

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



KHOA KHOA HỌC MÁY TÍNH

BÁO CÁO ĐỒ ÁN MÔN
NHẬP MÔN THỊ GIÁC MÁY TÍNH

Makeup Application



Lớp CS231.K11.KHCL
GVHD: TS. Nguyễn Vinh Tiệp
Nhóm thực hiện:
Đặng Quốc Quy - 17520960
Đặng Hoàng Sang - 17520967
Trần Vũ Hoàng Tú - 17521209

Thành phố Hồ Chí Minh, tháng 11 năm 2019

Mục lục

Mục lục	2
1. Thông tin chung.....	3
2. Mô tả bài toán:	3
3. Khó khăn, hướng phát triển.....	3
3.1 Khó khăn	3
3.2 Hướng phát triển.....	4
4. Tiến hành thực nghiệm	4
4.1 Chuyển ảnh về ảnh mức xám	4
4.2 Xác định các bộ phận trên gương mặt.....	4
4.3 Tô son môi	5
4.4 Kẻ Eyeliner	7
4.5 Kẻ lông mày	8
4.6 Nhuộm tóc	8
4.6.1 Giới thiệu về Keras:.....	8
4.6.2 Giới thiệu MobileNet	9
4.6.3 Fully Convolutional MobileNet cho Hair Segmentation	10
4.6.4 Hàm cv2.addWeighted trong OpenCV	11
4.6.5 Thay đổi màu tóc.....	11
5. Kết quả thực nghiệm.....	12
6. Hướng dẫn sử dụng giao diện.....	13
7. Cải tiến so với lần báo cáo trước	16
8. Source code.....	16
9. Tài liệu tham khảo	16

1. Thông tin chung

- GVHD: Nguyễn Vinh Tiệp
- Môn học: Nhập môn thị giác máy tính
- Nhóm 1
- Đề tài: Virtual makeup
- Danh sách thành viên nhóm

MSSV	HỌ TÊN
17520960	Đặng Quốc Quy
17520967	Đặng Hoàng Sang
17521209	Trần Vũ Hoàng Tú

2. Mô tả bài toán:

Ngày nay nhu cầu của con người không chỉ dừng lại ở việc “ăn cho no, mặc cho ấm” nữa mà là “ăn cho ngon, mặc cho đẹp”. Về bề ngoài ngày càng chiếm vị thế quan trọng trong việc nhìn người. Vì vậy, không chỉ riêng chị em phụ nữ mà cả nam giới cũng dành nhiều thời gian cho việc chăm sóc ngoại hình hơn đặc biệt trong lĩnh vực trang điểm. Thế nhưng việc lựa chọn mỹ phẩm hay thay đổi ngoại hình thế nào sao cho phù hợp với mình thì lại rất khó, công việc này càng khó hơn khi mục đích là tư vấn khách hàng hay mua quà tặng.

Nắm bắt như cầu đó, trong phạm vi môn học này, nhóm đề xuất làm đề tài nghiên cứu xây dựng ứng dụng trang điểm ảo. Với đầu vào là một bức ảnh, đầu ra là bức ảnh đã được trang điểm theo ý người dùng.

Bài báo cáo được trình bày theo nội dung như sau:

3. Khó khăn, hướng phát triển

3.1 Khó khăn

- Trang điểm có quá nhiều bước
- Trang điểm sao cho nhìn tự nhiên

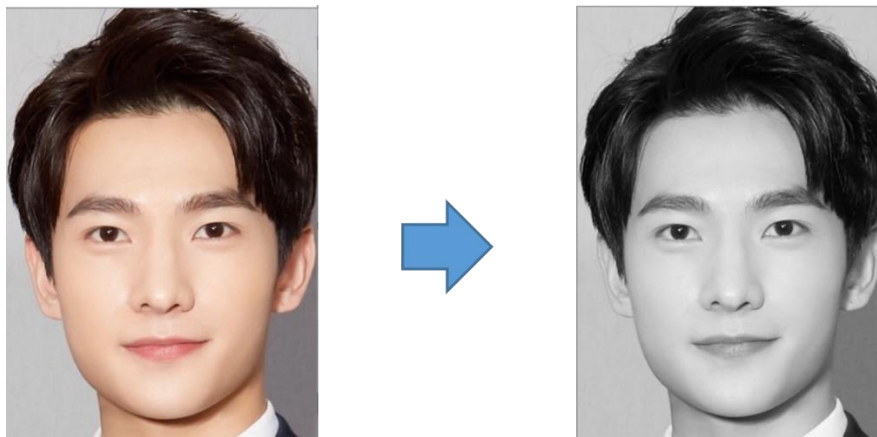
- Khó khăn trong việc xác định các điểm trên gương mặt

3.2 Hướng phát triển

- Thay đổi màu da
- Đánh phấn mắt
- Đánh má hồng
- Thay đổi hình dáng gương mặt
- Thay đổi màu tóc cho tự nhiên hơn
- Cho phép nhập vào màu mong muốn
- App makeup với nhiều filter
- Xây dựng app makeup có thể sử dụng trên web
- Dùng docker để có thể cài đặt trên nhiều hệ điều hành khác nhau

4. Tiến hành thực nghiệm

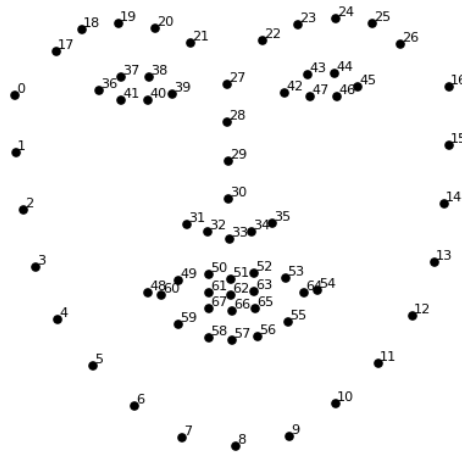
4.1 Chuyển ảnh về ảnh mức xám



Hình 1 Chuyển sang ảnh mức xám

4.2 Xác định các bộ phận trên gương mặt

Sử dụng model Landmark 68 đã được train sẵn để xác định được các bộ phận trên gương mặt.



Hình 2 Face Landmark

0 → 16: cằm

17 → 21: Lông mày trái

22 → 26: Lông mày phải

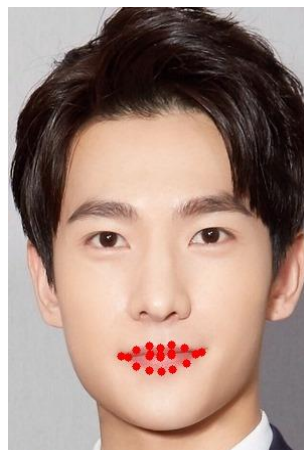
27 → 35: Mũi

36 → 41: Mắt trái

42 → 47: Mắt phải

48 → 68: Miệng

4.3 Tô son môi



Hình 3 Landmark môi

Bước 1: Xác định Landmark vùng môi

Bước 2: Vẽ đường cong theo các điểm vừa xác định được bằng cách sử dụng hàm nội suy interpolated được hỗ trợ trong thư viện numpy.

Bước 3: Phát sinh ngẫu nhiên các điểm nằm trong vùng đã được bao quanh ở bước 2 (các điểm của vùng môi)

Bước 4: Gán màu cho các điểm ở trên theo màu môi mình mong muốn.



Hình 4 Tô son

Sau khi hoàn thành bước 4, về cơ bản việc tô son đã hoàn thành. Tuy nhiên, màu son quá đậm, nhìn không tự nhiên nên chúng ta sẽ làm thêm các bước để làm màu son trông tự nhiên hơn.

Bước 5: Tạo mặt nạ môi

Bước 6: Sử dụng bộ lọc Gaussian để làm mờ mặt nạ. Làm mờ theo hiệu ứng 3D, ở ngoài nhợt và đậm dần khi vào trong lòng môi.



Hình 5 Mặt nạ môi trước và sau khi sử dụng bộ lọc Gaussian

Bước 7: Đè mặt nạ môi lên hình ảnh tạo được sau khi thực hiện bước 4.



Hình 6 Ảnh trước và sau khi làm mượt màu môi

4.4 Kẻ Eyeliner

Bước 1: Xác định eyes landmark



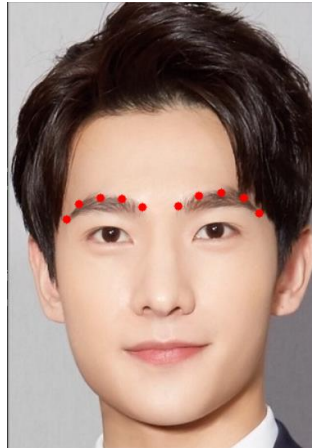
Bước 2,3,4 giống như mục 4.3 Tô son môi.



Hình 7 Sau khi kẻ eyeliner

4.5 Kẻ lông mày

Bước 1: Xác định landmark của lông mày



Bước 2,3,4 giống như mục 4.3 Tô son môi.

4.6 Nhuộm tóc

4.6.1 Giới thiệu về Keras:

Keras là một thư viện được phát triển vào năm 2015 bởi François Chollet, là một kỹ sư nghiên cứu Deep Learning tại Google. Nó là một open source cho neural network được viết bởi ngôn ngữ python. Keras là một API bậc cao có thể sử dụng chung với các thư viện Deep Learning nổi tiếng như TensorFlow (được phát triển bởi Google), CNTK (được phát triển bởi Microsoft), Theano (người phát triển chính Yoshua Bengio). Keras được coi là một thư viện ‘high-level’ với phần ‘low-level’ (còn được gọi là back-end) có thể là TensorFlow, CNTK, hoặc Theano. Keras có cú pháp đơn giản hơn TensorFlow rất nhiều. Keras có một số ưu điểm như:

- Dễ sử dụng, xây dựng model nhanh.
- Có thể hoạt động trên cả CPU và GPU.
- Hỗ trợ xây dựng CNN, RNN và có thể kết hợp cả 2.
- Hỗ trợ tính toán với GPU và các hệ thống phân tán. Điều này là tối quan trọng vì việc huấn luyện các mô hình Deep Learning yêu cầu khả năng tính toán rất mạnh.
- Hỗ trợ các ngôn ngữ lập trình phổ biến: C/C++, Python, Java, R,....

- Có thể chạy được trên nhiều hệ điều hành.
- Thời gian từ ý tưởng tới xây dựng và huấn luyện mô hình ngắn.
- Có thể chạy trên trình duyệt và các thiết bị di động.
- Có khả năng giúp người lập trình can thiệp sâu vào mô hình và tạo ra các mô hình phức tạp.
- Chứa nhiều model zoo, tức các mô hình Deep Learning thông dụng đã được huấn luyện.
- Hỗ trợ tính toán backpropagation tự động.
- Có cộng đồng hỏi đáp lớn.

Keras hoạt động tốt trên Python phiên bản 3.6

4.6.2 Giới thiệu MobileNet

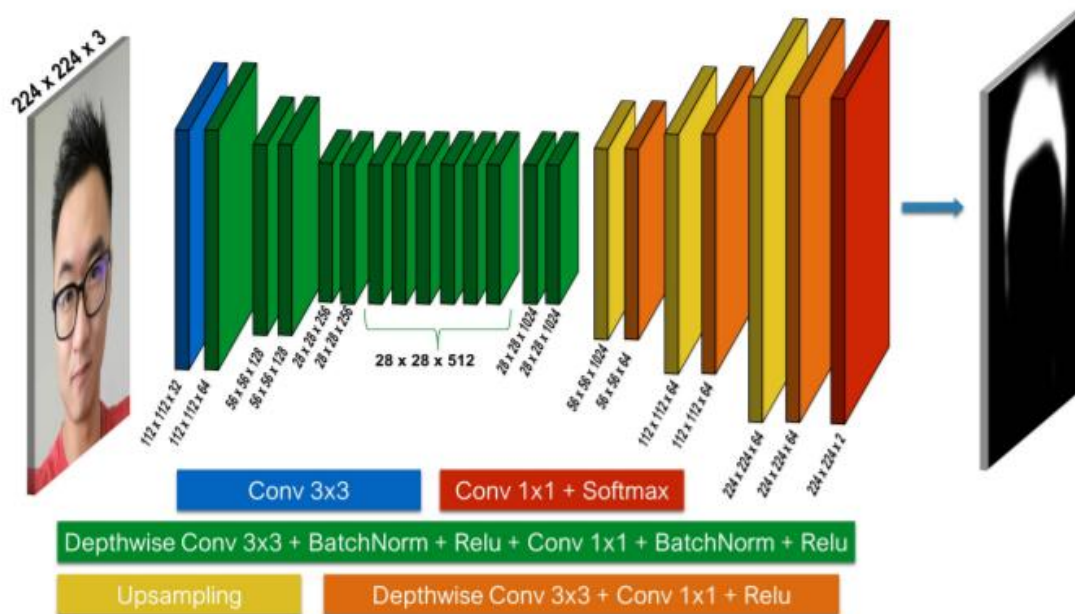
- Các mô hình pre-trained CNN như: VGG-16, ResNet, InceptionNet, InceptionResNet,... tuy có độ chính xác cao, nhưng chúng đều có một điểm hạn chế chung đó là không phù hợp với các ứng dụng trên mobile hay các hệ thống nhúng có khả năng tính toán thấp. Nếu muốn deploy các mô hình trên cho các ứng dụng real time, ta cần phải có cấu hình cực kì mạnh mẽ (GPU / TPU) còn đối với các hệ thống nhúng (Raspberry Pi, Nano pc, etc) hay các ứng dụng chạy trên smart phone, ta cần có một mô hình "nhẹ" hơn.
- Yếu tố chính giúp MobileNet có được độ chính xác cao trong khi thời gian tính toán thấp nằm ở sự cải tiến Conv layer bình thường. Trong MobileNet có 2 Conv layer được sử dụng là: SeparableConv và DepthwiseConv. Thay vì thực hiện phép tích chập như thông thường, SeparableConv sẽ tiến hành phép tích chập depthwise spatial, sau đó là phép tích chập pointwise. Còn DepthwiseConv sẽ chỉ thực hiện phép tích chập depthwise spatial (không tính pointwise convolution). Việc chia phép tích chập ra như vậy giúp giảm đáng kể khối lượng tính toán và số lượng tham số của mạng. Với sự thay đổi này, MobileNet có thể hoạt động một cách mượt mà ngay cả trên phần cứng cấu hình thấp.
- Ta có thể cài đặt MobileNet với thư viện Keras
- Ưu điểm Mobilenet:
 - + Giảm thời gian huấn luyện mô hình

- + Giảm tham số, giảm số lượng phép tính toán
- + Hiệu quả trên các thiết bị nhúng, smartphone,....
- Hạn chế: Độ chính xác không cao như các pre-trained model khác

4.6.3 Fully Convolutional MobileNet cho Hair Segmentation

Mô hình mạng Fully convolutional MobileNet được tạo ra dựa trên mô hình MobileNet gốc để phù hợp cho việc phân đoạn tóc bằng cách:

- Bỏ 3 lớp cuối cùng trong mô hình MobileNet: Avg Pool, FC, và Softmax
 - Tăng độ phân giải của Output feature bằng cách thay đổi step size của 2 layer cuối từ 2 thành 1.
 - Kernel cho các lớp tăng thêm (kernel 2 tăng lên thành 4, kernel 4 tăng lên thành 8). Điều này mang lại độ phân giải cuối cùng là 28×28 .
 - Xây dựng bộ giải mã trên CNN features dựa trên tính năng như đầu vào và upsample chúng lên thành các Hair Mark với độ phân giải 224×224 .
 - Ở mỗi giai đoạn, chúng ta lấy mẫu của lớp trước theo một yếu tố bằng 2 bằng cách sao chép từng pixel trong vùng lân cận 2×2 .
- Sau đó, áp dụng phép tích chập theo chiều sâu, theo sau là các điểm tương ứng 1×1 với 64 bộ lọc, sau đó là lớp ReLu.



Hình 8 Fully Convolutional MobileNet cho Hair Segmentation

4.6.4 Hàm cv2.addWeighted trong OpenCV

- Hàm cv2.addWeighted dùng để trộn 2 bức ảnh lại với nhau bằng cách tính tổng trọng số trên 2 ảnh.

- Hàm cv2.addWeighted trên python để trộn hai ảnh:

`cv2.addWeighted(src1, alpha, src2, beta, gamma)`

Trong đó:

+ Src1, src2: Mảng biểu diễn cho 2 ảnh dạng ma trận

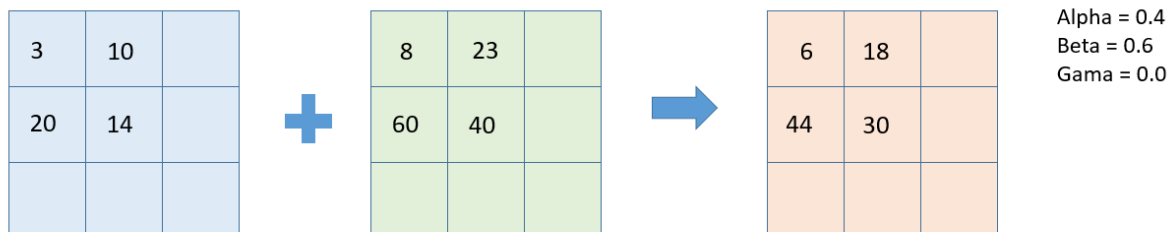
+ Alpha, beta: Lần lượt là trọng số cho ảnh một và ảnh hai. Nếu trọng số ảnh nào càng lớn thì khi trộn 2 ảnh, ảnh có trọng số lớn hơn sẽ thể hiện rõ hơn và ngược lại. ($\text{Alpha} + \text{Beta} = 1$)

+ Gamma: scalar được thêm vào mỗi tổng (thường dùng 0.0)

- Khi dùng cv2.addWeighted để trộn 2 ảnh (bắt buộc cùng kích thước), ảnh mới sẽ được tạo ra bằng cách: mỗi điểm ảnh mới sẽ bằng tổng của 2 điểm tương ứng của ảnh 1 và 2, theo công thức:

$\text{Ảnh mới} = (\text{ảnh 1}) * \alpha + (\text{ảnh 2}) * \beta + \gamma$

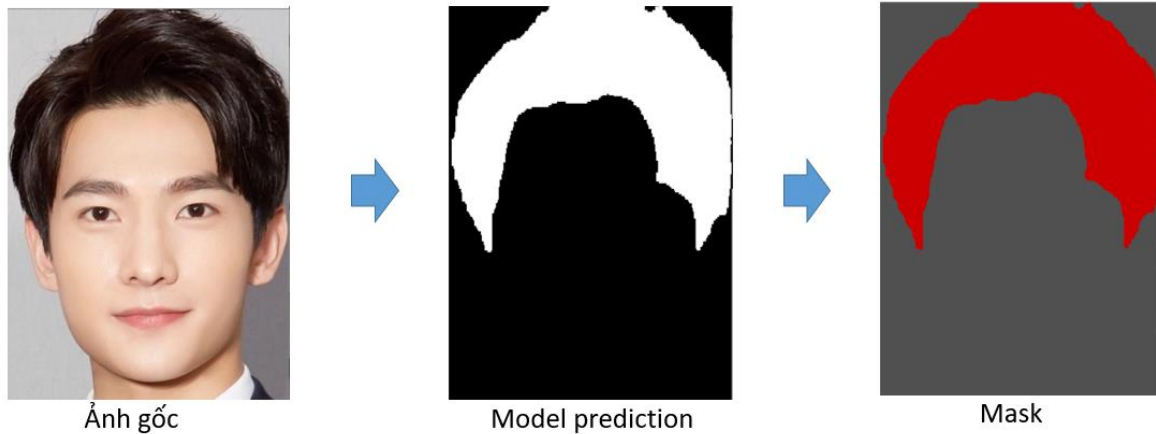
- Đối với ảnh màu, ta thực hiện độc lập cho từng kênh màu



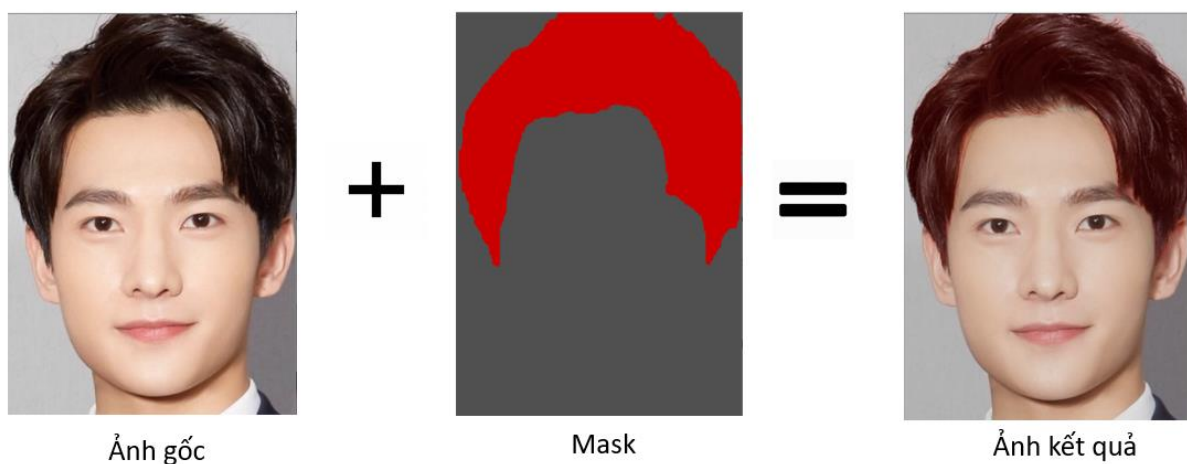
4.6.5 Thay đổi màu tóc

B1: Sử dụng mạng Fully Convolutional MobileNet để huấn luyện HairSegNet model

B2: Sử dụng HairSegNet Model để dự đoán phần tóc, tạo 1 Hair mark .



B3: Sử dụng `cv2.addWeighted` để blend Ảnh và Hair mark



5. Kết quả thực nghiệm

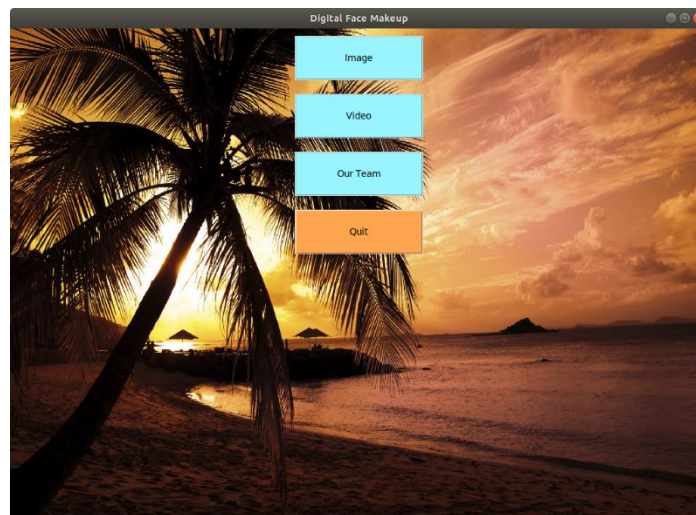
Nhóm đã cơ bản hoàn thành các bước trang điểm:

- Tô son
- Nhuộm tóc
- Kẻ eyeliner
- Kẻ lông mày



Hình 9 Trước và sau trang điểm

6. Hướng dẫn sử dụng giao diện

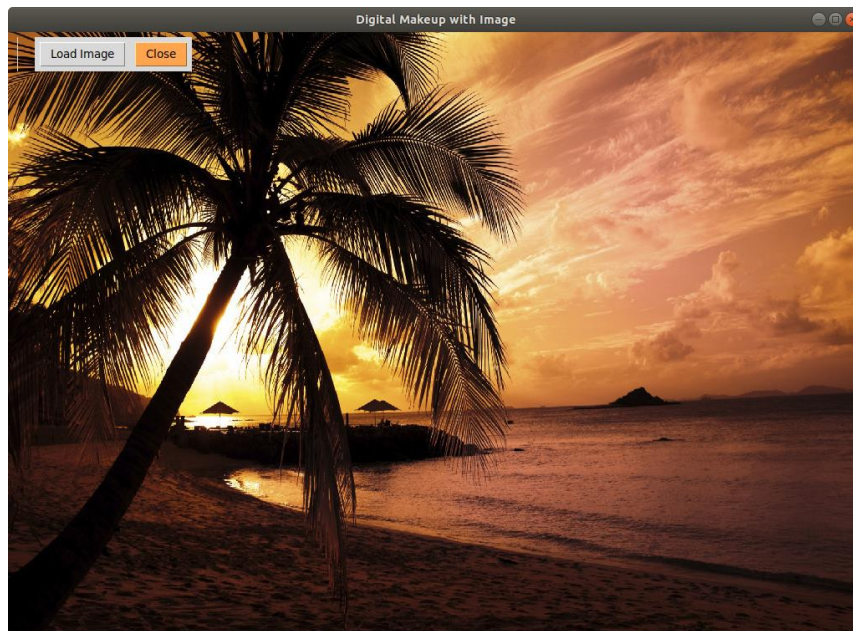


Hình 10 Giao diện chính

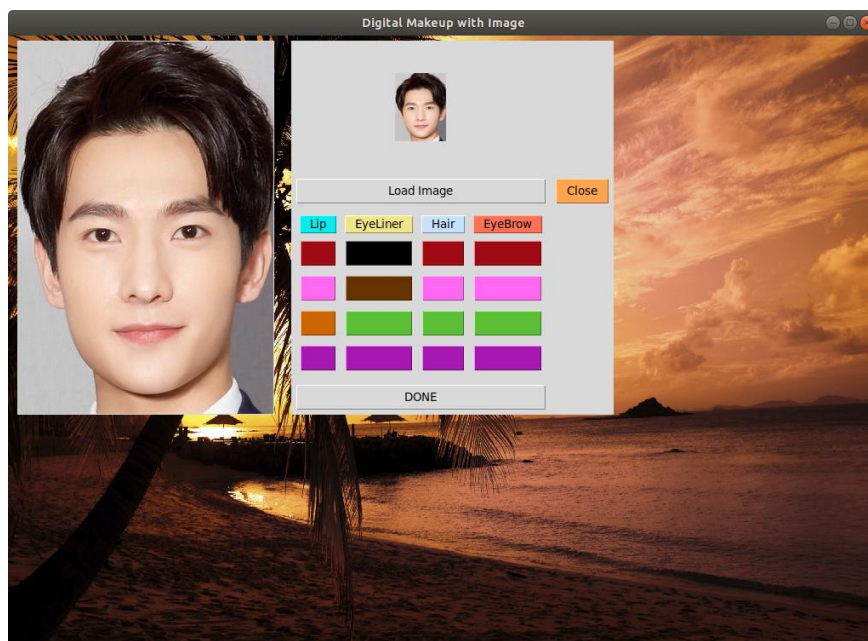
Image	Makeup trên ảnh
Video	Makeup trực tiếp trên camera
Our team	Giới thiệu về nhóm
Quit	Thoát chương trình

1. Image

- Chọn button Image



- Chọn Load image để mở một ảnh và makeup trên ảnh

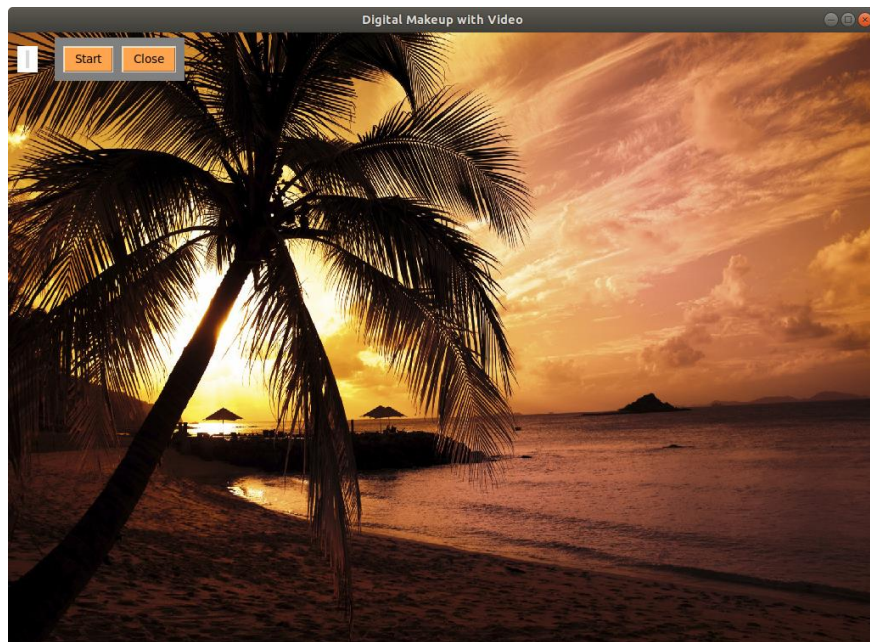


Tên	Chức năng
Lip	Tô son bằng cách chọn màu bên dưới
Eyeline	Kẻ mắt bằng cách chọn màu bên dưới
Hair	Đổi màu tóc bằng cách chọn màu bên dưới
Eyebrow	Kẻ lông mày bằng cách chọn màu bên dưới

- Chọn Close để thoát

2. Video

- Chọn Video



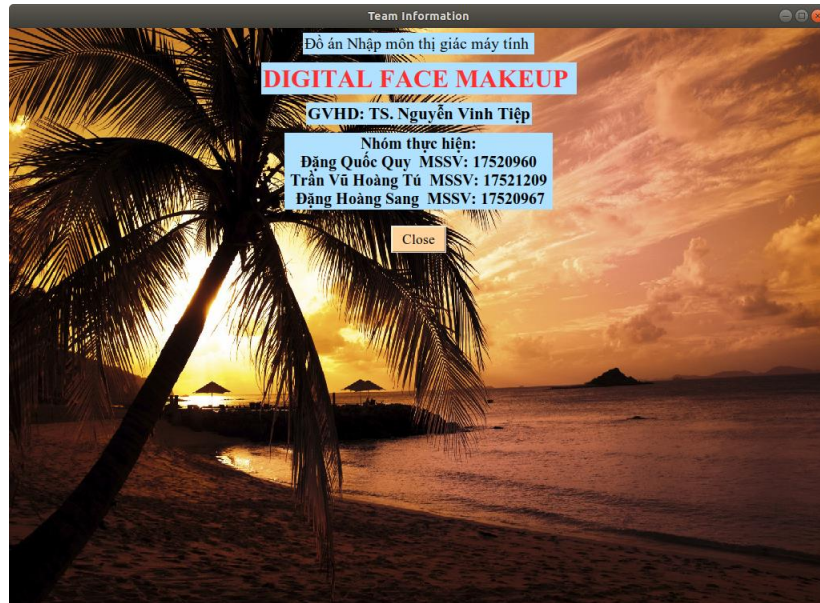
- Chọn Start để bắt đầu



Tên	Chức năng
Lip	Tô son bằng cách chọn màu bên dưới
Eyeliner	Kẻ mắt bằng cách chọn màu bên dưới

- Chọn Done để kết thúc

3. Our team



Hình 11 Our team

7. Cải tiến so với lần báo cáo trước

- Tăng tốc độ xử lý với tính năng tóc và chân mày nhanh hơn
- Trình bày code theo hướng đối tượng
-

8. Source code

Sour code : <https://github.com/quyvsquy/cs231doAn>

9. Tài liệu tham khảo

[1] <https://github.com/srivatsan-ramesh/Virtual-Makeup>

[2] <https://github.com/thangtran480/hair-segmentation>