

1.4.4 Brightness enhancement

Implementation process: synchronously amplify the three channel values of each pixel, while keeping the channel value between 0-255.

map (f, list) applies the function f to the entire list and returns the new list.

np.clip (a, a_min, a_max, out = None) limit the elements in a to the minimum and maximum values

Code path:

[/home/pi/Yahboom_Project/1.OpenCV_course/04image_beautification/04_Brightness_enhancement.ipynb](#)

```
import cv2

import numpy as np

import matplotlib.pyplot as plt


img = cv2.imread('yahboom.jpg',1)

imgInfo = img.shape

height = imgInfo[0]

width = imgInfo[1]

#cv2.imshow('src',img)

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) + 40

        gg = int(g) + 40

        rr = int(r) + 40
```

```
        if bb>255:

            bb = 255

        if gg>255:

            gg = 255

        if rr>255:

            rr = 255

        dst[i,j] = (bb,gg,rr)

# cv2.imshow('dst',dst)

# cv2.waitKey(0)

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

dst = cv2.cvtColor(dst, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(14, 6), dpi=100) # Set the size and pixels of the drawing area

plt.subplot(121) # The first in a row and two columns

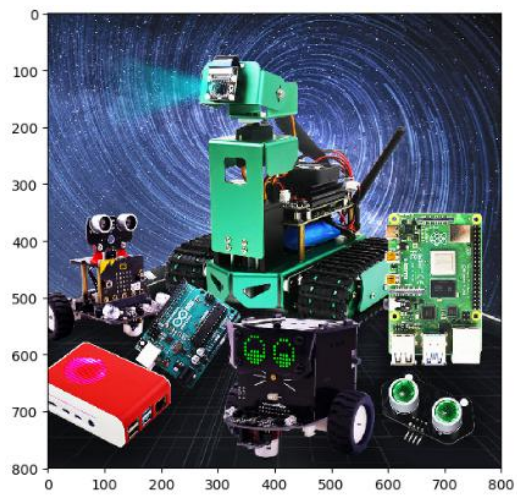
plt.imshow(img)

plt.subplot(122) # The second in a row and two columns

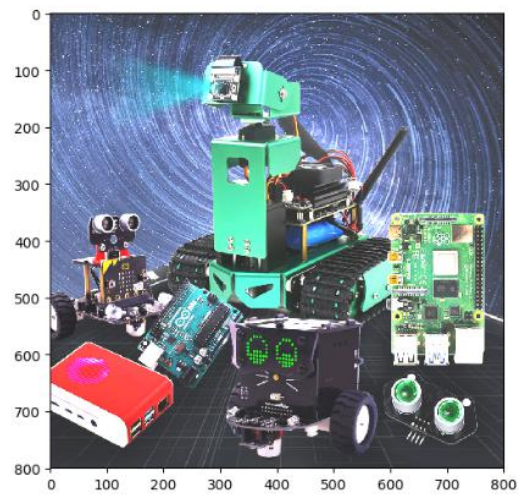
plt.imshow(dst)

plt.show()
```

After running the above program, two pictures will be displayed in the jupyterLab control interface, as shown below.



[Original picture]



[Brightness enhancement

picture]