Image zoom

In OpenCV, the function that implements image scaling is: cv2.resize(InputArray src, OutputArray dst, Size, fx, fy, interpolation)

- 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/01_zoom_pic.ipynb

Parameter explanation:

InputArray src	**Input image**
OutputArray dst	Output picture
Size	Output image size
fx, fy	Scaling factors along the x-axis, y-axis
interpolation	insertion method

Interpolation method used by options:

INTER_NEAREST	**Nearest neighbor interpolation**
INTER_LINEAR	Bilinear interpolation (default setting)
INTER_AREA	Resampling using pixel area relationships.
INTER_CUBIC	Bicubic interpolation of 4x4 pixel neighborhoods
INTER_LANCZOS4	Lanczos interpolation of 8x8 pixel neighborhoods

Notice:

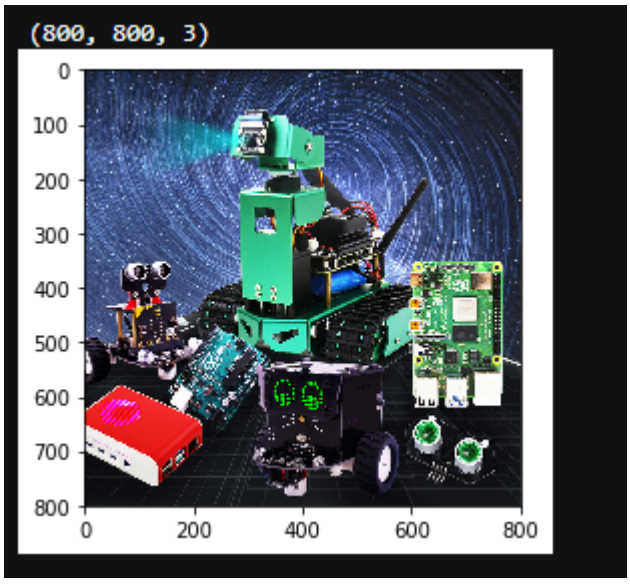
1. The output size format is (width, height)
2. The default interpolation method is: bilinear interpolation

The main code is as follows:

```
# 1 load 2 info 3 resize 4 check
import cv2
import matplotlib.pyplot as plt # Python's 2D drawing library

# Read in the original image
```

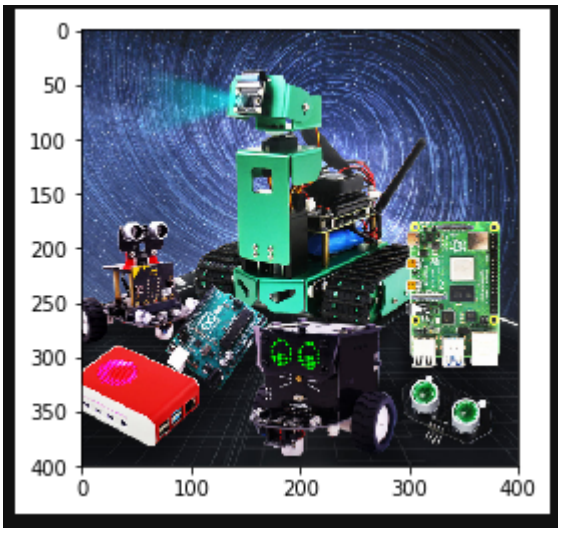
```
img = cv2.imread('yahboom.jpg')
#Print the image size
print(img.shape)
#Assign the height and width of the image to x and y respectively
x, y = img.shape[0:2]


# Scale to one-half of the original size, and the output size format is (width,
height)
img_test1 = cv2.resize(img, (int(y / 2), int(x / 2)))
# cv2.imshow('resize0', img_test1)
#cv2.waitKey()

# Nearest neighbor interpolation scaling

# Zoom to a quarter of the original size
img_test2 = cv2.resize(img, (0, 0), fx=0.25, fy=0.25,
interpolation=cv2.INTER_NEAREST)
# cv.imshow('resize1', img_test2)
#cv.waitKey()
#cv.destroyAllWindows()
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
dst1 = cv2.cvtColor(img_test1, cv2.COLOR_BGR2RGB)
dst2 = cv2.cvtColor(img_test2, cv2.COLOR_BGR2RGB)
# show original image
plt.imshow(img)
plt.show()
```

After execution, you can see that the image is 800*800



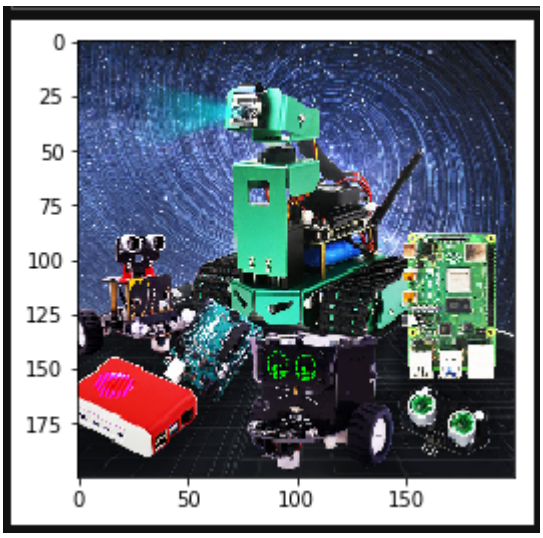

```
# Display zoom 1/2
plt.imshow(dst1)
plt.show()
```

After the execution is completed, you can see that the image is 400*400, scaled by half.




```
#Display scaling 1/4 neighbor interpolation scaling
plt.imshow(dst2)
plt.show()
```

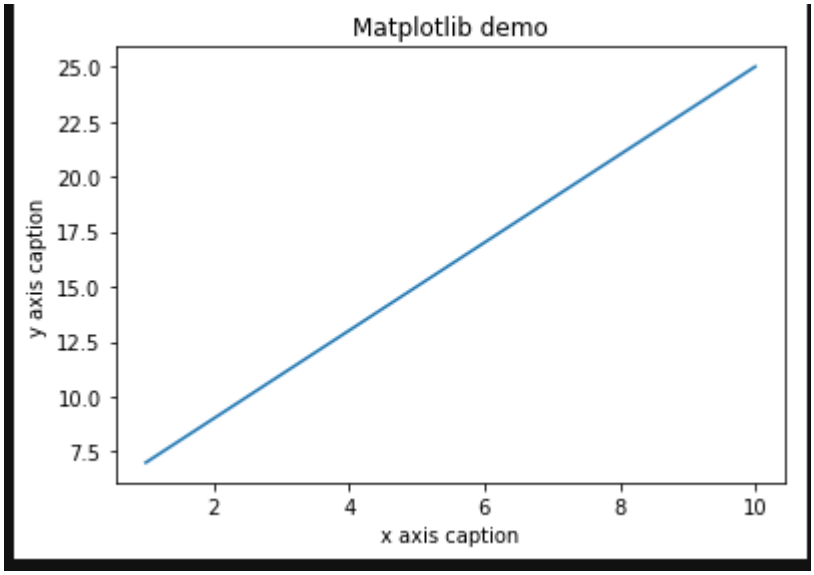
After execution, you can see that the image is 200*200, scaled by a quarter.



Next let's talk about matplotlib: Python's 2D drawing library.

Reference tutorial: https://www.runoob.com/numpy/numpy-matplotlib.html

```
import numpy as np
from matplotlib import pyplot as plt
x = np.arange(1,11)
y=2*x+5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y)
plt.show()
```


Matplotlib demo
y axis caption
x axis caption
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
2
4
6
8
10