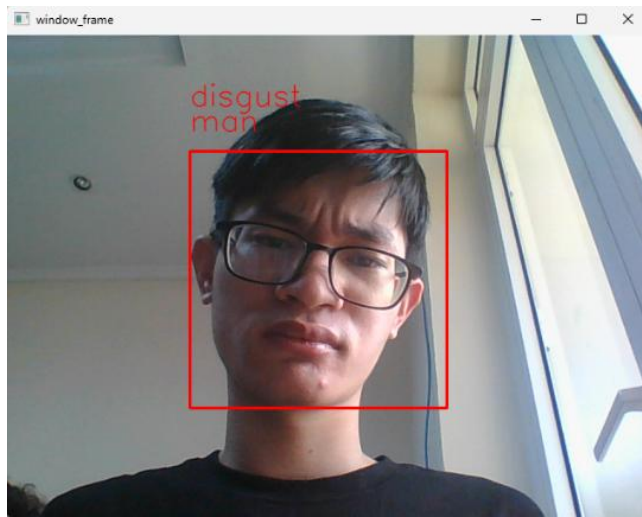
- Describe: Giới tính sẽ có 2 màu riêng biệt

- run: `python video_emotion_gender_demo.py`

+ **Kết quả:**







- Kết thúc chương trình: ấn phím **q**