

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



BÁO CÁO MÔN HỌC

CS231 & CS406

ĐỀ TÀI: Phát hiện và làm đẹp khuôn mặt người Việt Nam

Giáo viên hướng dẫn: TS.Mai Tiến Dũng

Sinh viên thực hiện: Đào Văn Tài – 19522148

Ngô Gia Kiệt – 19521725

Nguyễn Thành Trọng – 19522410 (CS231)

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

Mục lục

1. Giới thiệu.....	1
1.1. Lý do chọn đề tài.....	1
1.2. Mô tả.....	1
2. Bộ dữ liệu.....	2
3. Phát hiện khuôn mặt và phân loại.....	3
3.1. Phát hiện khuôn mặt.....	3
3.2. Tiền xử lý bằng blob	5
3.3. Trích xuất đặc trưng ảnh	6
3.4. Phân loại khuôn mặt người Việt.....	7
3.4.1. Các mô hình phân loại.....	7
3.4.1.1. Support Vector Machine (SVM).....	7
3.4.1.2. K-Nearest Neighbors (K-NN).....	8
3.4.1.3. Mạng nơ-ron phân loại.....	9
3.4.2. Kết quả.....	9
3.4.2.1. Chưa tiền xử lý blob.....	9
3.4.2.2. Có tiền xử lý blob	10
3.5. Nhận xét: Một số trường hợp sai.....	11
4. Làm đẹp trên khuôn mặt.....	12
4.1. Phát hiện vùng da.....	12
4.1.1. Phương pháp Sobel edge detection.....	12
4.1.2. Phương pháp lấy ngưỡng trong không gian màu HSV	14
4.1.3. So sánh.....	17
4.2. Làm mịn vùng da.....	19
4.2.1. Average Blur.....	19
4.2.2. Gaussian Blur.....	20
4.2.3. Bilateral Blur.....	21
4.2.4. So sánh và nhận xét.....	22
4.3. Trang điểm khuôn mặt.....	23
5. Kết luận và hướng phát triển.....	26
6. Nguồn tham khảo.....	27
7. Phân công công việc.....	27

1. Giới thiệu.

1.1. Lý do chọn đề tài.

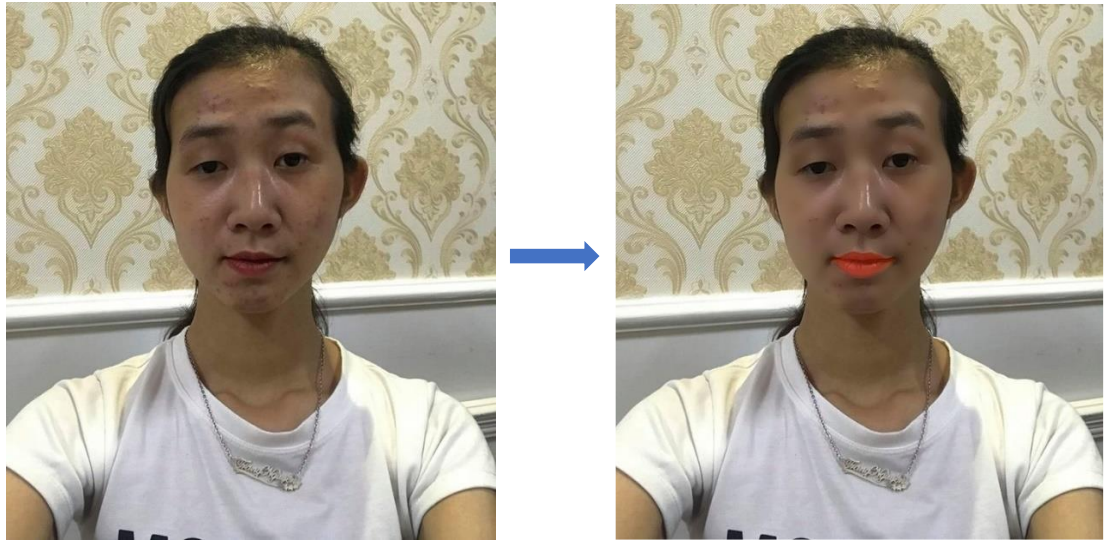
Trong thời kỳ dịch bệnh Covid-19 hiện nay, học và làm việc trực tuyến không còn là việc quá xa lạ với mọi người. Việc chuyển đổi từ học trực tiếp sang học trực tuyến quá nhanh chóng đã gây ra rất nhiều khó khăn cho cả học sinh, sinh viên lẫn đội ngũ giáo viên, giảng viên. Có rất nhiều học sinh, sinh viên không mở webcam trong quá trình học dẫn tới việc giao tiếp, trao đổi giữa mọi người gặp nhiều khó khăn. Nguyên nhân một phần lớn chính là do không họ tự tin vì trên khuôn mặt có những khuyết điểm như: mụn, vết thâm,...

Vậy nên chúng tôi thực hiện làm mờ các khuyết điểm và trang điểm trên khuôn mặt để giúp mọi người có thể tự tin hơn trước webcam. Giúp cho việc giao tiếp qua màn hình nhỏ tự tin hơn, tăng thêm sự tương tác giữa những người tham gia với nhau, đem lại hiệu quả công việc cao.

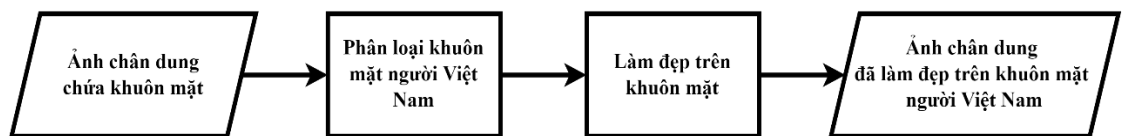
1.2. Mô tả.

Với mỗi vùng khác nhau đều có đặt trưng khác nhau. Người Đông Nam Á- Đông Á có màu da vàng. Người châu Âu, châu Mỹ có màu da trắng. Người châu Phi có màu da đen. Việc áp dụng một thông số lên toàn bộ người trên mọi châu lục thì rất khó đạt được kết quả tốt nhất. Vậy nên ở báo cáo chúng tôi giới hạn chỉ áp dụng trên khuôn mặt người Việt Nam.

Đầu vào của bài toán là một bức hình chân dung chứa khuôn mặt trong đó khuôn mặt chiếm phần lớn trong khung hình, để đạt được kết quả tốt nhất. Đầu ra của bài toán là ảnh chân dung mà người Việt có trong ảnh đã được làm đẹp khuôn mặt.



Hình 1: Ảnh đầu vào và đầu ra của bài toán



Hình 2: Sơ đồ khối các bước thực hiện.

2. Bộ dữ liệu.

2.1. Thu thập dữ liệu.

Để phân loại người Việt Nam, chúng tôi sử dụng bộ dữ liệu gồm 2 lớp bao gồm: Ảnh người Việt Nam, ảnh người nước ngoài.

Với bộ dữ liệu người Việt Nam, chúng tôi thực hiện thu thập danh sách tên những người Việt có năm sinh từ năm 1890 có trên Wikipedia. Kết quả thu được là danh sách hơn 1300 tên người Việt sau đó sử dụng danh sách tên để thu thập ảnh những người có trong danh sách đó từ Google Images.

Với bộ dữ liệu người nước ngoài, chúng tôi sử dụng bộ dữ liệu WFLW.

Từ hai bộ dữ liệu đề cập bên trên chúng tôi sử dụng mạng MTCNN (Việc sử dụng MTCNN sẽ được đề cập ở phần bên dưới) để có thể phát hiện, trích xuất những khuôn mặt có trong những bức hình đó. Cuối cùng là loại bỏ những khuôn mặt có kích thước quá nhỏ, hoặc các trường hợp phát hiện sai.

2.2. Kết quả.

Số người Việt trong khoảng trên 2000 người là các ca sĩ, vận động viên, người nổi tiếng, chính trị gia,... Việc chênh lệch giữa danh sách tên và số lượng người là do với một cái tên thì kết quả tìm kiếm hình ảnh ra nhiều người hoặc trong một bức ảnh có nhiều người. Bộ dữ liệu này sẽ đại diện cho các đặc trưng trên khuôn mặt của người Việt

Bộ dữ liệu sau khi thu thập bao gồm: 10000 ảnh khuôn mặt người Việt, 9589 ảnh khuôn mặt người nước ngoài. Ta có thể thấy

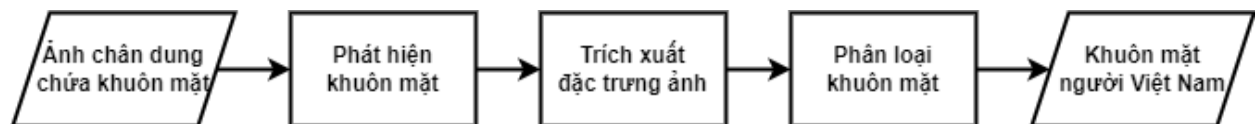


Hình 3: Ảnh khuôn mặt người Việt Nam



Hình 4: Ảnh khuôn mặt người nước ngoài

3. Phát hiện khuôn mặt và phân loại.



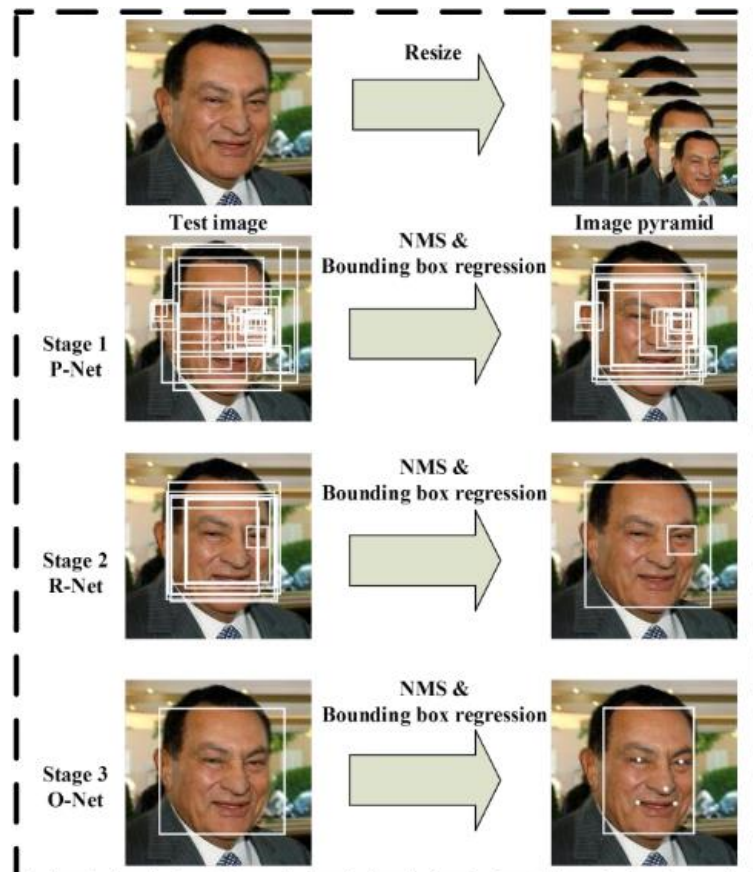
Hình 5: Sơ đồ khối thức hiện bài toán phát hiện và phân loại khuôn mặt

3.1. Phát hiện khuôn mặt.

MTCNN bao gồm 3 mạng nơ-ron dành cho việc phát hiện khuôn mặt. Đầu tiên nó thay đổi ảnh đầu vào thành nhiều kích thước khác nhau trở thành một kim tự tháp ảnh (image pyramid) giúp mô hình phát hiện được với các kích thước khác nhau:

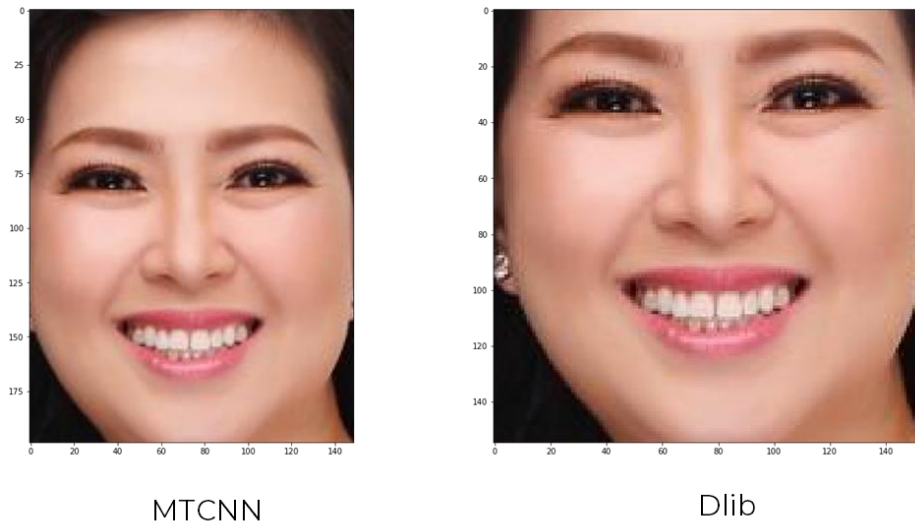
- Mạng thứ nhất là Proposal Network (P-Net) có nhiệm vụ đề xuất các cửa sổ (windows) tiềm năng và bounding box regression vectors của chúng.
- Mạng thứ hai là Refine Network (R-Net) tất cả các bounding box ở P-Net sẽ là đầu vào cho R-Net. R-Net có nhiệm vụ loại bỏ một lượng lớn các cửa sổ sai, hiệu chuẩn lại bounding box regression vectors.

- Mạng thứ ba là Output Network (O-Net) đầu ra của R-Net là đầu vào của O-Net. O-Net có nhiệm vụ tương tự như R-Net nhưng tập trung mô tả chi tiết khuôn mặt hơn và tìm kiếm 5 điểm trên khuôn mặt bao gồm: mắt trái, mắt phải, chóp mũi, khóe môi trái, khóe môi phải.
- Cả ba mạng ở trên đều sử dụng thuật toán Non-Maximum Suppression giành cho việc loại bỏ đi các bounding box dư thừa của cùng một đối tượng trong ảnh.



Hình 6: Mô tả nhiệm vụ các mạng của MTCNN

MTCNN và thư viện Dlib đều được sử dụng rộng rãi cho việc phát hiện khuôn mặt. Tuy nhiên bounding box kết quả mà MTCNN phát hiện được chứa bao gồm cả phần trán. Thay vì không chứa phần trán như của Dlib. Việc có thêm phần trán sẽ chứa nhiều thông tin hơn cho việc phân loại sau này nên chúng tôi lựa chọn MTCNN.



Hình 7: Kết quả phát hiện khuôn mặt từ MTCNN và Dlib

3.2. Tiền xử lý bằng blob

Blob là một tập hợp các hình ảnh có cùng kích thước không gian (tức là chiều rộng và chiều cao), cùng độ sâu (số kênh), tất cả đều được xử lý trước theo cùng một cách. Mục đích của việc sử dụng ảnh blob là để giảm nhiễu cho ảnh do bị chiếu sáng. Chúng tôi sử dụng hàm:

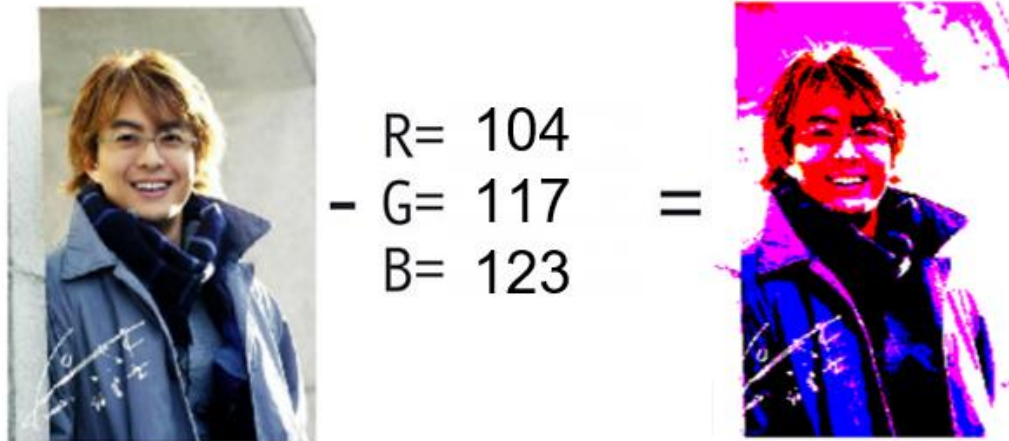
- `cv2.dnn.blobFromImage`, Hàm trả về một blob là hình ảnh đầu vào của chúng ta sau khi trừ trung bình, chuẩn hóa và hoán đổi kênh. Hàm này thực hiện gồm 2 chức năng chính là Mean subtraction và Scaling
- Mean subtraction được dùng để chống lại các thay đổi về độ sáng trong ảnh
- Sử dụng các biến giá trị pixel trung bình của các kênh màu R, G và B lần lượt là: 104, 117 và 123. Đây là các giá trị pixel trung bình thường được sử dụng trong các model pretrain ImageNet

Sau đó, trừ giá trị pixel trung bình μ ứng với mỗi kênh màu R, G và B

$$R = R - \mu_R$$

$$G = G - \mu_G$$

$$B = B - \mu_B$$



Hình 8: Ví dụ Mean subtraction

Sau khi Mean subtraction thì ta sẽ tiếp tục chuẩn hóa với Scaling

Tham số scaleFactor σ là độ lệch của tập dữ liệu hoặc có thể được cài đặt thủ công. Mặc định thì tham số $\sigma = 1$, tương ứng với việc sẽ không thực hiện Scaling sau khi meansubtraction

3.3. Trích xuất đặc trưng ảnh

Trích xuất đặc trưng là một phần của quy trình giảm kích thước, trong đó, tập hợp dữ liệu thô ban đầu được chia và giảm thành các nhóm để quản lý hơn. Vì vậy, khi bạn muốn xử lý sẽ dễ dàng hơn. Đặc điểm quan trọng nhất của các tập dữ liệu lớn này là chúng có một số lượng lớn các biến. Các biến này yêu cầu nhiều tài nguyên máy tính để xử lý chúng.

Vì vậy, việc trích xuất giúp có được tính năng tốt nhất từ các tập dữ liệu lớn đó bằng cách chọn và kết hợp các biến thành các tính năng, do đó, giảm lượng dữ liệu một cách hiệu quả. Các tính năng này rất dễ xử lý, nhưng vẫn có thể mô tả tập dữ liệu thực tế với độ chính xác và độc đáo.

Để xuất vector đặc trưng nhóm sử dụng 3 pre-trained model của mạng nơ-ron OpenFace để trích xuất các đặc trưng của khuôn mặt thành những vector 128 chiều.

- nn4.v2
- nn4.small1.v1
- nn4.small2.v1

Các pre-trained model bên trên được đào tạo theo nhiều cách khác nhau với các bộ dữ liệu khác nhau. Các mô hình được đào tạo trước được tạo phiên bản và nên được phát hành cùng với định nghĩa mô hình tương ứng.

Model	Accuracy
nn4.small2.v1 (Default)	0.9292 ± 0.0134
nn4.small1.v1	0.9210 ± 0.0160
nn4.v2	0.9157 ± 0.0152

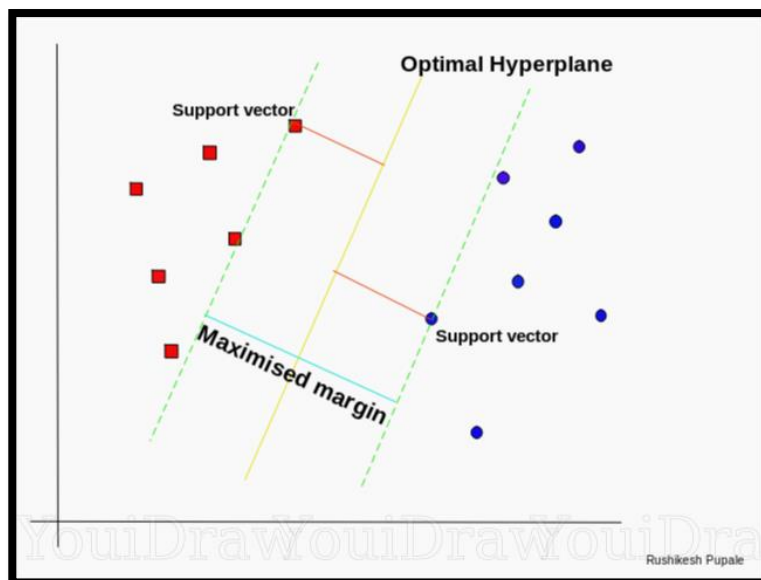
Hình 9: Accuracy của các pre-trained model

3.4. Phân loại khuôn mặt người Việt.

3.4.1. Các mô hình phân loại.

3.4.1.1. Support Vector Machine (SVM).

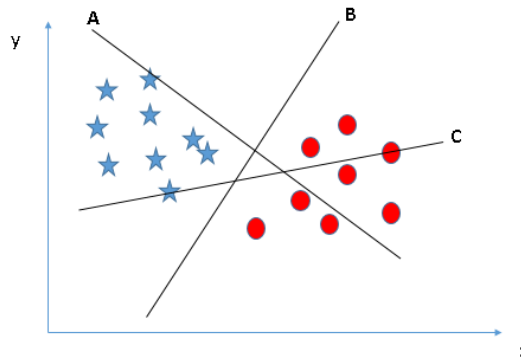
SVM là một thuật toán hỗ trợ chủ yếu bài toán phân loại. Nó hoạt động dựa trên nguyên lý tìm một siêu mặt phẳng sao cho nó chia 2 lớp trở thành tách biệt và tối ưu khoảng cách từ mặt phẳng đến 2 lớp đang xét.



Hình 10: Mô phỏng thuật toán SVM

Theo như thuật toán SVM sẽ tìm các điểm nằm gần đường thẳng nhất, những điểm đấy được gọi là support vector. Tiếp theo là tính khoảng cách từ các support vector đến

đường thẳng, khoảng cách ấy được gọi là margin, Mục đích của thuật toán là tìm margin lớn nhất. Mặt phẳng mà có margin lớn nhất là siêu mặt phẳng cần tìm.

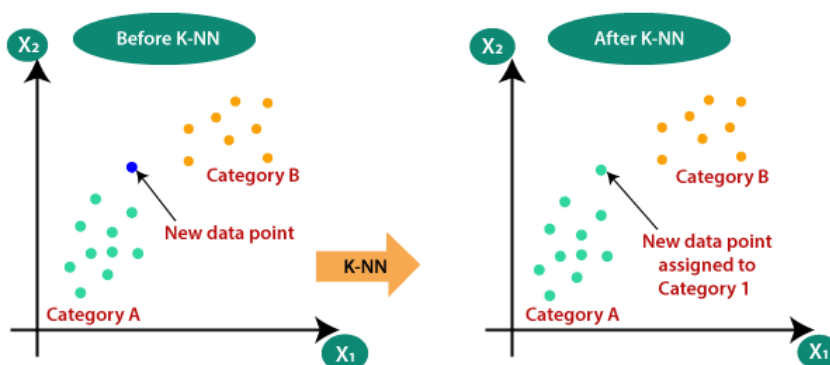


Hình 11: Ví dụ về SVM

Ví dụ những ngôi sao là người Việt còn hình tròn là người nước ngoài thì trong trường hợp này đường thẳng phù hợp nhất là đường thẳng B.

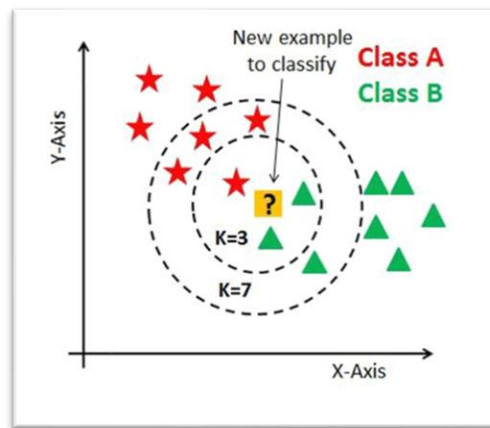
3.4.1.2. K-Nearest Neighbors (K-NN).

Thuật toán K-NN lưu trữ tất cả dữ liệu có sẵn và phân loại một điểm dữ liệu mới dựa trên sự tương đồng. K-NN là một thuật toán xuất phát từ thực tế cuộc sống. Mọi người có xu hướng bị ảnh hưởng bởi những người xung quanh họ. Hành vi của chúng ta được hướng dẫn bởi những người bạn mà chúng ta đã lớn lên cùng. Cha mẹ chúng ta cũng hình thành nhân cách của chúng ta theo một số cách. Nếu bạn lớn lên với những người yêu thể thao, rất có thể bạn sẽ yêu thể thao. Tất nhiên có sự ngoại lệ. K-NN hoạt động tương tự.



Hình 12: Giải thích thuật toán KNN

Điều này có nghĩa là khi dữ liệu mới xuất hiện thì nó có thể dễ dàng được phân loại thành một loại tốt bằng cách sử dụng thuật toán K-NN



Hình 13: Minh họa tham số k

Giá trị của một điểm dữ liệu được xác định bởi các điểm dữ liệu xung quanh nó. Nếu bạn luôn đi chơi với một nhóm 3 người, mỗi người trong nhóm có ảnh hưởng đến hành vi của bạn và bạn sẽ nhận được điểm trung bình là 3. Đó là K-NN với $k = 3$. Nếu bạn luôn đi chơi với một nhóm 7 người, mỗi người trong nhóm có ảnh hưởng đến hành vi của bạn và bạn sẽ nhận được điểm trung bình là 7. Đó là K-NN với $k = 7$

3.4.1.3. Mạng nơ-ron phân loại.

Sau nhiều lần chỉnh sửa thì chúng tôi tìm được mô hình mạng nơ-ron phân loại đơn giản gồm 2 lớp ẩn (hidden layer) có kiến trúc như hình bên dưới cho kết quả tốt nhất. Nếu đầu vào là vector embedding được trích xuất từ pre-trained model của Openface, thì $input_dim=128$. Nếu đầu vào là đặc trưng histogram thì $input_dim=256$. Sử dụng hàm kích hoạt là ReLU ở hai lớp ẩn giúp mô hình hội tụ nhanh hơn.

```
def create_model():
    model = Sequential()
    model.add(Dense(10, input_dim=256, activation='relu'))
    model.add(Dense(5, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

Hình 14: Kiến trúc mạng tự tạo

3.4.2. Kết quả

3.4.2.1. Chưa tiền xử lý blob

Kết quả	SVM	KNN	CUSTOM NETWORK
Histogram	60.64%	58.58%	61.74%
Pre-Trained: nn4.v2	60.67%	56.76%	60.44%
Pre-Train: nn4.small1.v1	63.43%	58.63%	61.84%
Pre-Trained: nn4.small2.v1	64.96%	60.69%	64.45%

Bảng 1: Kết quả chưa tiền xử lý

Nhận xét:

Sau khi train ta được kết quả như hình. Histogram cho kết quả không tốt do dữ liệu khá phức tạp, nó phụ thuộc khá nhiều vào giá trị cực đại và cực tiểu. Trong 3 pretrain model thì small2.v1 có kết quả tốt nhất ở 3 model SVM, KNN và Custom Network. Do small2.v1 là model mặc định và có điểm pre-trained cao nhất trong 3 model

Vì vậy nhóm tiếp tục xử lý ảnh và huấn luyện cho mô hình này.

3.4.2.2. Có tiền xử lý blob

Kết quả	SVM	KNN	CUSTOM NETWORK
Pre-Trained: nn4.small2.v1	81%	79%	82%

Bảng 2: Kết quả đã tiền xử lý

Nhận xét:

Sau khi tiến hành tiền xử lý blob thì ta thấy kết quả đánh giá cao hơn ở cả 3 model trên phương pháp lấy vector embedding pre-trained nn4.small2.v1. Cao nhất là Custom Network với 82%.

3.5. Nhận xét: Một số trường hợp sai

Những trường hợp người nước ngoài có nét giống người Việt và ngược lại sẽ cho kết quả sai.



Hình 15: Ví dụ

Những trường hợp màu ảnh bị ám vàng do người Việt chúng ta có màu da vàng nên model sẽ nhận diện sai.



Hình 16: Ví dụ

Những trường hợp khuôn mặt bị che bởi các phụ kiện kính như kính, khẩu trang,... thì model không thể phân biệt được người Việt.

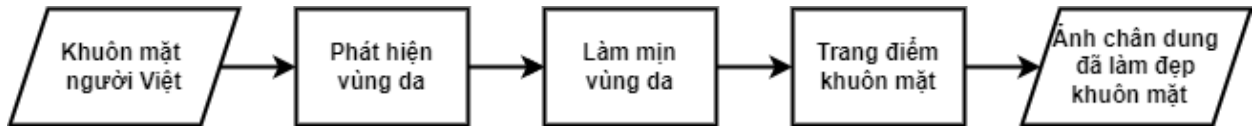


Hình 17: Ví dụ

Ngoài ra do số lượng dữ liệu lớn nên không tránh khỏi việc lọc dữ liệu còn sai sót khi trong dữ liệu còn các ảnh tạp không phải hình khuôn mặt

4. Làm đẹp trên khuôn mặt.

Như đã đề cập ở phần giới thiệu, để cho mọi người tự tin khi sử dụng webcam giao tiếp, chúng tôi tiến hành thực hiện bước trang điểm, làm đẹp khuôn mặt cho người dùng. Dưới đây là sơ đồ các thực hiện việc làm đẹp.



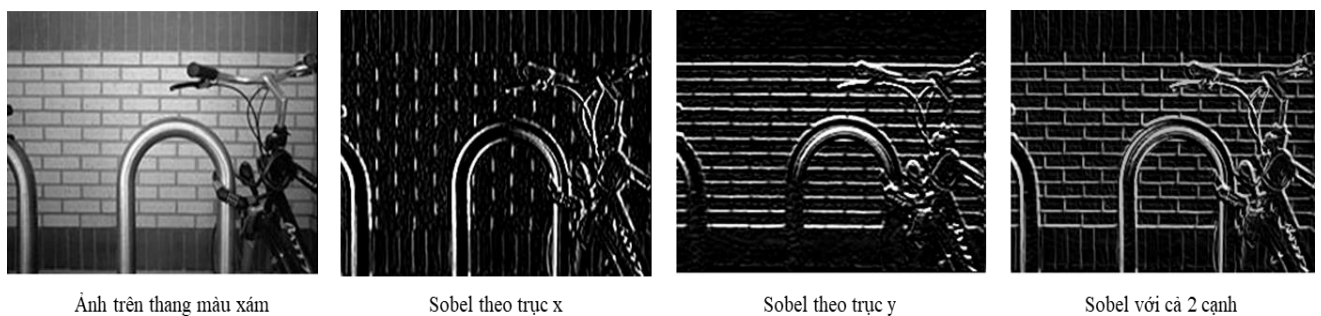
Hình 18: Sơ đồ thực hiện bài toán trang điểm.

4.1. Phát hiện vùng da

Ở bài báo cáo này, chúng tôi thực hiện hai phương pháp phát hiện da là: Sobel edge detection và chọn ngưỡng trong không gian màu HSV. Hai phương pháp này có hướng tiếp cận và cách sử dụng tương đối dễ cho người mới bắt đầu. Sau đây, chúng tôi sẽ trình bày về hai phương pháp này, so sánh và đánh giá một cách khách quan về hoạt động của chúng.

4.1.1. Phương pháp Sobel edge detection.

Sobel edge detection là một thuật toán được dùng rộng rãi trong các bài toán phát hiện cạnh của vật thể. Các cạnh được đánh dấu theo sự thay đổi đột ngột cường độ của điểm ảnh



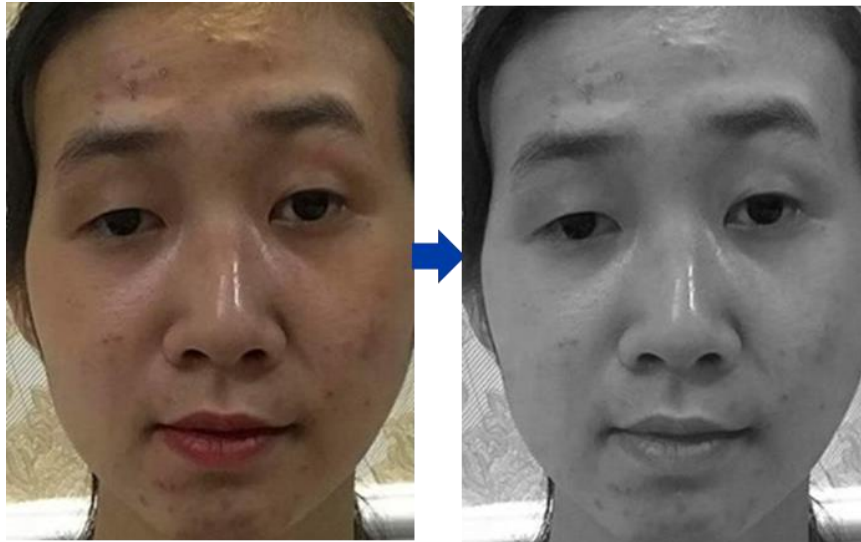
Hình 19: Một số ví dụ Sobel edge detection

Trong bài toán phát hiện da, Sobel edge detection sẽ coi toàn bộ các vùng đồng màu đều là vùng da, kể cả vùng gồm phần tóc đen và vùng nền phía sau khuôn mặt mà ta đang

xét. Mục đích là sẽ làm mịn các phần đồng màu và giữ được nguyên được các đường nét trên khuôn mặt

Để sử dụng phương pháp này, ta cần phải thực hiện các bước sau:

- Chuyển ảnh gốc về thang màu xám để thực hiện bước tiếp theo.



Hình 20: Ảnh thang màu xám

- Áp dụng Sobel edge detection theo 2 trục x, y để xác định các cạnh trên khuôn mặt.
 - Lấy Threshold sao cho vùng da được xác định một cách rõ ràng nhất. (Mặc định màu trắng là vùng da và ngược lại)

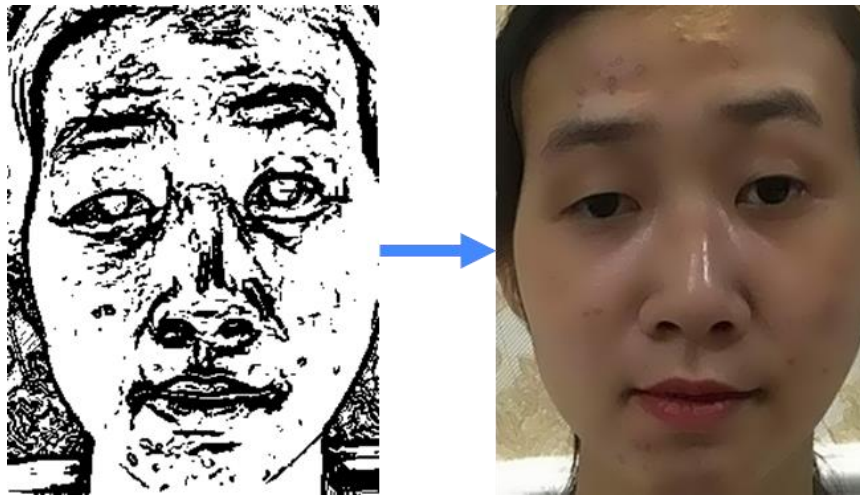
$$Skinmap[i, j] = \begin{cases} 1 & (if \ x[i, j] < thresholdX \ \& \ y[i, j] < thresholdY \ \& \ (x[i, j]^2 + y[i, j]^2) < (thresholdX^2 + thresholdY^2)) \\ 0 & otherwise \end{cases}$$

Trong đó:

- o thresholdX là threshold theo trục X mặc định là 40
- o thresholdY là threshold theo trục Y mặc định là 40
- o X[i,j] là giá trị sobel theo trục X tại điểm i,j của ảnh
- o Y[i,j] là giá trị sobel theo trục Y tại điểm i,j của ảnh

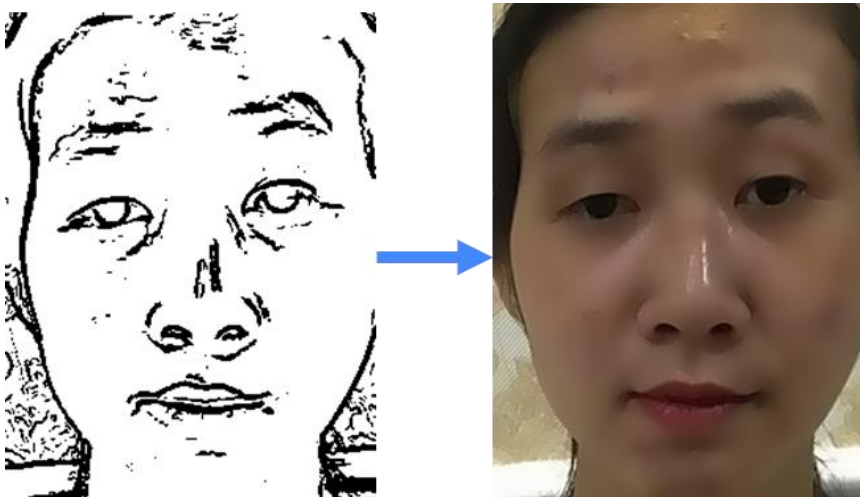
Threshold là tham số ảnh hưởng tới độ chi tiết của vùng da. Cụ thể hơn, Threshold càng lớn thì các cạnh của nốt mụn, nếp nhăn sẽ được lờ đi, tuy nhiên các chi tiết của hoa văn phía sau nền sẽ không còn nguyên vẹn như ban đầu.

Thực nghiệm ảnh hưởng Threshold:



Hình 21: Threshold = 20

- Ở threshold = 20, các chi tiết hoa văn phía sau hầu như nguyên vẹn, tuy nhiên những viên mụn vẫn còn nhiều trên vùng trán của khuôn mặt do việc lọc nhiễu còn tương đối thấp. Điều này dẫn đến da mặt chưa được tốt (không tự nhiên) sau khi làm mịn.

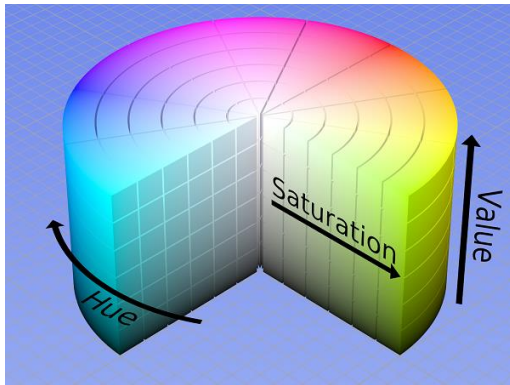


Hình 22: Threshold = 40

- Ở threshold = 40, các chi tiết hoa văn ở phía sau đã không còn nguyên vẹn, nó đã bị làm mờ đi. Bên cạnh đó, việc lọc nhiễu trên vùng trán là rất tốt → các viên mụn hầu như không còn. Đây là sự đánh đổi của phương pháp Sobel edge detection.

4.1.2. Phương pháp lấy ngưỡng trong không gian màu HSV

Không gian màu HSV (HSB) là một cách tự nhiên hơn để mô tả màu sắc, dựa trên 3 số liệu:



H: (Hue) Vùng màu,

S: (Saturation) Độ bão hòa màu

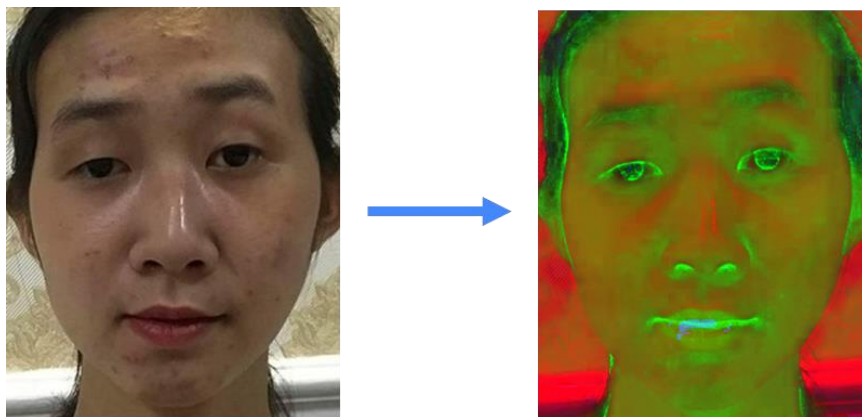
B (hay V): (Bright hay Value) Độ sáng

Hình 23: Mô phỏng HSV

Bên cạnh đó, ứng dụng điển hình nhất của hệ màu này là trong việc lọc màu. Vì vậy, đây là một điểm nổi bật để sử dụng trong việc phát hiện da trong khuôn mặt.

Để thực hiện phương pháp lấy ngưỡng trong không gian màu HSV, chúng tôi thực hiện các bước sau:

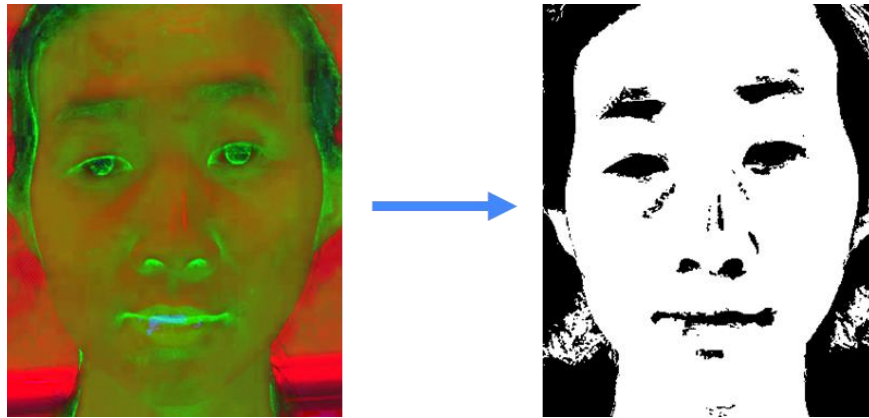
- Chuyển ảnh gốc sang ảnh trong hệ màu HSV



Hình 24: Chuyển ảnh sang hệ màu HSV

Kết quả cho thấy vùng da trên khuôn mặt của người này được tách biệt rõ so với các bộ phận mắt, mũi, miệng,...Điều đó sẽ góp phần tối ưu cho các bước thực hiện xử lý tiếp theo.

- Lấy ngưỡng của ảnh trong không gian màu vừa chuyển để xác định vùng da cần tìm.



Hình 25: Sau khi lấy ngưỡng

Để cho ra được một kết quả xác định vùng da tốt, nó phụ thuộc vào việc ta chọn ngưỡng phải tốt. Công việc này rất quan trọng trong bài toán phát hiện da nói riêng và cả bài toán trang điểm nói chung.

- Nếu kết quả xác định da không tốt, nghĩa là vùng da không được tìm thấy hoặc kết quả tìm không chính xác, sẽ cho ra kết quả làm mịn bị sai vùng, làm sai bài toán cần đạt được.
- Bên cạnh đó, việc giữ lại cạnh của các bộ phận trên khuôn mặt như mắt, mũi, miệng, cằm,... là một việc tương đối quan trọng. Việc giữ lại này sẽ làm cho khuôn mặt sau khi làm mịn đạt được một kết quả tự nhiên.

Để thực hiện việc chọn ngưỡng một cách hiệu quả và đáp ứng các nhu cầu nói trên, chúng tôi sử dụng hàm `inRange()` trong thư viện `Opencv` hỗ trợ với ràng buộc: không dùng kênh H (Hue) với lý do khi tinh chỉnh kênh này, sẽ ảnh hưởng đến việc xác định vùng da bị sai.



(0.0, 60.0, 60.0)
(100.0, 255.0, 255.0)



(0.0, 75.0, 75.0)
(100.0, 255.0, 255.0)



(0.0, 90.0, 90.0)
(100.0, 255.0, 255.0)

Hình 26: Ví dụ chọn ngưỡng.

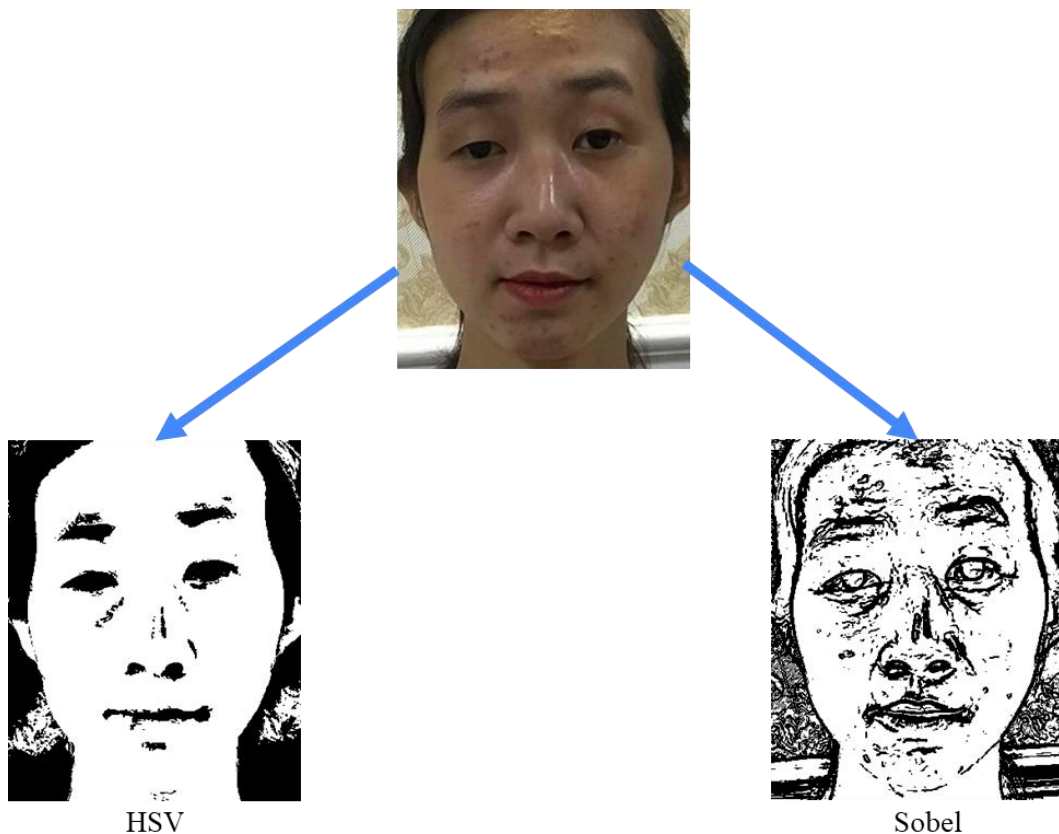
Nhận xét quá trình chọn ngưỡng:

- Ở hình bên trái, các cạnh trên khuôn mặt bị mất nhiều nhiều, làm cho khuôn mặt sau khi làm mịn bị mất tự nhiên .
- Hình bên phải, cạnh và nhiễu trên mặt con nhiều, sẽ làm ảnh hưởng đến việc phát hiện da sai và ảnh sau khi làm mịn sẽ bị không tốt.
- Ngược lại hai trường hợp trên thì ảnh ở giữa cho tỷ lệ cân bằng nhất về việc phát hiện vùng da đồng thời giữ lại các cạnh trên khuôn mặt → Chúng tôi chọn thông số này làm mặc định.

4.1.3. So sánh.

Ở phần này, để đánh giá một cách khách quan ta sẽ xét ảnh trong ba trường hợp đủ sáng, thiếu sáng và dư sáng để đánh giá hoạt động ở hai phương pháp này.

Trường hợp ảnh đủ sáng:



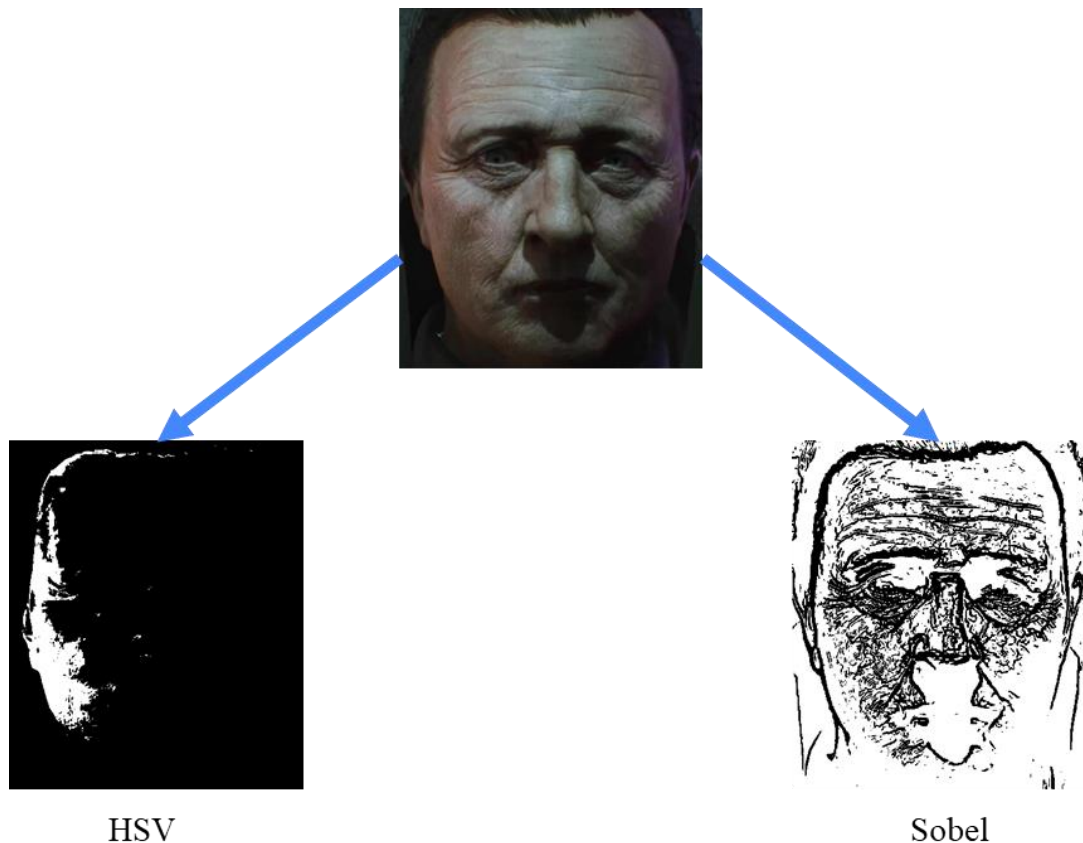
Hình 27: Trường hợp đủ sáng

- Trong trường hợp này, ta có thể quan sát thấy ở phương pháp Sobel edge detection, việc giữ lại các chi tiết trên khuôn mặt và ở phía sau rất tốt. Đó cũng là

nhược điểm khi những vùng viền mụn hay nếp nhăn trên khuôn mặt cũng xuất hiện tương đối.

- Ở phương pháp lấy ngưỡng trong không gian màu HSV hoạt động khá tốt trong trường hợp này, vùng da được xác định rõ ràng và loại được nhiều đen trên khuôn mặt khá tốt.

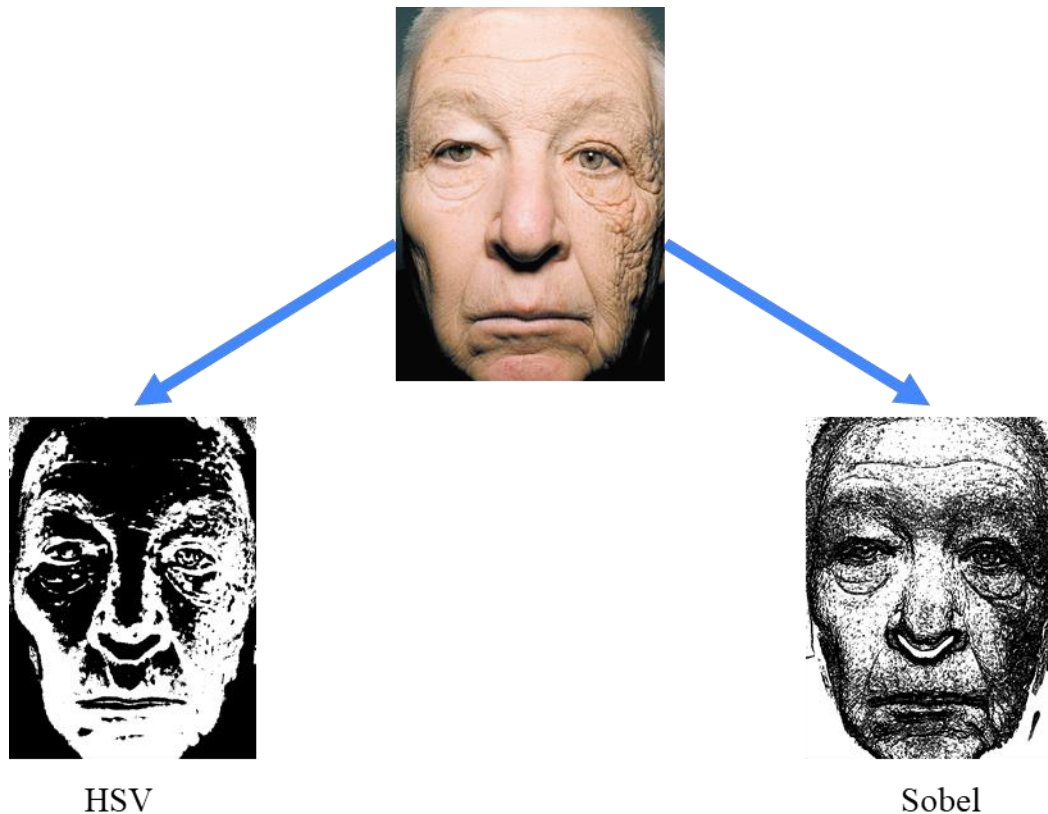
Trường hợp ảnh thiếu sáng:



Hình 28: Trường hợp thiếu sáng

- Tuy nhiên ở trong trường hợp thiếu sáng, kết quả hoàn toàn nghiêng về phương pháp Sobel edge detection.
- Phương pháp chọn ngưỡng trong không gian màu HSV bị ảnh hưởng lớn bởi việc thiếu sáng. Bằng chứng là nó chỉ có thể phát hiện được phần da bên trái của khuôn mặt là do phần da có ánh sáng tốt hơn so với các vùng còn lại trên khuôn mặt.
- Về phương pháp Sobel edge detection, việc phát hiện cạnh để quyết định vùng da là khá tốt → Việc tìm kiếm vùng da không bị ảnh hưởng nhiều do sự thiếu sáng.

Trường hợp ảnh dư sáng:



Hình 29: Trường hợp dư sáng

- Ở trường hợp này, cả hai phương pháp đều có nhược điểm riêng.
- Ở phương pháp lấy ngưỡng trong không gian HSV, việc cháy sáng vùng trán đã làm cho vùng da này bị phát hiện sai.
- Cả hai kết quả đều có kết quả lọc nhiễu thấp, rõ nhất là vùng dưới mắt, những chấm đen to, nhỏ xuất hiện rất nhiều.

4.2. Làm mịn vùng da.

4.2.1. Average Blur.

Average Blur là phương pháp làm mịn đơn giản. Nó hoạt động dựa trên xét từng pixel, lấy giá trị trung bình của tất cả các pixel xung quanh pixel đang xét và thay thế giá trị của pixel đó.

$$\begin{bmatrix} 0 & 3 & 1 \\ 4 & 2 & 2 \\ 1 & 1 & 4 \end{bmatrix} \times A = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{0 + 3 + 1 + 4 + 2 + 2 + 1 + 1 + 4}{9} = 2$$

Hình 30: ví dụ xét kernel 3x3 với Average Blur.

Average Blur được hỗ trợ bởi thư viện Opencv: cv2.blur() hoặc cv2.boxFilter().



Kernel = 3x3



Kernel = 6x6



Kernel = 9x9

Hình 31: Ví dụ Average Blur

4.2.2. Gaussian Blur.

Gaussian Blur là một phương pháp làm mịn được sử dụng rộng rãi, điển hình nhất là việc Gaussian Blur làm mịn ảnh và giảm nhiễu rất tốt, thường được sử dụng ở giai đoạn tiền xử lý trước khi áp dụng mô hình học sâu.

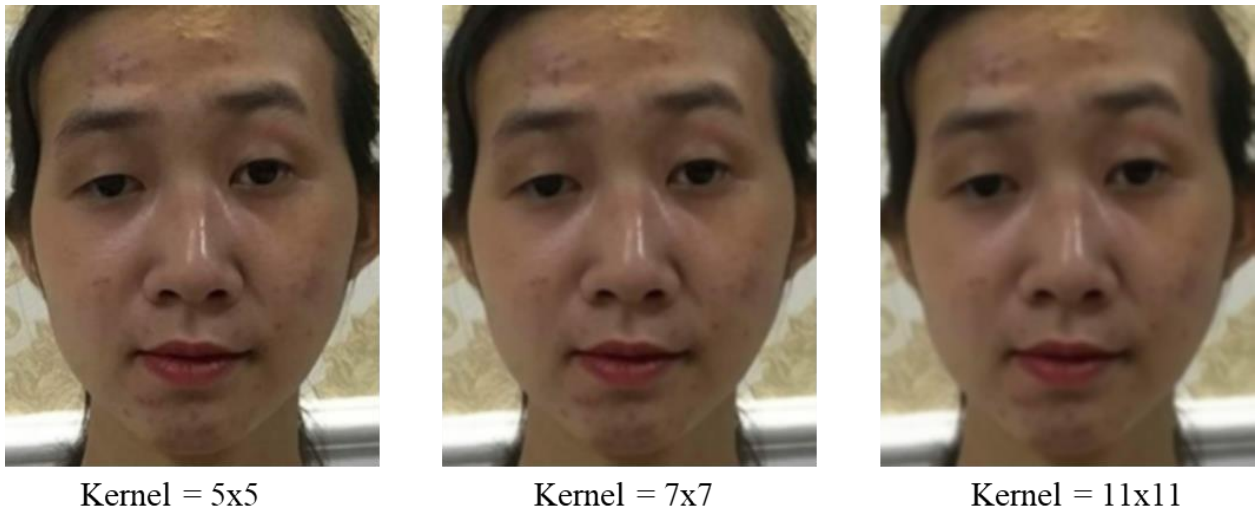
Gaussian Blur hoạt động tương tự như Average Blur nhưng khác ở điểm các pixel của kernel đều có trọng số nhất định và số chiều của kernel xét phải là số lẻ, lớn hơn 0.

$$\begin{bmatrix} 0 & 3 & 1 \\ 4 & 2 & 2 \\ 4 & 6 & 4 \end{bmatrix} \times G = \frac{1}{18} \begin{bmatrix} 1 & 3 & 1 \\ 2 & 4 & 2 \\ 1 & 3 & 1 \end{bmatrix}$$

$$\frac{0 * 1 + 3 * 3 + 1 * 1 + 4 * 2 + 2 * 3 + 2 * 2 + 4 * 1 + 6 * 3 + 4 * 1}{18} = 3$$

Hình 32: ví dụ xét kernel 3x3 với Gaussian Blur.

Gaussian Blur được hỗ trợ bởi thư viện Opencv: `cv2.GaussianBlur()`.



Hình 33: Ví dụ Gaussian Blur

4.2.3. Bilateral Blur.

Bilateral Blur là một phương pháp có độ hiệu quả cao trong việc loại bỏ nhiễu trong khi vẫn giữ cho các cạnh sắc nét. Nhưng tốc độ của nó chậm hoàn toàn so với hai phương pháp nói trên.

Về cách hoạt động của Bilateral Blur cũng giống Gaussian Blur nhưng lại xét trong một không gian. Điều này có nghĩa là trong không gian, chỉ các pixel lân cận với nhau mới được xem xét và làm mờ.

Tương tự như 2 phương pháp trên, Bilateral Blur cũng được hỗ trợ bởi thư viện Opencv: `cv2.bilateralFilter(d, sigmaColor, sigmaSpace)`

Ở đây chúng ta chỉ quan tâm 3 tham số sau:

- `d` là đường kính vùng lân cận của mỗi pixel, được sử dụng trong quá trình lọc
- Đối với tham số `sigmaColor`, khi giá trị của tham số này càng lớn thì những màu ở xa so trong vùng lân cận thì sẽ được trộn lẫn nhau.
- Còn đối với tham số `sigmaSpace`, khi tăng giá trị thì pixel xa hơn sẽ ảnh hưởng lẫn nhau miễn là màu của chúng gần giống nhau.



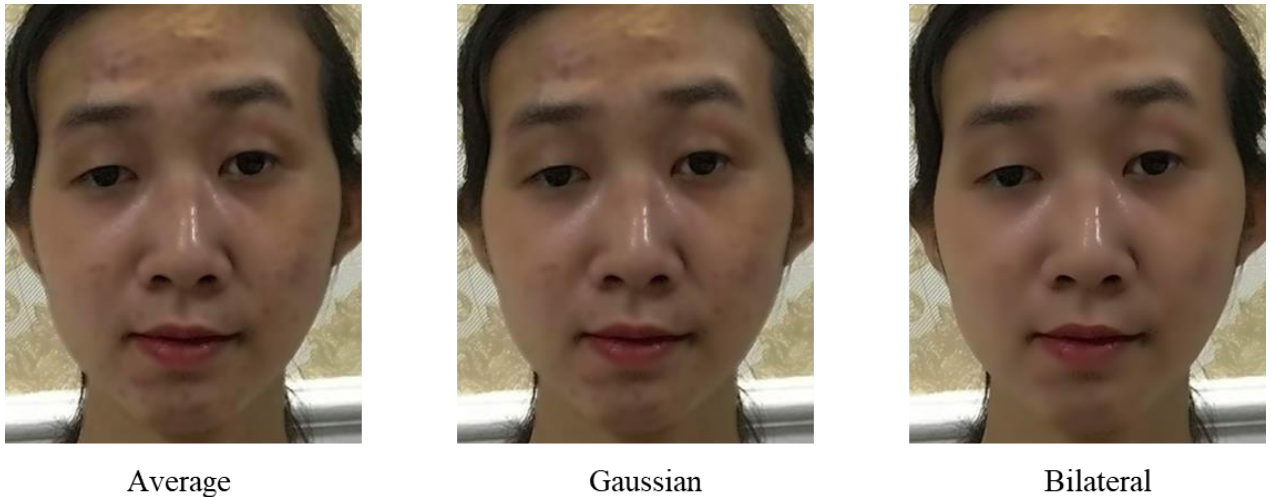
Hình 34: Ví dụ Bilateral Blur

4.2.4. So sánh và nhận xét.



Hình 35: Quá trình xác định thành phần ảnh.

Đối với 3 phương pháp mà chúng tôi đã trình ở trên, qua kết quả thực nghiệm, chúng tôi sẽ chọn những thông số tốt nhất, tiến hành xác định thành phần của ảnh cho ra kết quả hoàn thiện và đi so sánh chúng với nhau.



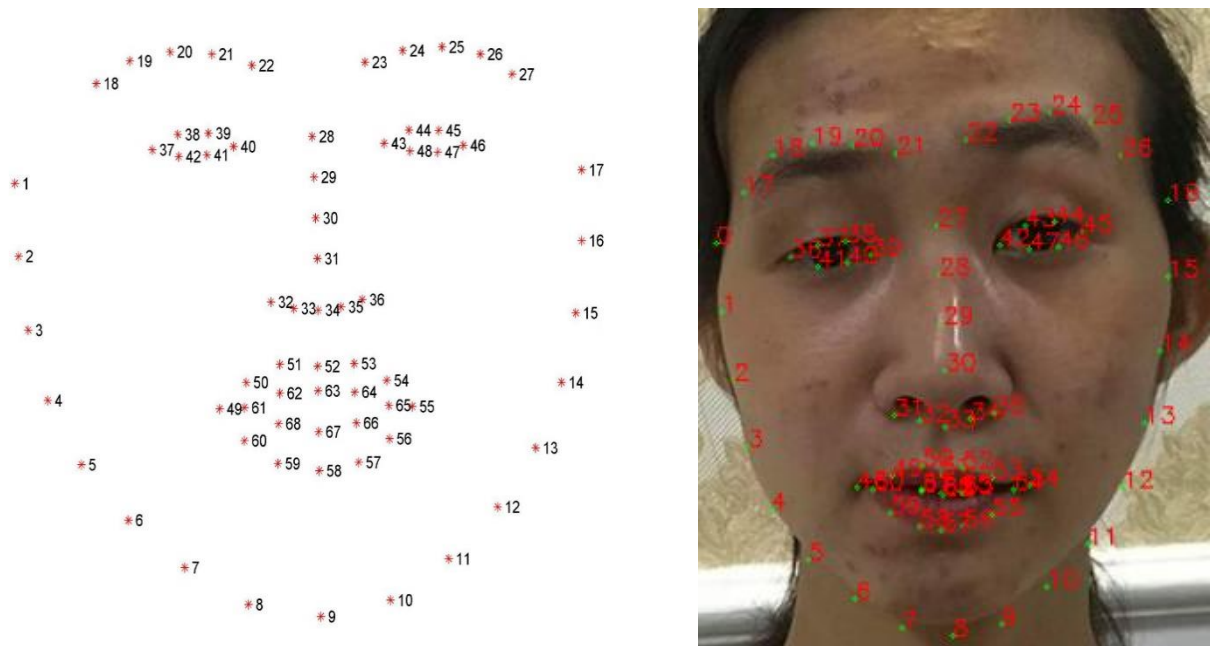
Hình 36: Kết quả so sánh.

Nhận xét:

- Ở phương pháp Average, độ làm mờ của nó tương đối thấp, không có độ hiệu quả so với ảnh ban đầu.
- Để khắc phục trường hợp trên, phương pháp Gaussian sẽ tăng những trọng số cho các pixel. Kết quả đạt được có phần mờ và mịn đi so với phương pháp đầu rất tốt. Nhưng đối với hai phương pháp trên, việc làm mịn da được cải thiện nhưng vấn đề lọc nhiễu vẫn chưa được thấy trên hai phương pháp này. Rõ nhất là những vết thâm đen vẫn còn nhiều trên hai kết quả.
- Vì vậy, chúng tôi tiếp tục tìm kiếm một phương pháp có thể đáp ứng được hai nhu cầu làm mịn và lọc nhiễu tốt. Và ở kết quả Bilateral đã thực hiện hai công việc đó rất tốt, khác biệt hoàn toàn so với hai phương pháp đầu tiên. Mặc dù lọc nhiễu nhưng các cạnh trên khuôn mặt vẫn giữ tốt → sự tự nhiên của tấm ảnh được cải thiện.

4.3. Trang điểm khuôn mặt.

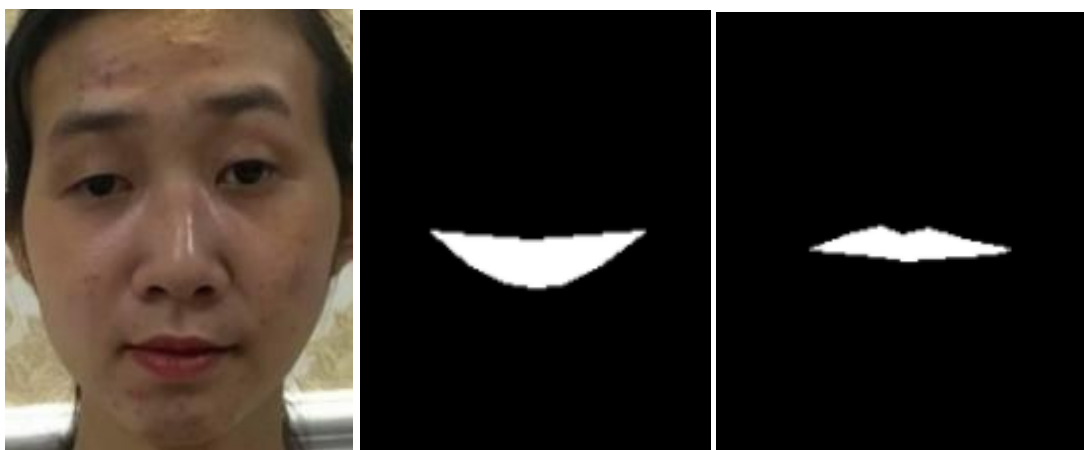
Trong bài báo cáo này, chúng tôi chỉ tô son trên phần môi của khuôn mặt. Để làm được điều này, ta cần phải biết được vị trí của môi trên mỗi khuôn mặt. Ta tạm phân chia vị trí các điểm khuôn mặt thành 68 điểm.



Hình 37: 68 điểm trên khuôn mặt

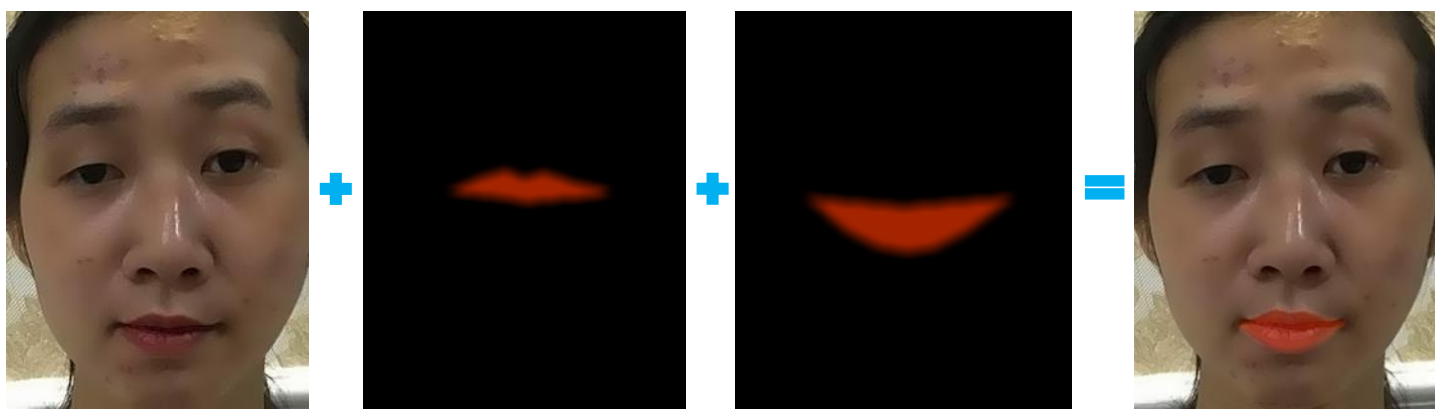
- Như hình vẽ trên có thể thấy vị trí của môi trên khuôn mặt là từ điểm 49 cho tới điểm 68.
- Cần chia thành 2 phần là môi trên và môi dưới. Việc chia thành 2 phần để khi thêm lớp màu lên ảnh gốc không bị lem vào phần răng:
 - o Môi trên bao gồm: 49,50,51,52,53,54,55,65,64,63,62,61
 - o Môi dưới bao gồm: 61,68,67,66,65,55,56,57,58,59,60,49
- Chúng tôi sử dụng pre-trained model phát hiện điểm trên khuôn mặt được tích hợp sẵn bên trong thư viện Dlib.

Sau khi xác định được các điểm của 2 phần môi, chúng tôi tạo 2 lớp(layer) riêng biệt.



Hình 38: 2 lớp (layer) môi được tạo từ khuôn mặt.

- Thêm màu, làm mờ đi để làm mịn các phần cạnh và cuối cùng là đưa 2 lớp này đè lên ảnh gốc.



Hình 39: Kết quả sau khi thêm màu son môi.

5. Kết luận và hướng phát triển.

Kết luận:

- Việc phân loại người Việt và người nước ngoài mang tính tương đối chứ chưa thực sự chính xác. Trên thực tế việc phân loại người Việt chỉ dựa trên khuôn mặt không thể chính xác. Vậy nên với những người nước ngoài nhưng có các đặc điểm giống với người Việt Nam đều được mô hình nhận diện là người Việt. Mô hình chỉ phân loại tốt với đối với người Việt với người châu Âu, châu Mỹ, châu Phi là những người có đặc trưng khác hẳn so với người Việt. Tuy nhiên việc những người nước ngoài có nét đặc trưng giống với người Việt bị phân loại sai thì việc phát hiện da cũng không bị ảnh hưởng nhiều.
- Từ việc khoanh vùng được nhóm người có đặc trưng giống với người Việt nên việc phát hiện da cho kết quả khá tốt. Tuy nhiên việc tô son ảo lên khuôn mặt còn thiếu tự nhiên. Nguyên nhân có thể do kỹ thuật tô màu son còn khá đơn giản.

Hướng phát triển:

- Để cải thiện hơn về độ chính xác trong phân loại. Chúng tôi sẽ thu thập thêm dữ liệu đa dạng hơn về các khuôn mặt người Việt, tìm kiếm các phương pháp tiền xử lý ảnh, huấn luyện mô hình trích xuất đặc trưng ảnh dành riêng cho bộ dữ liệu người Việt.
- Với phần làm đẹp khuôn mặt, sử dụng mô hình học sâu để có thể phát hiện da tốt hơn. Tìm kiếm và áp dụng các kỹ thuật tô son ảo để kết quả tự nhiên hơn.

6. Nguồn tham khảo.

- [1] - OpenCV - https://docs.opencv.org/tutorial_py_filtering.html
- [2] - 5starkarma - <https://github.com/5starkarma>
- [3] - Pyimagesearch - <https://www.pyimagesearch.com/deep-learning-opencvs-blobfromimage>
- [4] - Pham Dinh Khanh - <https://phamdinhhkhanh.github.io/faceNet.html>
- [5] - OpenFace - <https://cmusatyalab.github.io/openface>
- [6] - Face Beautification and Color Enhancement with Scene Mode Detection - <https://www.csie.ntu.edu.tw/~personal/FaceBeautificationandColorEnhancement>

7. Phân công công việc.

STT	Họ tên	MSSV	Công việc
1	Đào Văn Tài (*Nhóm trưởng)	19522148	<ul style="list-style-type: none">- Thu thập bộ dữ liệu người Việt Nam.- Lọc dữ liệu.- Tìm hiểu, trình bày mô hình MTCNN.- Hỗ trợ trích xuất đặc trưng ảnh.- Tìm hiểu, xây dựng và thực nghiệm mô hình mạng nơ-ron phân loại.- Tìm hiểu, trình bày và cài đặt phương pháp hiện da Sobel edge detection.- Tìm hiểu, trình bày và cài đặt lớp trang điểm ảo.- Tổng hợp, cài đặt code demo.- Viết phần giới thiệu, trang điểm khuôn mặt, kết luận và hướng phát triển- Làm slide thuyết trình, đóng góp ý kiến và duyệt báo cáo.
2	Ngô Gia Kiệt	19521725	<ul style="list-style-type: none">- Xuất và lọc dữ liệu người Việt Nam- Trích xuất và thực hiện phân loại trên đặc trưng Histogram

			<ul style="list-style-type: none"> - Tìm hiểu và thực nghiệm phân loại với mô hình SVM. - Tìm nguồn tài liệu pre-train model trích xuất đặc trưng. - Tìm hiểu, trình bày và cài đặt phương pháp lấy ngưỡng trong không gian màu HSV. - Tìm hiểu, trình bày và cài đặt phương pháp làm mịn da. - Tổng hợp code thực nghiệm - Tìm ảnh của các trường hợp sai trong bài toán phân loại. - Viết phần làm đẹp khuôn mặt. - Làm silde thuyết trình, định dạng hoàn thiện báo cáo
3	Nguyễn Thành Trọng	19522410	<ul style="list-style-type: none"> - Thu thập và cắt khuôn mặt người nước ngoài từ bộ dữ liệu - Lọc dữ liệu người nước ngoài - Trích xuất đặc trưng ảnh trên 3 pre-trained model - Thực nghiệm mô hình K-NN - Tìm hiểu và trình bày lý thuyết mô hình SVM và K-NN - Tìm hiểu, trình bày tiền xử lý Blob. - Làm silde thuyết trình. - Viết báo cáo phân phần phát hiện và phân loại khuôn mặt người Việt

===== HẾT =====