ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
-oOo-

ĐỒ ÁN CUỐI KỲ

MÔN: CS231 - Nhập môn thị giác máy tính

Đề tài: Nhận diện kiểu khuôn mặt

Giảng viên hướng dẫn: Mai Tiến Dũng

Thành phố Hồ Chí Minh, ngày 9 tháng 6 năm 2024

I. Danh sách thành viên:

Bảng 1.1. Danh sách thành viên nhóm

STT	Mã số sinh viên	Họ và tên	Mức độ hoàn thành
1	22521200	Hồ Trọng Duy Quang	100%
2	22520917	Nguyễn Hữu Nam	100%
3	22521135	Nguyễn Trần Phúc	100%

II. Giới thiệu bài toán

2.1. Lý do thực hiện bài toán:

2.1.1. Trong vấn đề thẩm mỹ:

- Việc xác định kiểu khuôn mặt giúp các chuyên gia thẩm mỹ đưa ra lời khuyên và phương pháp điều trị phù hợp nhất để cân bằng hoặc nổi bật các đặc điểm của khuôn mặt. Điều này bao gồm việc lựa chọn kiểu tóc, cách trang điểm, và thậm chí là các thủ thuật phẫu thuật thẩm mỹ.

2.1.2. Trong vấn đề về nhân tướng học:

- Trong lĩnh vực nhân tướng học, việc phân tích khuôn mặt được coi là cách để hiểu rõ hơn về tính cách và vận mệnh của một người. Mỗi kiểu khuôn mặt được cho là mang lại những đặc điểm riêng biệt về tính cách và cách thức tương tác với thế giới xung quanh. Những nhận định này có thể hỗ trợ trong các mối quan hệ cá nhân và nghề nghiệp.

2.2. Phát biểu bài toán:

2.2.1. Input:

- Ảnh khuôn mặt chính diện của 1 người lấy phần trên cơ thể.
- Không đeo mắt kính, thấy rõ trán.

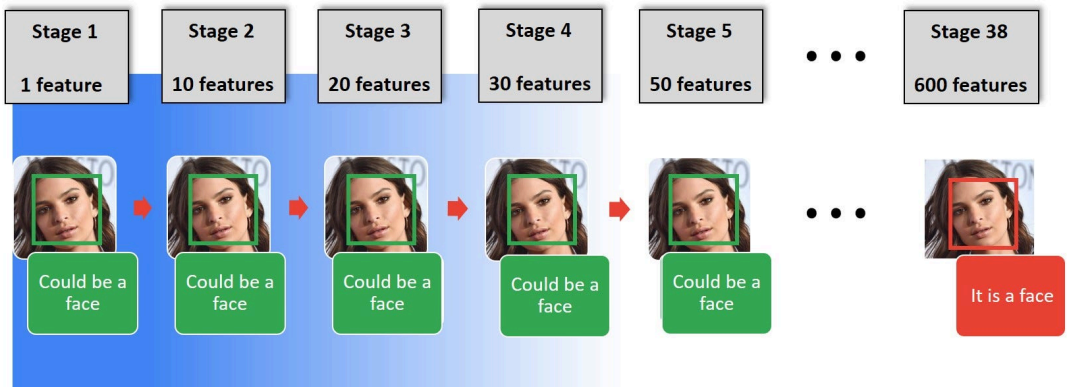
2.2.2. Output:

- Kiểu khuôn mặt được dự đoán (Heart, Oval, Round, Square).

III. Phương pháp giải quyết:

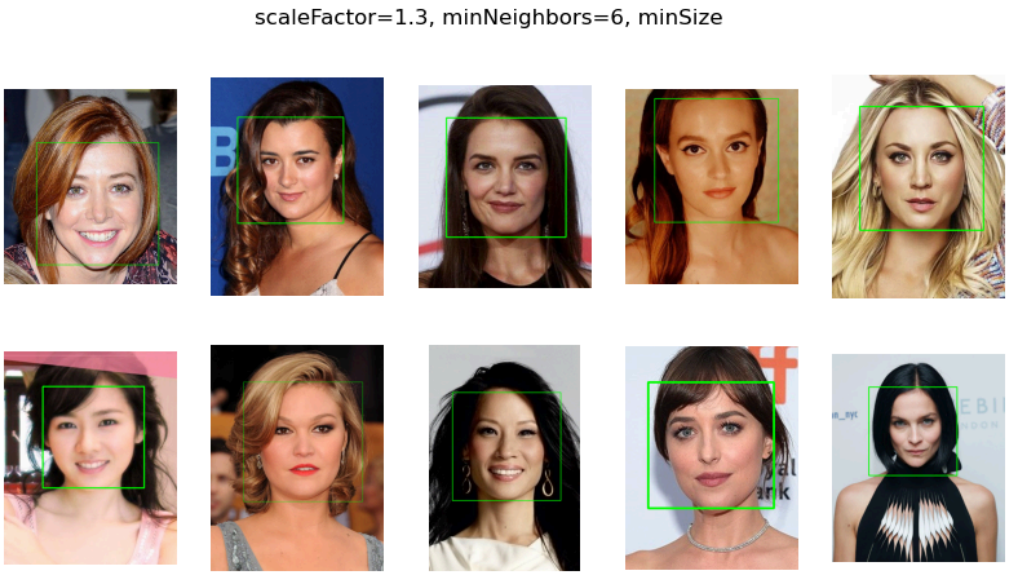
3.1. Tiền xử lý dữ liệu:

3.1.1. Haar Cascade và cắt ảnh:



Hình 3.1. Hình ảnh minh họa Haar Cascade

- Haar Cascade là một phương pháp phát hiện đối tượng trong ảnh từ thư viện OpenCV. Thuật toán này sử dụng các đặc trưng Haar và một cây phân loại cascade để nhận diện khuôn mặt hoặc các đối tượng khác. Ở đây chúng em sử dụng hàm này với mục đích phát hiện và cắt khuôn từ trong ảnh.
- Các tham số chúng ta cần quan tâm:
    + *scaleFactor*: giá trị trong đoạn [1.1, 2.0], cửa sổ trượt sau mỗi lần xử lí xong sẽ tăng thêm kích thước theo scaleFactor để xử lí lại bức ảnh 1 lần nữa. Giá trị càng lớn thuật toán sẽ chạy càng nhanh nhưng có thể sẽ bỏ lỡ các chi tiết khác.
    + *minNeighbors*: tham số này quyết định số cửa sổ gần kề để xác định rằng đây có phải đối tượng cần tìm không, trong bài này là khuôn mặt. Tham số càng lớn càng ít lỗi nhưng cũng có thể sẽ bỏ lỡ luôn khuôn mặt chúng ta cần tìm.
    + *minSize*: kích cỡ ban đầu cho cửa sổ.
- Sau một thời gian nghiên cứu, chúng em tìm được tham số tối ưu cho bài toán này:
    + *scaleFactor*: 1.3
    + *minNeighbors*: 6
    + *minSize*: (width / 2, height / 2) hoặc (width / 2.6, height / 2.6)



Hình 3.2. Kết quả sau khi sử dụng các tham số tối ưu

- Sau khi cắt ảnh xong, chúng em bỏ vào thư mục mới, chuẩn bị cho bước trích xuất đặc trưng HOG.

3.1.2. Đặc trưng HOG:

- Tất cả các ảnh đều được chuyển về ảnh xám và resize về kích thước 128x128.
- Tham số sử dụng trong hàm hog để trích xuất đặc trưng:
  + *Orientations*: 9
  + *Pixels per cell*: (8, 8)
  + *Cells per block*: (4, 4)
  + *Block norm*: L2
- Bộ tham số trên là bộ tham số cho ra kết quả tốt nhất sau khi chạy Grid Search với các bộ tham số.

```
param_grid = {'orientations': [9],
              'pixels_per_cell': [(8, 8), (16, 16)],
              'cells_per_block': [(2, 2), (4, 4)]}
```

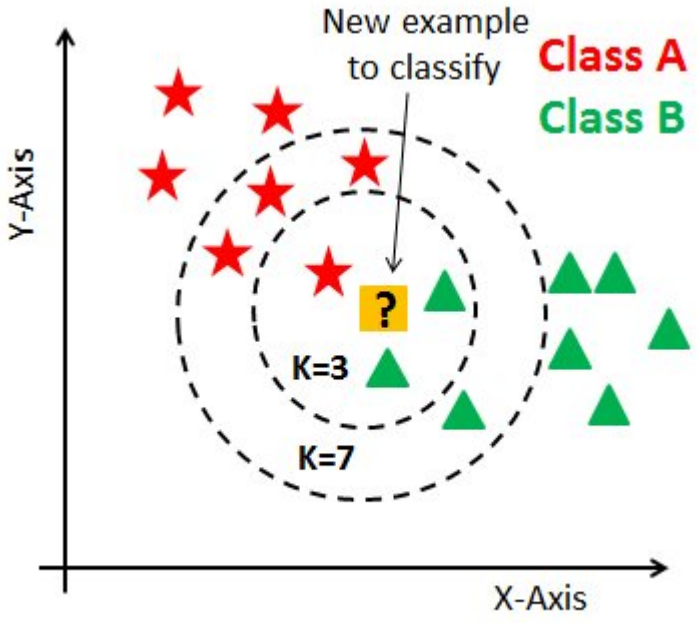
```
Best parameters: {'cells_per_block': (4, 4), 'orientations': 9, 'pixels_per_cell': (8, 8)}
Best accuracy: 0.6525096525096525
```

Hình 3.3. Tham số tốt nhất khi dùng HOG

3.2. Các mô hình máy học:

3.2.1. K-nearest neighbour (KNN):

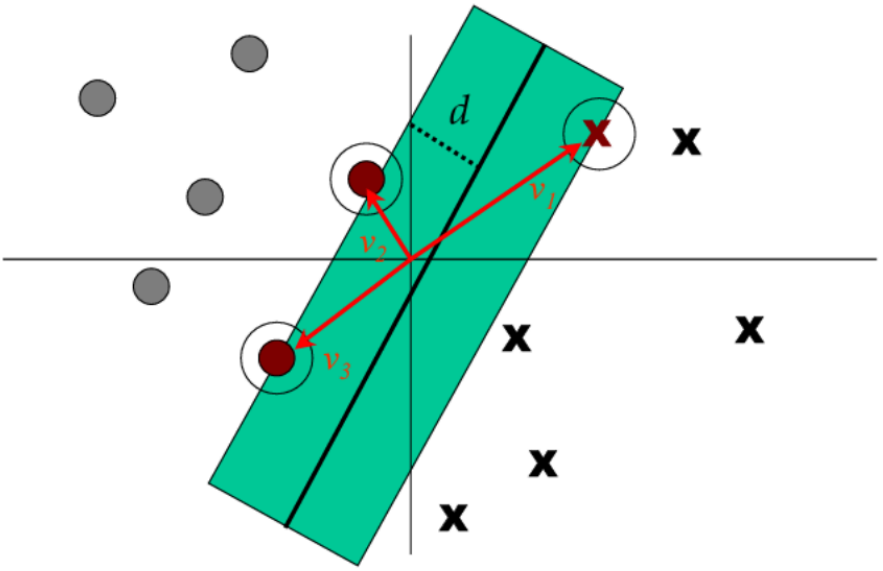
- KNN là một trong những thuật toán supervised-learning đơn giản nhất, thuật toán này không học một điều gì từ dữ liệu training, mọi tính toán được thực hiện khi nó cần kết quả của dữ liệu mới. Có thể dùng cho Classification và Regression.



Hình 3.4. Hình ảnh minh họa thuật toán KNN

3.2.2. SVM classification:

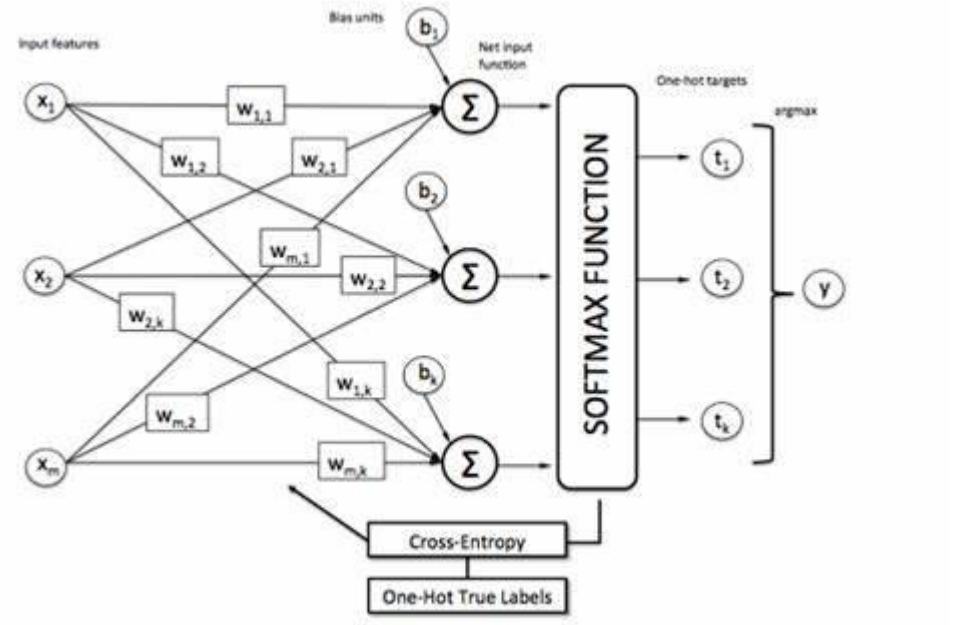
- SVM là thuật toán học có giám sát dùng để phân loại bằng cách chia các điểm bằng siêu phẳng N chiều thành các lớp khác nhau.



Hình 3.5. Hình ảnh minh họa cho thuật toán SVM

3.2.3. Softmax classification:

- Một softmax classifier (hay còn được gọi là logistic classifier đa lớp) là một mô hình phân loại thường được sử dụng trong các bài toán có nhiều lớp nhãn khác nhau.



Hình 3.6. Minh họa cho thuật toán Softmax classification

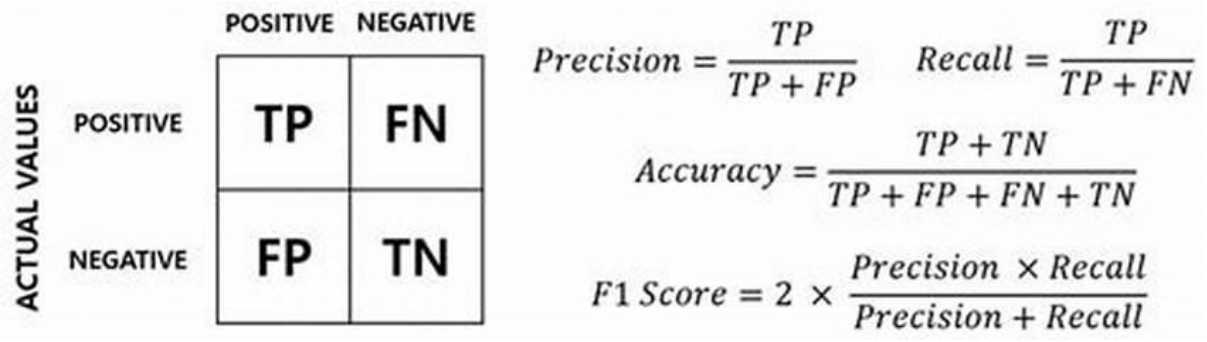
IV. Kết quả thực nghiệm:

4.1. Dataset:

- Bộ dữ liệu được lấy từ Kaggle, gồm tập train và test (Mỗi tập được chia thành 4 nhóm bao gồm Heart, Oval, Round, Square).

4.2. Độ đo:

- Precision được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive.
- Recall được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive.
- F1-score, là harmonic mean của precision và recall.
- Accuracy là tỉ lệ giữa số điểm được phân loại đúng và tổng số điểm.



Hình 4.1. Công thức Precision, Recall, Accuracy, F1 score

4.2. Lập trình:

4.2.1. K-Nearest Neighbor

- Sau khi thực hiện tìm tham số tốt nhất bằng Grid-Search, ta được bộ tham số tốt nhất: **n_neighbors: 6, 'weights': 'distance'.**

```
{'n_neighbors': 10, 'weights': 'distance'} mean cross-validation score: 0.656595126486654
{'n_neighbors': 10, 'weights': 'uniform'} mean cross-validation score: 0.6374722039282531
Best parameters: {'n_neighbors': 6, 'weights': 'distance'}
Best cross-validation score: 0.6618950399669418
```

Hình 4.2. Bộ tham số tốt nhất của KNN trên bộ dataset

- Thực hiện cài đặt KNN:

```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors = 6, weights = 'distance')
model.fit(x_train, y_train)
```

Hình 4.3. Cài đặt KNN

4.2.2. Support Vector Machine:

- Sau khi thực hiện tìm tham số tốt nhất bằng Grid-Search, ta được bộ tham số tốt nhất: **C = 5, 'gamma': 'scale'.**

```
    best_params = search.best_params_
    best_score = search.best_score_

    print(f"Best parameters: {best_params}")
    print(f"Best cross-validation score: {best_score}")
]

Best parameters: {'C': 5, 'gamma': 'scale'}
Best cross-validation score: 0.5479999999999999
```

Hình 4.4. Bộ tham số tốt nhất của SVM trên bộ dataset

- Thực hiện cài đặt SVM:

```
from sklearn.svm import SVC

model = SVC(kernel = 'rbf', C = 5, gamma="scale")
model.fit(x_train, y_train)
```

Hình 4.5. Cài đặt SVM

4.2.3. Soft-max classification:

- Sau khi thực hiện tìm tham số tốt nhất bằng Grid-Search, ta được bộ tham số tốt nhất: **C = 0.1, 'solver': 'lbfgs'.**

```
{'C': 1, 'solver': 'sag'} mean cross-validation score: 0.505
{'C': 1, 'solver': 'saga'} mean cross-validation score: 0.505
Best parameters: {'C': 0.1, 'solver': 'lbfgs'}
Best cross-validation score: 0.515
```

Hình 4.6. Bộ tham số tốt nhất của Soft-Max trên bộ dataset

- Thực hiện cài đặt Soft-max:

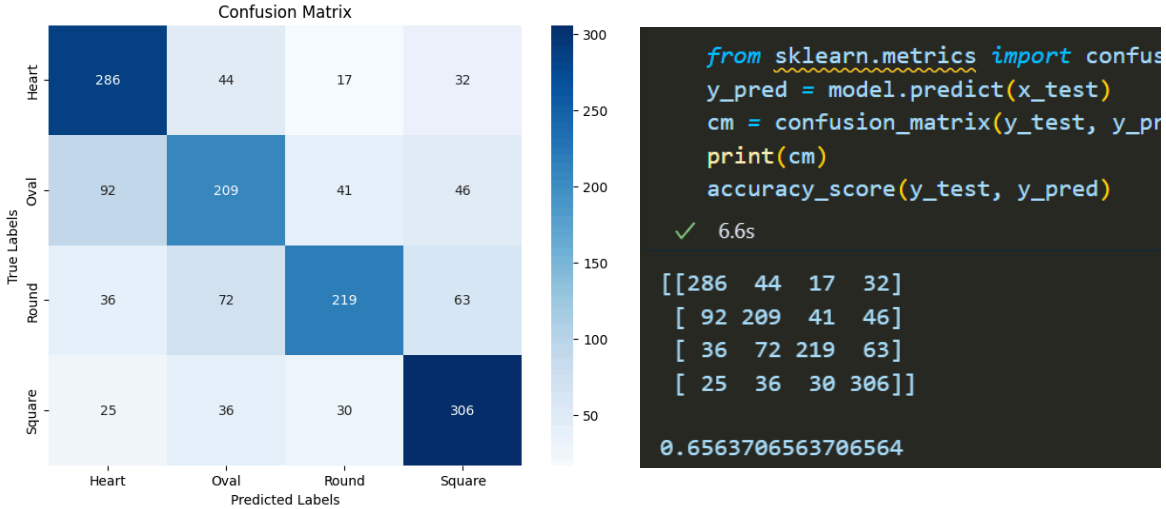
```
model = LogisticRegression(C=0.1, solver='lbfgs', max_iter=1000, random_state=42)
model.fit(x_train, y_train)
```

Hình 4.7. Cài đặt Soft-max

4.3. Kết quả:

4.3.1. K-Nearest Neighbor:

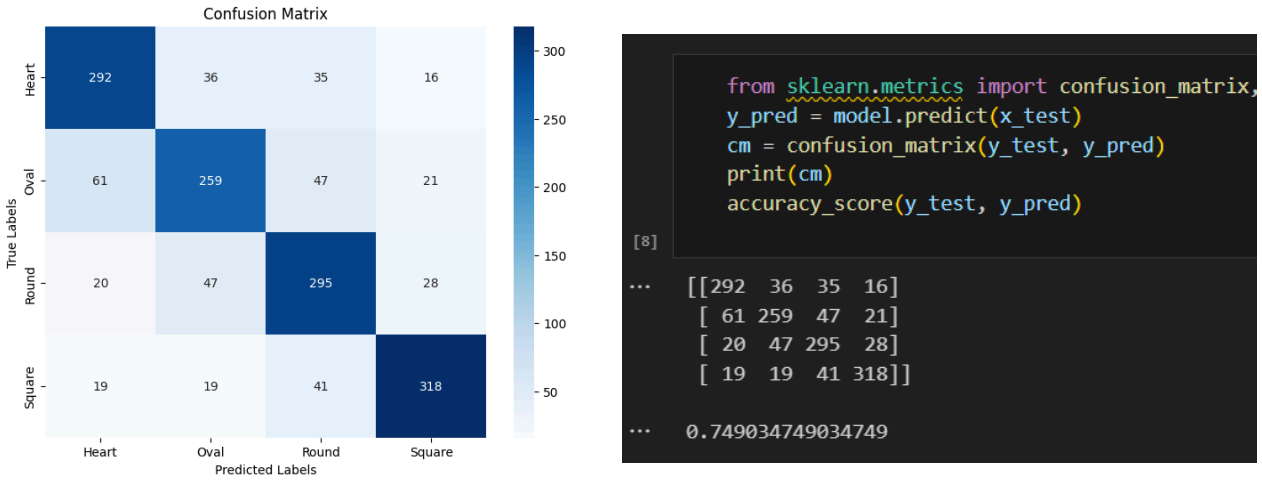
- Accuracy: 0.65



Hình 4.8. Accuracy và confusion matrix của KNN

4.3.2. Support Vector Machine:

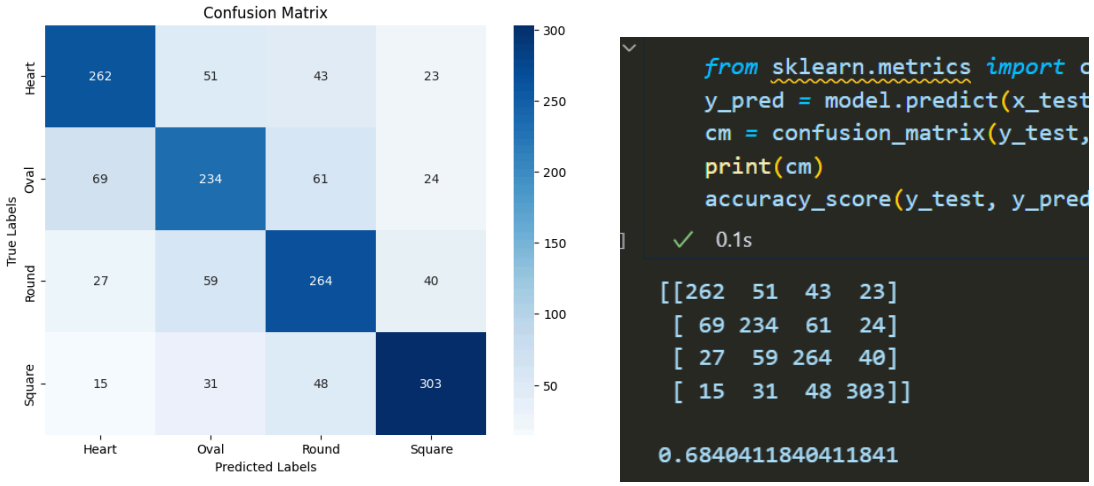
- Accuracy: 0.75



Hình 4.9. Accuracy và confusion matrix của SVM

4.3.3. Soft-max classification:

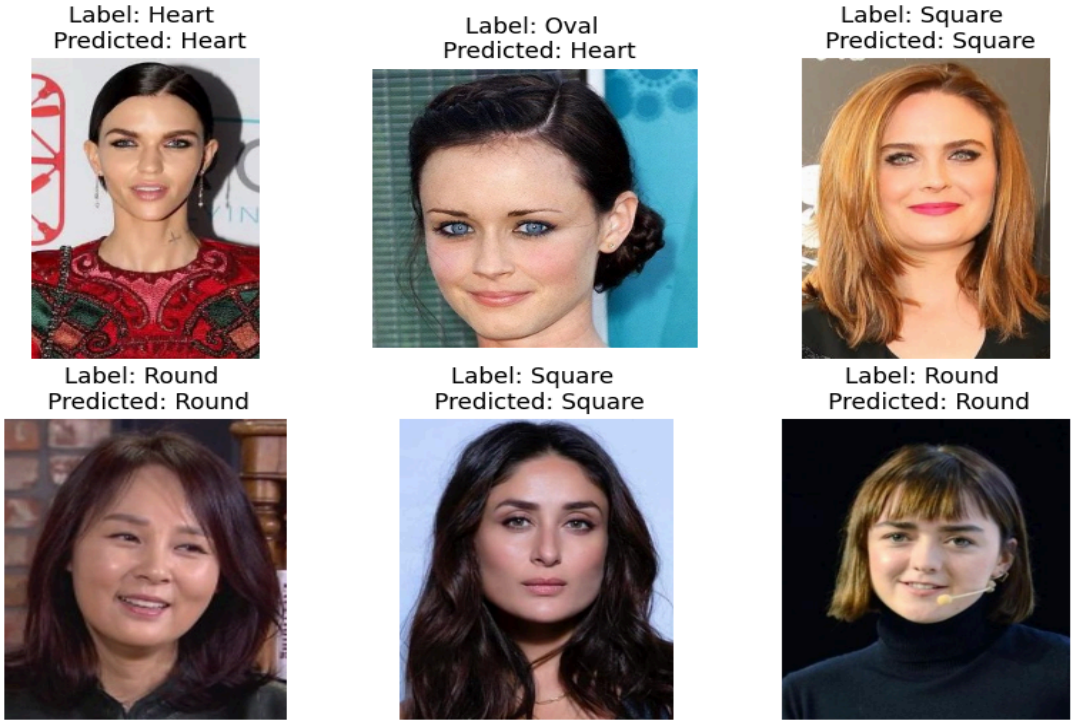
- Accuracy: 0.68



Hình 4.10. Accuracy và confusion matrix của Soft-max

4.3.4. Một số kết quả chạy thử từ tập test:

- Nhìn chung khả năng nhận diện của chương trình khá tốt, trong 6 bức ảnh lấy từ tập test chỉ sai 1 ảnh.



Hình 4.11. Một số kết quả chạy thử từ tập test

V. Đánh giá:

5.1. Ưu điểm:

- Mô hình cài đặt nhận diện được khá tốt, với độ chính xác ổn định.
- Có khả năng phân biệt tốt các kiểu mặt khác nhau.

5.2. Hạn chế:

- Chỉ mới nhận diện được 4 kiểu khuôn mặt (Heart, Round, Oval, Square) còn những kiểu mặt khác nằm ngoài dataset.
- Độ chính xác còn chưa tốt bằng các mô hình học sâu.

TÀI LIỆU THAM KHẢO

[1] Giới thiệu về SVM [Trực tuyến].
Địa chỉ: https://machinelearningcoban.com/2017/04/09/smv/

[2] Giới thiệu về KNN [Trực tuyến].
Địa chỉ: K-Nearest Neighbor(KNN) Algorithm - GeeksforGeeks

[3] Giới thiệu về Soft-max [Trực tuyến].
Địa chỉ: Machine Learning cơ bản (machinelearningcoban.com)

[4] Giới thiệu về các độ đo [Trực tuyến].
Địa chỉ: F1 Score in Machine Learning - GeeksforGeeks

[5] Face shape dataset [Trực tuyến].
Địa chỉ: Face Shape Dataset (kaggle.com)