4. OpenCV image beautification

Before running the sample program, you need to switch to the directory where the code is located. SSH connects the car, runs at the terminal,

```
cd /userdata/yahboomcar_ws/src/yahboomcar_astra/opencv_examples
```

4.1. Image repair

> Image repair is a class of algorithms in computer vision whose goal is to fill areas within an image or video. The region is identified using a binary mask, and filling is usually done according to the region boundary information that needs to be filled. The most common application of image restoration is to restore old scanned photos. It is also used to remove small unwanted objects from images.

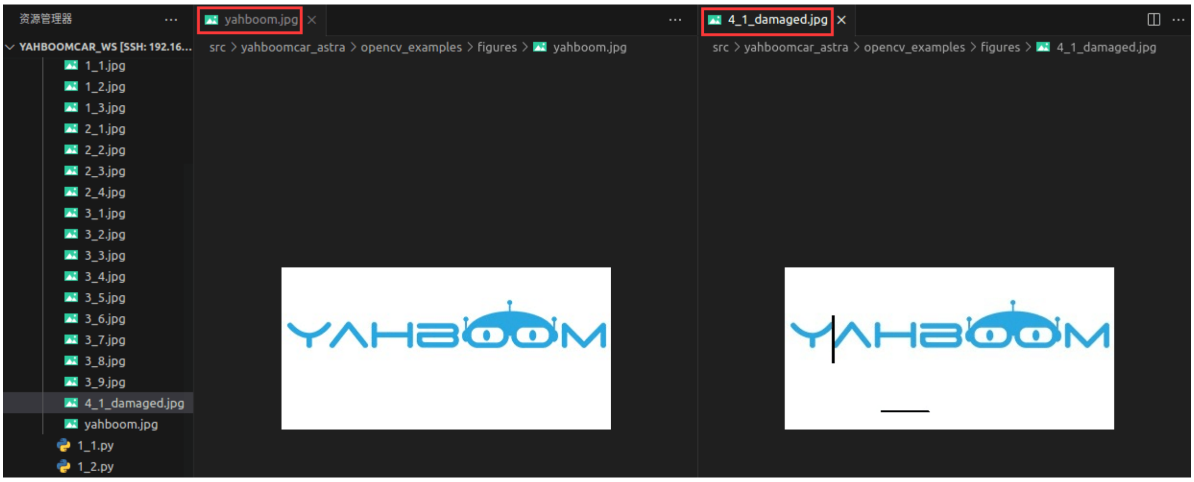
(1) Firstly, according to the intact picture, the damage is added to it, which can be understood as modifying the pixel value of a specific part of it

After the car terminal switches to the directory where the code is located, run the program,

```
python3 4_1_1.py
```

```python
import cv2
if __name__ == '__main__':
    img = cv2.imread('figures/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('figures/4_1_damaged.jpg',img)
```

The resulting image can be seen as a damaged image of the original image.



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

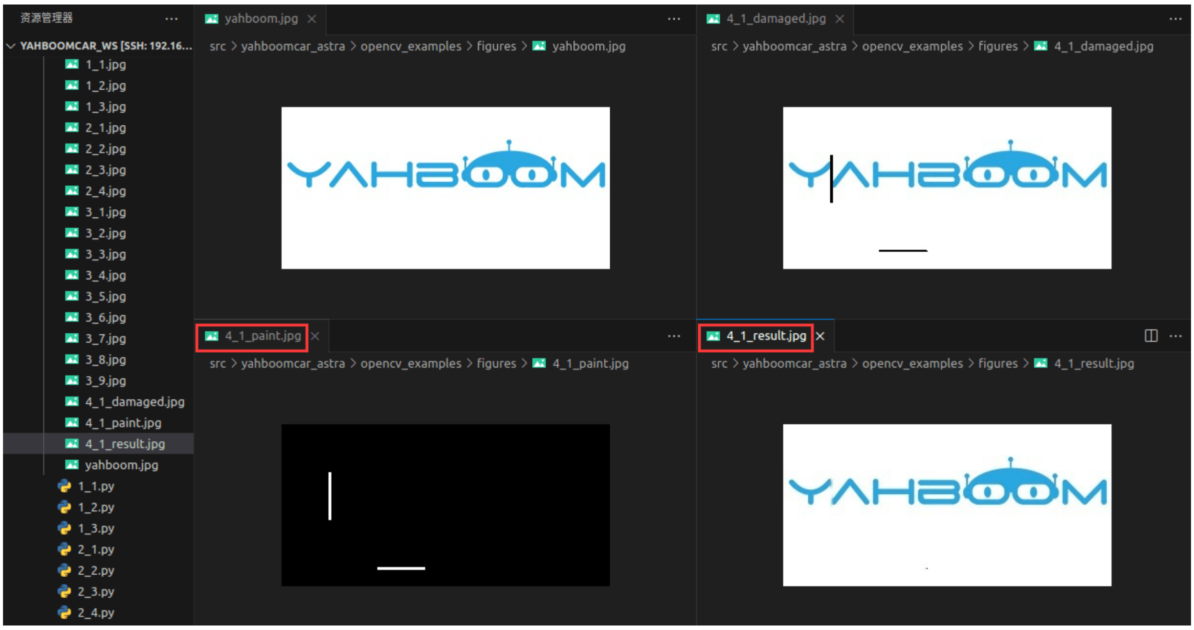
After the car terminal switches to the directory where the code is located, run the program,

```
python3 4_1_2.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    dam_img = cv2.imread('figures/4_1_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)
    cv2.imwrite("figures/4_1_paint.jpg",paint)
    cv2.imwrite("figures/4_1_result.jpg",dst_img)

'''
绘制文字：dst = cv2.inpaint(src,inpaintMask,inpaintRadius,flags)
    src：源图像，也就是需要修复的图像
    inpaintMask：二进制掩码，指示要修复的像素
    dst：结果图像
    inpaintRadius：表示修复的半径
    flags: 修复算法，主要有，
        INPAINT_NS (Navier-Stokes based method)
        INPAINT_TELEA (Fastmarching based method)
Draw text: dst = cv2.inpaint(src,inpaintMask,inpaintRadius,flags)
    src: The source image, which is the image that needs to be repaired
    inpaintMask: Binary mask indicating the pixel to be repaired
    dst: result image
    inpaintRadius: indicates the radius of repair
    flags: Repair algorithm, mainly have,
        INPAINT_NS (Navier-Stokes based method)
        INPAINT_TELEA (Fastmarching based method)
'''
```

The resulting image is shown here,


资源管理器
YAHBOOMCAR_WS [SSH: 192.16...
1_1.jpg
1_2.jpg
1_3.jpg
2_1.jpg
2_2.jpg
2_3.jpg
2_4.jpg
3_1.jpg
3_2.jpg
3_3.jpg
3_4.jpg
3_5.jpg
3_6.jpg
3_7.jpg
3_8.jpg
3_9.jpg
4_1_damaged.jpg
4_1_paint.jpg
4_1_result.jpg
yahboom.jpg
1_1.py
1_2.py
1_3.py
2_1.py
2_2.py
2_3.py
2_4.py
yahboom.jpg
src > yahboomcar_astra > opencv_examples > figures > yahboom.jpg
4_1_damaged.jpg
src > yahboomcar_astra > opencv_examples > figures > 4_1_damaged.jpg
4_1_paint.jpg
src > yahboomcar_astra > opencv_examples > figures > 4_1_paint.jpg
4_1_result.jpg
src > yahboomcar_astra > opencv_examples > figures > 4_1_result.jpg
YAHBOOM

The lower left corner is the mask image, and the lower right corner is the repaired image, which is consistent with the original image in the upper left corner.

4.2. Brightness enhancement

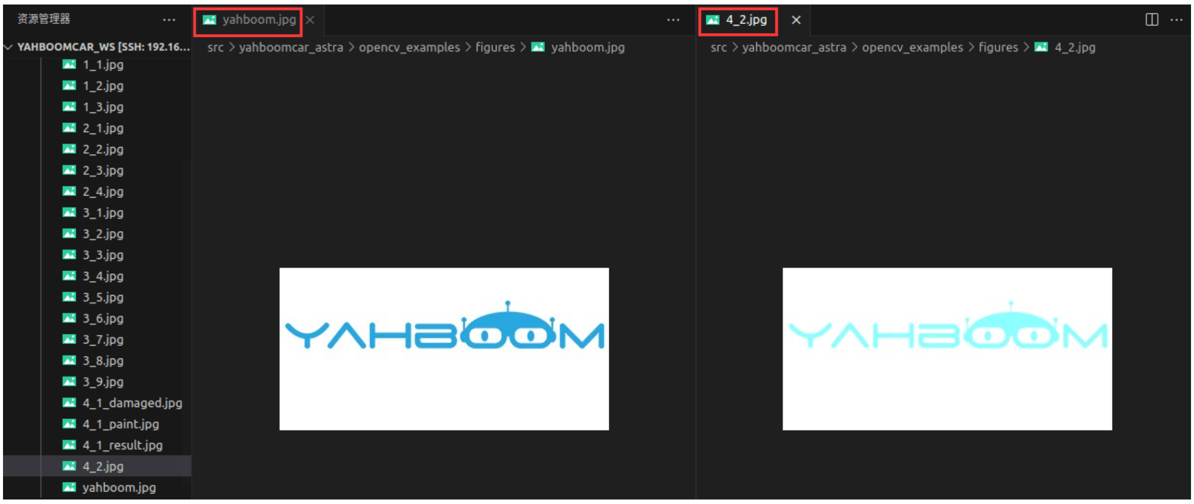
> Implementation process: The three-channel value of each pixel is amplified synchronously, while keeping the channel value between 0-255, in fact, it is to traverse each pixel, add or subtract values to them, and then determine whether the rgb of the three channels is in the 0-255 range, greater than or less than the value 255 or 0.

After the car terminal switches to the directory where the code is located, run the program,

```
python3 4_2.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('figures/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)
    cv2.imwrite('figures/4_2.jpg',dst)
```

The resulting image is shown here,



4.3. Exfoliate and whitening

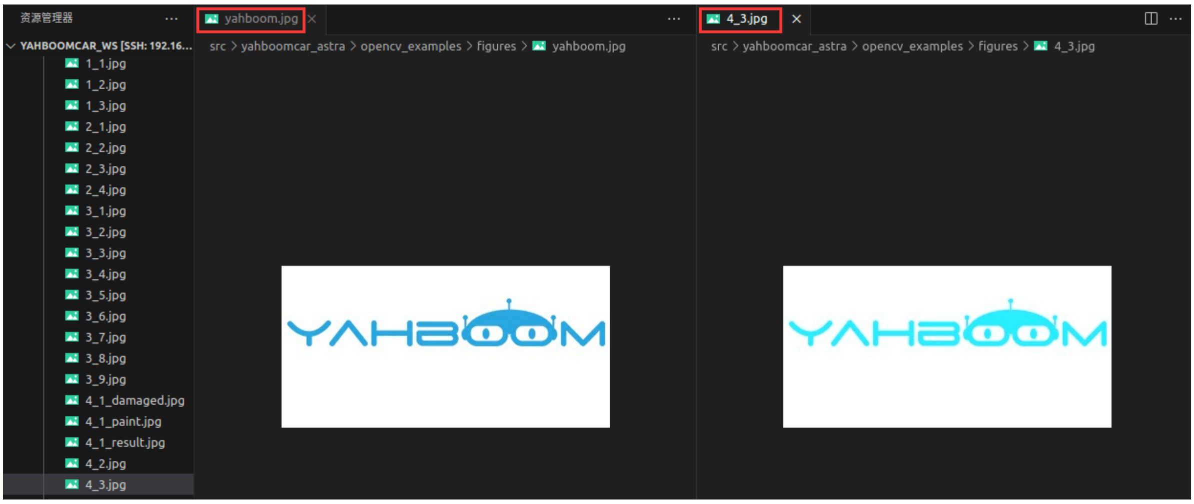
> OpenCV implements the function of image skin whitening, and the principle of realization is basically the same as that of brightness enhancement, but here we do not need to deal with r value, only need to follow this formula, p = p(x)*1.4+ y, where p(x) represents b channel or g channel, y represents the value that needs to be increased or decreased, similarly, After adding the value, we need to make a judgment about the value.

After the car terminal switches to the directory where the code is located, run the program,

```
python3 4_3.py
```

```
import cv2
import numpy as np
if __name__ == '__main__':
    img = cv2.imread('figures/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)
    cv2.imwrite('figures/4_3.jpg',dst)
```

The resulting image is shown here,



You can change the input picture to try.