Open Source CV Image beautification

1、 OpenCV Fix images

1.1、Image restoration is a type of algorithm in computer vision，

Its goal is to fill the area within the image or video. This area is identified using binary masks, and filling is usually done based on the boundary information of the area 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.

1.2、In OpenCV, dst=cv2. inpint (src, inpaintMask, inpaintradius, flags) is provided to achieve image restoration，

Parameter meanings：

src：Source image, which is the image that needs to be repaired

inpaintMask：Binary mask, indicating the pixels to be repaired.

dst：Result image

inpaintRadius：Indicates the radius of repair

flags : Repair algorithm，major have INPAINT_NS (Navier-Stokes based method) or INPAINT_TELEA (Fast marching based method)

The repair based on Navier Stokes should be slower and tend to produce more blurry results than the fast marking method. In practice, we have not found this situation. INPAINT-NS produced better results in our testing, and the speed was slightly higher than INPAINT-TELEA。

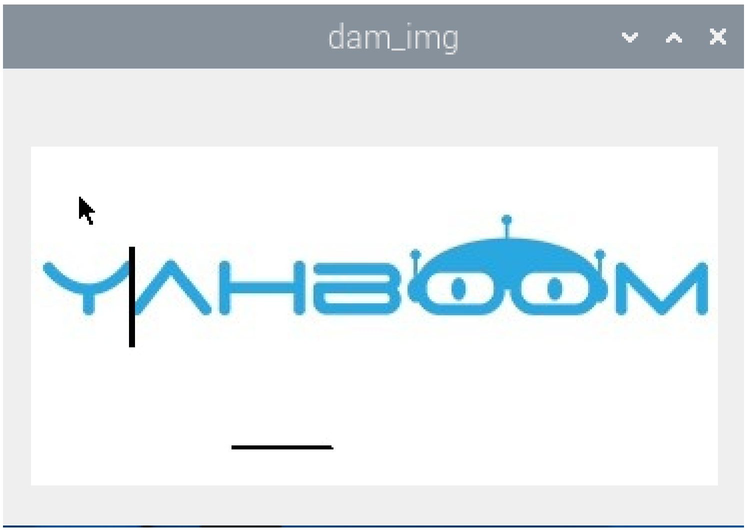
1.3、Code and actual effect display

（1）、Firstly, based on the intact image, we will add damage to it, which can be understood as modifying the pixel values of specific parts of it

```
cd ~/yahboomcar_ws/src/yahboom_esp32ai_car/scripts/opencv/
python3 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()
```

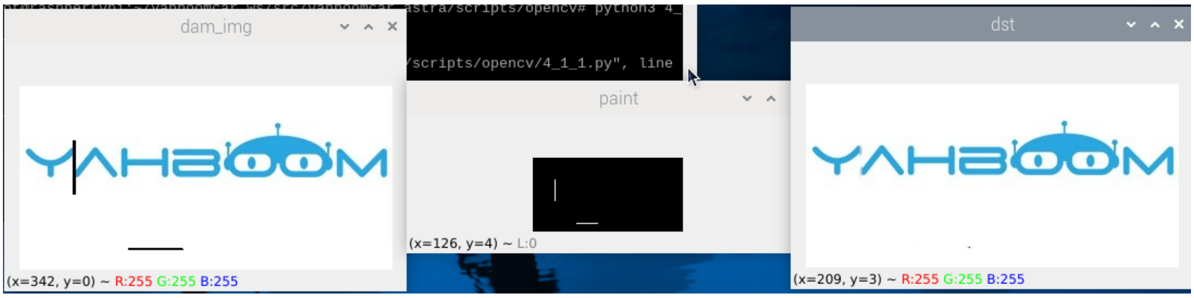
After running, an image will be generated, which is considered as a damaged image of the original image



（2）、To fix the photo you just created, first read it, then create a mask, and finally use a function to fix it

```
# Running programs
cd ~/yahboomcar_ws/src/yahboom_esp32ai_car/scripts/opencv/
python3 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()
```



```
astra/scripts/opencv# python3 4_
/scripts/opencv/4_1_1.py", line
```

dam_img
paint
dst
(x=342, y=0) ~ R:255 G:255 B:255
(x=126, y=4) ~ L:0
(x=209, y=3) ~ R:255 G:255 B:255

As shown in the figure, the left side is the pre repair image, the middle is the mask image, and the right side is the original image after repair。

2、 OpenCV image brightness enhancement

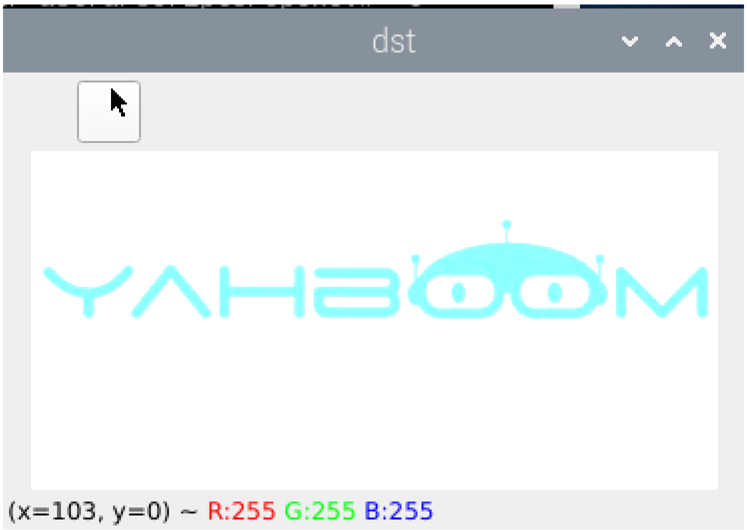
2.1、Implementation process:

Synchronously amplify the three channel values of each pixel while keeping the channel values between 0-255. In fact, it is to traverse each pixel, add or subtract numerical values, and then determine whether the three channel RGB values are in the 0-255 range. If they are greater than or less than, the values are 255 or 0.

2.2、Code and actual effect display

```
cd ~/yahboomcar_ws/src/yahboom_esp32ai_car/scripts/opencv/
python3 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()
```



(x=103, y=0) ~ R:255 G:255 B:255

The picture is a photo with increased brightness

3、OpenCV Image Peeling and Whitening

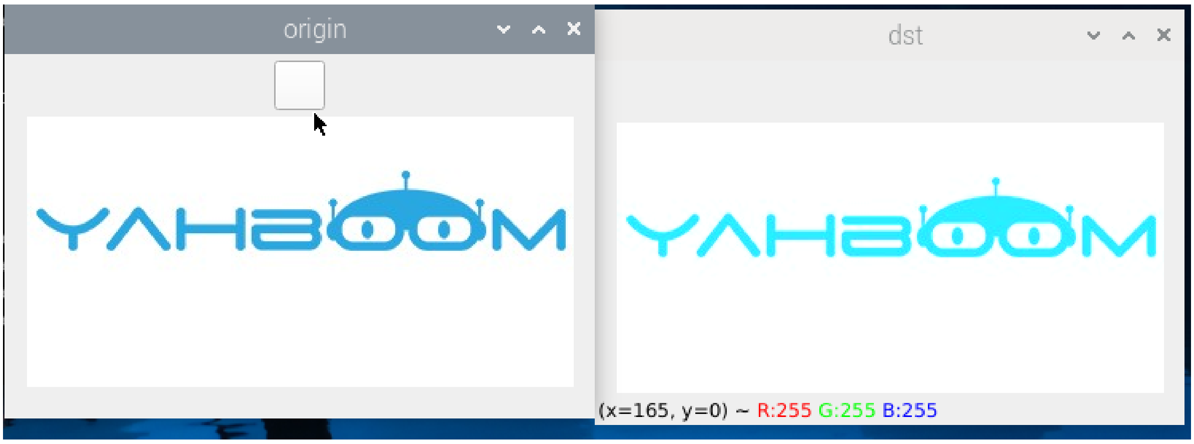
3.1、OpenCV realizes the function of image peeling and whitening,

The principle of implementation is basically the same as that of "1.20 OpenCV image brightness enhancement", but here we do not need to process the r value. We only need to follow this formula, p=p (x) * 1.4+y, where p (x) represents channel b or channel g, and y represents the number of values that need to be increased or decreased. Similarly, after adding a value, we need to make a judgment on the value.

3.2、Code and actual effect display

```
cd ~/yahboomcar_ws/src/yahboom_esp32ai_car/scripts/opencv/
python3 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()
```



(x=165, y=0) ~ R:255 G:255 B:255