ĐẠI HỌC QUỐC GIA TP.HCM

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

--------



**BÁO CÁO ĐỒ ÁN**

**Ứng dụng tạo kỹ xảo hình ảnh**

**CHROMA-KEY**

**Khoa:** Khoa học máy tính

**Môn:** Nhập môn thị giác máy tính

***GV lý thuyết:***

Nguyễn Vĩnh Tiệp

***Thành viên nhóm:***

Nguyễn Huỳnh Đức – 16520240

Mục lục

[1. Giới thiệu: 2](#_Toc14623622)

[1.1. Chroma key là gì ? 2](#_Toc14623623)

[1.2. Xử lý: 3](#_Toc14623624)

[a. Phông nền xanh lục: 3](#_Toc14623625)

[b. Các yếu tố chính: 3](#_Toc14623626)

[c. Ánh sáng: 3](#_Toc14623627)

[2. Chương trình Chroma Key Visual Effect: 5](#_Toc14623628)

[2.1. Giới thiệu thuật toán K-means clustering: 5](#_Toc14623629)

[2.2. Áp dụng thuật toán vào bài toán Chroma Key: 6](#_Toc14623630)

[a. Object segmentation (Tách vật thể): 7](#_Toc14623631)

[b. Trộn vật thể tách với background: 11](#_Toc14623632)

**Mục lục hình ảnh**

[Hình 1: Ảnh một người đang dự báo thời tiết trước nền xanh. 2](#_Toc14627025)

[Hình 2: Sử dụng phông nền màu xanh dương (bluescreen) 4](#_Toc14627026)

[Hình 3: sử dụng phông nền màu xanh lá (greenscreen) 4](#_Toc14627027)

[Hình 4: Bài toán với 3 clusters 5](#_Toc14627028)

[Hình 5: Phân vùng lãnh hải của mỗi đảo. Các vùng khác nhau có màu sắc khác nhau. 6](#_Toc14627029)

[Hình 6: input và output sau khi sử dung chroma key. 7](#_Toc14627030)

[Hình 7: Object segmentation (tách vật thể). 7](#_Toc14627031)

[Hình 8: Tạo ảnh mask 11](#_Toc14627032)

[Hình 9: Kết quả trộn ảnh mask và background. 12](#_Toc14627033)

[Hình 10: Kết quả cuối cùng sau khi trộn 2 ảnh lại. 13](#_Toc14627034)

[Hình 11: Ứng dụng trên web. 13](#_Toc14627035)

# Giới thiệu:

## Chroma key là gì ?

**Chroma key** là một [hiệu ứng hình ảnh](https://en.wikipedia.org/wiki/Visual_effects" \o "Hiệu ứng hình ảnh) / [post-sản xuất](https://en.wikipedia.org/wiki/Post-production) kỹ thuật cho [hợp lại](https://en.wikipedia.org/wiki/Compositing" \o "Tổng hợp) hai [hình ảnh](https://en.wikipedia.org/wiki/Image) hoặc [đoạn video](https://en.wikipedia.org/wiki/Video" \o "Video) với nhau dựa trên màu sắc màu ([chroma](https://en.wikipedia.org/wiki/Colorfulness) range). Kỹ thuật này đã được sử dụng trong nhiều lĩnh vực để xóa [nền](https://en.wiktionary.org/wiki/background" \o "nền: nền) khỏi chủ đề của ảnh hoặc video - đặc biệt là [bản tin](https://en.wikipedia.org/wiki/News" \o "Tin tức) , [hình ảnh chuyển động](https://en.wikipedia.org/wiki/Motion_picture" \o "Hình ảnh chuyển động) và [trò chơi video](https://en.wikipedia.org/wiki/Video_game" \o "Tro chơi điện tử) các ngành công nghiệp.

Một dải màu trong các cảnh quay phía trước được làm trong suốt, cho phép các cảnh quay nền được quay riêng biệt hoặc một hình ảnh tĩnh được chèn vào cảnh. Kỹ thuật chroma keying thường được sử dụng trong [sản xuất video](https://en.wikipedia.org/wiki/Video_production" \o "Sản xuất video) và hậu kỳ. Kỹ thuật này cũng được gọi là color keying, lớp phủ màu tách, hoặc theo cụm khác nhau cho các biến thể màu sắc cụ thể liên quan như màn hình xanh và màn hình xanh.

Color keying có thể được thực hiện với nền của bất kỳ màu nào đồng nhất và khác biệt, nhưng nền màu xanh lá cây và màu xanh lam được sử dụng phổ biến hơn vì chúng khác biệt nhất về màu sắc với hầu hết [các màu da người](https://en.wikipedia.org/wiki/Human_skin_color" \o "Màu da người).

Nó thường được sử dụng để [phát sóng](https://en.wikipedia.org/wiki/Broadcast" \o "Broadcast)[dự báo thời tiết](https://en.wikipedia.org/wiki/Weather_forecasting) , trong đó người [dẫn chương trình tin tức](https://en.wikipedia.org/wiki/News_presenter" \o "News presenter) thường được nhìn thấy đứng trước bản đồ [CGI](https://en.wikipedia.org/wiki/Computer-generated_imagery) lớn trong các [bản tin](https://en.wikipedia.org/wiki/Newscast" \o "Newscast) truyền hình trực tiếp , mặc dù trên thực tế, nó là một nền lớn màu xanh hoặc màu xanh lá cây. Khi sử dụng màn hình màu xanh, các bản đồ thời tiết khác nhau được thêm vào các phần của hình ảnh có màu xanh lam. Nếu người trình bày tin tức mặc quần áo màu xanh, quần áo của người đó cũng sẽ được thay thế bằng video nền. Chroma Key cũng phổ biến trong ngành công nghiệp giải trí cho hiệu ứng hình ảnh trong [phim](https://en.wikipedia.org/wiki/Movies" \o "Movies) và trò chơi video.



Hình 1: Ảnh một người đang dự báo thời tiết trước nền xanh.

## Xử lý:

Đối tượng chính được quay hoặc chụp ảnh trên nền bao gồm một màu hoặc một dải màu tương đối hẹp, thường là xanh dương hoặc xanh lục vì những màu này được coi là xa nhất với màu da.

### Phông nền xanh lục:

Màu xanh lá cây được sử dụng làm nền cho TV và quay phim điện tử hơn bất kỳ màu nào khác bởi vì những người dự báo thời tiết trên truyền hình có xu hướng mặc bộ đồ màu xanh dương. Chroma Key lần đầu tiên được sử dụng trong sản xuất truyền hình, màn hình xanh dương vốn là chuẩn mực trong ngành công nghiệp điện ảnh đã bị sử dụng theo thói quen, cho đến khi những cân nhắc thực tế khác khiến ngành công nghiệp truyền hình chuyển từ màn hình xanh dương sang xanh lá.

Từ góc độ công nghệ, có thể sử dụng kênh màu xanh lam hoặc xanh lục như nhau, nhưng vì quần áo màu xanh dương là một thách thức đang diễn ra. Bởi vì quần áo màu xanh lá cây ít phổ biến hơn màu xanh lam. Ngoài ra, do mắt người nhạy cảm hơn với các bước sóng màu lục, nằm ở giữa phổ ánh sáng khả kiến, kênh video tương tự màu xanh lá cây thường mang nhiều cường độ tín hiệu hơn, cho tín hiệu tốt hơn về tỷ lệ nhiễu so với các kênh video thành phần khác, vì vậy các phím màn hình màu xanh lá cây có thể tạo ra Key sạch nhất. Vì vậy màn hình màu xanh lá cây được sử dụng phổ biến.

### Các yếu tố chính:

Yếu tố quan trọng nhất đối với Key là sự phân tách màu của tiền cảnh (chủ thể) và nền (màn hình) - một màn hình màu xanh dương sẽ được sử dụng nếu đối tượng chủ yếu là màu xanh lá cây (ví dụ như thực vật).

Trong [analog color TV](https://en.wikipedia.org/wiki/Analog_television),  màu được biểu thị bằng pha của [sóng mang con](https://en.wikipedia.org/wiki/Subcarrier" \l "Television" \o "Xe con) sắc độ so với bộ dao động tham chiếu. Phím Chroma đạt được bằng cách so sánh pha của video với pha tương ứng với màu được chọn trước. Các phần trong pha của video được thay thế bằng video nền thay thế.

Trong [digital color TV](https://en.wikipedia.org/wiki/Digital_television), màu sắc được thể hiện bằng ba số (mức cường độ đỏ, lục, lam). Phím Chroma đạt được bằng cách so sánh số đơn giản giữa video và màu được chọn trước. Nếu màu tại một điểm cụ thể trên màn hình khớp (chính xác hoặc trong một phạm vi), thì video tại điểm đó được thay thế bằng nền thay thế.

### Ánh sáng:

Để tạo ảo giác rằng các nhân vật và đối tượng được quay có mặt trong cảnh nền dự định, ánh sáng trong hai cảnh phải là một kết hợp hợp lý. Đối với các cảnh ngoài trời, những ngày u ám tạo ra ánh sáng khuếch tán, có màu sắc đồng đều, có thể dễ dàng phù hợp hơn trong studio, trong khi ánh sáng mặt trời trực tiếp cần phải được kết hợp theo cả hướng và màu tổng thể dựa trên thời gian trong ngày.

Một studio được chụp trước màn hình màu xanh lá cây sẽ tự nhiên có ánh sáng xung quanh cùng màu với màn hình, do sự tán xạ ánh sáng của nó. Hiệu ứng này được gọi là tràn. Điều này có thể trông không tự nhiên hoặc khiến các phần của các nhân vật biến mất, do đó phải được bù hoặc tránh bằng cách sử dụng màn hình lớn hơn đặt xa các diễn viên.



Hình 2: Sử dụng phông nền màu xanh dương (bluescreen)



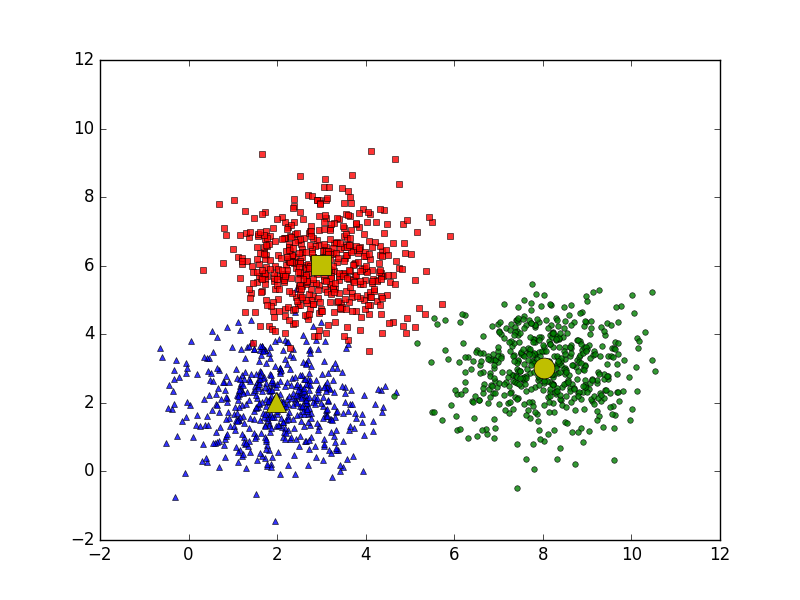
Hình 3: sử dụng phông nền màu xanh lá (greenscreen)

# Chương trình Chroma Key Visual Effect:

## Giới thiệu thuật toán K-means clustering:

Trong thuật toán K-means clustering ([Unsupervised learning](https://machinelearningcoban.com/2016/12/27/categories/#unsupervised-learning-hoc-khong-giam-sat)), chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thể nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau. [[2]](https://machinelearningcoban.com/2017/01/01/kmeans/?fbclid=IwAR1WVf7g3SqcBepN0rfewlvZOc99ArH7yxgm7CBq37fpX6Z9Nbib5bqyyhg)

Ý tưởng đơn giản nhất về cluster (cụm) là tập hợp các điểm ở gần nhau trong một không gian nào đó (không gian này có thể có rất nhiều chiều trong trường hợp thông tin về một điểm dữ liệu là rất lớn). Hình bên dưới là một ví dụ về 3 cụm dữ liệu (từ giờ tôi sẽ viết gọn là cluster).

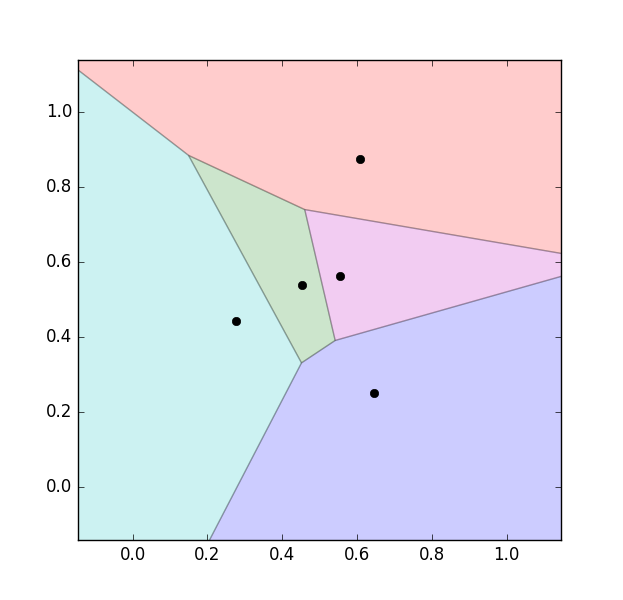


Hình 4: Bài toán với 3 clusters

Giả sử mỗi cluster có một điểm đại diện (center) màu vàng. Và những điểm xung quanh mỗi center thuộc vào cùng nhóm với center đó. Một cách đơn giản nhất, xét một điểm bất kỳ, ta xét xem điểm đó gần với center nào nhất thì nó thuộc về cùng nhóm với center đó.

Tới đây, chúng ta có một bài toán thú vị: Trên một vùng biển hình vuông lớn có ba đảo hình vuông, tam giác, và tròn màu vàng như hình trên. Một điểm trên biển được gọi là thuộc lãnh hải của một đảo nếu nó nằm gần đảo này hơn so với hai đảo kia. Hãy xác định ranh giới lãnh hải của các đảo.

Hình dưới đây là một hình minh họa cho việc phân chia lãnh hải nếu có 5 đảo khác nhau được biểu diễn bằng các hình tròn màu đen:



Hình 5: Phân vùng lãnh hải của mỗi đảo. Các vùng khác nhau có màu sắc khác nhau.

Chúng ta thấy rằng đường phân định giữa các lãnh hải là các đường thẳng (chính xác hơn thì chúng là các đường trung trực của các cặp điểm gần nhau). Vì vậy, lãnh hải của một đảo sẽ là một hình đa giác.

## Áp dụng thuật toán vào bài toán Chroma Key:

Xây dựng ứng dụng Chroma Key có:

* ***Đầu vào:*** 2 ảnh có cùng kích thước, ảnh nguồn và ảnh background.
* ***Đầu ra:*** ảnh trộn giữa ảnh nguồn và ảnh background, tại các vùng do người dùng lựa chọn.



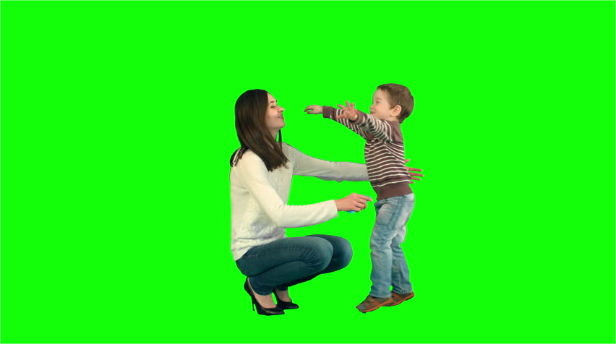
Hình 6: input và output sau khi sử dung chroma key.

Có 2 bước chính cần giải quyết:

* Tách vật thể (object segmentation) khỏi nền xanh.
* Trộn vật thể đã tách với ảnh background.

### Object segmentation (Tách vật thể):

Giả sử chúng ta có bức ảnh dưới đây và muốn một thuật toán tự động nhận ra vùng nhân vật trong ảnh và tách nó ra. [[3]](https://machinelearningcoban.com/2017/01/04/kmeans2/?fbclid=IwAR2DQAr9EADR6vrt7kleuDGlxyNNlLkkWVmMR7zagB8hsu2hV0z6iPDkFME)



Hình 7.1: Ảnh chưa sử dụng K-means.

Đây là một bức ảnh màu, mỗi điểm ảnh sẽ được biểu diễn bới 3 giá trị tương ứng với màu Red, Green, và Blue (mỗi giá trị này cũng là một số tự nhiên không vượt quá 255). Nếu ta coi mỗi điểm dữ liệu là một vector 3 chiều chứa các giá trị này, sau đó áp dụng thuật toán K-means clustering, chúng ta có thể có kết quả mong muốn.

Làm việc trên Python:

Khai báo thư viện và load bức ảnh:

**import** matplotlib.image **as** mpimg

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**from** sklearn.cluster **import** KMeans

img **=** mpimg**.**imread('girl3.jpg')

plt**.**imshow(img)

imgplot **=** plt**.**imshow(img)

plt**.**axis('off')

plt**.**show()

Biến đổi bức ảnh thành 1 ma trận mà mỗi hàng là 1 pixel với 3 giá trị màu:

X **=** img**.**reshape((img**.**shape[0]**\***img**.**shape[1], img**.**shape[2]))

Hàm train này sử dụng Kmeans với k = 2.

# Save the model with the center to facilitate the distance at the next time

def train(self, image):

X = image.reshape((image.shape[0]\*image.shape[1], image.shape[2]))

kmeans = KMeans(n\_clusters=2).fit(X)

pickle.dump(kmeans, open('model\_kmeans.sav', 'wb'))

áp dụng K-means để thực hiện quá trình tách đối tượng, được lưu dưới dạng file nhị phân

Loại bỏ màn hình xanh:

kmeans = pickle.load(open('Collage/model\_kmeans.sav', 'rb'))

X = image.reshape((image.shape[0]\*image.shape[1], image.shape[2]))

Y = kmeans.predict(X)

X[Y == 1] = 0

result = X.reshape((image.shape[0], image.shape[1], image.shape[2]))

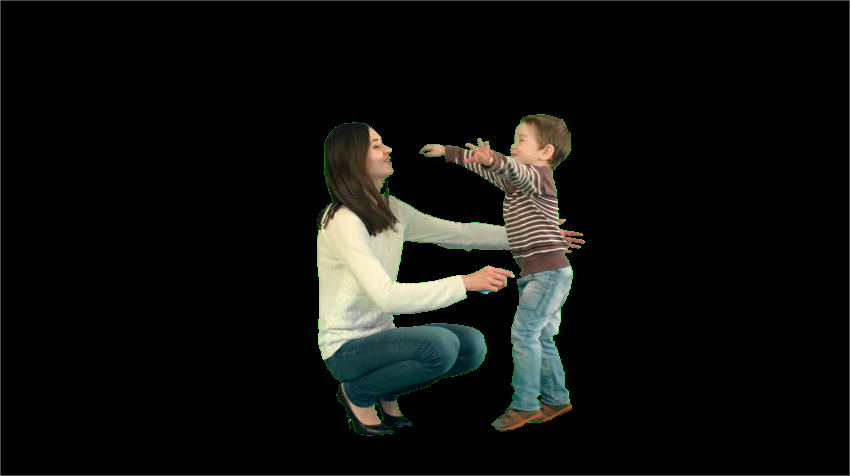
return result

Sau khi tìm được các cluster, tôi thay giá trị của mỗi pixel bằng center của cluster chứa nó, và được kết quả như sau:



Hình 7.2: Kết quả sau khi sử dụng K-means

Sau khi sử dụng K-means Clustering để tách đối tượng như hình 7.2. Ta bắt đầu xóa nền xanh và thu được kết quả như sau:



Hình 7.3: Kết quả sau khi xóa nền xanh lá (green).

Kết quả trên đã tách được đối tượng. Nhưng kết quả cho ra không tốt lắm nó vẫn còn tồn tại những viền xanh lá xung quanh.

Để loại bỏ những viền xanh còn sót lại ta làm như sau:

#remove noise.

if flag:

kmeans = pickle.load(open('Collage/model\_kmeans\_overfit.sav', 'rb'))

X = image.reshape((image.shape[0]\*image.shape[1], image.shape[2]))

Y = kmeans.predict(X)

# the jamming labels

X[Y == 18] = 0

X[Y == 33] = 0

X[Y == 47] = 0

X[Y == 51] = 0

X[Y == 54] = 0

X[Y == 61] = 0

X[Y == 64] = 0

X[Y == 76] = 0

X[Y == 78] = 0

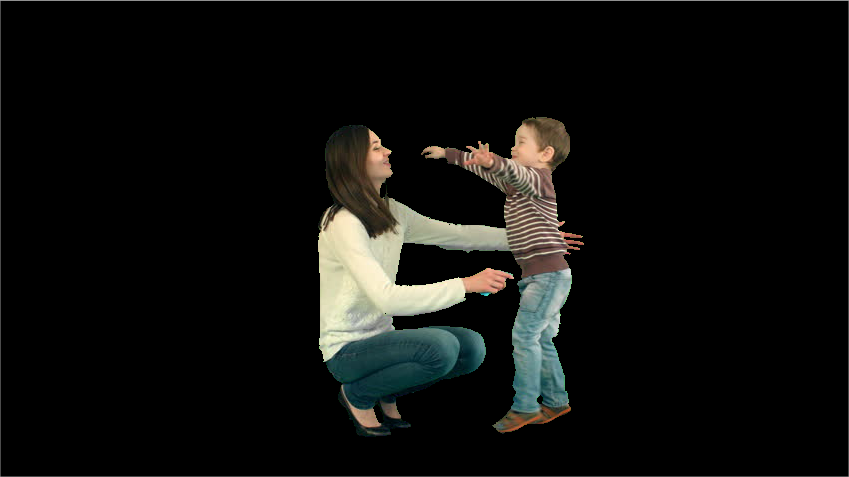
X[Y == 91] = 0

X[Y == 98] = 0

result = X.reshape((image.shape[0], image.shape[1], image.shape[2]))

return result

sau khi lại bỏ noise green (các viền xanh lá còn xót lại) ta thu được kết quả như sau:



Hình 7.4: Kết quả sau khi loại bỏ noise xanh lá

### Blend image (trộn vật thể tách với nền):

Để trộn 2 ảnh lại với nhau ta cần phải tạo ra ảnh mask giống như hình 8. [[4]](https://medium.com/fnplus/blue-or-green-screen-effect-with-open-cv-chroma-keying-94d4a6ab2743)

Mask là cách rất phổ biến để cô lập a selected area of intrest. Chúng ta có thể tạo mặt nạ trên vùng màu xanh bằng inRange() chức năng Open CV.

Đoạn code dưới đây sẽ thực thực hiện điều đó và trộn 2 ảnh object và background với nhau:

# combine 2 photos

mask = cv2.inRange(remove\_overfit, np.array([0,0,0]), np.array([10,10,10]))

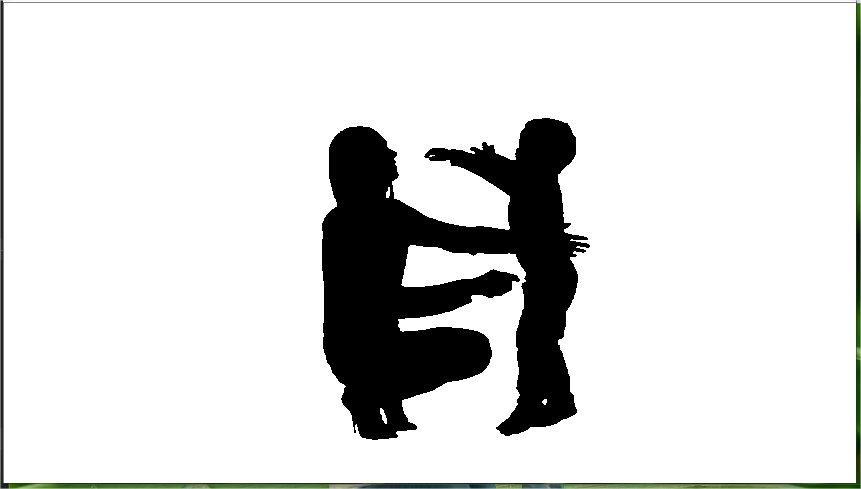
mask\_inv = cv2.bitwise\_not(mask)

bg = cv2.bitwise\_and(self.background\_image,self.background\_image, mask=mask)

fg = cv2.bitwise\_and(self.objects\_image, self.objects\_image, mask=mask\_inv)

final = bg + fg

Đây là kết quả ảnh mask được tạo ra:



Hình 8: Tạo ảnh mask

Toàn bộ khu vực màu trắng là nơi hình ảnh sẽ được phép hiển thị xuyên qua và màu đen sẽ bị chặn.

Mỗi tọa độ trong mặt nạ có giá trị 255 cho màu trắng và 0 cho màu đen, giống như hình ảnh thang độ xám.

Chúng ta cần phải chọn vùng để trộn nhân vật tách với ảnh background. Để làm điều này ta sẽ sử dụng bitwise\_and trong openCV với mask = mask (hình 8).

bg = cv2.bitwise\_and(self.background\_image,self.background\_image, mask=mask)

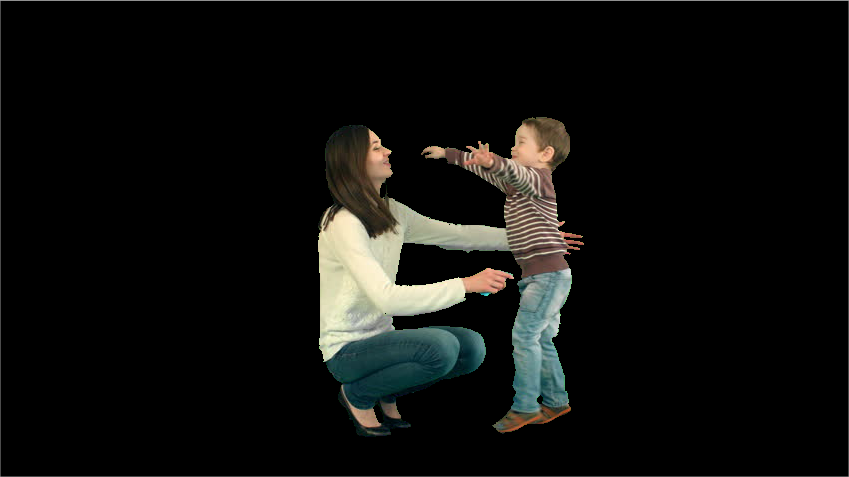


Hình 9: Kết quả trộn ảnh mask và background.

Bây giờ, chúng ta chỉ còn một bước cuối cùng là cộng 2 hình ảnh lại với nhau.

Vì vùng màu đen tương đương với số không trong giá trị màu pixel, nên một phép cộng đơn giản sẽ hoạt động.

Kết quả Hình 7.4 và Hình 9 sau khi cộng lại.



Hình 7.4. Hình 9.

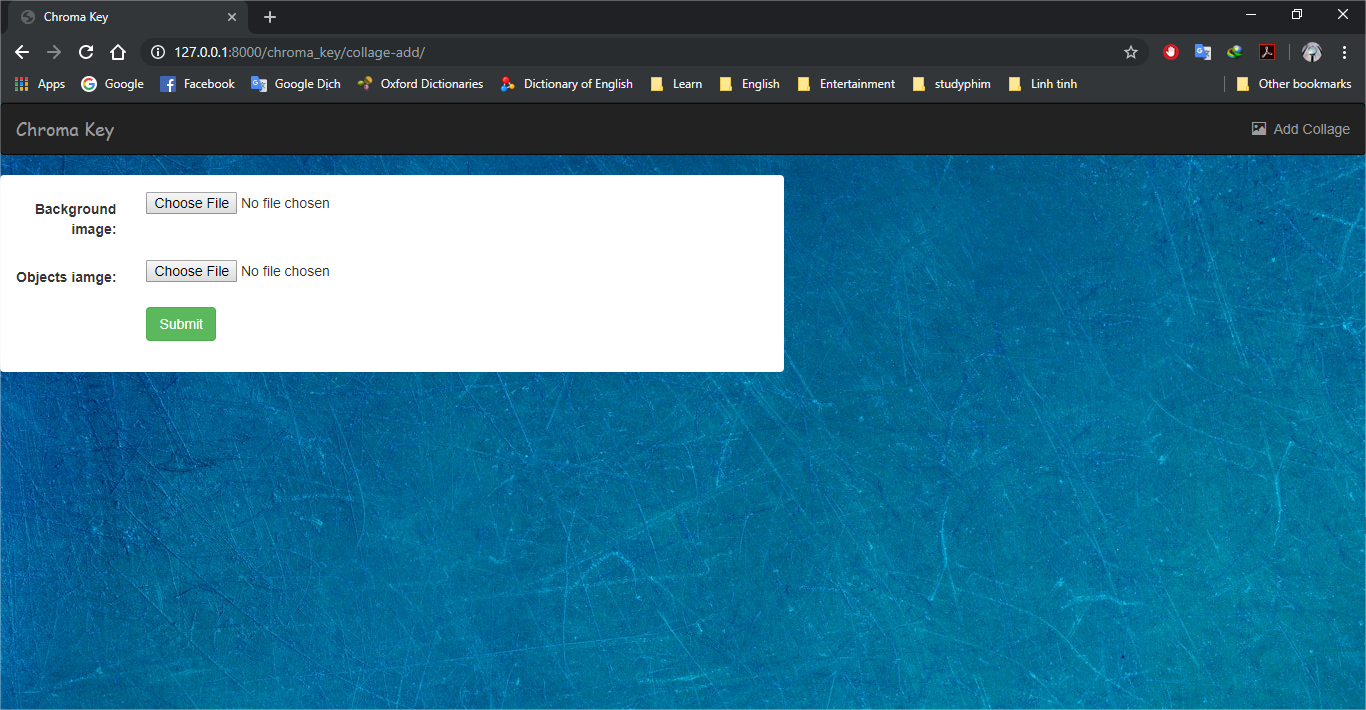
Và cuối cùng chúng ta thu được hình ảnh như dưới đây:



Hình 10: Kết quả cuối cùng sau khi trộn 2 ảnh lại.

# Giao diện chương trình:

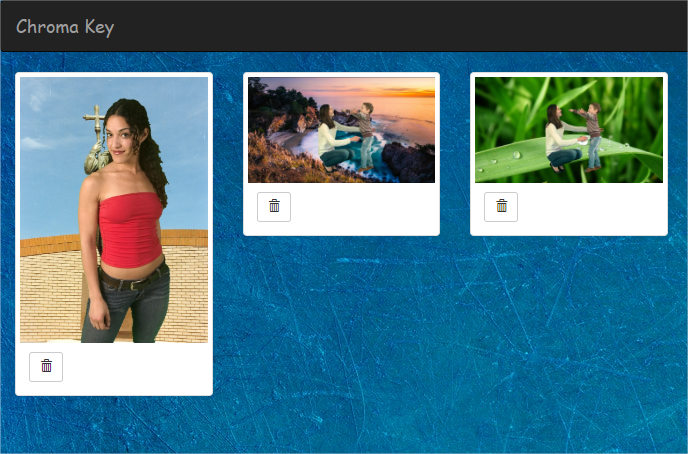
Chúng ta sẽ xây dựng giao diện chương trình trên nên tảng web sử dụng Django. [[5]](https://simpleisbetterthancomplex.com/series/beginners-guide/1.11/?fbclid=IwAR2Szf35eQQ24SMNWCR9AplfSUK9fQBPMRj9PBWrJmboTFiaFVT8s66NaV0)



Hình 11.1: Giao diện ứng dụng trên web.

Chỉ cần chọn 2 hình ảnh có cùng kích thước.

Chọn Submit. Hình ảnh sẽ tự lưu vào máy.



Hình 11.2: Ví dụ ba hình ảnh sau khi sử dụng Chroma Key

Khi click vào hình nó hiện thị 2 input ra kết quả.



Hình 11.3: Thông tin ảnh đã chọn.

***Tài liệu tham khảo:***

[1]<https://en.wikipedia.org/wiki/Chroma_key>

[2]<https://machinelearningcoban.com/2017/01/01/kmeans/?fbclid=IwAR1WVf7g3SqcBepN0rfewlvZOc99ArH7yxgm7CBq37fpX6Z9Nbib5bqyyhg>

[3]<https://machinelearningcoban.com/2017/01/01/kmeans/?fbclid=IwAR1WVf7g3SqcBepN0rfewlvZOc99ArH7yxgm7CBq37fpX6Z9Nbib5bqyyhg>

[4]<https://medium.com/fnplus/blue-or-green-screen-effect-with-open-cv-chroma-keying-94d4a6ab2743>

[5]<https://simpleisbetterthancomplex.com/series/beginners-guide/1.11/?fbclid=IwAR2Szf35eQQ24SMNWCR9AplfSUK9fQBPMRj9PBWrJmboTFiaFVT8s66NaV0>