Lab Practice 02

1、 Gaussian Blur

（1） Modules needed：

(a) *from PIL import Image*: image processing library, used to read the data in the image, such as, *Image.open()*

(b) *import numpy*: used for processing array objects and linear algebra functions, such as, *numpy.zero()* [The *array()* method is usually called to convert the image into a *numpy()* array object]

(c) *from scipy.ndimage import filters*: The *scipy.ndimage.filter* module is used to filter hype,.[*im2=filters.gaussian_filter(im1, σ)*]

(d) *from pylab import **: Draw image array into image processing, such as, imshow( )

(e) *from matplotlib import pyplot*: such as, *subplot( )*

（2） Practice one:

（a） a1 image grayscale stored to *im1*

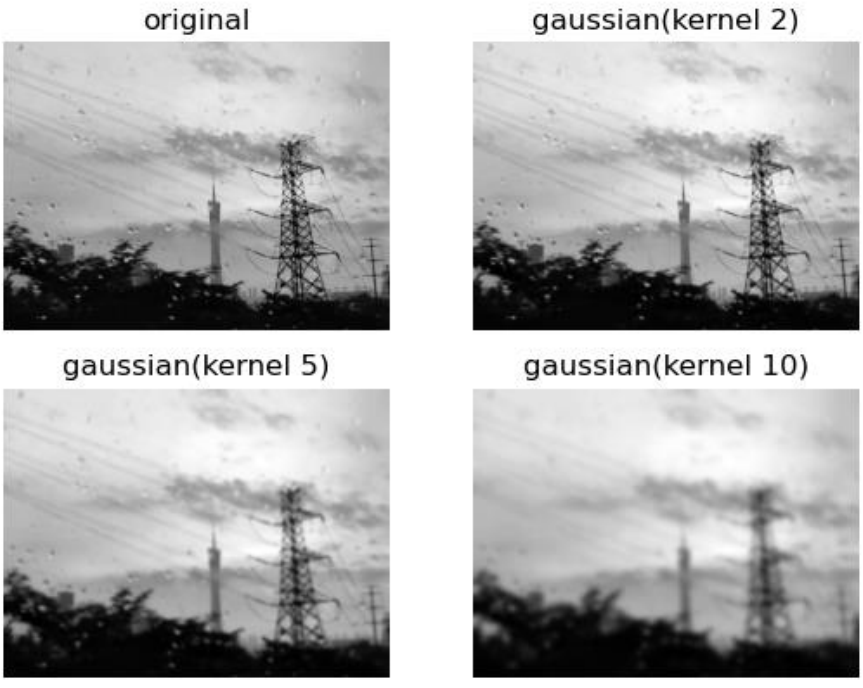
（b） Gaussian blur the array(σ =2, 5, 10), data is stored to *im2*, *im5*, *im10*.

（c） Output four images in the same image(Use the

subplot() function)

```
from PIL import Image
import numpy as np
from scipy.ndimage import filters
from matplotlib import pyplot as plt

im1 = np.array(Image.open('a1.jpg').convert('L'))
im2 = filters.gaussian_filter(im,2)
im5 = filters.gaussian_filter(im,5)
im10 = filters.gaussian_filter(im,10)

plt.subplot(2,2,1)
plt.axis('off')
plt.imshow(im1,cmap='gray')
plt.title('original')

plt.subplot(2,2,2)
plt.axis('off')
plt.imshow(im2,cmap='gray')
plt.title('gaussian(kernel 2)')

plt.subplot(2,2,3)
plt.axis('off')
plt.imshow(im5,cmap='gray')
plt.title('gaussian(kernel 5)')

plt.subplot(2,2,4)
plt.axis('off')
plt.imshow(im10,cmap='gray')
plt.title('gaussian(kernel 10)')
```

Examples are shown below:



2、 Sobel filter

（1） **Modules needed：** Same as above

Functions involved in Sobel filter:

Numpy.zeros(im.shape): Create array

filters.sobel(im, 0/1,imy/x):[0: y-axis;1: x-axis]

（2） **Practice two:**

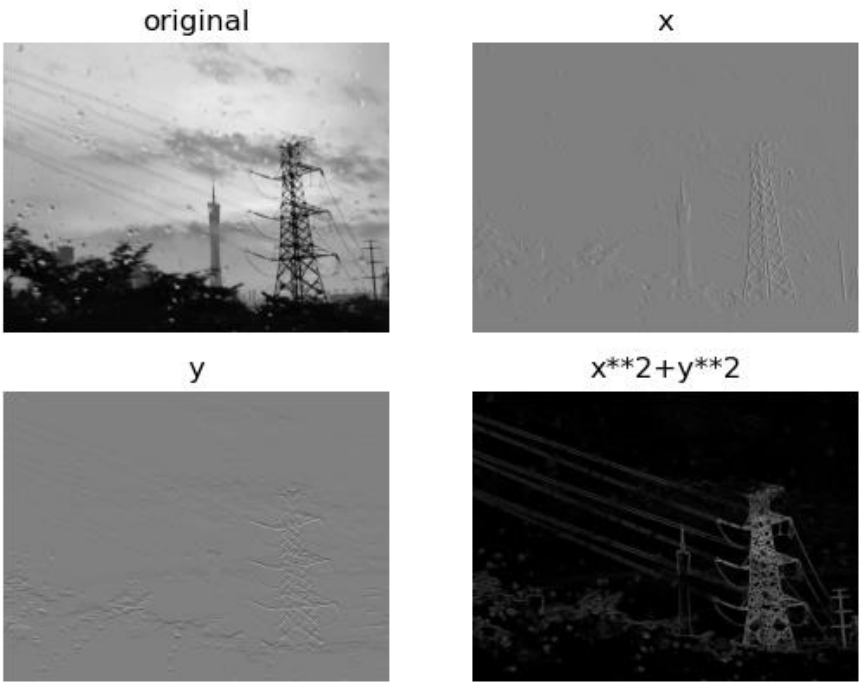
(a) a1 image grayscale stored to im1

(b) Filter processing on the x-axis and y-axis directions of the image, Stored in *imx* and *imy*

(c) The gradient size image is stored in the *magnitude*

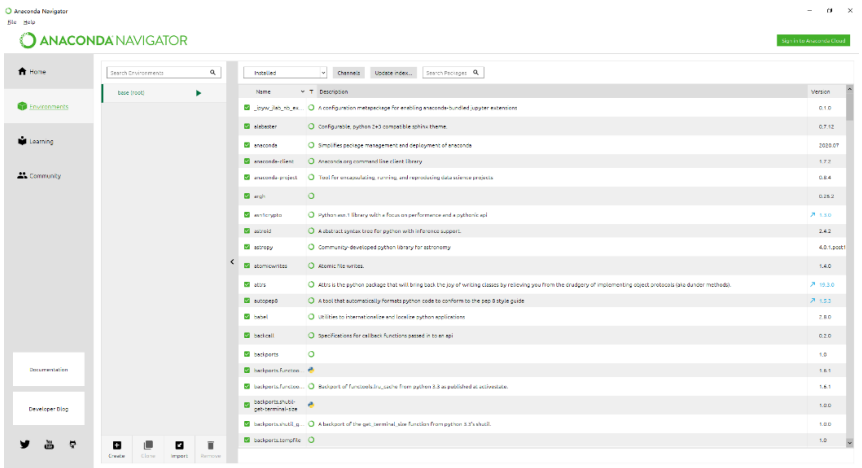
```
from PIL import Image
import numpy as np
from scipy.ndimage import filters
from matplotlib import pyplot as plt

im = np.array(Image.open('a1.jpg').convert('L'))

imx = np.zeros(im.shape)
filters.sobel(im,1,imx)

imy = np.zeros(im.shape)
filters.sobel(im,0,imy)

magnitude = np.sqrt(imx**2+imy**2)


plt.subplot(2,2,1)
plt.axis('off')
plt.imshow(im,cmap='gray')
plt.title('original')

plt.subplot(2,2,2)
plt.axis('off')
plt.imshow(imx,cmap='gray')
plt.title('x')

plt.subplot(2,2,3)
plt.axis('off')
plt.imshow(imy,cmap='gray')
plt.title('y')

plt.subplot(2,2,4)
plt.axis('off')
plt.imshow(magnitude,cmap='gray')
plt.title('x**2+y**2')
```

Examples are shown below:



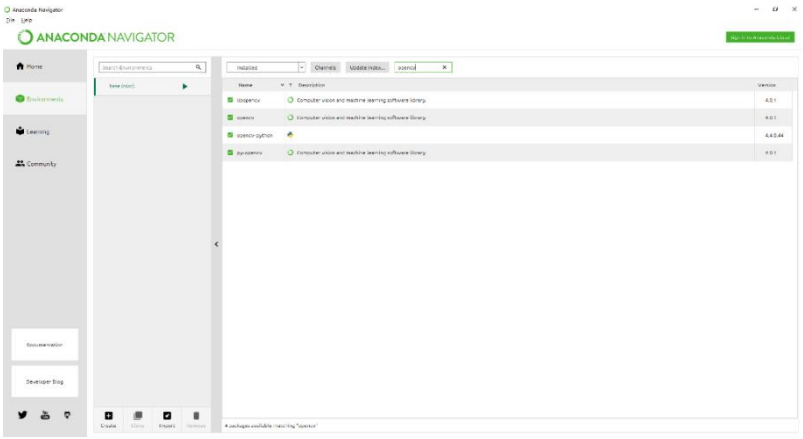
3、 Install opencv in anaconda

（1） Select environment in anaconda



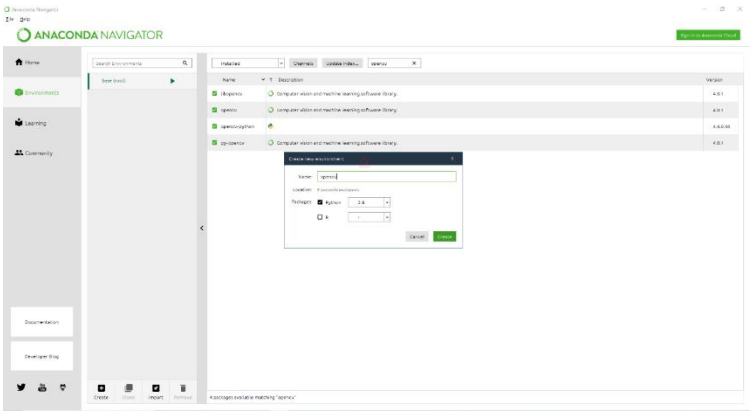
（2） Check if OpenCV is installed in anaconda

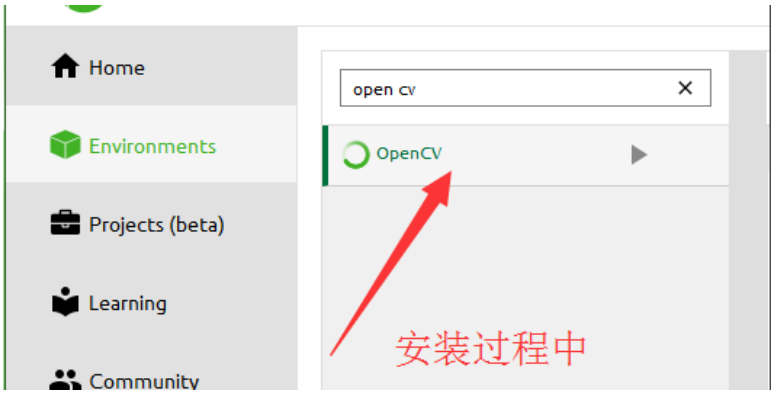
If you search for opencv under “installed” and display the following related environments, it proves that the installation has been successful.



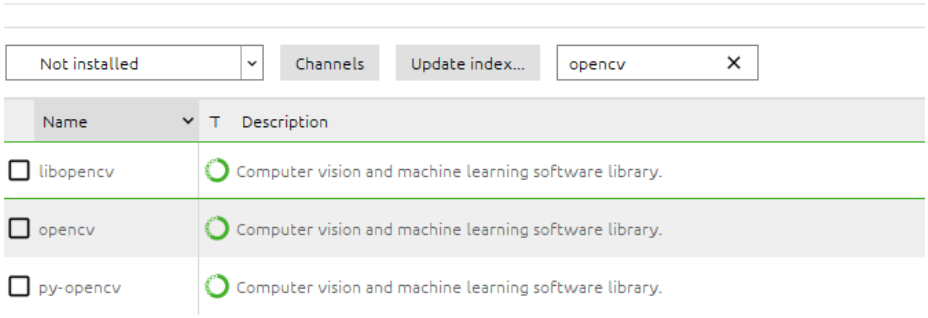
Otherwise, you need to install the OpenCV environment.

（3） Choose to create a new environment





（4） **Search for opencv, the three libraries (libopencv, opencv, py-opencv) that appear are all installed**



4、 OpenCV-python: read a picture and display

（1）cv2.imread(“a1.jpg or E:\....\a1.jpg ”): read a picture

（2）cv2.imshow( ) ： Show a picture

（3）cv2.waitKey(0)：# Waiting for user response

（4）cv2.destroyAllwindows()：Release all windows

Practice three:

(a) Use *cv2.namedWindow()* function, an image window that can be adjusted

(b) Press the s key or the Esc key to exit the image window, Save the image when exiting and convert it to “png” format

```python
#Import opencv module
import cv2

#Read a picture
tupian = cv2.imread("a1.jpg")

"""
Create a new window and set the size to be adjustable
cv2.nemedWindow("picture window name", parameter)
Parameter 1: WINDOW_AUTOSIZE cannot adjust the window size
Parameter 2: WINDOW_NORMAL adjustable window size
"""
cv2.namedWindow("img",cv2.WINDOW_NORMAL)

#Show the picture
cv2.imshow("img",tupian)

#Waiting for user response
k = cv2.waitKey(0)&0XFF

#If  press Esc
if k ==27:
    #Release all windows
    cv2.destroyAllWindows()

#If press the s key
elif k==ord("s"):
    #Save the picture and release all windows
    cv2.imwrite("a2.png",tupian)
    cv2.destroyAllWindows()

```

5、 Canny boundary detection in OpenCV.

project four:

(a) Use the *imread()* function in opencv to save the image as a grayscale image

(b) Learn and use the *canny()* function, Setting: *minVal*=100, *maxVal*=200

(c) Store the original image and the image after canny edge detection in the same image

```python
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('a1.jpg',0)
edges = cv2.Canny(img,100,200)

plt.subplot(1,2,1)
plt.imshow(img,cmap='gray')
plt.title('original')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(edges,cmap='gray')
plt.title('edge')
plt.axis('off')

plt.show()
```

Examples are shown below: