

NHẬN DIỆN NGƯỜI ĐEO KHẨU TRANG

Bùi Xuân Thành- 14520830 - CS114.K21

Link Github:

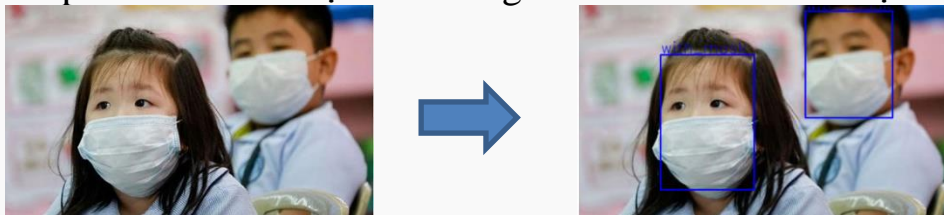
<https://github.com/buixuanthanh96/CS114.K21>

Tóm tắt

Tên đề tài: Nhận diện người đeo khẩu trang

Tóm tắt về đồ án:

- Input: Một bức ảnh về chưa một hay nhiều khuôn mặt.
- Output: Bức ảnh được bounding box cho các khuôn mặt và gắn nhãn cho nó.



Kết quả đạt được: Thực hiện test trên một số model(svm, knn và Random Forest) và model KNN cho kết quả tốt nhất khoảng 93%

Ảnh thành viên của nhóm:



Giới thiệu bài toán

- Hiện nay tình hình dịch bệnh covid-19 diễn ra rất phức tạp, chính phủ ban hành một số điều luật để phòng chống trong đó có bắt buộc người dân phải đeo khẩu trang khi ra nơi công cộng.
- Bài toán “nhận diện người đeo khẩu trang” giúp phát hiện khi nào có người không đeo khẩu trang để tiến hành xử lý.

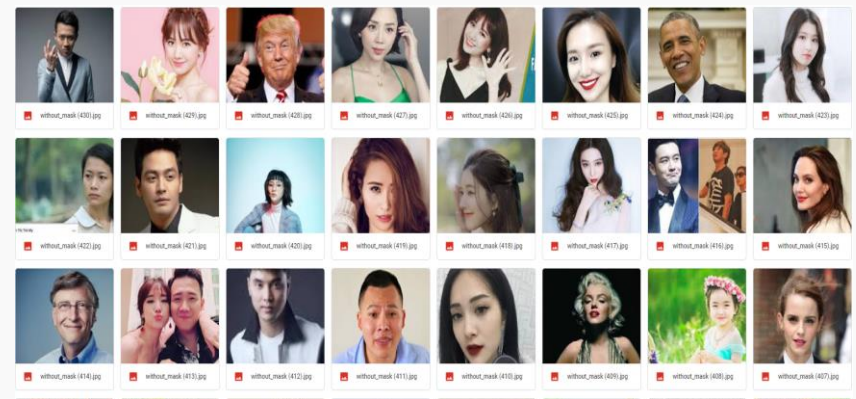
Cách giải quyết

Sử dụng thư viện face_detection để trích xuất khuôn mặt từ bức ảnh đầu vào. Sau đó sử dụng Histogram of Oriented Gradients để trích xuất khuôn mặt thành vector đặc trưng 5940 chiều. Sau đó áp dụng một số thuật toán Machine Learning để huấn luyện model. Chọn model tốt nhất để tiến hành chạy dự đoán

Mô tả dữ liệu

Cách thức xây dựng bộ dữ liệu: dữ liệu tự thu thập từ google image chia làm with_mask(đeo khẩu trang) và without_mask(không đeo khẩu trang)

<https://drive.google.com/drive/folders/1Kqmw2jTk1mJIAaXbg3qSNLYluFIBPYGR?usp=sharing>



Mô tả dữ liệu

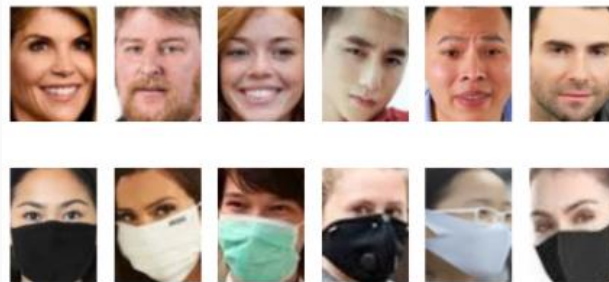
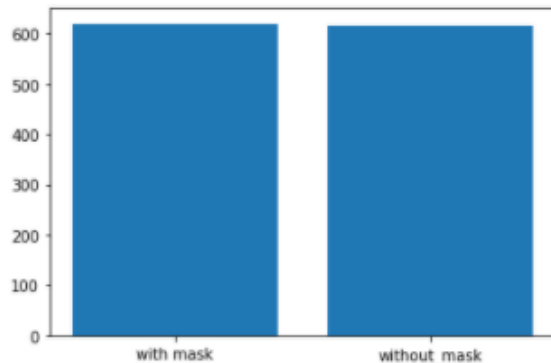
Dữ liệu sau khi tách face:

<https://drive.google.com/drive/folders/12tCRyodTKmrSiRPbMZPGfVLBPuzUnrm7?usp=sharing>

Số lượng ảnh sau khi xử lý cắt khuôn mặt:

- 620 ảnh người đeo khẩu trang
- 615 ảnh người không mang khẩu trang

with mask: 620
without mask: 615



Một số ảnh sau khi xử lý cắt khuôn mặt

Tiền xử lý dữ liệu và rút trích đặc trưng

Tiền xử lý dữ liệu:

- Dùng face_detection để xác định khuôn mặt sau đó cắt khuôn mặt trong ảnh.
- Chuyển ảnh về ảnh xám
- Resize ảnh thành (96,128)
- Dán nhãn dữ liệu

Rút trích đặc trưng: Dùng Histogram of Oriented Gradients trích xuất ảnh thành vector đặc trưng 5940 chiều (11x15x9x4)

Phân chia dataset và chọn Model

- Dataset được chia theo tỉ lệ train-test là: 80%-20%
- Chọn một số thuật toán máy học là: Linear SVM, K – Nearest Neighbors và Random Forest để giải quyết bài toán

Kết quả và đánh giá

kết quả đánh giá model SVM

	precision	recall	f1-score	support
with_mask	0.90	0.87	0.89	126
without_mask	0.87	0.90	0.89	121
accuracy			0.89	247
macro avg	0.89	0.89	0.89	247
weighted avg	0.89	0.89	0.89	247

kết quả đánh giá model Random Forest

	precision	recall	f1-score	support
with_mask	0.90	0.90	0.90	126
without_mask	0.89	0.90	0.90	121
accuracy			0.90	247
macro avg	0.90	0.90	0.90	247
weighted avg	0.90	0.90	0.90	247

kết quả đánh giá model KNN

	precision	recall	f1-score	support
with_mask	0.95	0.92	0.93	135
without_mask	0.91	0.94	0.92	112
accuracy			0.93	247
macro avg	0.93	0.93	0.93	247
weighted avg	0.93	0.93	0.93	247

Kết quả và đánh giá

	Accuracy
SVM	89%
KNN	93%
Random Forest	90%

Các model đều có độ chính xác khá cao, ta thấy model KNN cho kết quả đánh giá cao nhất là 93%.

=>Ta chọn SVM để làm model chạy dự đoán

Chạy dự đoán

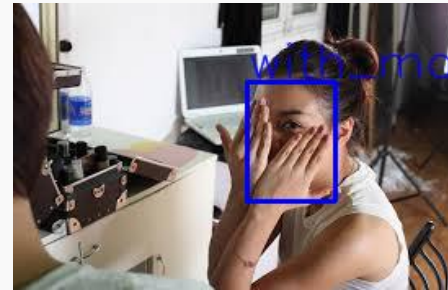
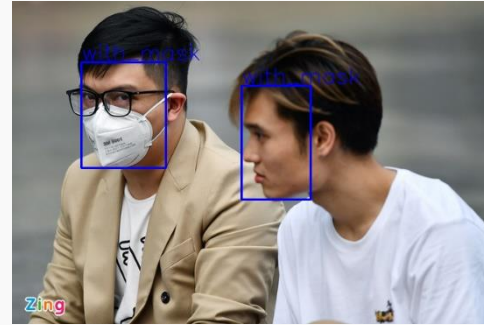
Một số kết quả chạy dự đoán trên model SVM

Chú thích: màu xanh dương là người có đeo khẩu trang và màu đỏ là không



Chạy dự đoán

Một số kết quả chạy dự đoán trên model SVM



Khó khăn và hướng phát triển

Những khó khăn :

- Model bị nhầm lẫn các khuôn mặt bị tay hay vật thể che.
- Nhận diện sai các khuôn mặt bị mất một phần mặt , các khuôn mặt bị nghiêng nhiều.

Hướng phát triển:

- Thêm các ảnh dự đoán bị sai và thu thập thêm nhiều dataset hơn để train model.
- Tìm hiểu một số phương pháp trích xuất đặc trưng khác.
- Thử nghiệm với một số deep learning model khác.

Tài liệu tham khảo

https://docs.opencv.org/2.4/modules/gpu/doc/object_detection.html

<https://viblo.asia/p/sift-scale-invariant-feature-transform-huan-luyen-mo-hinh-cho-cac-bai-toan-phan-loai-924lJqJaZPM>

<https://machinelearningcoban.com/2017/01/08/knn/>

<https://viblo.asia/p/tim-hieu-ve-phuong-phap-mo-ta-dac-trung-hog-histogram-of-oriented-gradients-V3m5WAwxZO7>

<https://viblo.asia/p/phan-lop-bang-random-forests-trong-python-djeZ1D2QKWz>

<https://machinelearningcoban.com/2017/01/08/knn/#:~:text=M%E1%BB%99t%20c%C3%A1ch%20ng%E1%BA%AFn%20g%E1%BB%8Dn%2C%20KNN,g%E1%BA%A7n%20nh%E1%BA%A5t%20n%C3%A0y%20l%C3%A0%20nh%E1%BB%85u>