

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN MÔN HỌC**  
**CS406 – XỬ LÝ ẢNH VÀ ỨNG DỤNG**

# **BACKGROUND REPLACEMENT**

Giảng viên hướng dẫn: Mai Tiến Dũng

Sinh viên thực hiện: Phạm Thành Luân - 19520154

Huỳnh Văn Hùng - 19521564

**TP. HỒ CHÍ MINH – 12/2021**

# Mục Lục

|  |   |
|--|---|
| 1. Giới thiệu đề tài .....                                   | 1 |
| 2. Đề xuất phương pháp .....                                 | 1 |
| 2.1. MediaPipe.....  | 1 |
| 2.1.1. Giới thiệu MediaPipe: .....                           | 1 |
| 2.1.2. Selfie-segmentation: .....                            | 1 |
| 2.1.3. Các bước thực hiện:.....                              | 2 |
| 2.1.4. Ưu, khuyết điểm:.....                                 | 4 |
| 2.2. Background subtraction: .....                           | 4 |
| 2.2.1. Tổng quan về Background subtraction: .....            | 4 |
| 2.2.2. Các bước thực hiện:.....                              | 4 |
| 2.2.3. Ưu, khuyết điểm:.....                                 | 5 |
| 2.3. Tìm biên cạnh và viền .....                             | 5 |
| 2.3.1. Tổng quan về phương pháp tìm biên cạnh và viền: ..... | 5 |
| 2.3.2. Các bước thực hiện:.....                              | 6 |
| 2.3.3. Ưu, khuyết điểm:.....                                 | 8 |
| 2.4. Mở rộng bài toán .....                                  | 8 |
| 3. Bảng phân công nhiệm vụ:.....                             | 9 |
| 4. Tài liệu tham khảo .....                                  | 9 |

# 1. Giới thiệu đề tài

Trong cuộc sống hiện nay có nhiều nhu cầu của người sử dụng chỉ muốn giữ lại ảnh chụp của họ và xóa toàn toàn bộ cảnh vật xung quanh, hay cao hơn là thay đổi phong nền với các khung cảnh khác tùy theo sở thích và sự sáng tạo của họ. Đặc biệt trong mùa covid này việc mở camera khi học online là vấn đề thiết yếu, rất nhiều người có nhu cầu muốn xóa hoặc thay đổi cách vật phía sau để đảm bảo quyền riêng tư. Vì vậy nhóm em quyết định chọn đề tài này để nghiên cứu.

- Input: một ảnh nhân vật và một ảnh nền muốn thay đổi trong ảnh nhân vật.
- Output: ảnh nhân vật đã thay đổi phong nền.

## 2. Đề xuất phương pháp

### 2.1. MediaPipe

#### 2.1.1. Giới thiệu MediaPipe:

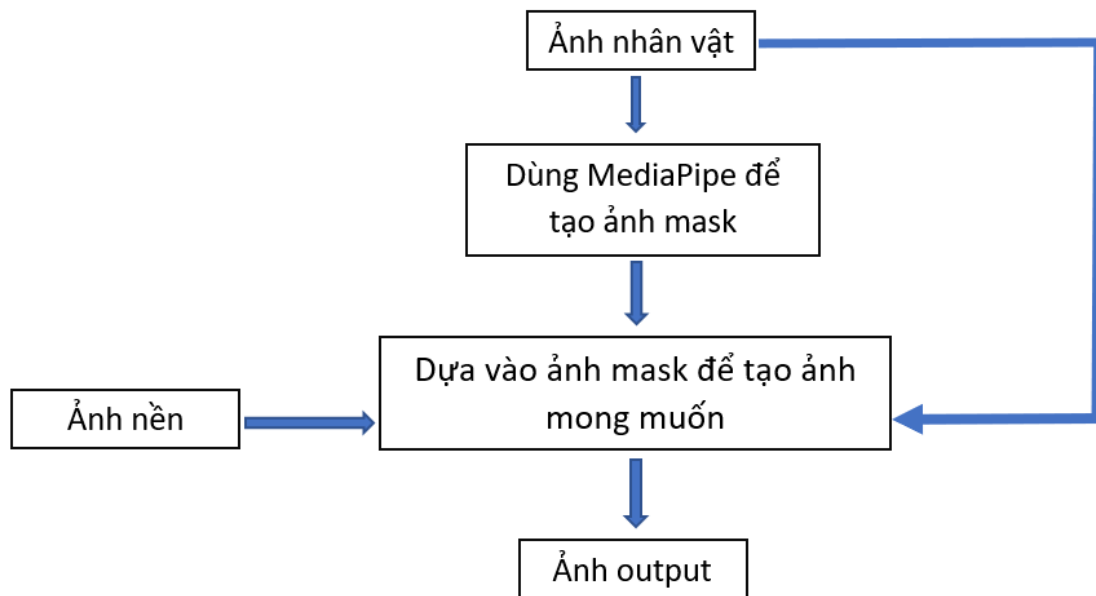
MediaPipe là một framework tập hợp của một loạt các giải pháp Machine Learning đa nền tảng, có thể can thiệp được và cực kỳ lightweight và nó được phát triển bởi Google. Hầu hết các giải pháp giải quyết các bài toán nổi bật trong lĩnh vực Computer Vision - Thị giác máy tính như Face detection, object detection,... đều được Google cài đặt trong MediaPipe. Chi tiết: <https://google.github.io/mediapipe/>

#### 2.1.2. Selfie-segmentation:

Trong MediaPipe có một mô hình xóa phong nền, họ gọi nó là selfieSegmentation. Nhóm sẽ sử dụng mô hình này để xây dựng ứng dụng của mình.

Mediapipe's selfie segmentation tạo ra một ma trận có cùng kích thước với ảnh với giá trị các phần tử trong  $[0;1]$ . Giá trị của phần tử càng lớn thì khả năng pixel tại vị trí đó là của nhân vật càng cao.

### 2.1.3. Các bước thực hiện:



*Lưu đồ thuật toán*

### Mô tả cụ thể và ví dụ:

Bước 1: Chọn ảnh nhân vật và nền:



*Ảnh nhân vật*



*Ảnh nền*

Sau khi chọn ảnh tiến hành resize ảnh nền sao cho cùng kích thước với ảnh nhân vật.

Bước 2 : Dùng MediaPipe để tìm tạo ảnh mask:



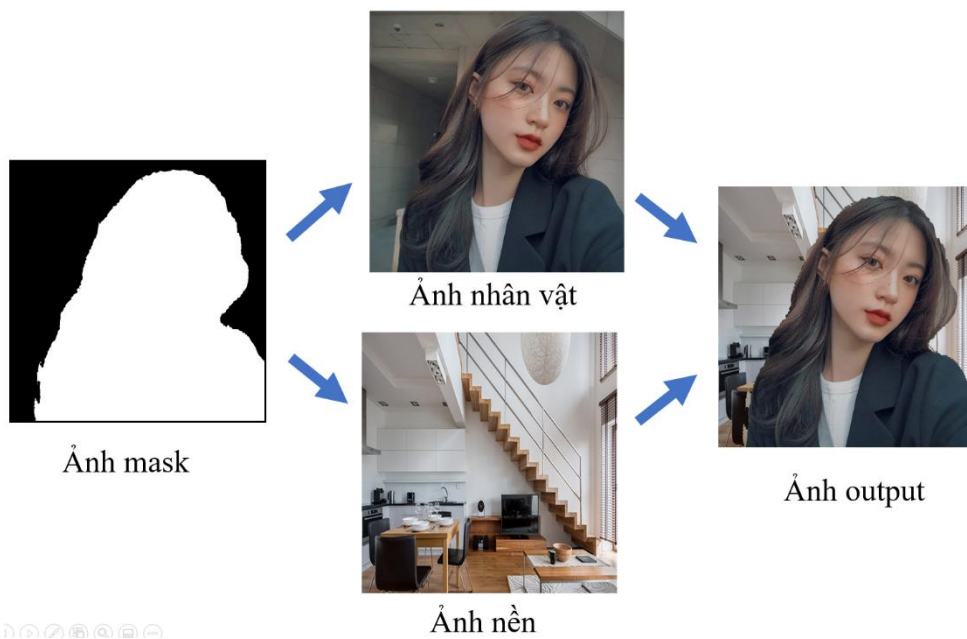
Ảnh nhân vật



Ảnh mask

Mediapipe's selfie segmentation tạo ra một ma trận có cùng kích thước với ảnh với giá trị các phần tử trong  $[0;1]$ . Các giá có giá trị bé hơn ngưỡng đã chọn trước thì cho điểm ảnh có giá trị bằng 0 (màu đen), lớn hơn ngưỡng có giá trị bằng 255 (màu trắng). Từ đó ta sẽ thu được ảnh mask.

Bước 3: Tạo ảnh output:



Ảnh minh họa cách tạo ảnh output

Trên ảnh mask các điểm ảnh có giá trị bằng 0 thì chọn điểm ảnh của ảnh nền, ngược lại thì chọn điểm ảnh của ảnh nhân vật. Từ đó ta sẽ thu được ảnh output mong muốn.

#### 2.1.4. Ưu, khuyết điểm:

##### Ưu điểm:

- Cung cấp một giải pháp inference nhanh chóng: Google khẳng định rằng bộ công cụ này có thể chạy ổn định trên hầu hết các cấu hình phần cứng thông dụng.
- Dễ dàng cài đặt và triển khai: Việc cài đặt cực kỳ dễ dàng và tiện lợi, có thể triển khai trên nhiều nền tảng khác nhau
- Mã nguồn mở và miễn phí: Toàn bộ source code được công khai, người dùng hoàn toàn có thể sử dụng và tùy chỉnh trực tiếp để phù hợp với bài toán của mình.

##### Khuyết điểm:

- Mediapipe gây khó hiểu: MediaPipe không quá phức tạp đối với người dùng, tuy nhiên để thực sự hiểu mediapipe cần phải đi sâu vào mã nguồn của nó, mã nguồn này cực kỳ phức tạp và gây khó hiểu cho người đọc.
- Chỉ xử lý tốt trường hợp ảnh người, những đối tượng khác hiệu suất mang lại không cao.

## 2.2. Background subtraction:

### 2.2.1. Tổng quan về Background subtraction:

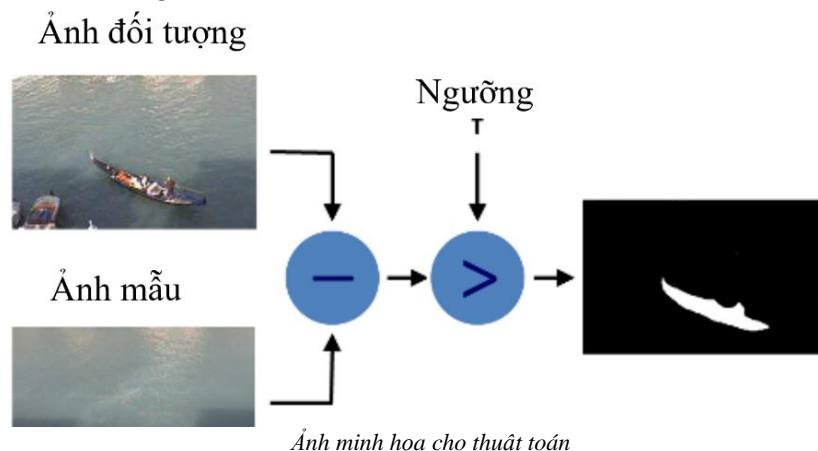
Đây là một kỹ thuật phổ biến và được sử dụng rộng rãi để tạo foreground mask. Có nhiều phương pháp để tìm foreground mask, trong bài báo cáo này nhóm sẽ giới thiệu phương pháp đơn giản nhất đó là thực hiện phép trừ giữa khung hình hiện tại (current frame) và ảnh nền mẫu (background model).

### 2.2.2. Các bước thực hiện:

Bước 1: Xác định ảnh nền mẫu.

Bước 2: Thực hiện phép trừ giữa khung hình hiện tại và ảnh nền mẫu.

Bước 3: Chọn ngưỡng để tạo foreground mask.



### **2.2.3. Ưu, khuyết điểm:**

#### **Ưu điểm:**

- Chi phí giải thuật thấp
- Tách được nhiều đối tượng khác nhau
- Đơn giản dễ hiểu

#### **Khuyết điểm:**

- Cần xác định được ảnh nền mẫu
- Có nhiều ràng buộc:
  - Nền là cảnh tĩnh
  - Máy ảnh phải được đặt cố định
  - Ánh sáng môi trường không thay đổi

## **2.3. Tìm biên cạnh và viền**

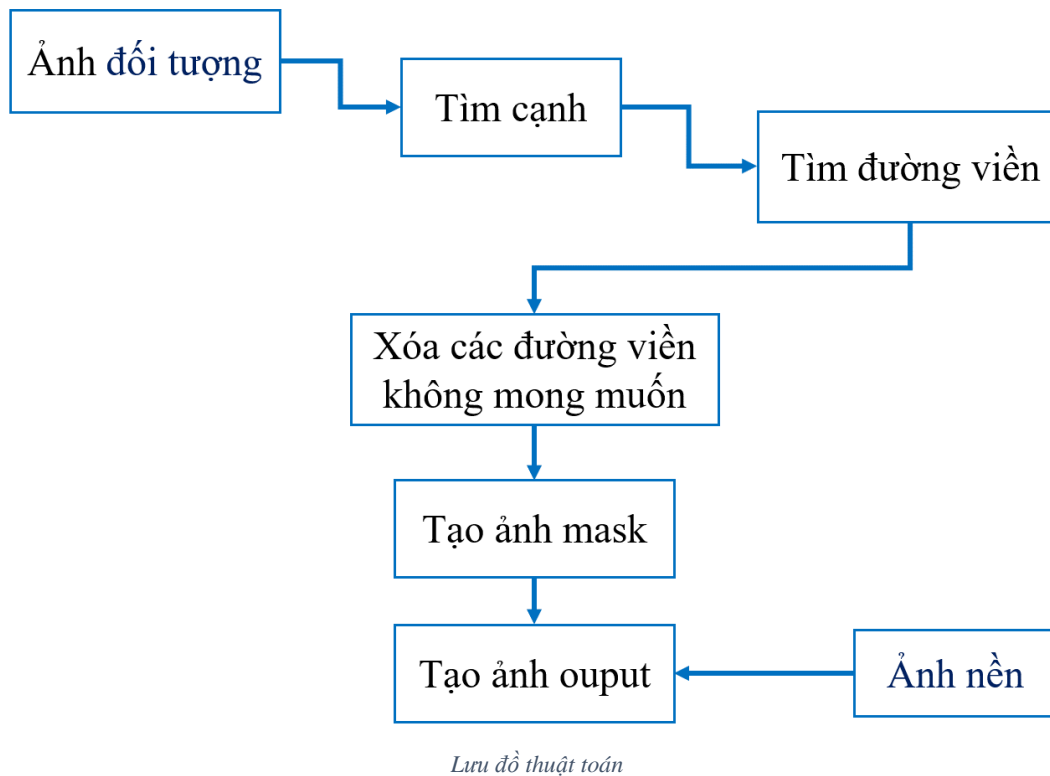
### **2.3.1. Tổng quan về phương pháp tìm biên cạnh và viền:**

Phương pháp này được phát triển dựa trên hai khái niệm: phát hiện biên cạnh và viền.

- Phát hiện cạnh là phương pháp tìm các đường ranh giới tương phản hoặc các cạnh trong ảnh. Có nhiều phương pháp để tìm biên cạnh nhưng phương pháp phổ biến nhất đó là Canny và đây cũng thuật toán mà nhóm sử dụng với sự hỗ trợ từ thư viện OpenCV.
- Khi các cạnh được tìm thấy việc tìm các đường viền bao quanh vật thể sẽ trở nên dễ dàng và chính xác hơn. Có được các đường viền ta sẽ xác định được chủ thể của ảnh. Ở phương pháp này nhóm sử dụng hàm findContours của thư viện OpenCV.

Mục đích của phương pháp này đó là sử dụng các kiến thức xử lý ảnh đã học để giải quyết bài toán mà không cần các mô hình máy học.

### 2.3.2. Các bước thực hiện:



### Mô tả cụ thể và ví dụ:

Bước 1: Chọn ảnh đối tượng và nền:



Ảnh nhân vật

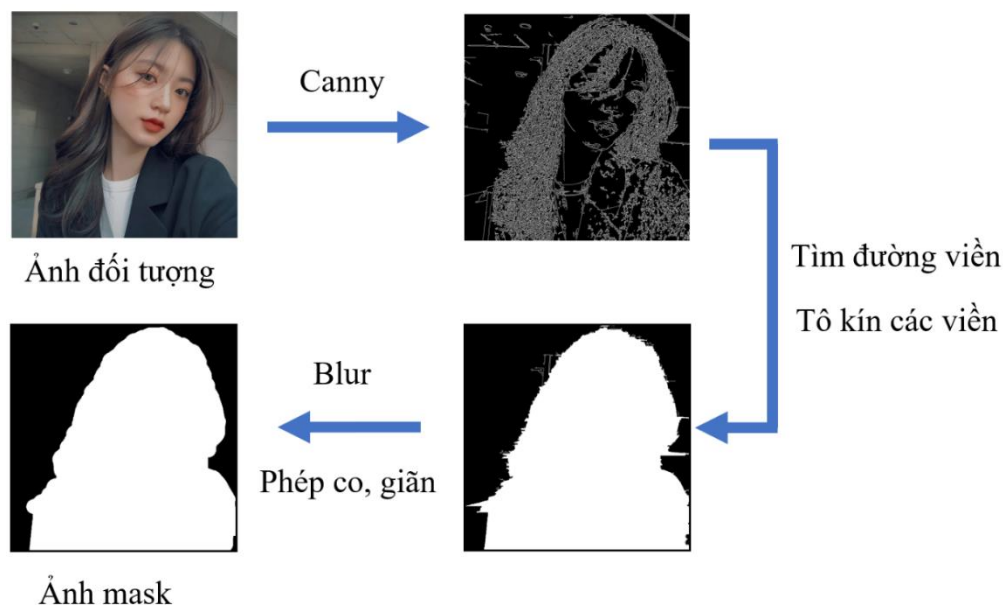


Ảnh nền

Sau khi chọn ảnh tiến hành resize ảnh nền sao cho cùng kích thước với ảnh nhân vật.



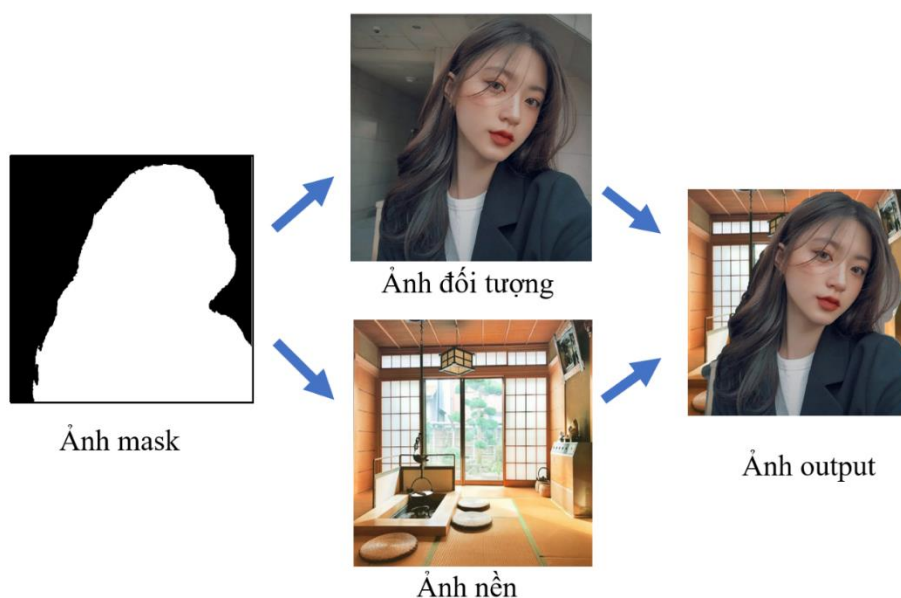
## Bước 2: Tạo ảnh mask:



Ảnh minh họa cách tạo ảnh mask

Sử dụng phương pháp Canny để tìm biên cạnh, sau đó dùng hàm findContours để tìm các đường viền. Khi đã có các đường viền ta sẽ tô kín các điểm ảnh bên trong viền. Để cho ảnh mask được chính xác và mềm mại hơn ta sẽ tiến hành blur và thực hiện các phép co giãn trên ảnh.

## Bước 3: Tạo ảnh output:



Ảnh minh họa cách tạo ảnh output

Trên ảnh mask các điểm ảnh có giá trị bằng 0 thì chọn điểm ảnh của ảnh nền, ngược lại thì chọn điểm ảnh của ảnh nhân vật. Từ đó ta sẽ thu được ảnh output mong muốn.

### 2.3.3. Ưu, khuyết điểm:

#### Ưu điểm:

- Không cần dùng các mô hình máy học.
- Linh hoạt cho nhiều chủ thể khác nhau.

#### Khuyết điểm:

- Khó xử lí cho các trường hợp phức tạp
- Khó xử lí realtime
- Có nhiều bước tinh chỉnh các tham số thủ công

## 2.4. Mở rộng bài toán

Các phương pháp trên chỉ là giải pháp về phần mềm và không thể nào chính xác hoàn toàn. Cách tốt nhất đó chính là nâng cấp phần cứng, sử dụng thêm cảm biến đo khoảng cách từ máy ảnh đến vật thể. Khi đó ta sẽ tách được chủ thể ra khỏi nền một cách hoàn hảo nhất. Tiêu biểu chính là cảm biến LiDAR trên iPhone.



*Vị trí cảm biến LiDAR trên iPhone*

### 3. Bảng phân công nhiệm vụ:

|                             | Công việc   | Đánh giá  |
|-----------------------------|---|---|
| Huỳnh Văn Hùng<br>19521564  | <ul style="list-style-type: none"><li>• Code thuật toán sử dụng mediapipe và một phần giao diện</li><li>• Viết và báo cáo về thuật toán sử dụng mediapipe</li></ul>       | <ul style="list-style-type: none"><li>• Thuật toán hoạt động tốt</li><li>• Hoàn thành báo cáo</li></ul> |
| Phạm Thành Luân<br>19520154 | <ul style="list-style-type: none"><li>• Code thuật toán Background subtraction; Tìm biên cạnh và viền; giao diện</li><li>• Viết và báo cáo các nội dung còn lại</li></ul> | <ul style="list-style-type: none"><li>• Thuật toán hoạt động tốt</li><li>• Hoàn thành báo cáo</li></ul> |

### 4. Tài liệu tham khảo

[1] *How to Remove Background of Images in Python*

[https://data-flair.training/blogs/python-remove-image-background/?fbclid=IwAR1I4ax0\\_ZWNNwYPMcegvNbztqOfkQPSPm6EcolvPMIE-kHSNQ4o5Hb9zN4](https://data-flair.training/blogs/python-remove-image-background/?fbclid=IwAR1I4ax0_ZWNNwYPMcegvNbztqOfkQPSPm6EcolvPMIE-kHSNQ4o5Hb9zN4)

[2] *MediaPipe Selfie Segmentation*

[https://google.github.io/mediapipe/solutions/selfie\\_segmentation#python-solution-api](https://google.github.io/mediapipe/solutions/selfie_segmentation#python-solution-api)

[3] *How to Use Background Subtraction Methods?*

[https://docs.opencv.org/4.x/d1/dc5/tutorial\\_background\\_subtraction.html](https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html)

[4] *Background Removal with Python*

<https://towardsdatascience.com/background-removal-with-python-b61671d1508a>