

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH  
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

-----\*\*-----



**HCMUTE**

## **BÁO CÁO DỰ ÁN CUỐI HỌC KỲ**

Môn học: Trí tuệ nhân tạo

Nội dung: Sử dụng CNN

nhận diện giới tính con người

theo thời gian thực

GVHD: Nguyễn Trường Thịnh

SVTH: Phan Thanh Trường

MSSV: 19146293

Tp. Hồ Chí Minh, tháng 05, năm 2023

## Mục lục

A. Tổng quan .....	3
I. Giới thiệu .....	3
II. Phương pháp thực hiện .....	4
III. Thực hiện .....	4
1. Huấn luyện mô hình trên Google Colab .....	4
1.1. Tải lên các thư viện cần thiết .....	4
1.2. Tải lên tập dữ liệu và kiểm tra dữ liệu Input .....	5
1.3. Gắn nhãn cho dữ liệu .....	6
1.4. Xây dựng mạng CNN .....	6
2. Chương trình Python nhận diện thời gian thực .....	8
2.1. Nhập thư viện .....	8
2.2. Sử dụng ‘harrircascades’ để xác định vị trí khuôn mặt trong mỗi bức hình .....	8
2.3. Xử lý nhận dạng giới tính .....	8
IV. Kết quả .....	9
1.1. Kiểm thử chất lượng model bằng một số hình ảnh .....	9
1.2. Tiếp tục kiểm tra model với chương trình Python đơn giản .....	10
1.3. Chạy thời gian thực trên Visual Code .....	12
V. Kết luận .....	12
B. Tài liệu tham khảo .....	13

## A. Tổng quan

Đề tài nhận diện giới tính sử dụng Python và mạng thần kinh tích chập (CNN) là một chủ đề thú vị trong lĩnh vực trí tuệ nhân tạo, được ứng dụng vô cùng nhiều trong đời sống thực tiễn. CNN là một thuật toán học sâu đã thay đổi hoàn toàn lĩnh vực thị giác máy tính và đạt được thành tích đáng kể trong nhiều ứng dụng, bao gồm phân loại ảnh, phát hiện đối tượng và phân đoạn ngữ nghĩa. Trong đề tài này, bạn có thể sử dụng Python để xây dựng một mô hình CNN để nhận diện giới tính từ ảnh. Chúng ta có thể sử dụng các thư viện như Keras hoặc TensorFlow để xây dựng mô hình CNN của mình. Từ đó hiểu được những kiến thức cơ bản nhất trong việc thực hiện 1 chương trình máy tính có chức năng ‘nhận diện giới tính’.

## I. Giới thiệu

Mạng thần kinh tích chập (CNN) là một thuật toán học sâu đã thay đổi hoàn toàn lĩnh vực thị giác máy tính và đạt được thành tích đáng kể trong nhiều ứng dụng, bao gồm phân loại ảnh, phát hiện đối tượng và phân đoạn ngữ nghĩa. CNN được lấy cảm hứng từ vỏ não thị giác của con người và có khả năng tự động học và trích xuất đặc trưng từ ảnh. Trong bài báo này, chúng tôi trình bày một tổng quan về CNN và các ứng dụng của nó trong thị giác máy tính. Bài làm đề cập đến các khối xây dựng cơ bản của CNN, bao gồm lớp tích chập, lớp gộp và lớp kết nối đầy đủ. Việc sử dụng CNN đã mở ra những khả năng mới trong thị giác máy tính và được kỳ vọng sẽ tiếp tục thúc đẩy sự tiến bộ trong lĩnh vực này.

Bài làm sử dụng CNN để làm lớp huấn luyện mô hình, mô hình này sau đó dùng để nhận diện giới tính người (trong bài làm chỉ giới hạn 2 loại giới tính là ‘male - nam’ và ‘female - nữ’, độ tuổi từ 18 - 60), nhận diện được trong thời gian thực với tốc độ nhận diện ở mức tương đối, và độ chính xác không quá cao (do dữ liệu được làm hoàn toàn thủ công mà không có lấy dữ liệu có sẵn, nên nguồn data để train không dồi dào, dẫn đến độ chính xác không thật sự cao, tuy nhiên đủ dùng để vận dụng và hiểu được cách mạng CNN được dùng để nhận diện giới tính và khuôn mặt người).

Bài làm còn sử dụng thêm một công cụ phát hiện khuôn mặt (công cụ này có sẵn trong thư viện open-CV có tên 'cvlib'.

## II. Phương pháp thực hiện

Trong bài làm này, tôi sử dụng mạng CNN huấn luyện một mô hình tạo mô hình xác định chính xác giới tính của một người xuất hiện trong 1 khung hình được quét bởi camera máy tính. File này sau đó được lưu với đuôi '.h5'

Trước tiên một công cụ hỗ trợ nhận diện khuôn mặt có sẵn trong thư viện open-CV là 'cvlib' được dùng để xác định vị trí khuôn mặt trong một tấm hình bất kỳ, sau đó cắt tấm hình chứa khuôn mặt đó ra, có kích thước là 100x100 (pixel). Hình này sẽ được dùng để thực hiện công việc dự đoán.

Giới tính được dự đoán có thể là 'male - nam' hoặc 'female - nữ'.

Rất khó để tự tạo được một bộ dữ liệu lớn để huấn luyện (trên 10 000 tấm hình). Bài làm có tổng số 300 tấm hình (gồm 150 male, 150 female) đủ để dùng làm một mô hình mang tính chất học thuật.

## III. Thực hiện

### 1. Huấn luyện mô hình trên Google Colab

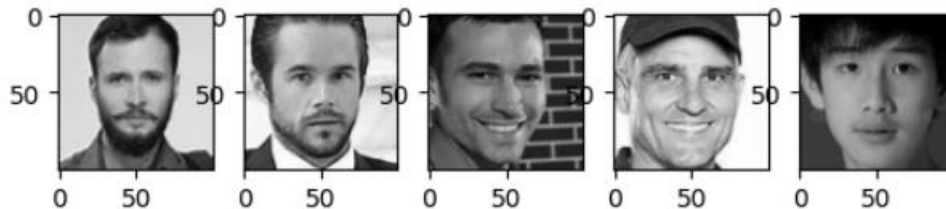
#### 1.1. Tải lên các thư viện cần thiết

```
1 # Import dependencies
2 import matplotlib.pyplot as plt
3 from matplotlib.image import imread
4 from keras.models import Sequential,load_model,Model
5 from tensorflow.keras.callbacks import ModelCheckpoint
6 from tensorflow.keras.layers import Input
7 from tensorflow.keras.optimizers import Adam

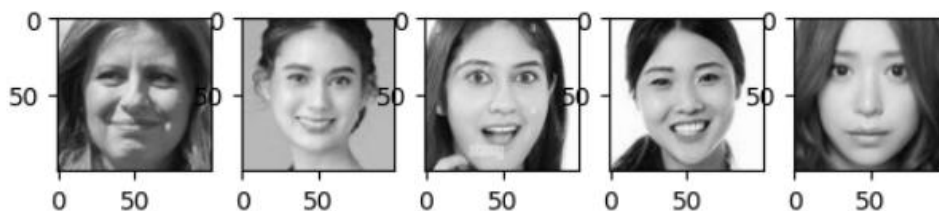
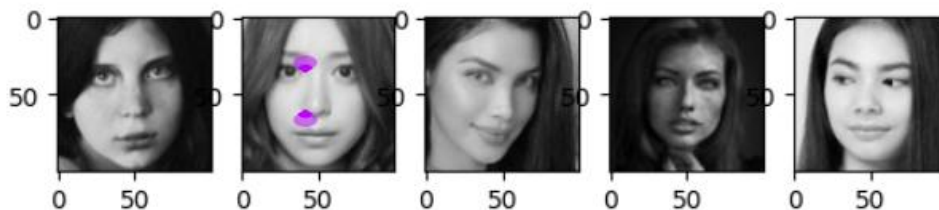
1 from tensorflow import keras
2 import matplotlib.pyplot as plt
3 from keras.utils import load_img
4 from keras.utils.image_utils import img_to_array
5 import numpy as np
```

## 1.2. Tải lên tập dữ liệu và kiểm tra dữ liệu Input

```
1 folder = '/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Intelligence/Code_AI/pi
2 for i in range(10):
3     plt.subplot(2,5,i+1)
4     filename = folder + 'male_0_'+str(i)+'.jpg'
5     img = plt.imread(filename)
6     plt.imshow(img)
7 plt.show()
```



```
1 folder = '/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Intelligence/Cc
2 for i in range(10):
3     plt.subplot(2,5,i+1)
4     filename = folder + 'female_1_'+str(i)+'.jpg'
5     img = plt.imread(filename)
6     plt.imshow(img)
7 plt.show()
```



### 1.3. Gắn nhãn cho dữ liệu

```
1 from os import listdir
2 from os.path import isdir
3 from numpy import asarray
4 from numpy import save
5 from keras.utils import load_img, img_to_array
6 folder = '/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Intelligence/Code_AI/project_1'
7 photos, labels = list(), list()
8 for file in listdir(folder):
9     output = 0.0
10    if file.startswith('female'):
11        output = 1
12    if file.startswith('male'):
13        output = 2
14    img = load_img(folder+file)
15    photo = img_to_array(img)
16    photos.append(photo)
17    labels.append(output)
18 photos = asarray(photos)
19 labels = asarray(labels)
20 print(photos.shape, labels.shape)
21 save('/content/drive/Code_AI/projects/gender_detection_photos.npy', photos)
22 save('/content/drive/Code_AI/projects/gender_detection_labels.npy', labels)
```

(300, 100, 100, 3) (300,)

```
1 import numpy as np
2 x_train = np.load('/content/drive/MyDrive/projects/gender_detection_photos.npy')
3 y_train = np.load('/content/drive/MyDrive/projects/gender_detection_labels.npy')
4 print(photos.shape, labels.shape)
```

(300, 100, 100, 3) (300,)

```
1 x_train = x_train.astype('float32')/255
2 from keras.utils import to_categorical
3 y_train = to_categorical(y_train,10)
```

### 1.4. Xây dựng mạng CNN

```
1 from keras.models import Sequential
2 from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D, Normalization, LeakyReLU
3 from keras.optimizers import Adam
4
5 #32 lan tích chập
6 model = Sequential()
7 model.add(Conv2D(32, kernel_size = (3,3), activation = 'relu', input_shape=(100,100,3), padding='Same'))
8 model.add(MaxPooling2D((2,2), padding='same'))
9 model.add(Dropout(0.25))
10
11 #64 lan tích chập
12 model.add(Conv2D(64, (3,3), activation = 'relu', padding = 'same'))
13 model.add(MaxPooling2D((2,2), padding='same'))
14 model.add(Dropout(0.25))
15
16 #128 lan tích chập
17 model.add(Conv2D(128, (3,3), activation = 'relu', padding = 'same'))
18 model.add(MaxPooling2D((2,2), padding='same'))
19 model.add(Dropout(0.25))
```



```

20
21 # #256 lan tich chap
22 # model.add(Conv2D(256,(3,3),activation='relu',padding='same'))
23 # model.add(MaxPooling2D((2,2),padding='same'))
24 # model.add(Dropout(0.25))
25
26 model.add(Flatten())
27 model.add(Dense(128,activation='relu'))
28 model.add(Dropout(0.25))
29 model.add(Dense(10,activation='sigmoid'))
30
31 from keras.losses import categorical_crossentropy
32 model.compile(loss = categorical_crossentropy,optimizer = Adam(),metrics=['accuracy'])
33 model.summary()
34 train = model.fit(x_train,y_train,batch_size=200,epochs = 60,verbose = 1)
35 model.save('/content/drive/MyDrive/projects/gender_detection2.h5')

```

Model: "sequential\_15"

Layer (type)	Output Shape	Param #
=====		
conv2d_52 (Conv2D)	(None, 100, 100, 32)	896
max_pooling2d_52 (MaxPooling2D)	(None, 50, 50, 32)	0
dropout_67 (Dropout)	(None, 50, 50, 32)	0
conv2d_53 (Conv2D)	(None, 50, 50, 64)	18496
max_pooling2d_53 (MaxPooling2D)	(None, 25, 25, 64)	0
dropout_68 (Dropout)	(None, 25, 25, 64)	0
conv2d_54 (Conv2D)	(None, 25, 25, 128)	73856
max_pooling2d_54 (MaxPooling2D)	(None, 13, 13, 128)	0
dropout_69 (Dropout)	(None, 13, 13, 128)	0
flatten_15 (Flatten)	(None, 21632)	0
dense_30 (Dense)	(None, 128)	2769024
dropout_70 (Dropout)	(None, 128)	0
dense_31 (Dense)	(None, 10)	1290

```

=====
Total params: 2,863,562
Trainable params: 2,863,562
Non-trainable params: 0

```

```

1 model = keras.models.load_model('/content/drive/projects/gender_detection2.h5')
2 model.save('/content/drive/projects/gender_detection2.model')
3

```

## 2. Chương trình Python nhận diện thời gian thực

### 2.1. Nhập thư viện

```
18 import matplotlib.pyplot as plt
19 from keras.utils import load_img
20 from keras.utils.image_utils import img_to_array
21 from tensorflow.keras.models import load_model
22 import numpy as np
23 import cv2
24 import cvlib
25 vat = {1: 'female', 2: 'male'}
26 # load model
27 model = load_model('gender_detection2.h5')
```

### 2.2. Sử dụng 'harrrcascades' để xác định vị trí khuôn mặt trong mỗi bức hình

```
29 while True:
30     status, test_image = webcam.read()
31     face, confidence = cvlib.detect_face(test_image)
32     # test_image = cv2.imread('R.jpeg')
33     gray = cv2.cvtColor(test_image, cv2.COLOR_BGR2GRAY)
34     face_cascade = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')
35     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
```

### 2.3. Xử lý nhận dạng giới tính

```
36 i = 0
37 for idx, f in enumerate(face):
38     # get corner points of face rectangle
39     (startX, startY) = f[0], f[1]
40     (endX, endY) = f[2], f[3]
41     cv2.rectangle(test_image, (startX-30, startY-30), (endX+30, endY+30), (203, 12, 255), 2)
42     img = gray[(startY-30):(endY+30), (startX-30):(endX+30)]
43     img = cv2.resize(img, (100, 100))
44     cv2.imwrite('img.jpg', img)
45     print('shape2', img.shape)
46     img = load_img('img.jpg', target_size=(100, 100))
47     img = img_to_array(img)
48     img = img.reshape(1, 100, 100, 3)
49     img = img.astype('float64')
50     img = img/255
51     result = np.argmax(model.predict(img), axis=1)
52     text = str(vat[result[0]])
53     cv2.putText(test_image, text, (startX, startY), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
54
55     cv2.imshow('out', test_image)
56     # press "Q" to stop
57     k = cv2.waitKey(1)
58     if k == 27 or 0xFF == ord('q'):
59         break
60 # release resources
61 webcam.release()
62 cv2.destroyAllWindows()
```

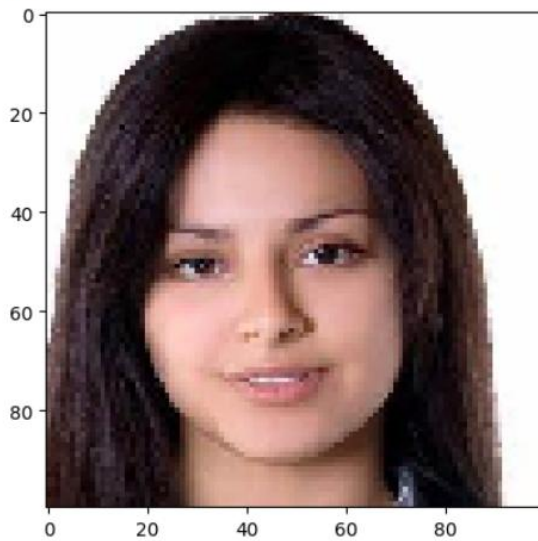


## IV. Kết quả

### 1.1. Kiểm thử chất lượng model bằng một số hình ảnh

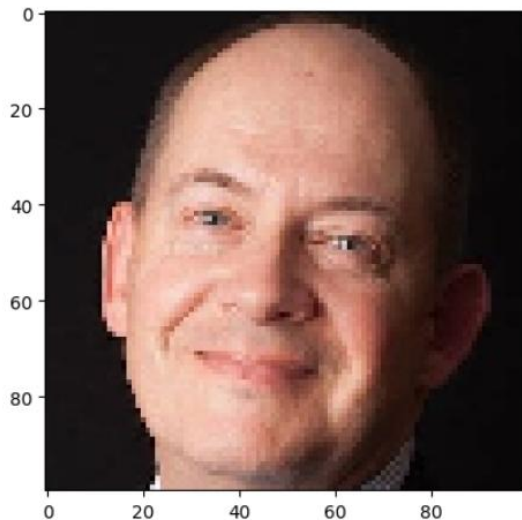
```
1 gender = {1: 'female',2:'male'}
2 img = load_img("/content/drive/MyDrive/projects/test_image/female2.jpeg",target_size=(100,100))
3 plt.imshow(img)
4 img = img_to_array(img)
5 img=img.reshape(1,100,100,3)
6 img = img.astype('float32')
7 img =img/255
8 result = np.argmax(model.predict(img),axis=1)
9 gender[result[0]]
```

1/1 [=====] - 0s 388ms/step  
'female'



```
1
2 img = load_img("/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Intelligence/Cod
3 plt.imshow(img)
4 img = img_to_array(img)
5 img=img.reshape(1,100,100,3)
6 img = img.astype('float32')
7 img =img/255
8 result = np.argmax(model.predict(img),axis=1)
9 gender[result[0]]
```

1/1 [=====] - 0s 17ms/step  
'male'

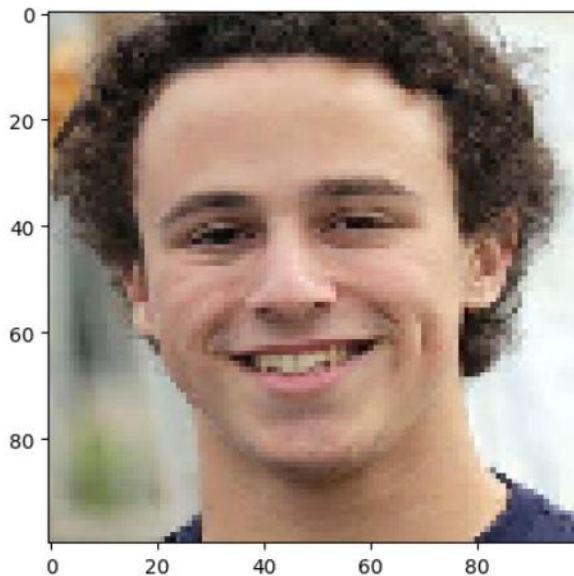


```

1 img = load_img("/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Intelli
2 plt.imshow(img)
3 img = img_to_array(img)
4 img=img.reshape(1,100,100,3)
5 img = img.astype('float32')
6 img =img/255
7 result = np.argmax(model.predict(img),axis=1)
8 gender[result[0]]

```

1/1 [=====] - 0s 32ms/step  
'male'



## 1.2. Tiếp tục kiểm tra model với chương trình Python đơn giản

```

1 # làm được hiển thị chữ trên hình
2
3 import matplotlib.pyplot as plt
4 from keras.utils import load_img
5 from keras.utils.image_utils import img_to_array
6 import numpy as np
7 vat = {1: 'female',2:'male'}
8
9 test_image = cv2.imread('/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Artificial_Intelli
10 gray = cv2.cvtColor(test_image,cv2.COLOR_BGR2GRAY)
11 face_cascade = cv2.CascadeClassifier('/content/drive/MyDrive/A_HOC_TAP/Nam_4_Hoc_ky_2/Arti
12 faces = face_cascade.detectMultiScale(gray, 1.3, 5)
13
14 for (x,y,w,h) in faces:
15     cv2.rectangle(test_image,(x,y),(x+w,y+h),(203,12,255),2)
16
17     img_gray = gray[(y-30):(y+h+30),(x-30):(x+w+30)]
18     cv2.imwrite('/content/drive/MyDrive/projects/test_image/fema2.jpg',img_gray)
19     img = load_img("/content/drive/MyDrive/projects/test_image/fema2.jpg",target_size=(100,100))
20     # img = img_gray
21     plt.imshow(img)
22     img = img_to_array(img)
23     img = img.reshape(1,100,100,3)
24     img = img.astype('float64')
25     img =img/255
26     result = np.argmax(model.predict(img),axis=1)

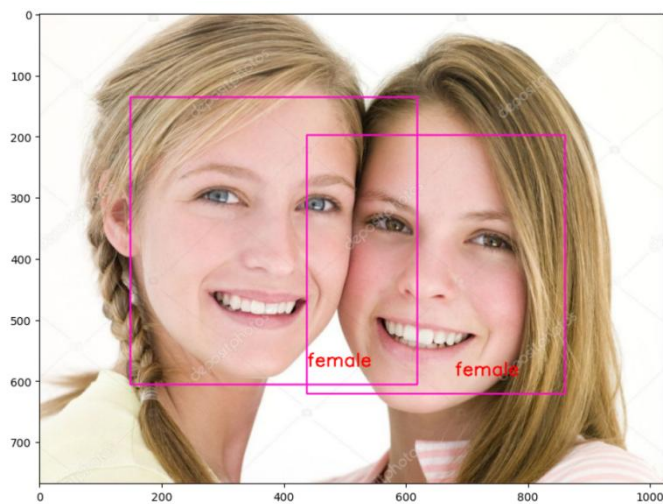
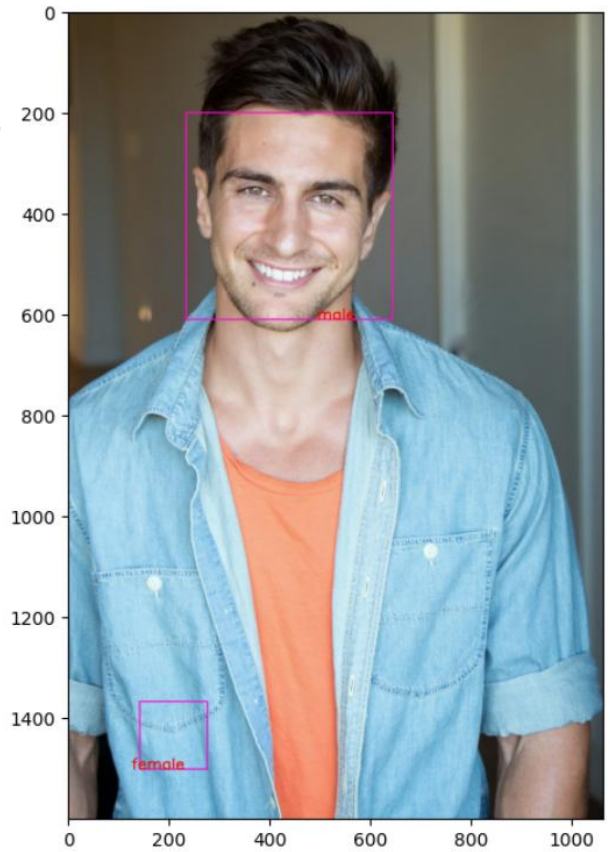
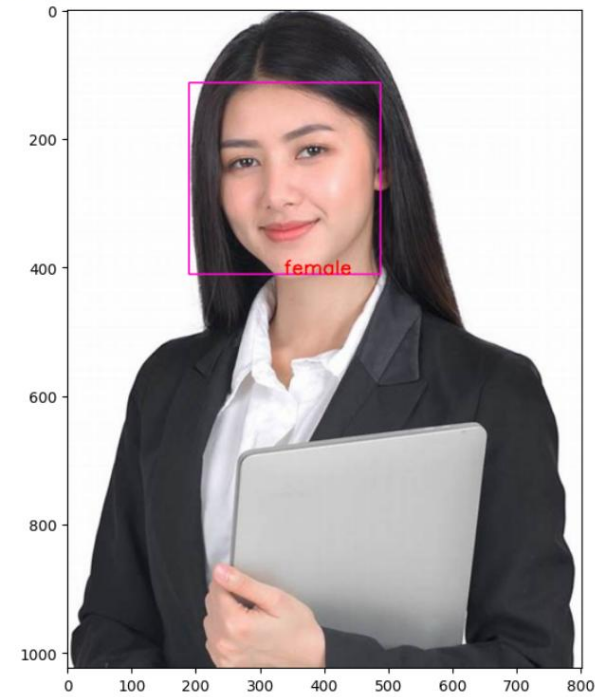
```

```

27 # plt.imshow(cv2.cvtColor(img_gray, cv2.COLOR_BGR2RGB))
28 text = str(vat[result[0]])
29 cv2.putText(test_image, text, (x+w-150, y+h), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
30 plt.imshow(cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB))
31
32 print('text', text)

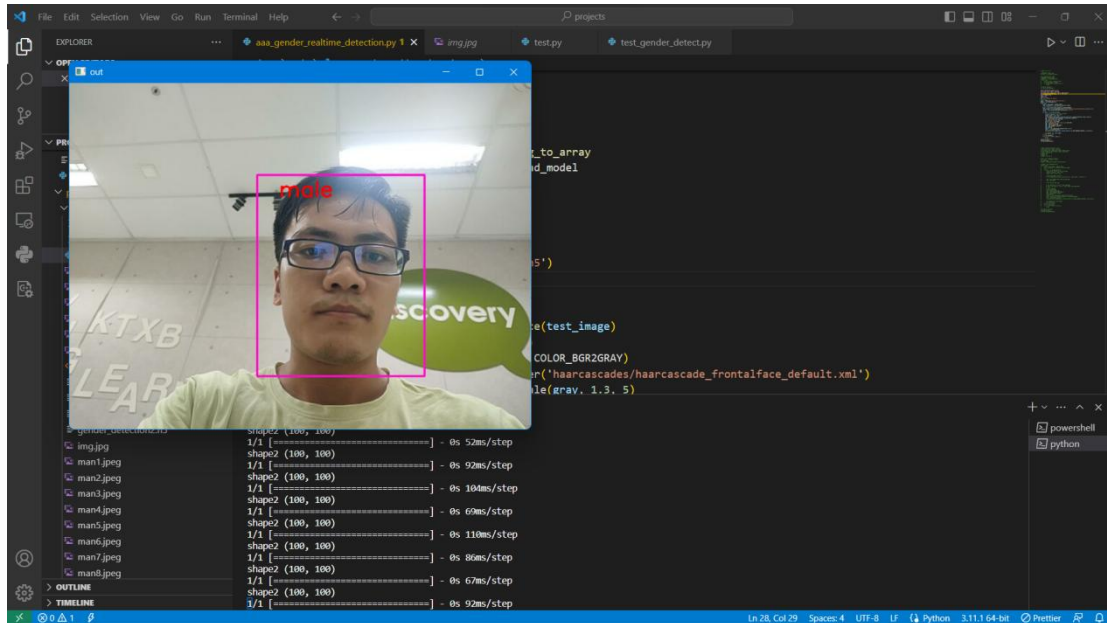
```

1/1 [=====] - 0s 111ms/step  
text female





### 1.3. Chạy thời gian thực trên Visual Code



## V. Kết luận

Bài làm đã thực hiện đúng và đủ chỉ tiêu đề ra đó là nhận diện giới tính con người theo thời gian thực. Kết quả trả về có độ chính xác tương đối, phù hợp với một đề tài với mục đích học hiểu.

Với lượng dữ liệu đầu vào ít, chất lượng của ‘model’ được huấn luyện có độ chính xác không cao, với những trường hợp như nữ để tóc ngắn hoặc nam có khuôn mặt tròn thì ‘model’ có thể cho ra kết quả sai.

‘Model’ nếu dùng nhận diện người lớn tuổi (>65 tuổi) hoặc người dưới 18 tuổi cũng có tỉ lệ sai cao.

Để cải thiện chất lượng của ‘model’, cần một lượng lớn dữ liệu đầu vào (>10,000 tấm hình). Một nguồn dữ liệu hay được biết đến trong lĩnh vực nhận diện khuôn mặt có thể là ‘UTK face datasets’ với hơn 23,000 tấm hình làm dữ liệu.

## B. Tài liệu tham khảo

- [1] UTK face datasets (<https://susanqq.github.io/UTKFace/>)
- [2] Age and gender detection using Deep learning  
(<https://www.analyticsvidhya.com/blog/2021/07/age-and-gender-detection-using-deep-learning/>)
- [3] Age, gender and ethnicity  
(<https://www.kaggle.com/datasets/nipunarora8/age-gender-and-ethnicity-face-data-csv/>)
- [4] Introduction to Convolution neural network  
(<https://www.geeksforgeeks.org/introduction-convolution-neural-network/>)
- [5] Convolution neural network (<https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network>)
- [6] How Convolution neural network (CNN) work?  
(<https://medium.com/mlearning-ai/how-does-the-convolutional-neural-network-cnn-work-dcc46d68cd1c>)
- [7] Cascade classifier  
([https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html))



## Thông tin đính kèm

- [1] Link Google Drive ([https://drive.google.com/drive/u/o/folders/1XoqS4Nlp-sXEfTsypLNADbf\\_ioVJqf6R](https://drive.google.com/drive/u/o/folders/1XoqS4Nlp-sXEfTsypLNADbf_ioVJqf6R))
- [2] Link Github (<https://github.com/thanhtruongphan>)
- [3] Link Youtube (<https://www.youtube.com/@truonglaptrinh7972/featured>)
- [4] Link Data  
([https://drive.google.com/drive/u/o/folders/1fQ4yRwlo27JTujnMLouAumdgGB\\_R3oIA](https://drive.google.com/drive/u/o/folders/1fQ4yRwlo27JTujnMLouAumdgGB_R3oIA))