

**Trường Đại Học Công Nghệ Thông Tin  
Đại Học Quốc Gia TP.HCM**



**BÁO CÁO**

**ỨNG DỤNG XỬ LÝ ẢNH CHO CÁC TÁC VỤ SELFIE**

**Môn học: Nhập môn Thị giác máy tính - CS231.M11**

|                     |          |
|---------------------|----------|
| Nguyễn Văn Thành    | 19522243 |
| Hoàng Xuân Vũ       | 19522531 |
| Đinh Trọng Tùng Sơn | 19522132 |

# Mục lục

|  |    |
|--|----|
| I. Giới thiệu đề tài .....                                 | 3  |
| II. Thuật toán và cách xử lý chức năng trong ứng dụng..... | 3  |
| 1. Yêu cầu ràng buộc đối với các chức năng.....            | 3  |
| 2. Xóa và thay thế phong nền.....                          | 4  |
| 3. Thay đổi khuôn mặt từ một ảnh cho trước.....            | 9  |
| 4. Tạo filter ở mũi lên khuôn mặt.....                     | 12 |
| III. Hướng phát triển ứng dụng.....                        | 13 |
| IV. Tài liệu tham khảo.....                                | 14 |

## **I. Giới thiệu đề tài**

Trong bối cảnh xã hội hiện nay nhu cầu chụp ảnh ngày càng tăng và càng phổ biến, kéo theo đó là nhu cầu về các chức năng xử lý hình ảnh cũng trở nên bùng nổ đặc biệt là các chức năng làm đẹp, livestream ...

Hiểu rõ nhu cầu đó nên đối với môn học này chúng em thực hiện một vài chức năng cơ bản trong xử lý ảnh selfie hay livestream. Đồ án được triển khai trên web app với 03 chức năng chính là xóa và thay thế phông nền, thay đổi khuôn mặt trong bức ảnh với khuôn mặt đã mặc định sẵn, cuối cùng tạo filter mũi lên khuôn mặt được chỉ định.

### **1. Lý do thực hiện**

Các thuật toán dễ sử dụng, có độ chính xác cao không yêu cầu quá nặng về kiến thức toán và lập trình, khả năng có thể phát triển cao trong tương lai.

### **2. Tổng quát các chức năng**

- Xóa và thay đổi phông nền: Chức năng sẽ nhận diện chủ thể của bức ảnh, sau đó tiến hành tách chủ thể khỏi phông nền phía sau. Sử dụng các phông nền mặc định hoặc phông nền người dùng tự thêm vào để tạo ra bức ảnh được thay thế phông nền.
- Thay đổi khuôn mặt với bức ảnh đã mặc định trước: Người dùng ghép khuôn mặt của mình với các khuôn mặt hệ thống đã chuẩn bị trước.
- Tạo filter mũi lên khuôn mặt người dùng: Chức năng giúp người dùng thay đổi mũi của khuôn mặt thay mũi của các hình có trong ứng dụng (ưu tiên dùng các hình ảnh mặc định).

## **II. Thuật toán và cách xử lý chức năng trong ứng dụng.**

### **1. Yêu cầu ràng buộc với các chức năng**

- Xóa và thay đổi phông nền:
- Thay đổi khuôn mặt với bức ảnh đã mặc định trước: Người dùng sử dụng hình ảnh hay video rõ nét, khuôn mặt hiển thị đầy đủ các bộ phận, nếu người dùng muốn thay đổi bức ảnh mặc định thì bức ảnh đó phải rõ các bộ phận trên khuôn mặt, ảnh không quá nhỏ.

- Tạo filter mũi lên khuôn mặt người dùng: Ảnh hoặc video người được tạo filter phải rõ nét, đầy đủ các bộ phận trên khuôn mặt. Ảnh filter rõ ràng thường chọn ảnh filter có phong trắng hoặc đen để dễ dàng và chính xác trong quá trình xử lý.

## **2. Xóa và thay thế phong nền**

Tại phần này nhóm đề xuất 3 phương pháp giải quyết bài toán là:

- Sử dụng model DeepLabV3.
- Sử dụng model Selfie Segmentation.
- Sử dụng phương pháp tách nền xanh.

### **2.1. Model DeepLabV3.**

#### **2.2.1. Giới thiệu.**

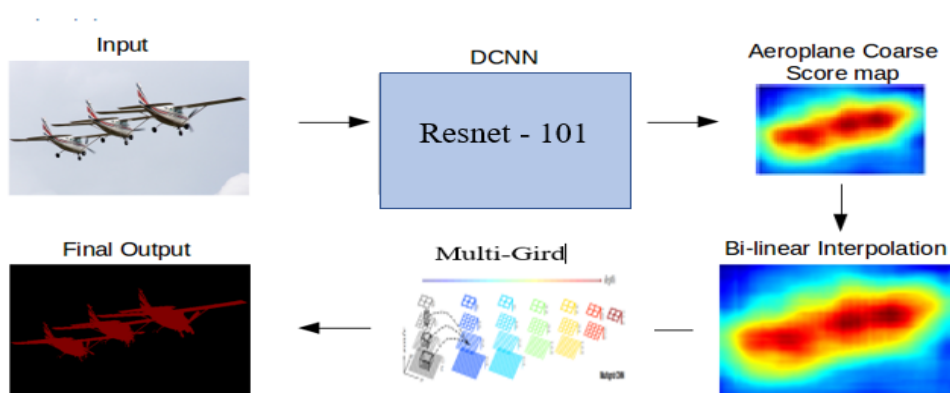
Model DeeplabV3 là một model pretrain sử dụng mạng ResNet-101 làm nhánh encoder. Nó được đào tạo trên một tập hợp con của COCO train2017, trên 20 danh mục có trong bộ dữ liệu VOC của Pascal.

#### **2.2.2. Hướng xử lý.**

Giải pháp xử lý mỗi hình ảnh: Tách nền ra khỏi ảnh ban đầu bằng cách sử dụng model “DeepLabV3” để tính toán mặt nạ (mặt nạ gồm 2 bit 0 và 1), sau khi có mặt nạ thì thực hiện phép tính “and” với ảnh gốc ban đầu thì ta sẽ thu được ảnh đã được tách nền.

#### **2.2.3. Quá trình xử lý.**

- Input: Ảnh đầu vào là ảnh có chứa người và background.
- Output: Ảnh chứa đối tượng người ban đầu nhưng đã được thay thế bằng background mới.



Hình 1. Mô tả các bước trích xuất mặt nạ trong DeepLabV3

- Các bước xử lý:

Bước 1: Đầu tiên một ảnh đầu vào sẽ được truyền vào một mạng CNN học sâu nhiều tầng (DCNN - Deep Convolutional Neural Network). Nhiệm vụ chính của DCNN là tạo ra một feature map là một biểu diễn không gian các đặc trưng của ảnh đầu vào. Kết quả sẽ thu được một feature map là một bản đồ đặc trưng của ảnh đầu vào chính là Aeroplane Coarse Score map.

Bước 2: Score map có kích thước nhỏ hơn nhiều so với ảnh gốc. Chúng ta sử dụng Bi-linear Interpolation để resize lại score map về kích thước gốc. Bố cục của ảnh sau khi resize không khác so với ảnh gốc, chỉ thay đổi về kích thước.

Bước 3: Để dự đoán feature map mask cho ảnh thì model sử dụng lớp mạng Multi-Grid. Thay vì mỗi một layer chỉ áp dụng một bộ lọc duy nhất. Trong một mạng Multi-Grid, mỗi một layer có thể được huấn luyện trên một tập hơn bộ lọc có kích thước khác nhau dưới dạng kim tự tháp (pyramid).

Bước 4: Sau khi có được mặt nạ (Mask) là một ma trận gồm bit 0 và 1 có cùng kích thước với ảnh gốc ban đầu thì tiến hành thực hiện phép tính “and” giữa mặt nạ và ảnh gốc ban đầu ta sẽ thu được ảnh đã tách nền.

Bước 5: Để thay thế ảnh nền thì chúng ta thay đổi kích thước của ảnh nền về kích thước ảnh của đối tượng sau đó “and” giữa ảnh nền và ảnh mặt nạ đã đảo bit sau đó cộng 2 ma trận là ảnh đã tách nền và kết

quả vừa thu được sau phép “and” thì chúng ta sẽ thu được ảnh đã thay thế nền.

## 2.2. Model Selfie Segmentation.

### 2.2.1. Giới thiệu.

Selfie Segmentation là model dựa trên model “MobileNetV3”, cho phép chúng ta nhanh chóng và dễ dàng loại bỏ nền từ các bức ảnh.

### 2.2.2. Hướng xử lý.

Giải pháp xử lý mỗi hình ảnh: Tách background ra khỏi ảnh ban đầu bằng cách sử dụng model “MobileNetV3” để tính toán mặt nạ, sau đó tinh chỉnh thêm mặt nạ để căn chỉnh nó với ranh giới hình ảnh. Mặt nạ được sử dụng để kết xuất với ảnh ban đầu và background muốn thay thế sau đó hiển thị hình ảnh đầu ra với nền được thay thế.

### 2.2.3. Quá trình xử lý.

- Input: Ảnh đầu vào là ảnh có chứa người và background.
- Output: Ảnh chứa đối tượng người ban đầu nhưng đã được thay thế bằng background mới.

- Các bước xử lý:

Bước 1: Ảnh ban đầu sẽ được làm giảm kích thước nhằm mục đích giảm thiểu số lượng tính toán.

Bước 2: Ảnh sẽ được xử lý tiếp bằng model Selfie segmentation gồm 2 nhánh đối xứng nhau:

- Nhánh encoder: Sử dụng “MobileNet V3” (sử dụng phương pháp tích chập tách biệt chiều sâu giúp giảm số lượng tham số) và model sẽ có kích thước giảm dần qua các layers để cuối cùng chúng ta thu được những đặc trưng bậc cao
- Nhánh Decoder: Mạng giải chập sẽ giúp chúng ta thực hiện chuyển từ đặc trưng sang ảnh. Nhánh này sẽ có kiến trúc là shape của các layers tăng dần.

Qua từng layer mạng sẽ giải mã các khối đặc trưng thành những thông tin không gian của từng điểm ảnh và tạo thành một bức ảnh mới ở output.

Quá trình gia tăng kích thước tại các layers của mạng giải chập còn được gọi là Upsampling và mạng giải chập này sử dụng tích chập chuyển vị (Transposed Convolution) trong quá trình này.

Bước 3: Sau khi xử lý sẽ nhận được 1 ma trận gọi là “mask”, mask sẽ có cùng kích thước với "input". Mỗi pixel của mask được gán một số thực có phạm vi trong khoảng  $[0,1]$  gọi là ngưỡng. Con số càng gần 1 thì độ tin cậy rằng pixel đại diện cho một người càng cao và ngược lại.

Bước 4: Từ ma trận “Mask” chúng ta có thể tạo ảnh nhị phân bằng cách lấy ngưỡng tin cậy mỗi pixel có giá trị nhỏ hơn ngưỡng sẽ được gán lại giá trị là 0 và ngược lại pixel có giá trị cao hơn ngưỡng sẽ được gán giá trị là 255.

Bước 5: Tiến hành làm mịn “Mask” bằng phương pháp bilateral filter. Sau đó tiến hành kết xuất để tạo ảnh được thay thế Background.

## 2.3. Phương pháp tách nền xanh.

### 2.3.1. Giới thiệu.

Phương pháp tách nền xanh là một kỹ thuật được sử dụng để cô lập và loại bỏ một màu sắc hoặc một giá trị độ sáng nhất định khỏi một bức ảnh khiến màu sắc hoặc giá trị độ sáng được trở nên trong suốt.

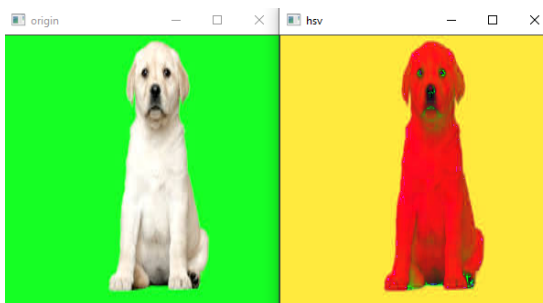
Có thể dùng bất cứ màu sắc nào để làm Background trong kỹ thuật này, nhưng ta lại chọn màu xanh vì chúng không trùng với bất kỳ tông màu da hay màu tóc tự nhiên nào, và cũng rất hiếm khi trùng với trang phục hay phụ kiện.

### 2.3.2. Phương pháp sử dụng.

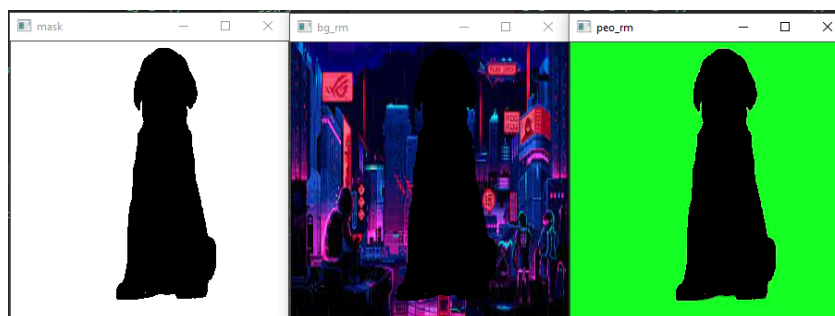
Lọc màu background ra khỏi ảnh bằng cách lấy ngưỡng lựa chọn phù hợp cho các giá trị màu của mỗi pixel.

### 2.3.3. Quá trình xử lý.

Bước 1: Lựa chọn kênh màu HSV trong quá trình xử lý, Ban đầu ảnh được chuyển đổi sang kênh màu HSV. Sau đó được chọn ngưỡng phù hợp để tách nền ra khỏi đối tượng.

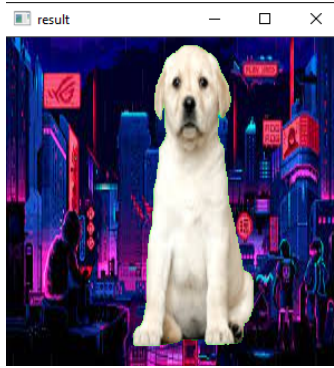


Bước 2: Tạo mask là ảnh nhị phân với mỗi pixel: Được cho là đối tượng sẽ nhận giá trị là 0, được cho là background thì sẽ nhận giá trị 255.



Bước 3: Từ mask có thể sử dụng phép toán bitwise\_and để tạo nên hình ảnh với đối tượng ban đầu nhưng đã được thay thế background.





## 2.4. Ưu điểm và nhược điểm của các phương pháp.

- Ưu điểm:

- DeepLab v3: cho chất lượng ảnh sau khi xử lý tốt, background được xóa chuẩn ko bị xóa nhầm vào phần chứa tối tương người.
- Selfie Segmentation: cho ra chất lượng ảnh ổn định, với thời gian xử lý thấp có thể dùng được trong stream video.
- Phương pháp tách nền xanh với khả năng tách nền tốt, đáp ứng được nhu cầu xử lý với các bức ảnh có phong nền xanh trong studio.

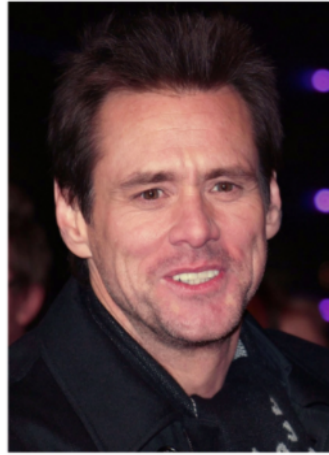
- Nhược điểm:

- Deeplab v3 xử lý phức tạp cho nên yêu cầu phần cứng cao vì thể độ trễ lớn.
- Selfie Segmentation: Khi dơ tay hoặc cầm nắm hay đội mũ thì ảnh được xử lý chưa tốt dẫn đến bị xóa nhầm các phần khác của cơ thể.
- Phương pháp tách nền xanh chỉ xử lý tốt đối với ảnh xanh và nếu màu sắc không giống với ngưỡng đã chỉ định thì sẽ ko xử lý chính xác.

## 3. Thay đổi khuôn mặt từ một bức ảnh cho trước

### 3.1. Thuật toán và các bước xử lý

Bước 1: Đầu vào sẽ là một bức ảnh chứa khuôn mặt sẽ được dùng để thay thế các khuôn mặt khác và bức ảnh hay video sẽ được thay thế khuôn mặt (Lưu ý: các hình ảnh,video phải tuân theo các ràng buộc ở phần trên)

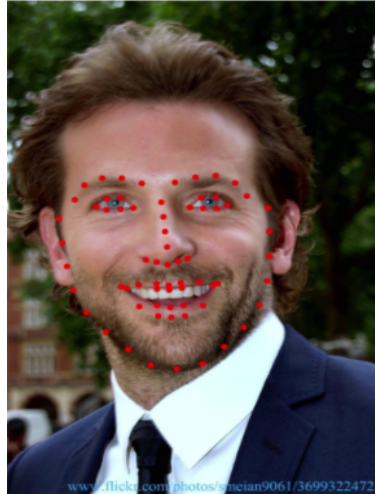


Bước 2: Sử dụng các phương pháp nhận diện khuôn mặt để nhận diện khuôn mặt có trong các hình ảnh, video. Sau đó ta cắt lấy phần khuôn mặt đã được nhận diện.

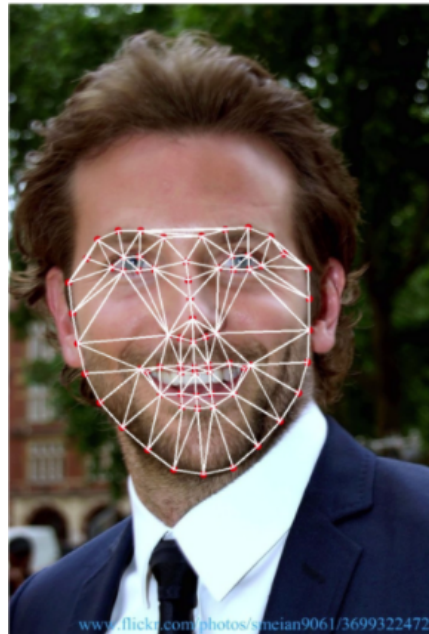
(Ở đây ta sử dụng máy dò khuôn mặt kết hợp giữa HOG và Linear SVM của Dlib nhanh và hiệu quả)



Bước 3: Sử dụng kỹ thuật facial landmark để xác định 68 điểm có trong khuôn mặt, kỹ thuật này giúp chúng nhận biết khung hình của khuôn mặt



Bước 4: Từ bức ảnh đã tìm được 68 điểm trước đó ta sử dụng thuật toán Delaunay để xác định các tam giác cấu tạo nên khuôn mặt (Các tam giác kết hợp với nhau sẽ bao phủ hết khuôn mặt và những tam giác đó không chồng lấn lên nhau)



Bước 5: Ta thực hiện lặp lại các quá trình với ảnh hay video dùng để thay khuôn mặt.

Bước 6: Lấy lần lượt các tam giác ở bức ảnh có khuôn mặt dùng để thay thế thay đổi kích thước giống tam giác của người được thay thế và làm thay đổi da mặt chứa trong tam giác đó (mục đích tạo cho khung giống khung khuôn mặt người được thay thế và phần da mặt khi thay thế không bị biến đổi quá nhiều)

Bước 7: Ta kết hợp các tam giác lại với nhau, sau đó cắt khuôn mặt người được thay thế, tiếp đó hợp nhất hay bức ảnh với nhau.



Bước 8: Sử dụng các kỹ thuật làm mờ, mịn để làm bức ảnh được tự nhiên nhất.

### 3.2. Hạn chế trong quá trình thực thi

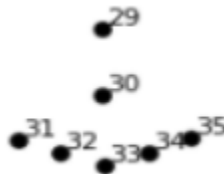
- Ảnh có nhiều đường nét nhiễu do các cạnh tam giác chồng lấn lên nhau làm bức ảnh không được tự nhiên.
- Trong quá trình nhận diện khuôn mặt cần phải rõ ràng các bộ phận của khuôn mặt, với các bức ảnh anime thông thường nó sẽ không nhận diện được.

## 4. Tạo filter mũi lên khuôn mặt người dùng

### 4.1. Thuật toán và các bước xử lý

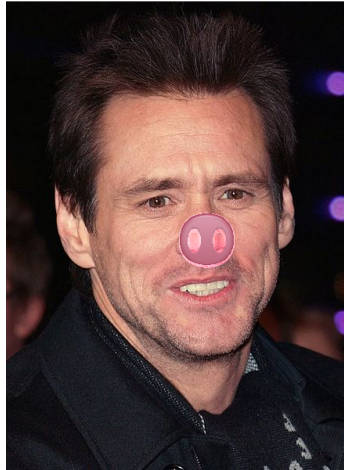
Bước 1: Đầu vào là bức ảnh hoặc video người cần được tạo filter và bức ảnh filter thỏa mãn các yêu cầu và ràng buộc ở trên.

Bước 2: Sử dụng kỹ thuật facial landmark xác định vùng mũi của đối tượng.



Bước 3: Đưa ảnh filter về kích thước của mũi đối tượng.

Bước 4: Tiến hành cắt vùng mũi của đối tượng và ghép filter mũi và phần cắt của đối tượng.



#### 4.2. Hạn chế trong quá trình triển khai

- Do sử dụng cùng bộ dò khuôn mặt như phần trước nên phần nhận diện khuôn mặt thì khuôn mặt cần được rõ ràng.

### III. Hướng phát triển

1. Hoàn thiện các chức năng, khắc phục các yếu điểm của các thuật toán.
2. Tiến hành phát triển trên các mô hình ứng dụng di động.
3. Tăng tốc độ của khả năng realtime.
4. Phát triển thêm nhiều các filter và trên các bộ phận khác nhau trên cơ thể con người.

#### IV. Tài liệu tham khảo

- [pysource.com/2019/05/14/swap-faces-face-swapping-opencv-with-python-part-5/](https://pysource.com/2019/05/14/swap-faces-face-swapping-opencv-with-python-part-5/)
- <https://pysource.com/2019/03/25/pigs-nose-instagram-face-filter-opencv-with-python/>
- <https://www.quora.com/Why-use-an-HSV-image-for-color-detection-rather-than-an-RGB-image#:~:text=The%20reason%20we%20use%20HSV,relative%20ly%20lesser%20than%20RGB%20values.>
- <https://developers.google.com/ml-kit/vision/selfie-segmentation>
- <https://phamdinhhkhanh.github.io/2020/06/18/DeepLab.html?fbclid=IwAR1PJELamNXE3GHYPZfRf6Bi-WIHliGvsk61V0BcAT-BTglOeGg-JIlr1lU>