

# BÁO CÁO DEMO 2

## THÔNG TIN NHÓM

**Thành viên 1:** Trần Quang Huy - 1512211

**Thành viên 2:** Phùng Tiến Hào - 1612174

**Thành viên 3:** Võ Quốc Huy – 1612269

## Link download:

- Demo: [Demo2.rar](#)
- Dataset: [Dataset.rar](#)

## I/ Giới thiệu:

Đây là phiên bản cải tiến và bổ sung thử nghiệm trên tập dữ liệu lớn. Nhóm đã cải tiến sử dụng Multi-task CNN để detect face thay vì sử dụng model SSD (Single Shot multibox Detector).

Kiểm tra kết quả nhận diện khuôn mặt của các nhân vật trong các frame ảnh lấy từ phim Avenger: Infinity War. Các frame ảnh này sẽ thể hiện được sự đa dạng về khoảng cách của các khuôn mặt đến camera (từ gần đến xa), cũng như sự đa dạng về hướng quay của các khuôn mặt và mức độ chiếu sáng,...

## II/ Datasets (nhóm tự tạo):

- 1) Tên dataset: Avenger.
- 2) Lý do tạo: phim có nhiều nhân vật, khoảng cách khuôn mặt trong các frame, hướng quay và độ chiếu sáng cũng rất đa dạng.
- 3) Cách tạo dataset: sử dụng Google images download [5][6] để crawl hình ảnh của một vài diễn viên trong phim. Sau đó chọn lọc lại những hình ảnh phù hợp.

#### 4) Thông tin dataset:

- Tổng số lượng ảnh: 1214 ảnh.
- Ảnh từng nhân vật:
- + Black widow: 79 ảnh.



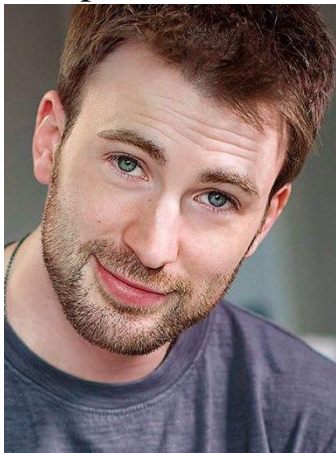
- + Black\_panther: 100 ảnh.



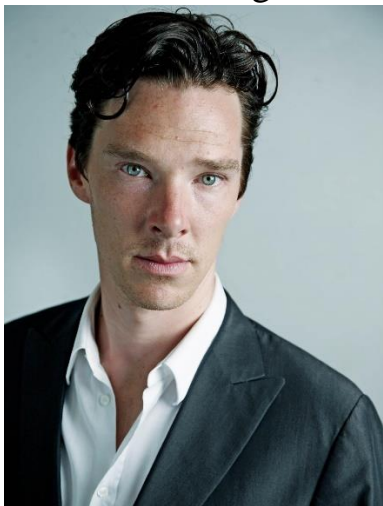
- + Bruce\_Banner: 127 ảnh.



+ Captain\_America: 94  nh.



+ Doctor\_Strange: 156  nh.



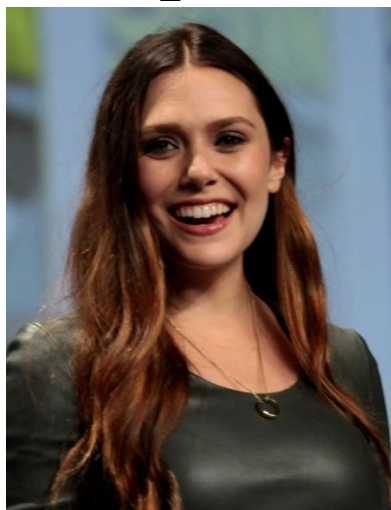
+ Iron\_man: 110  nh.



+ Loki: 127  nh.



+ Scarlett\_Witch: 142  nh.



+ Star\_lord: 134  nh.



+ Thor: 145 ảnh.



### **III/ Phương pháp đề xuất:**

#### **1. Encode các ảnh trong dataset:**

- Lưu các embedding của các ảnh vào một file pickle.

#### **2. Train SVM để phân loại ảnh nhân vật:**

- Train SVM dựa trên các thông tin có được trong file pickle trên.

#### **3. (Bước chính) Nhận dạng các khuôn mặt trong một frame ảnh bất kỳ:**

- Bước 1: Detect các khuôn mặt có trong ảnh input bằng MTCNN. [2]
- Bước 2: Từ các vùng khuôn mặt detect được, rút các embedding vector tương ứng ra bằng thư viện face\_recognition (thư viện được xây dựng theo bài báo FaceNet). [1]

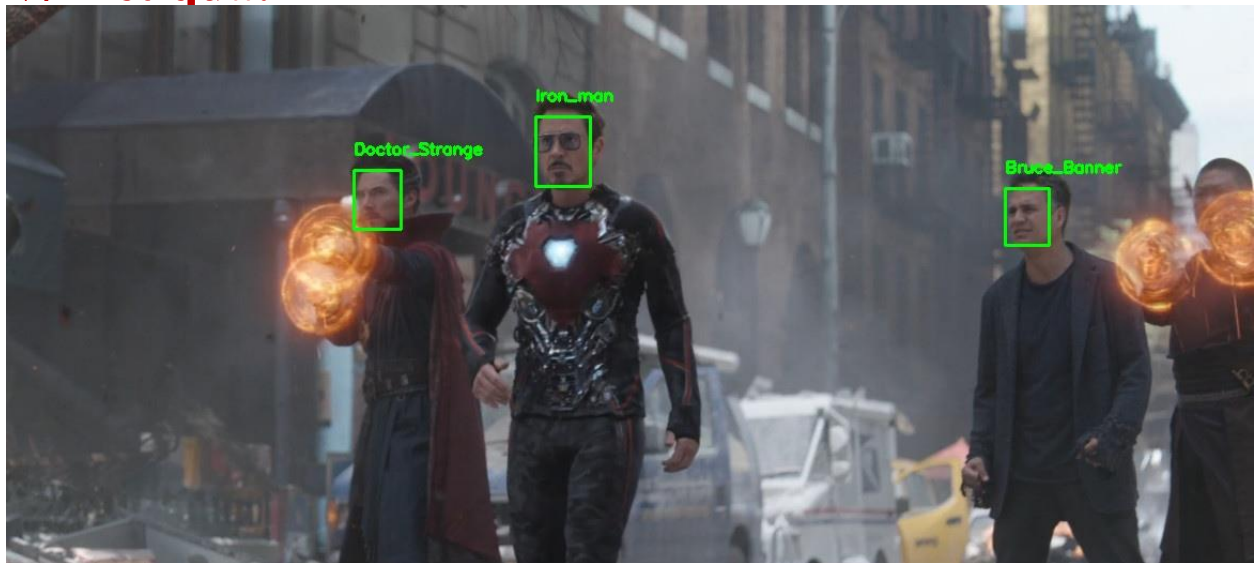
- Bước 3: Sử dụng SVM để phân loại các embedding vector trên vào lớp thích hợp. [4]
- Bước 4: Output thông tin tìm được lên ảnh.

#### **IV/ Tổ chức file:**

- **Thư mục Input**: chứa 25 ảnh input có kích thước khoảng 1100x500.
- **Thư mục Output**: chứa 25 ảnh output có đóng khung vùng khuôn mặt và tên nhân vật do mô hình nhận diện trả về.
- **Thư mục Pickle**: chứa các file pickle.
  - + avenger5.pickle: các embedding vector rút được từ dataset.
  - + label.pickle: chứa label encoder để hỗ trợ mô hình SVM.
  - + trainedSVM: mô hình SVM đã được train trên dataset.
- **Thư mục Src**: Chứa các file source.
  - + Thư mục ServiceMTCNN: chứa các tác vụ liên quan đến MTCNN [2].
  - + encode.py: để tạo file pickle chứa các embedding vector của dataset.
  - + SVM\_train.py: để train SVM.
  - + recognize\_faces.py: (file chính) để detect, nhận diện và xuất output.



## V/ Kết quả:



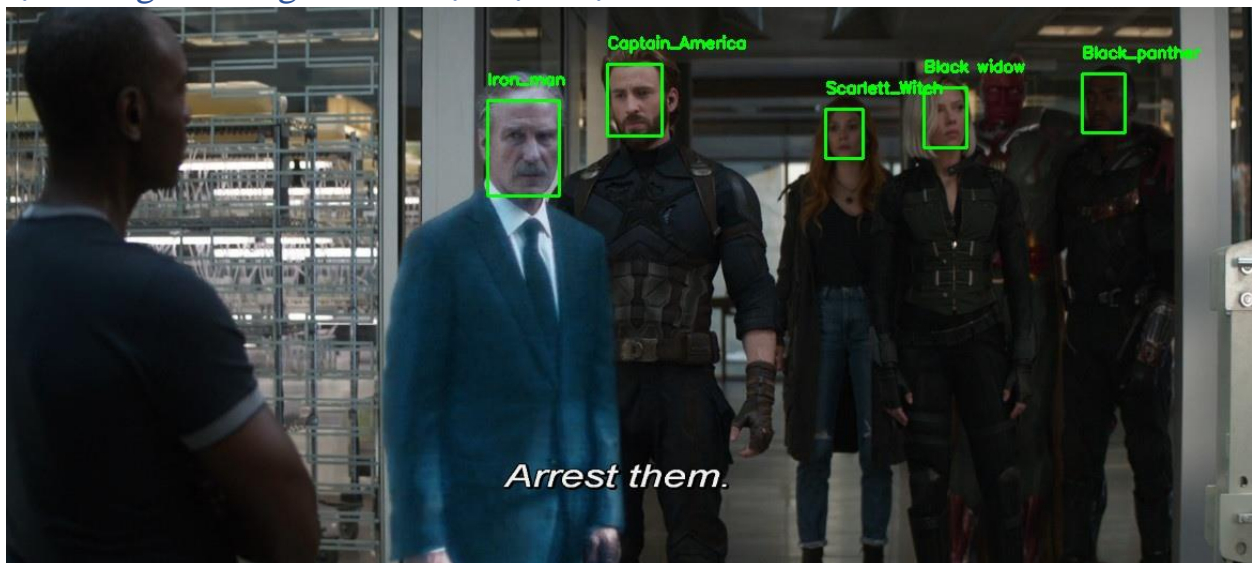
*Ảnh 1.* Nhân diện đúng tất cả nhân vật, 1 nhân vật không detect được mặt



*Ảnh 2.* Khi mặt khá xa và hướng xuống, nhận diện sai



**Ảnh 3.** Mặt trực diện, nhận diện đúng được Black panther, các nhân vật còn lại không có trong dataset bị nhận diện sai



**Ảnh 4.** Nhận diện đúng được Captain America, Black widow và Scarlett Witch. Hai nhân vật còn lại không có trong dataset bị nhận diện sai.





**Ảnh 5.** Nhận diện sai tất cả khuôn mặt khi chúng nằm quá xa.



**Ảnh 6.** Ảnh này cho kết quả rất tốt khi nhận diện được các khuôn mặt rất xa của Thor và Bruce Banner. Kết quả chỉ nhận diện sai một người là Iron man và detect nhầm một vùng không có khuôn mặt (tuy nhiên vùng ảnh bị detect sai này nhìn trông rất giống một khuôn mặt)

## **VI/ Đánh giá:**

- Kết quả detect bằng MTCNN rất tốt khi detect được các khuôn mặt rất nhỏ.
- Kết quả khi nhận dạng: Vẫn còn nhận diện sai khi mặt ở quá xa, bị mờ, chỉ thấy một phần khuôn mặt hay quá tối,...

## VII/ Hướng cải tiến:

- Sử dụng multi-frame super resolution để làm rõ lên những khuôn mặt ở quá xa. (nhóm có tìm hiểu nhưng chưa tìm được source code có sẵn thích hợp).
- Sử dụng các phương pháp Face super resolution như FSRNet,...

## VIII/ Hướng dẫn chạy demo:

```
python recognize_faces.py --image <path đến một ảnh input> --encodings <path đến file pickle chứa các embedding rút từ dataset>
```

- Chạy bằng command line.

## VIII/ Tham khảo:

- [1] [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)
- [2] <https://github.com/davidsandberg/facenet>
- [3] <https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>
- [4] <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC>
- [5] <https://forum.machinelearningcoban.com/t/crawl-nhanh-hinh-anh-tu-google-image-cho-bat-ki-chu-de-nao-ban-can/2510>
- [6] <https://github.com/hardikvasa/google-images-download>