You Are Beautiful

{Python Project}

Team : URBeautiful
Members : Tiffany Yen, Luhan Shen, Yueping Li, Xiaoyue Shen

You Are Beautiful is a project that combines:

- Applicability in Real Life
- Research into deeper python functions

We want to create a simple, beauty camera application that is easy to use

Abstract

Why We Choose This Topic

According to Zhicha Big Data, Beauty & Filter Camera users have exceeded **100 million** people since 2017

Rare to see an unedited picture

Decorated pictures are a MUST now!

People are always taking pictures

- Vacation
- Coffee Dates
- Casual Outing

Our group is all girls

What do we all love?

Picture

Picture

Picture

Introduction to Libraries

Recognize and manipulate the faces

99.38% Accuracy

Python Imaging Library = displays, process, and archives images

Strong focus in real-time application

It’s FREE

Face and Facial Feature Recognition

- Prepare the library

```
import face_recognition
```

- Load the file into a numpy array

```
image = face_recognition.load_image_file("picture.jpg")
```

- Get locations and outlines of facial features. Store the values into a dictionary

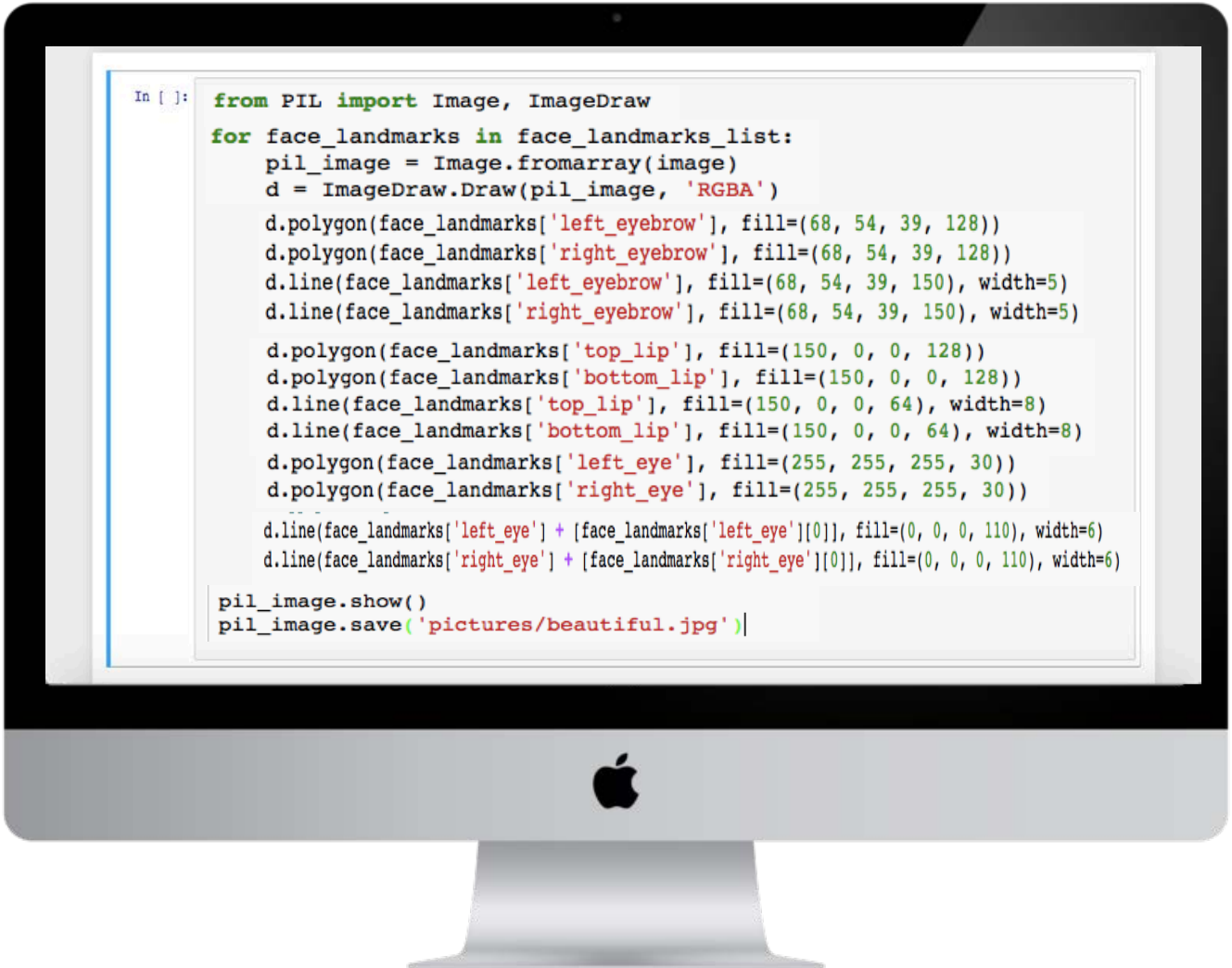
```
face_landmarks_list = face_recognition.face_landmarks(image)
```

Makeup (Still Image)

- Prepare the Library using PIL modules
- Use for-loop to beautify faces
- Use polygon function and line function to draw the eyebrows, gloss the lips, sparkle the eyes and apply the eyeliners.
- Display and save image

Makeup (Still Image)

CODE



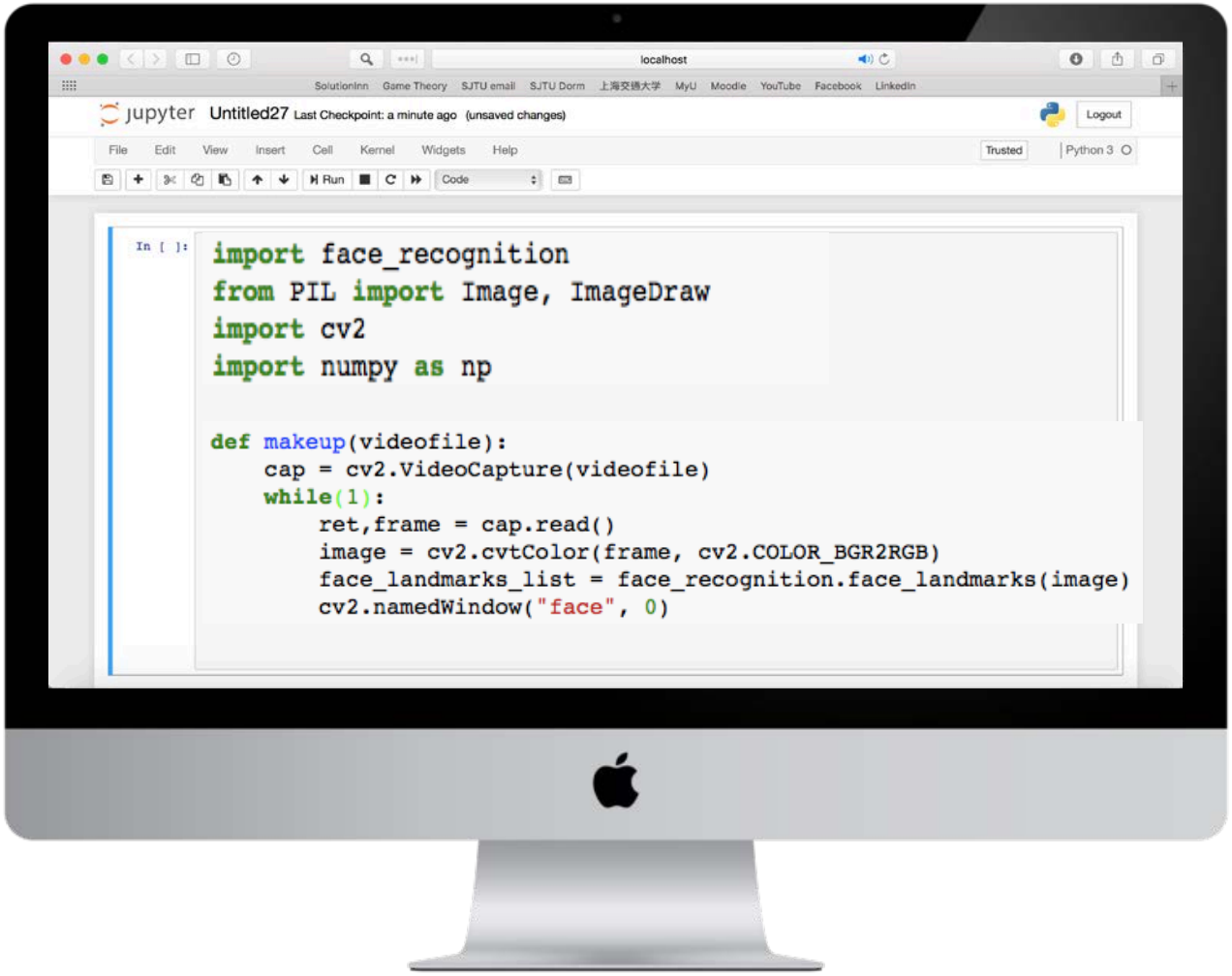
Makeup (examples)

Makeup (Moving Image)

- Prepare the Libraries
- Split and read the video using while-loop
- Apply makeup (same as still image)
- Present the outcome in video format
- Execute the functions defined

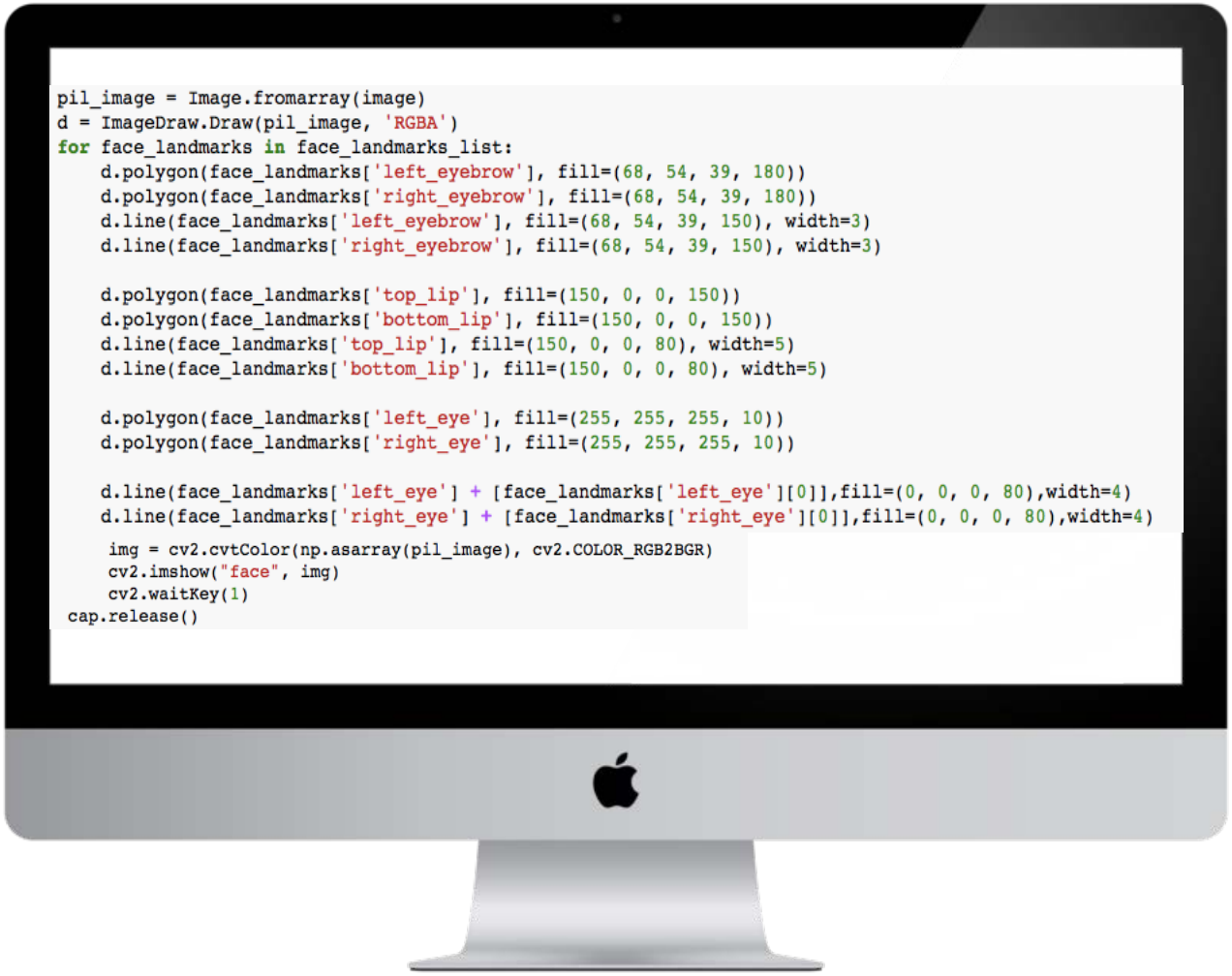
Makeup (Moving Image)

CODE



Makeup (Moving Image)

CODE



```
pil_image = Image.fromarray(image)
d = ImageDraw.Draw(pil_image, 'RGBA')
for face_landmarks in face_landmarks_list:
    d.polygon(face_landmarks['left_eyebrow'], fill=(68, 54, 39, 180))
    d.polygon(face_landmarks['right_eyebrow'], fill=(68, 54, 39, 180))
    d.line(face_landmarks['left_eyebrow'], fill=(68, 54, 39, 150), width=3)
    d.line(face_landmarks['right_eyebrow'], fill=(68, 54, 39, 150), width=3)

    d.polygon(face_landmarks['top_lip'], fill=(150, 0, 0, 150))
    d.polygon(face_landmarks['bottom_lip'], fill=(150, 0, 0, 150))
    d.line(face_landmarks['top_lip'], fill=(150, 0, 0, 80), width=5)
    d.line(face_landmarks['bottom_lip'], fill=(150, 0, 0, 80), width=5)

    d.polygon(face_landmarks['left_eye'], fill=(255, 255, 255, 10))
    d.polygon(face_landmarks['right_eye'], fill=(255, 255, 255, 10))

    d.line(face_landmarks['left_eye'] + [face_landmarks['left_eye'][0]],fill=(0, 0, 0, 80),width=4)
    d.line(face_landmarks['right_eye'] + [face_landmarks['right_eye'][0]],fill=(0, 0, 0, 80),width=4)
    img = cv2.cvtColor(np.asarray(pil_image), cv2.COLOR_RGB2BGR)
    cv2.imshow("face", img)
    cv2.waitKey(1)
cap.release()
```

Makeup (examples)

```
makeup('video.mp4')
```

Makeup (examples)

```
makeup('video.mp4')
```

Stickers

Stickers will make the images more interesting and cute. Two types of sticker function that we have included are:

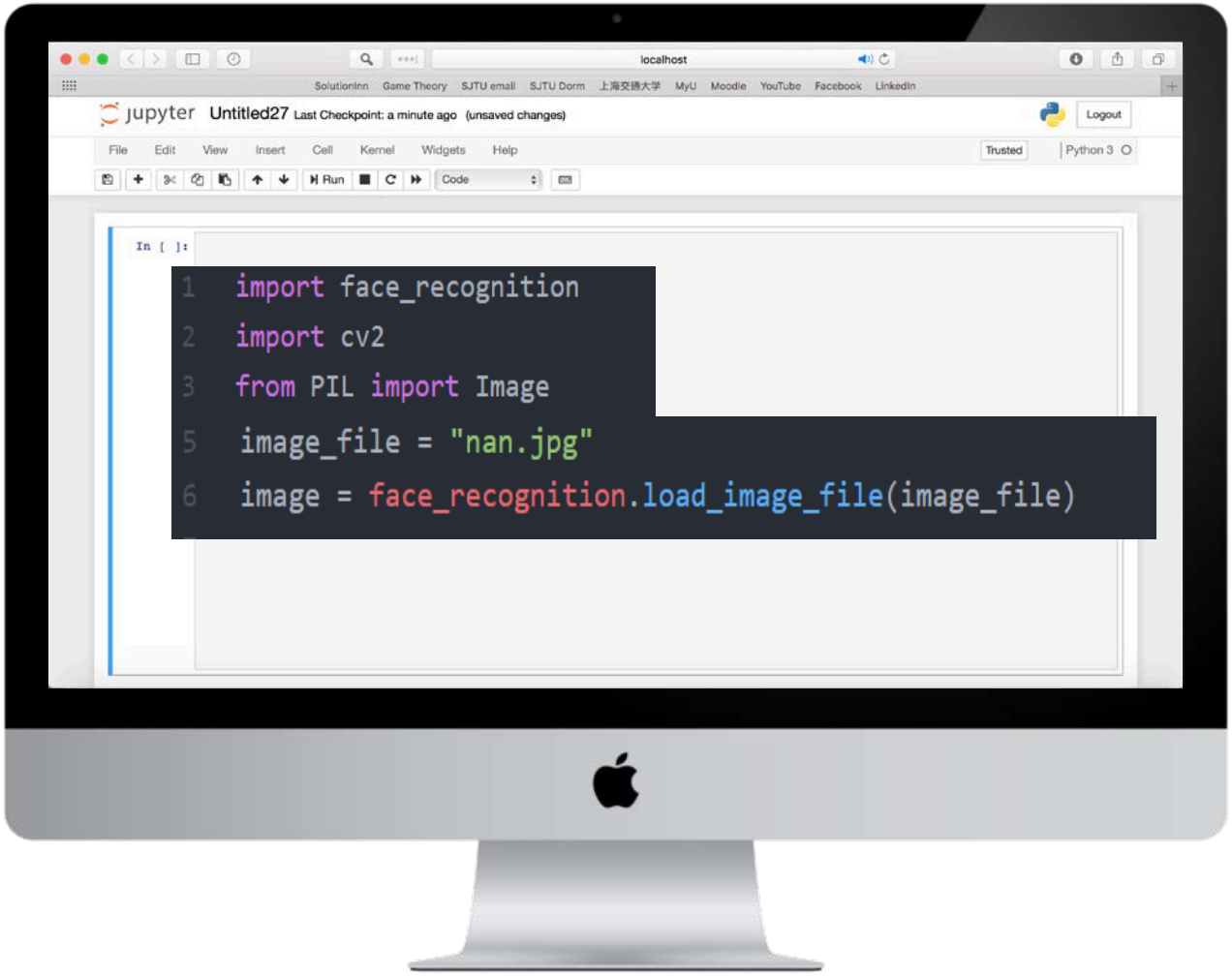
- Slogan
- Filter (similar to Snapchat's stickers)

How Slogan Stickers are Made

- Prepare the Library
- Find face locations in the picture
- Read, open , and detect the size of the picture
- Resize the slogan to fit the background image
- Past the stickers onto the picture

1.Prepare the library (face_recogniti on&PIL)

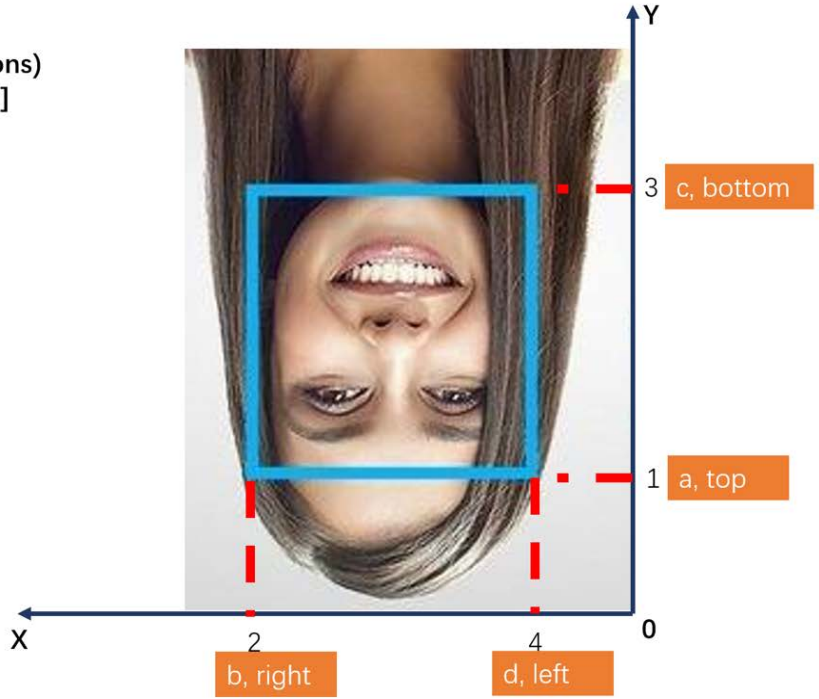
2. open the image with the face



3. Get the location of the face

print(face_locations)
Output: [(1,2,3,4)]

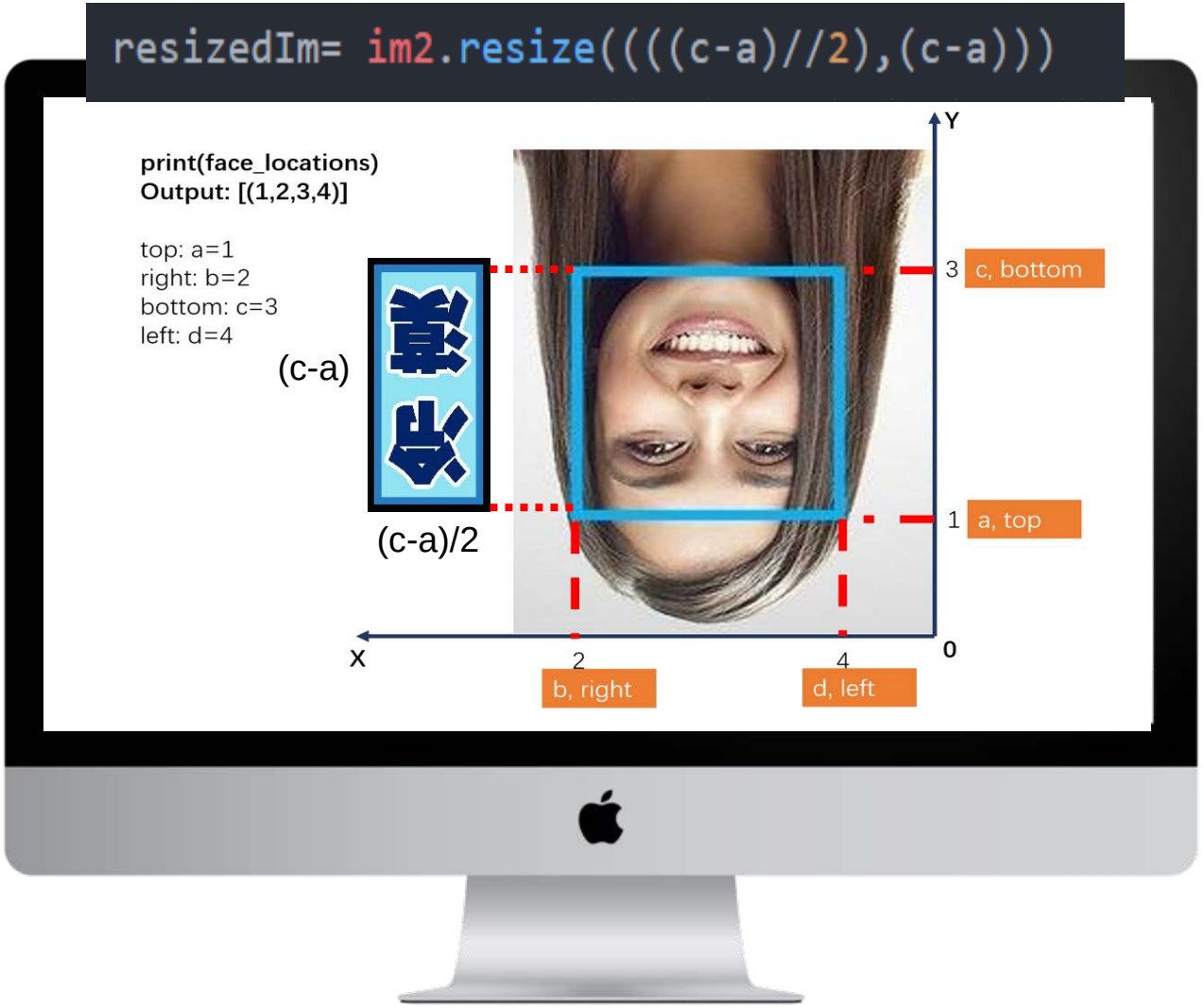
top: a=1
right: b=2
bottom: c=3
left: d=4



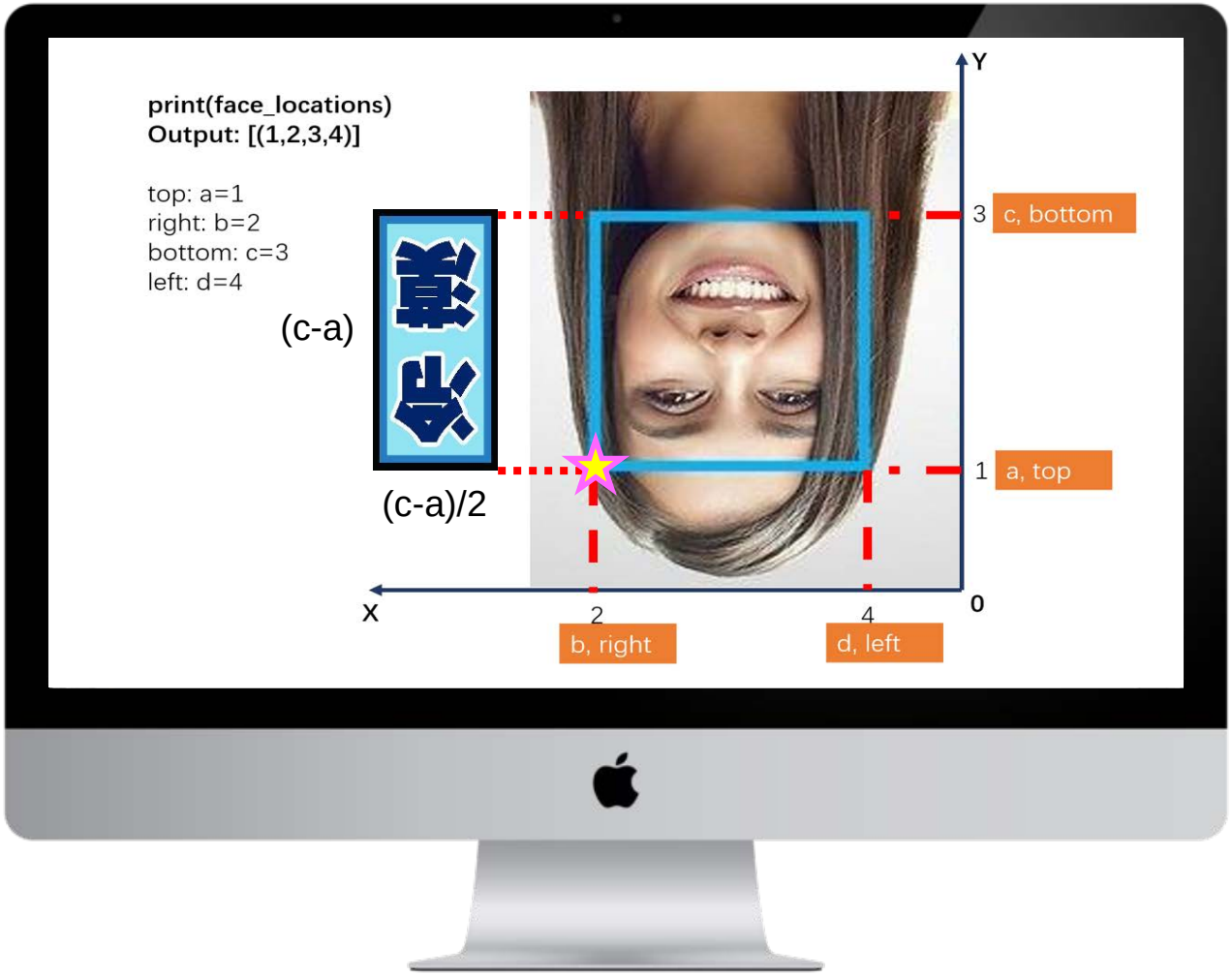

```
face_locations = face_recognition.face_locations(image)
print('face_locations:',face_locations)

a=int(face_locations[0][0])
b=int(face_locations[0][1])
c=int(face_locations[0][2])
d=int(face_locations[0][3])
```

4. Resize the sticker



5. Past stickers onto the background and show the result



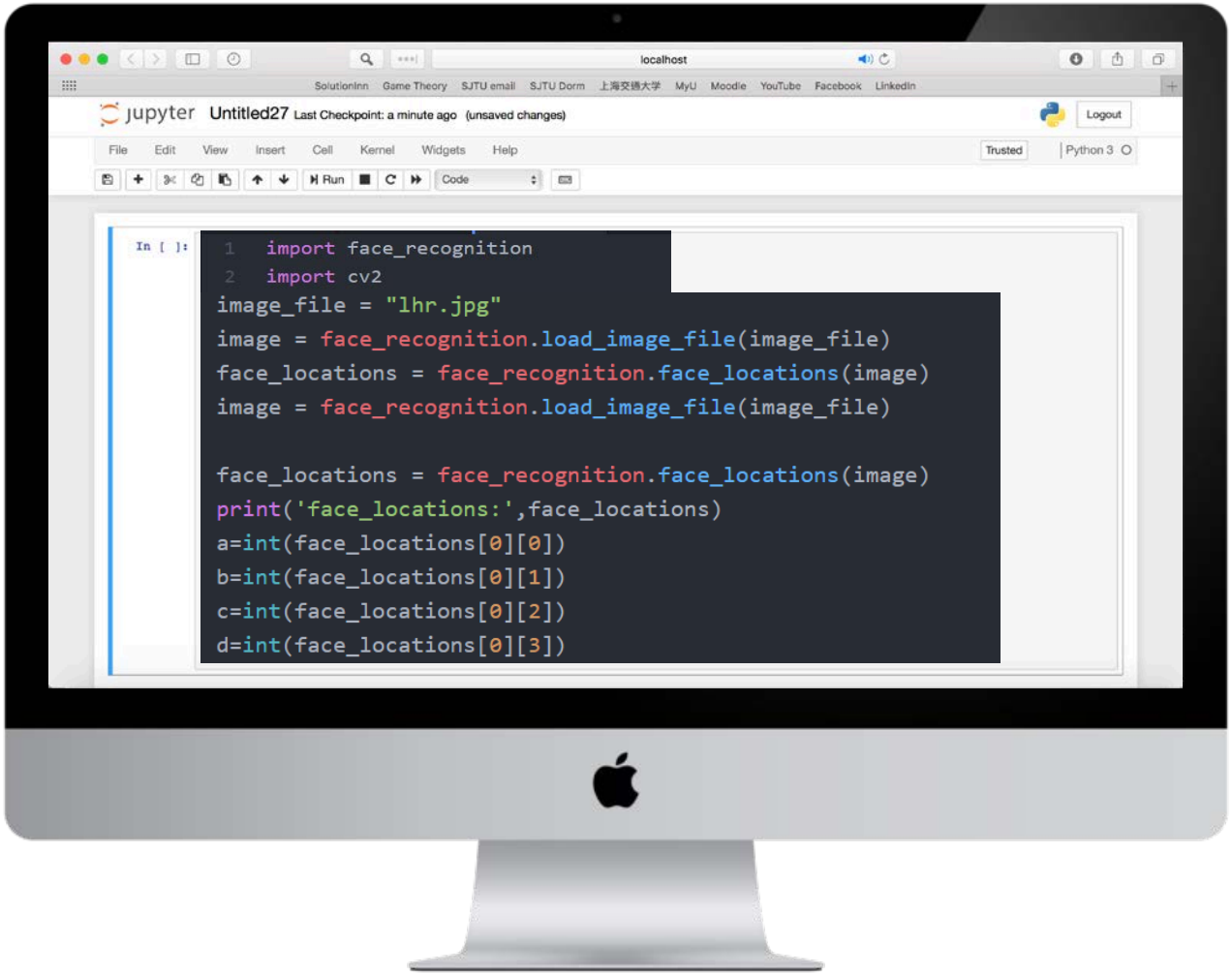
Finally, we get the result!

How Filter Stickers are Made

- Prepare the library
- Recognize the face location
- Resize the sticker
- Reformat the sticker and paste it
- Combine the picture and save/show result

1. Prepare the library.

2. Recognize the face location

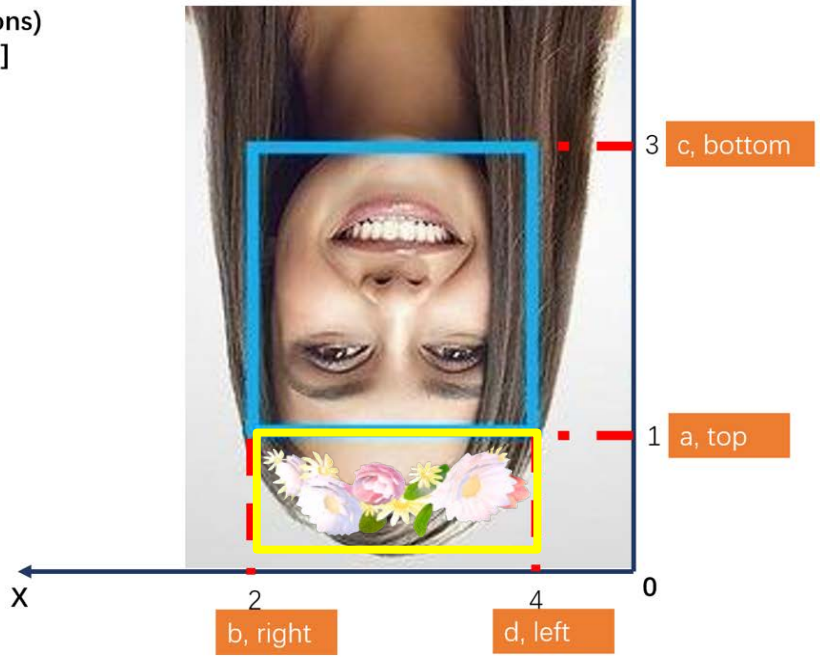


3. Resize the sticker

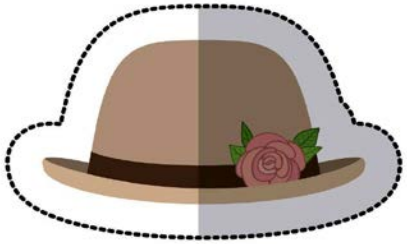
```
im1_path = r'C:\Users\Lucia\Documents\交大\Python\week11-code\f.png'
im2_path = r'C:\Users\Lucia\Documents\交大\Python\week11-code\lhr2.jpg'
im1 = Image.open(im1_path)
im2 = Image.open(im2_path)
if (abs(b-d)//2)>a:
    resizedIm = im1.resize((abs(b-d),a )) #防止边界过小
else:
    resizedIm= im1.resize((abs(b-d),(abs(b-d)//2))) #让贴纸的宽度和脸的宽度一致，贴纸的高度为其宽度的1/2
resizedIm.save(r'C:\Users\Lucia\Documents\交大\Python\week11-code\changeflower.png') #把改变过大小的贴纸
```

print(face_locations)
Output: [(1,2,3,4)]

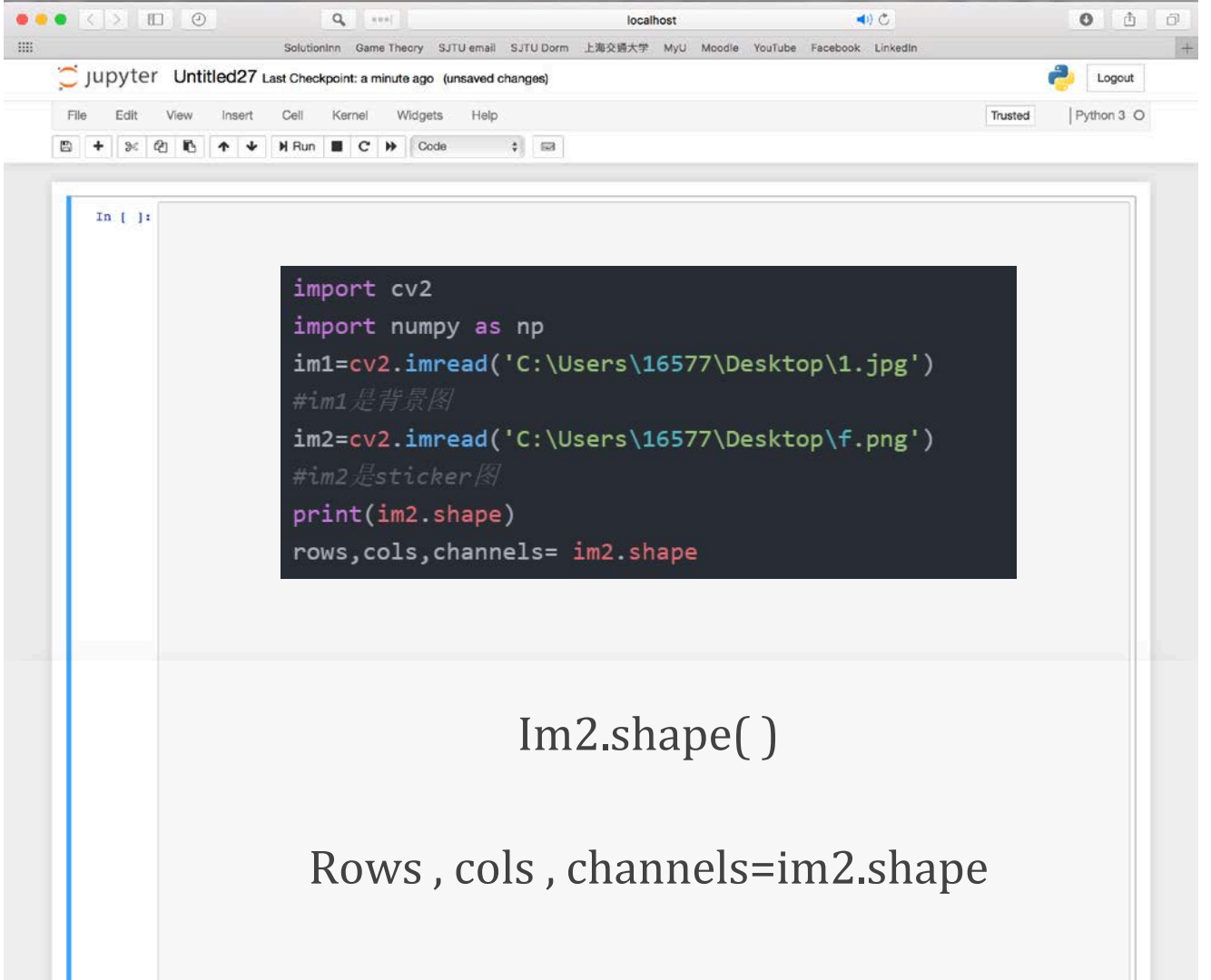
top: a=1
right: b=2
bottom: c=3
left: d=4



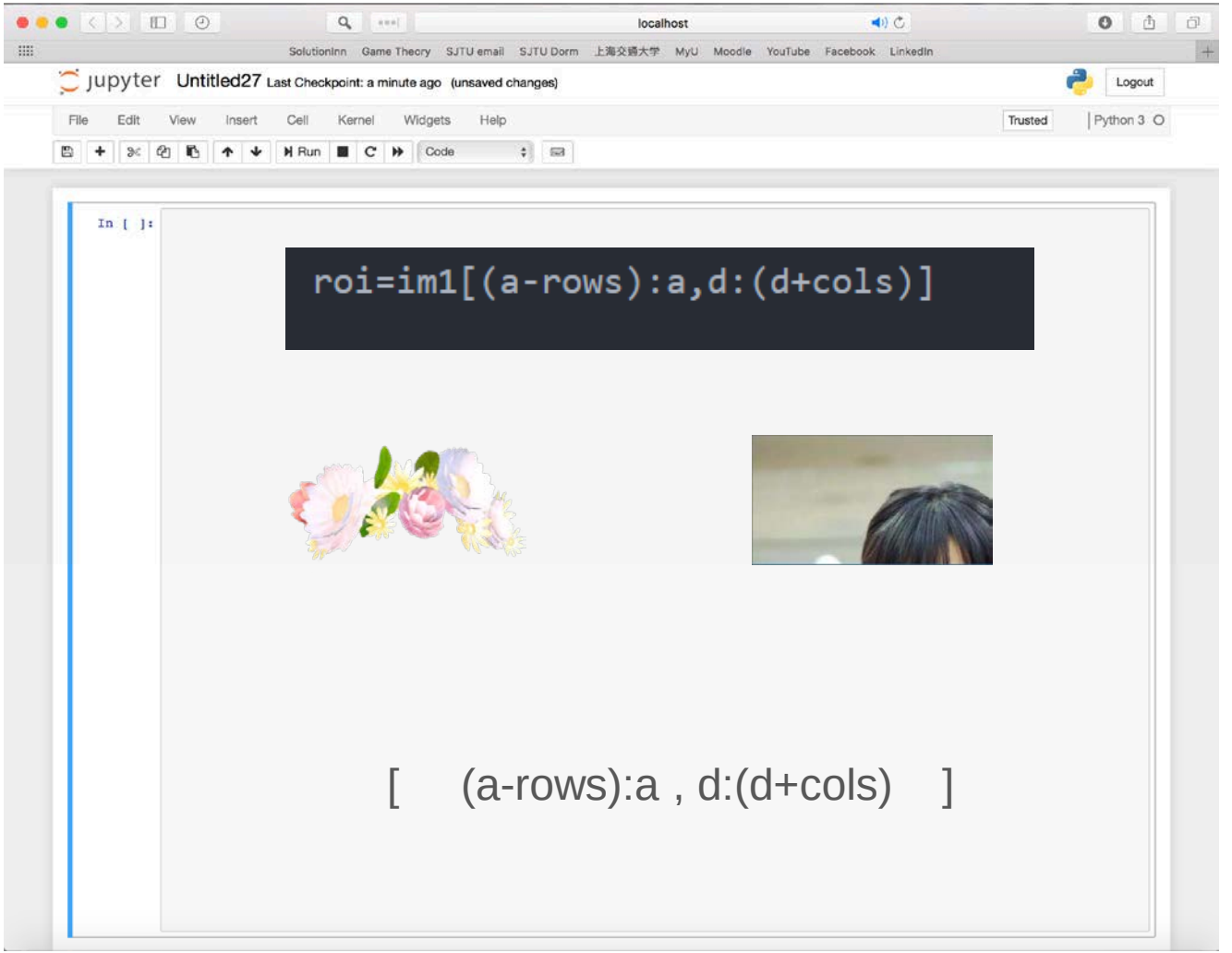
4. Sticker Matting and Pasting



Step 1: Preparation



Step 1: Preparation

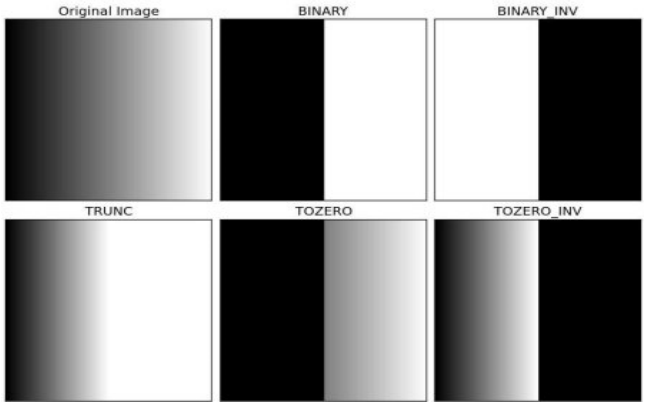


Step 2：

Change Sticker to Gray Scale image:

Step 3：
Fixed threshold binarization

```
_,mask=cv2.threshold(gray,50,255,cv2.THRESH_BINARY)
```

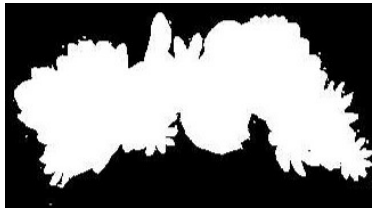


Binarization二值化:
Cv2.THRESH_BINARY
将图像上的像素点的灰度值设置为0或255，
整个图像呈现只有黑和白的视觉效果。

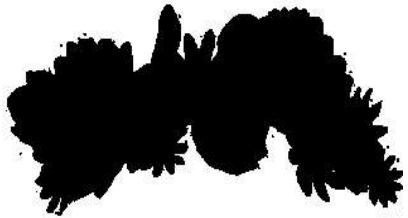
Threshold阈值：
CV2.Threshold（Gray，50,255）
寻找图像二值化阈值（50,255），
根据阈值将图像分为Foreground（White）,或者Background（Black）。

Step 3：
Fixed threshold binarization

```
_,mask=cv2.threshold(gray,50,255,cv2.THRESH_BINARY)
```

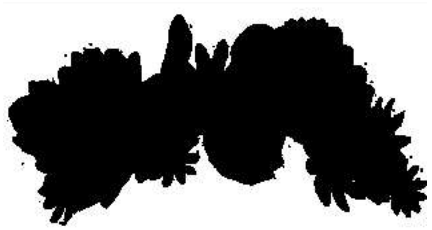


```
mask_inv=cv2.bitwise_not(mask)
```



Step 4:
Combine Pictures (1)

```
img1_bg=cv2.bitwise_and(roi,roi,mask=mask_inv)
```

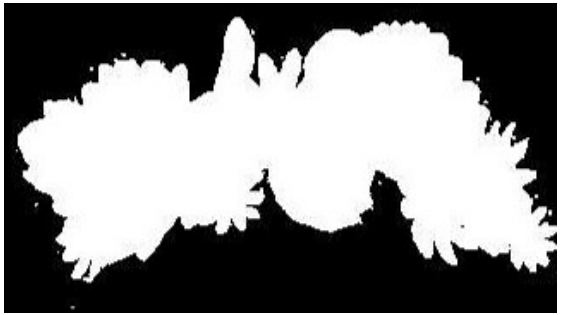


Cv2.bitwise_and( )

Mask=mask_inv

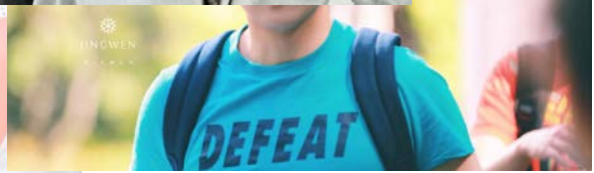
Step 4:
Combine Pictures (2)

```
img2_fg=cv2.bitwise_and(im2,im2,mask=mask)
```



Step 4:

Combine Pictures (3)



Combined effects

Weaknesses in Our Project

Lack of Accuracy

The position of makeup applied was not always accurate

Sometimes sticker gets placed at awkward position

Perfect Outcome not Guaranteed

Project was created using simpler python modules

Can’t compare with Meitu and B612 app

Conclusion

The project a good learning experience to:

- Apply what we learned in class to real use
- Cross our comfort zone to learn more
- Explore image processing power in python
- Feel proud with our own written program

Thank You