Lab Practice

1、 Gaussian Pyramid

In the downward sampling of images, the *gaussian pyramid* is used most. It will perform Gaussian kernel convolution on the image *Gi* and delete all even rows and columns in the original image, ultimately reducing the image. Among them, Gaussian kernel convolution operation is a weighted average process of the whole image. The value of each pixel is obtained by the weighted average of itself and other pixel values in the neighborhood (with different weights).

Practice One

In OpenCV, the down-sampling function is pyrDown( )：

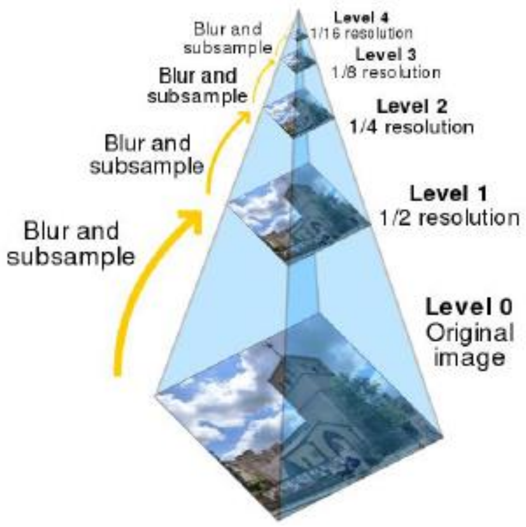
dst = pyrDown(src[, dst[, dstsize[, borderType]]]):

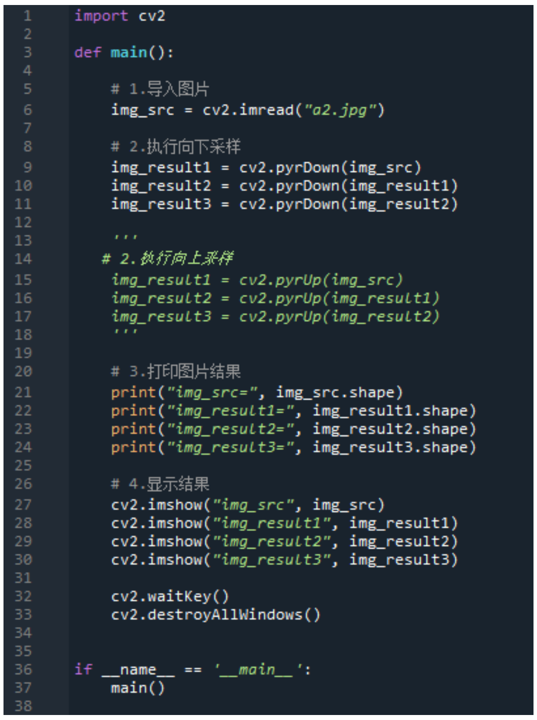
（1） *src* represents the input image

（2） *dst* represents the output image, which has the same size and type as the input image

（3） *dstsize* represents the Size of the output image. The default value is Size()

（4） *borderType* represents the pixel extrapolation method




```
import cv2

def main():

    # 1.导入图片
    img_src = cv2.imread("a2.jpg")

    # 2.执行向下采样
    img_result1 = cv2.pyrDown(img_src)
    img_result2 = cv2.pyrDown(img_result1)
    img_result3 = cv2.pyrDown(img_result2)

    '''
   # 2.执行向上采样
    img_result1 = cv2.pyrUp(img_src)
    img_result2 = cv2.pyrUp(img_result1)
    img_result3 = cv2.pyrUp(img_result2)
    '''

    # 3.打印图片结果
    print("img_src=", img_src.shape)
    print("img_result1=", img_result1.shape)
    print("img_result2=", img_result2.shape)
    print("img_result3=", img_result3.shape)

    # 4.显示结果
    cv2.imshow("img_src", img_src)
    cv2.imshow("img_result1", img_result1)
    cv2.imshow("img_result2", img_result2)
    cv2.imshow("img_result3", img_result3)

    cv2.waitKey()
    cv2.destroyAllWindows()


if __name__ == '__main__':
    main()

```

2、 Low pass filtering of Fourier transform

A low-pass filter is a filter that passes through low frequencies, attenuates high frequencies and passes through low frequencies, and is often used for blurring images.A low-pass filter, as opposed to a high-pass filter, smoothes the brightness of a pixel when the interpolation between the pixel and surrounding pixels is less than a specific value, and is often used in desiccating and blurring processes

Practice Two:

OpenCV implements the Fourier transform

The function prototype is shown below：

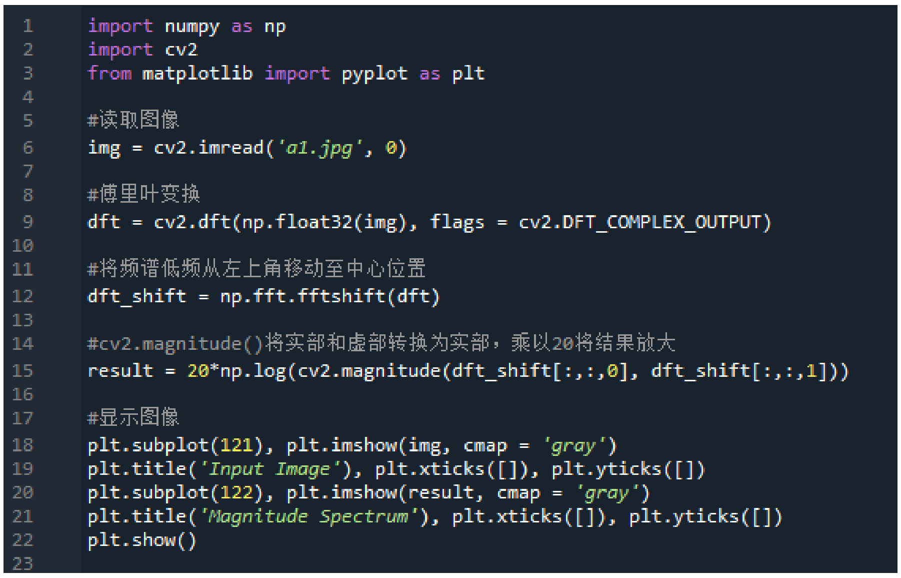
dst = cv2.dft(src, dst=None, flags=None, nonzeroRows=None)

（1） *scr* represents the input image that needs to be converted via NP.Float32 conversion format

（2） *dst* represents the output image, including the output size and size

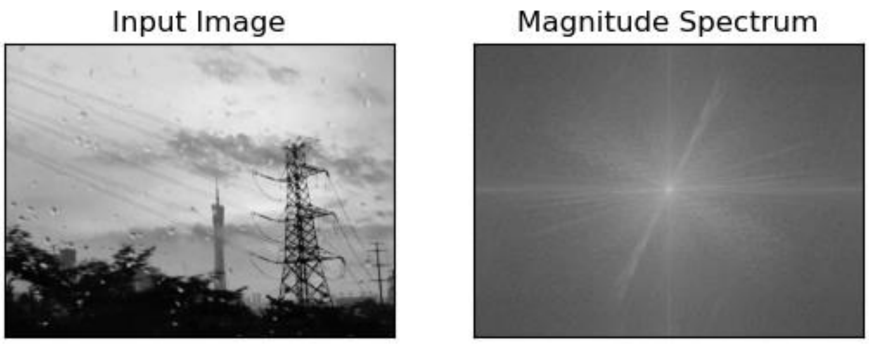
（3） *flags* represents for conversion flags. The *DFT _COMPLEX_OUTPUT* performs forward conversion of 1D or 2D real number groups, which is the fastest choice and the default function

（4） *nonzeroRows* means that when the argument is not zero, the function assumes that only the first row of the *nonzeroRows* input array (not set) or only the first row of the output array (set) contains non-zero



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

#读取图像
img = cv2.imread('a1.jpg', 0)

#傅里叶变换
dft = cv2.dft(np.float32(img), flags = cv2.DFT_COMPLEX_OUTPUT)

#将频谱低频从左上角移动至中心位置
dft_shift = np.fft.fftshift(dft)

#cv2.magnitude()将实部和虚部转换为实部，乘以20将结果放大
result = 20*np.log(cv2.magnitude(dft_shift[:,:,0], dft_shift[:,:,1]))

#显示图像
plt.subplot(121), plt.imshow(img, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(result, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()

```

Examples are shown below:



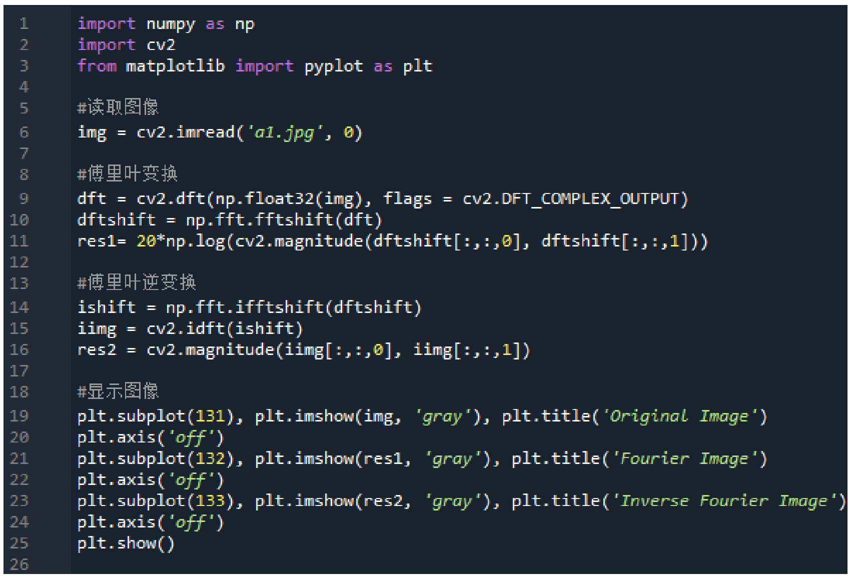
3、 *Opencv* implements *inverse Fourier transform*

In OpenCV, the inverse Fourier transform is realized by the function cv2.IDft (), and the return result depends on the type and size of the original image, which can be real or complex.

Practice Three:

dst = cv2.idft(src[, dst[, flags[, nonzeroRows]]])

(1) src: input image, including real or complex Numbers
(2) dst: output image
(3) flags: conversion flags
(4) nonzeroRows: the number of DST rows to be processed; the contents of the remaining rows are undefined.



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

#读取图像
img = cv2.imread('a1.jpg', 0)

#傅里叶变换
dft = cv2.dft(np.float32(img), flags = cv2.DFT_COMPLEX_OUTPUT)
dftshift = np.fft.fftshift(dft)
res1= 20*np.log(cv2.magnitude(dftshift[:,:,0], dftshift[:,:,1]))

#傅里叶逆变换
ishift = np.fft.ifftshift(dftshift)
iimg = cv2.idft(ishift)
res2 = cv2.magnitude(iimg[:,:,0], iimg[:,:,1])

#显示图像
plt.subplot(131), plt.imshow(img, 'gray'), plt.title('Original Image')
plt.axis('off')
plt.subplot(132), plt.imshow(res1, 'gray'), plt.title('Fourier Image')
plt.axis('off')
plt.subplot(133), plt.imshow(res2, 'gray'), plt.title('Inverse Fourier Image')
plt.axis('off')
plt.show()
```

Examples are shown below: