

Face Detection and Identification

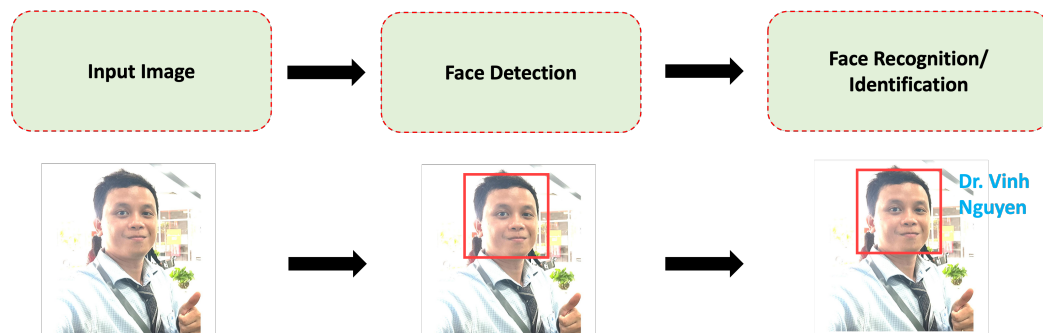
(Computer Vision Foundation)

Ngày 20 tháng 10 năm 2023

Giới thiệu về bài tập xây dựng ứng dụng nhận diện khuôn mặt sử dụng các kỹ thuật computer vision cơ bản

Công nghệ AI nhận diện khuôn mặt là công nghệ ứng dụng trí tuệ nhân tạo vào việc xác định danh tính thông qua việc phân tích đặc điểm khuôn mặt (có thể là tròng mắt, hình dạng, kích thước, khoảng cách và đặc tính giữa các bộ phận).

Ngày nay, với tốc độ phát triển mạnh mẽ của khoa học kỹ thuật, hệ thống AI nhận diện có thể dựa vào các điểm nổi bật trên khuôn mặt như những mô cứng và xương. Từ đó nhìn thấy rõ nhất đường cong của hốc mắt, mũi và cằm để nhận ra đối tượng. Các đặc điểm này với mỗi khuôn mặt là độc nhất và không thay đổi theo thời gian. Hệ thống có thể hoạt động trong nhiều điều kiện thời tiết khác nhau, không ngoại trừ ban ngày hay ban đêm, khi môi trường không đủ điều kiện ánh sáng hoặc trong bóng tối.

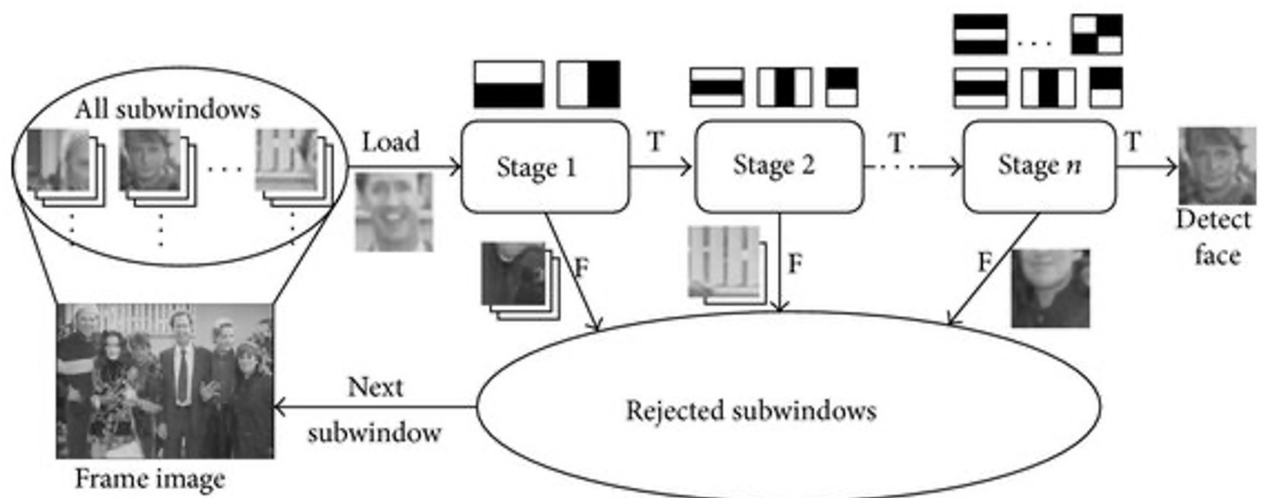


Hình 1: kết quả phát hiện và nhận diện khuôn mặt sử dụng các kỹ thuật computer vision cơ bản

Để hoàn thành bài tập này bạn cần từng bước hoàn thiện các bài tập nhỏ bên dưới.

Bài tập 1: (Phát hiện khuôn mặt từ hình ảnh sử dụng bộ phân loại Haar Cascade). Haar Cascade là một thuật toán được tạo ra dựa trên những tính năng đó để phát hiện đối tượng (có thể là khuôn mặt, mắt, tay, đồ vật,...) được đề xuất vào năm 2001 bởi Paul Viola và Michael Jones trong bài báo của họ với khẳng định “Rapid object detection using a boosted cascade of simple features”. Hình 2, thể hiện sơ đồ của giải thuật phát hiện khuôn mặt sử dụng bộ phân loại Haar cascade. Yêu cầu của bài tập này là bạn hãy hoàn thiện đoạn code bên dưới để phát hiện vị trí khuôn mặt trên ảnh sử dụng thư viện OpenCV.

```
1 #Problem 1
2 import os
3 import cv2
4 import pickle
5 import numpy as np
```



Hình 2: Sơ đồ giải thuật nhận diện khuôn mặt dựa trên Haar Cascade Classifier

```

6
7 #Read image from the file
8 image = cv2.imread("vinh.jpg", 1)
9
10 # ***** Your code here *****
11
12 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
13
14 face_coordinates = facecascade.detectMultiScale(gray, 1.3, 4)
15
16 for (a, b, w, h) in face_coordinates:
17     cv2.rectangle(image, (a, b), (a+w, b+h), (255, 0, 0), 2)
18
19 #Question 1:
20 print("Face detection's coordinate: ", face_coordinates)
21 cv2.imshow('frames', image)
22 cv2.waitKey(0)

```

Question 1: Hãy cho biết kết quả sau khi thực hiện đoạn code ở bài tập 1

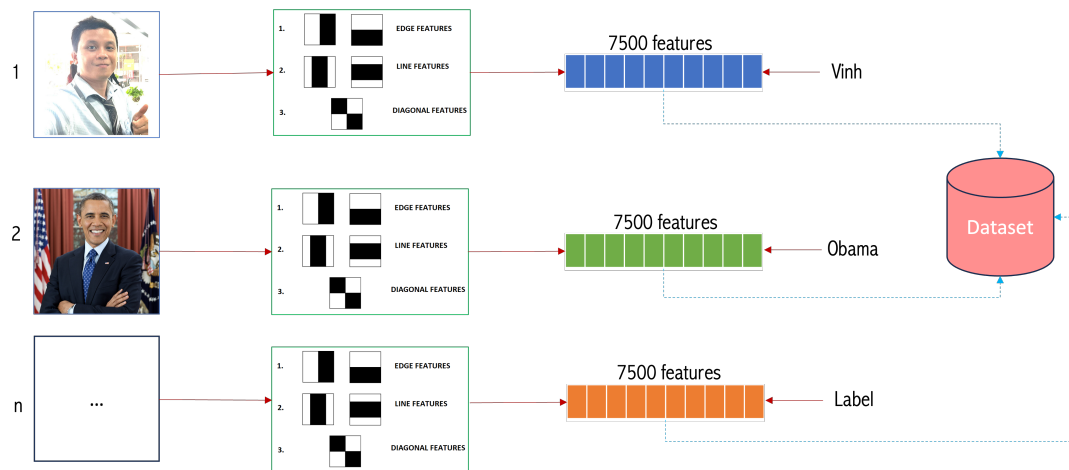
- a) Face detection s coordinate: [547 189 298 298]
- b) Face detection s coordinate: [647 189 298 298]
- c) Face detection s coordinate: [747 189 298 298]
- d) Face detection s coordinate: [947 189 298 298]

Bài tập 2: (Xây dựng cơ sở dữ liệu cho ứng dụng nhận diện khuôn mặt) Yêu cầu của bài tập này là hoàn thiện đoạn code bên dưới để chuẩn bị dữ liệu giải thuật nhận diện khuôn mặt. Hình 3 thể hiện quy trình rút trích đặc trưng của ảnh đầu vào thành 7500 features sử dụng kỹ thuật haar cascade classifier.

```

1
2 import os
3 import cv2
4 import pickle
5 import numpy as np
6

```



Hình 3: Quy trình xây dựng feature đặc trưng từ ảnh đầu vào sử dụng kỹ thuật Haar classifier

```

7 face_data = []
8 i = 0
9
10 camera = cv2.VideoCapture(0)
11
12 facecascade = cv2.CascadeClassifier(cv2.data.harcascades + '
    haarcascade_frontalface_default.xml')
13
14 name = input('Enter your name --> ')
15 ret = True
16
17 while(ret):
18     ret, frame = camera.read()
19     if ret == True:
20         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
21
22         face_coordinates = facecascade.detectMultiScale(gray, 1.3, 4)
23
24         ***** your code here *****
25
26
27         cv2.imshow('frames', frame)
28
29         if cv2.waitKey(1) == 27:
30             break
31     else:
32         print('error')
33         break
34
35
36 cv2.destroyAllWindows()
37 camera.release()
38 close_window_os()
39
40
41 face_data = np.asarray(face_data)
42 face_data = face_data.reshape(10, -1)
43
44 #Store face data to the file
45 if 'names.pkl' not in os.listdir('/dataset/'):

```

```

46     names = [name]*10
47     with open('/dataset/names.pkl', 'wb') as file:
48         pickle.dump(names, file)
49 else:
50     with open('/dataset/names.pkl', 'rb') as file:
51         names = pickle.load(file)
52
53     names = names + [name]*10
54     with open('/dataset/names.pkl', 'wb') as file:
55         pickle.dump(names, file)
56
57
58 if 'faces.pkl' not in os.listdir('/dataset/'):
59     with open('/dataset/faces.pkl', 'wb') as w:
60         pickle.dump(face_data, w)
61 else:
62     with open('/dataset/faces.pkl', 'rb') as w:
63         faces = pickle.load(w)
64
65     faces = np.append(faces, face_data, axis=0)
66     with open('/dataset/faces.pkl', 'wb') as w:
67         pickle.dump(faces, w)

```

Question 2: Hãy cho biết kết quả sau khi thực hiện đoạn code ở bài tập 2

- a) File faces.pkl contains label data and face data
- b) File faces.pkl contains face data, and file names.plk contains face labels
- c) File faces.pkl contains face labels, and file names.plk contains face data
- d) File faces.pkl and file names.plk do not contain face labels

Bài tập 3: (Sử dụng giải thuật KNN cho bài toán nhận diện khuôn mặt là ai) . Sử dụng kết quả file lưu trữ ở bài tập 2 (faces.pkl và names.plk) để xây dựng ứng dụng nhận diện khuôn mặt sử dụng camera từ máy tính bằng giải thuật KNN.

```

1 #KNN for face identification
2
3 import cv2
4 from sklearn.svm import SVC
5 import numpy as np
6 import pickle
7
8 facecascade = cv2.CascadeClassifier(cv2.data.haarcascades + '
9     haarcascade_frontalface_default.xml')
10
11 with open('/dataset/faces.pkl', 'rb') as w:
12     faces = pickle.load(w)
13
14 with open('/dataset/names.pkl', 'rb') as file:
15     labels = pickle.load(file)
16
17 camera = cv2.VideoCapture(0)
18
19 print('Shape of Faces matrix --> ', faces.shape)
20 # your knn code here *****
21
22 # Face recognition using Knn
23 while True:
24     ret, frame = camera.read()
25     if ret == True:

```

```

25     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
26     face_coordinates = facecascade.detectMultiScale(gray, 1.3, 5)
27
28     for (a, b, w, h) in face_coordinates:
29         fc = frame[b:b + h, a:a + w, :]
30         r = cv2.resize(fc, (50, 50)).flatten().reshape(1,-1)
31         text = knn.predict(r)
32         cv2.putText(frame, text[0], (a, b-10), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,
255, 0), 2)
33         cv2.rectangle(frame, (a, b), (a + w, b + w), (0, 0, 255), 2)
34
35     cv2.imshow('livetime face recognition', frame)
36     if cv2.waitKey(1) == 27:
37         break
38     else:
39         print("error")
40         break
41
42 cv2.destroyAllWindows()
43 camera.release()
44 close_window_os()

```

Question 3: Hãy cho biết lập luận nào sau đây là đúng về KNN

- a) KNN uses 7500 features to measure the face distance
- b) KNN uses 2500 features to measure the face distance
- c) KNN uses 5000 features to measure the face distance
- d) KNN uses 7000 features to measure the face distance

Bài tập 4: (Sử dụng giải thuật SVM cho bài toán nhận diện khuôn mặt là ai) . Sử dụng kết quả file lưu trữ ở bài tập 2 (faces.pkl và names.pkl) để xây dựng ứng dụng nhận diện khuôn mặt sử dụng camera từ máy tính sử dụng giải thuật SVM.

```

1
2 import cv2
3 from sklearn.neighbors import KNeighborsClassifier
4 import numpy as np
5 import pickle
6
7 facecascade = cv2.CascadeClassifier(cv2.data.haarcascades + '
haarcascade_frontalface_default.xml')
8
9 with open('/dataset/faces.pkl', 'rb') as w:
10     faces = pickle.load(w)
11
12 with open('/dataset/names.pkl', 'rb') as file:
13     labels = pickle.load(file)
14
15 camera = cv2.VideoCapture(0)
16
17 # ***** your svm code here *****
18
19 # Face recognition using svm
20 while True:
21     ret, frame = camera.read()
22     if ret == True:
23         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

```

```

24     face_coordinates = facecascade.detectMultiScale(gray, 1.3, 5)
25
26     for (a, b, w, h) in face_coordinates:
27         fc = frame[b:b + h, a:a + w, :]
28         r = cv2.resize(fc, (50, 50)).flatten().reshape(1,-1)
29         text = svm.predict(r)
30         cv2.putText(frame, text[0], (a, b-10), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,
255, 0), 2)
31         cv2.rectangle(frame, (a, b), (a + w, b + w), (0, 0, 255), 2)
32
33     cv2.imshow('livetime face recognition', frame)
34     if cv2.waitKey(1) == 27:
35         break
36     else:
37         print("error")
38         break
39
40 cv2.destroyAllWindows()
41 camera.release()
42 close_window_os()

```

Question 4: Hãy cho biết lập luận nào sau đây là đúng về SVM

- a) SVM might under fitting when setting C is large
- b) SVM might overfitting when setting C is small
- c) SVM might overfitting when setting C is large
- d) SVM might under when setting C is small

Bài tập 5: (Sử dụng giải thuật Decision Tree cho bài toán nhận diện khuôn mặt là ai) . Sử dụng kết quả file lưu trữ ở bài tập 2 (faces.pkl và names.pkl) để xây dựng ứng dụng nhận diện khuôn mặt sử dụng camera từ máy tính sử dụng giải thuật Decision Tree.

```

1 import cv2
2 from sklearn.tree import DecisionTreeClassifier
3 import numpy as np
4 import pickle
5
6 facecascade = cv2.CascadeClassifier(cv2.data.harcascades + '
    haarcascade_frontalface_default.xml')
7
8 with open('/dataset/faces.pkl', 'rb') as w:
9     faces = pickle.load(w)
10
11 with open('/dataset/names.pkl', 'rb') as file:
12     labels = pickle.load(file)
13
14 camera = cv2.VideoCapture(0)
15
16 # ***** Your decision tree code here *****8
17
18 # Face recognition using decision tree
19 while True:
20     ret, frame = camera.read()
21     if ret == True:
22         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

```

```
23     face_coordinates = facecascade.detectMultiScale(gray, 1.3, 5)
24
25     for (a, b, w, h) in face_coordinates:
26         fc = frame[b:b + h, a:a + w, :]
27         r = cv2.resize(fc, (50, 50)).flatten().reshape(1,-1)
28         text = tree_clf.predict(r)
29         cv2.putText(frame, text[0], (a, b-10), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,
255, 0), 2)
30         cv2.rectangle(frame, (a, b), (a + w, b + w), (0, 0, 255), 2)
31
32     cv2.imshow('lifetime face recognition', frame)
33     if cv2.waitKey(1) == 27:
34         break
35     else:
36         print("error")
37         break
38
39 cv2.destroyAllWindows()
40 camera.release()
41 close_window_os()
```

Question 5: Hãy cho biết lập luận nào sau đây là đúng về Decision Tree

- a) We might avoid overfitting by changing evaluation metric from entropy to gini
- b) We might avoid overfitting by increasing depth of the tree
- c) We might avoid overfitting by increasing data size
- d) We might avoid overfitting by using random forest

Bài tập 6: (In class discussion) Hãy cho biết ưu nhược điểm của KNN, SVM và Decision Tree cho bài toán nhận diện khuôn mặt? Giải pháp của bạn để khắc phục các nhược điểm này là gì?