

# 4. Open Source CV Image Beautification

## 4.1. OpenCV Image Repair

1. Image repair is a class of algorithms in computer vision whose goal is to fill in areas within an image or video. The area is identified using a binary mask, and the filling is usually done based on the boundary information of the area to be filled. The most common application of image repair is to restore old scanned photos. It is also used to remove small unwanted objects in an image.
2. In OpenCV, `dst = cv2.inpaint(src, inpaintMask, inpaintRadius, flags)` is provided to repair the image.

Parameter meaning:

`src`: source image, that is, the image to be repaired

`inpaintMask`: binary mask indicating the pixels to be repaired

`dst`: result image

`inpaintRadius`: indicates the radius of the repair

`flags`: repair algorithm, mainly `INPAINT_NS` (Navier-Stokes based method) or `INPAINT_TELEA` (Fast marching based method)

Navier-Stokes-based repair should be slower and tend to produce blurrier results than the fast marching method. In practice, we did not find this to be the case. `INPAINT_NS` produced better results in our tests and was slightly faster than `INPAINT_TELEA`.

### 3. Code and actual effect display

(1) First, we add damage to the intact image, which can be understood as modifying the pixel values of a specific part of it

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-00001/src/yahboom_navrobo_astra/scripts/opencv/4_1_1.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    for i in range(50,100):
        img[i,50] = (0,0,0)
        img[i,50+1] = (0,0,0)
        img[i,50-1] = (0,0,0)
    for i in range(100,150):
        img[150,i] = (0,0,0)
        img[150,i+1] = (0,0,0)
        img[150-1,i] = (0,0,0)
    cv2.imwrite("damaged.jpg",img)
    dam_img = cv2.imread('damaged.jpg')
    while True :
        cv2.imshow("dam_img",dam_img)
        action = cv2.waitKey(10) & 0xFF
```

```

        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()

```

After running, an image will be generated. This image is considered a damaged image of the original image.



(2) Repair the photo just created. First read it, then create a mask, and finally use the function to repair it.

Run the program  
python /home/yahboom/YBAMR-COBOT-EDU-00001/src/yahboom\_navrobo\_astra/scripts/opencv/4\_1\_2.py

```

import cv2
import numpy as np
if __name__ == '__main__':
    dam_img = cv2.imread('damaged.jpg')
    imgInfo = dam_img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    paint = np.zeros((height,width,1),np.uint8)
    for i in range(50,100):
        paint[i,50] = 255
        paint[i,50+1] = 255
        paint[i,50-1] = 255
    for i in range(100,150):
        paint[150,i] = 255
        paint[150+1,i] = 255
        paint[150-1,i] = 255
    dst_img = cv2.inpaint(dam_img,paint,3,cv2.INPAINT_TELEA)
    while True :
        cv2.imshow("dam_img",dam_img)
        cv2.imshow("paint",paint)
        cv2.imshow("dst",dst_img)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()

```



As shown in the figure, the left side is before restoration, the middle is the mask image, and the right side is the original image after restoration.

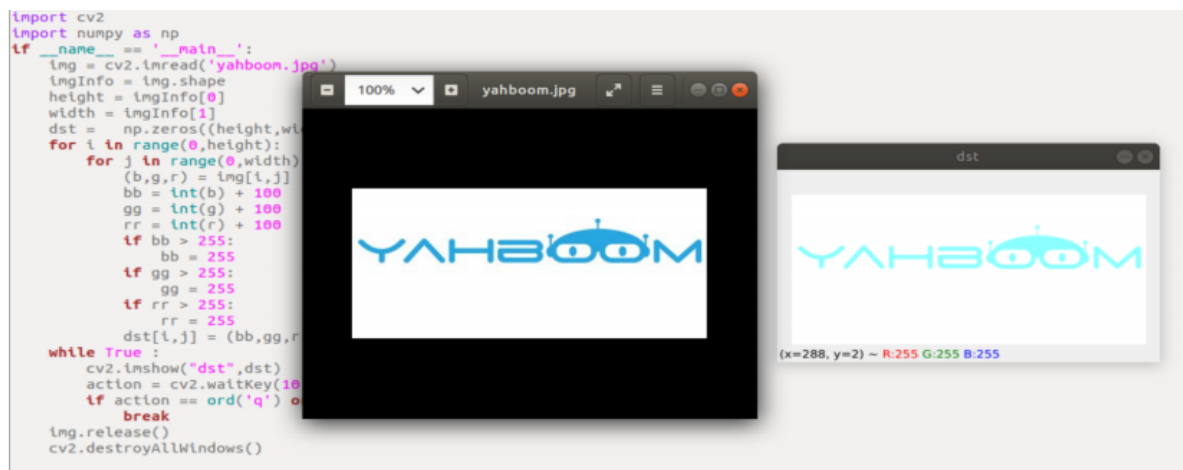
## 4.2. OpenCV image brightness enhancement

1. Implementation process: Synchronously amplify the three-channel values of each pixel point, while keeping the channel value between 0-255. In fact, it is to traverse each pixel point, add or subtract values to them, and then determine whether the three channels RGB are in the range of 0-255. If it is greater or less than, it takes the value of 255 or 0.
2. Code and actual effect display

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-00001/src/yahboom_navrobo_astra/scripts/opencv/4_2.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    imgInfo = img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    dst = np.zeros((height,width,3),np.uint8)
    for i in range(0,height):
        for j in range(0,width):
            (b,g,r) = img[i,j]
            bb = int(b) + 100
            gg = int(g) + 100
            rr = int(r) + 100
            if bb > 255:
                bb = 255
            if gg > 255:
                gg = 255
            if rr > 255:
                rr = 255
            dst[i,j] = (bb,gg,rr)
    while True :
        cv2.imshow("dst",dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```



The left picture is the original picture, and the picture behind is the picture with increased brightness.

### 4.3. OpenCV picture skin smoothing and whitening

1. OpenCV implements the function of picture skin smoothing and whitening. The principle of implementation is basically the same as the principle of "1.20 OpenCV picture brightness enhancement", but here we do not need to process the r value, just follow this formula,  $p = p(x) * 1.4 + y$ , where  $p(x)$  represents the b channel or g channel, and  $y$  represents the value to be increased or decreased. Similarly, after adding the value, we need to judge the value.
2. Code and actual effect display

Run the program

```
python /home/yahboom/YBAMR-COBOT-EDU-00001/src/yahboom_navrobo_astra/scripts/opencv/4_3.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('yahboom.jpg')
    imgInfo = img.shape
    height = imgInfo[0]
    width = imgInfo[1]
    dst = np.zeros((height,width,3),np.uint8)
    for i in range(0,height):
        for j in range(0,width):
            (b,g,r) = img[i,j]
            bb = int(b*1.4) + 5
            gg = int(g*1.4) + 5
            if bb > 255:
                bb = 255
            if gg > 255:
                gg = 255
            dst[i,j] = (bb,gg,r)
    while True :
        cv2.imshow("origin",img)
        cv2.imshow("dst",dst)
        action = cv2.waitKey(10) & 0xFF
        if action == ord('q') or action == 113:
            break
    img.release()
    cv2.destroyAllWindows()
```

