

Low pass và High pass dùng ma trận kernel

Ở phần này ta sẽ dùng:

Phần mềm : Vscode

Thư viện : Pillow, matplotlib, scipy (có thể dùng opencv có hàm hỗ trợ nhiều hơn, lúc đó sẽ không cần dùng matplotlib)

*Ảnh gốc:



*Import thư viện:

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

*Nhập đường dẫn đã tải tệp git: (lưu ý nhập đúng địa chỉ lưu thư mục git)

```
#Nhập đường dẫn thư mục đã tải tệp git về (lưu ý nhập đúng đầy đủ thư mục chứa tệp git không bao gồm tệp git)
dir = input("Xin moi nhap duong dan thu muc da tai git ve: ")
```

*Tạo đường dẫn cho thư mục:

```
#Tạo đường dẫn cho thư mục và ảnh:
image_path = dir + '/bai2/hinh2.JPG'
bai2_path = dir + '/bai2'
```

*Mở ảnh từ đường dẫn:

```
# Mở ảnh từ đường dẫn
img_01 = Image.open(image_path)
#img_01.show()
```

*Chuyển ảnh về ảnh xám (bước này rất quan trọng nếu không làm theo, để ảnh gốc chuyển về dạng ma trận sẽ không tương thích với các ma trận kernel)

```
#Chuyển ảnh về ảnh xám
gray = img_01.convert('L')
```

*Chuyển ảnh xám về dạng ma trận:

```
#Chuyển ảnh xám về mảng
array_data = np.array(gray)
```

*High pass kernel:

```
#High pass kernel
high_kernel = np.array([[0, -1, 0],
                        [-1, 4, -1],
                        [0, -1, 0]])
```

*Low pass kernel:

```
#Low pass kernel
Low_kernel = np.array([[1, 2, 1],
                       [2, 4, 2],
                       [1, 2, 1]])
```

*Average kernel:

```
#Average kernel
average_kernel = np.array([[1, 1, 1],
                           [1, 1, 1],
                           [1, 1, 1]])
```

*Laplace kernel:

```
#Laplace kernel
laplace_kernel = np.array([[0, 1, 0],
                           [1, -4, 1],
                           [0, 1, 0]])
```

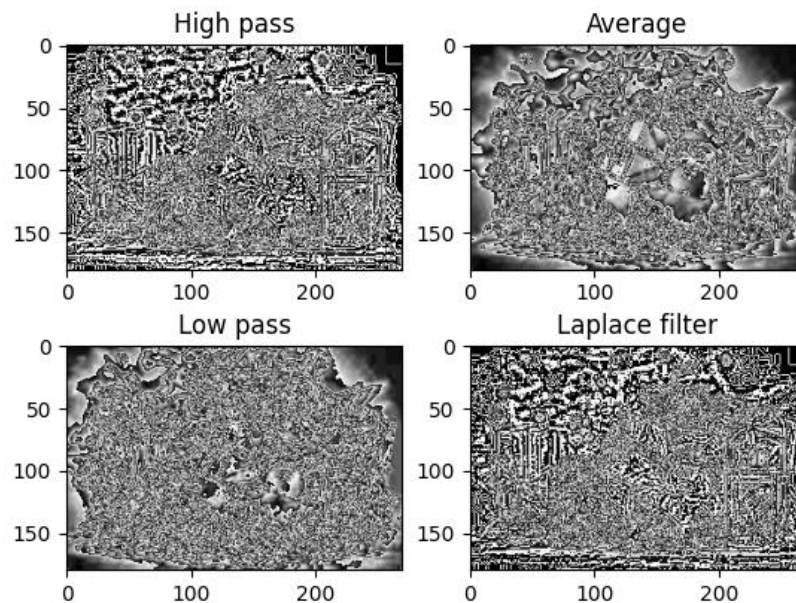
*Convolve các kernel với kênh gốc:

```
#Convolve các kernel với kênh gốc
high_pass = ndimage.convolve(array_data, high_kernel)
low_pass = ndimage.convolve(array_data, low_kernel)
average_filter = ndimage.convolve(array_data, average_kernel)
laplace_filter = ndimage.convolve(array_data, laplace_kernel)
```

*Hiển thị kết quả:

```
#Hiển thị kết quả
plt.subplot(2,2,1)
plt.title("High pass")
plt.imshow(high_pass, cmap='gray')
plt.subplot(2,2,2)
plt.title("Average")
plt.imshow(average_filter, cmap='gray')
plt.subplot(2,2,3)
plt.title("Low pass")
plt.imshow(low_pass, cmap='gray')
plt.subplot(2,2,4)
plt.title("Laplace filter")
plt.imshow(laplace_filter, cmap='gray')
plt.show()
```

*Kết quả:



Code đầy đủ:

```

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

#Nhập đường dẫn thư mục đã tải tệp git về (lưu ý nhập đúng đầy đủ thư mục chứa
tệp git không bao gồm tệp git)
dir = input("Xin moi nhap duong dan thu muc da tai git ve: ")

#Tạo đường dẫn cho thư mục và ảnh:
image_path = dir + '/bai2/hinh2.JPG'
bai2_path = dir + '/bai2'

# Mở ảnh từ đường dẫn
img_01 = Image.open(image_path)

#Chuyển ảnh về ảnh xám
gray = img_01.convert('L')

#Chuyển ảnh xám về mảng
array_data = np.array(gray)

#High pass kernel
high_kernel = np.array([[0, -1, 0],
                        [-1, 4, -1],
                        [0, -1, 0]])

#Low pass kernel
Low_kernel = np.array([[1, 2, 1],
                       [2, 4, 2],
                       [1, 2, 1]])

#Average kernel
average_kernel = np.array([[1, 1, 1],
                           [1, 1, 1],
                           [1, 1, 1]])

#Laplace kernel
laplace_kernel = np.array([[0, 1, 0],
                           [1, -4, 1],
                           [0, 1, 0]])

#Convolve các kernel với kênh gốc
high_pass = ndimage.convolve(array_data, high_kernel)
low_pass = ndimage.convolve(array_data, Low_kernel)

```

```
average_filter = ndimage.convolve(array_data, average_kernel)
laplace_filter = ndimage.convolve(array_data, laplace_kernel)

#Hiển thị kết quả
plt.subplot(2,2,1)
plt.title("High pass")
plt.imshow(high_pass, cmap='gray')
plt.subplot(2,2,2)
plt.title("Average")
plt.imshow(average_filter, cmap='gray')
plt.subplot(2,2,3)
plt.title("Low pass")
plt.imshow(low_pass, cmap='gray')
plt.subplot(2,2,4)
plt.title("Laplace filter")
plt.imshow(laplace_filter, cmap='gray')
plt.show()
```