

Image mirroring

There are two types of image mirroring transformation: horizontal mirroring and vertical mirroring. Horizontal mirroring takes the vertical centerline of the image as the axis and swaps the pixels of the image, that is, swapping the left and right halves of the image. Vertical mirroring takes the horizontal centerline of the image as the axis and swaps the upper and lower parts of the image.

Transformation principle: Let the width of the image be width and the length be height. (x,y) are the transformed coordinates, (x₀,y₀) are the coordinates of the original image

Horizontal mirror transformation

$$\begin{aligned}x &= \text{width} - x_0 - 1 \\ y &= y_0\end{aligned}$$

forward mapping

Its inverse transformation is

$$\begin{aligned}x_0 &= \text{width} - x - 1 \\ y_0 &= y\end{aligned}$$

backward mapping

Vertical Mirror Transformation

$$\begin{aligned}x &= x_0 \\ y &= \text{height} - y_0 - 1\end{aligned}$$

Its inverse transformation is

$$\begin{aligned}x_0 &= x \\ y_0 &= \text{height} - y - 1\end{aligned}$$

Summarize:

During horizontal mirror transformation, the entire image is traversed, and then each pixel is processed according to the mapping relationship. In fact, the horizontal mirror transformation is to change the image coordinate column to the right, and the right column to the left. The transformation can be done in column units. The same is true for vertical mirror transformation, which can be transformed in row units. Here we take vertical transformation as an example to see how Python is written:

- Start Docker

After entering the Raspberry Pi 5 desktop, open a terminal and run the following command to start the container corresponding to Dofbot:

```
./Docker_Ros.sh
```

Access Jupyter Lab within Docker:

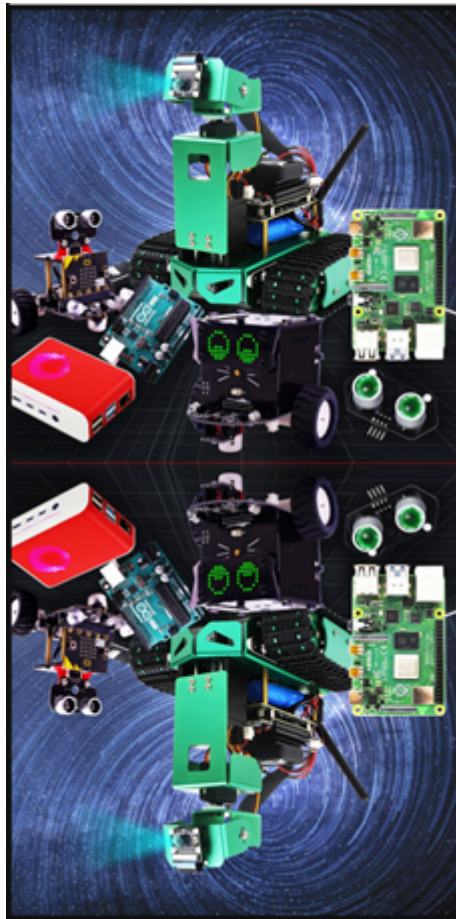
```
IP:9999 // Example: 192.168.1.11:9999
```

Code path: /root/Dofbot/4.opencv/2.Transform/04_mirror_image.ipynb

```
import cv2
import numpy as np
img = cv2.imread('yahboom.jpg',1)
#cv2.imshow('src',img)
imgInfo = img.shape
height = imgInfo[0]
width = imgInfo[1]
deep = imgInfo[2]
newImgInfo = (height*2,width,deep)
dst = np.zeros(newImgInfo,np.uint8)#uint8
for i in range(0,height):
    for j in range(0,width):
        dst[i,j] = img[i,j]
        #x y = 2*h - y -1
        dst[height*2-i-1,j] = img[i,j]
for i in range(0,width):
    dst[height,i] = (0,0,255) #BGR
```

```
#bgr8 to jpeg format
import enum
import cv2
def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

```
import ipywidgets.widgets as widgets
image_widget1 = widgets.Image(format='jpg', )
# image_widget2 = widgets.Image(format='jpg', )
# create a horizontal box container to place the image widget next to each other
# image_container = widgets.HBox([image_widget1, image_widget2])
# display the container in this cell's output
display(image_widget1)
#display(image_widget2)
image_widget1.value = bgr8_to_jpeg(dst)
```



You can see the mirror image from the picture.