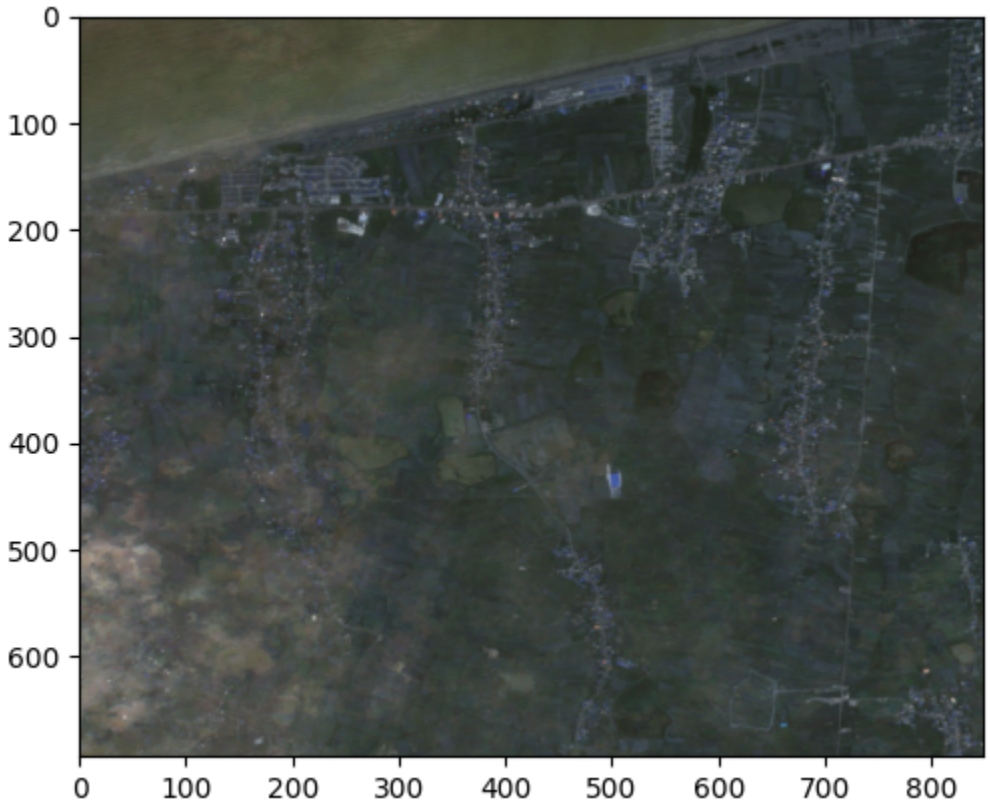
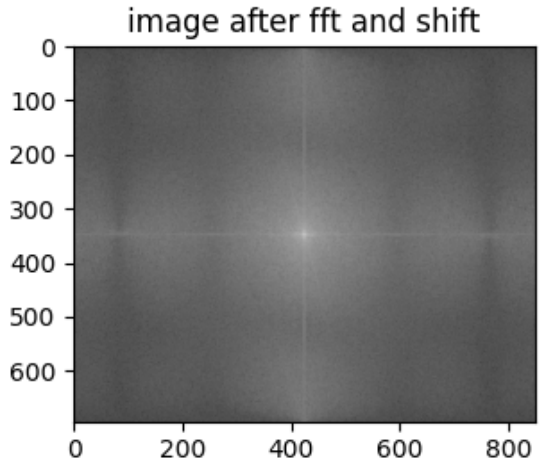
پروژه سوم
اعمال تبدیل فوریه و فیلتر در حیطه فرکانس

علیرضا ابراهیمی
۸۱۰۳۰۱۰۱۷

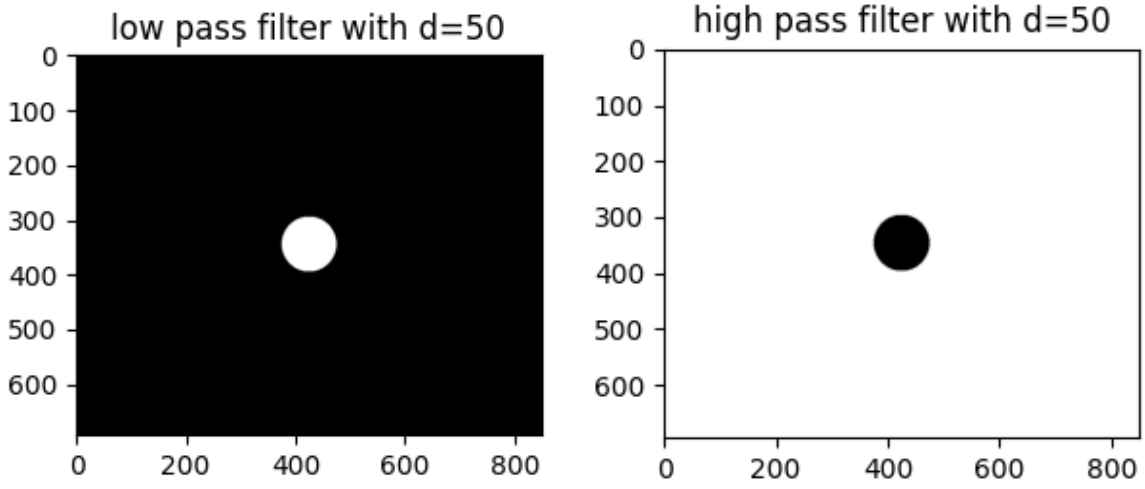
هدف در این پروژه ایجاد فضای فرکانس برای یک عکس و اعمال فیلترهای بالا گذر و پایین‌گذر بر روی آن است . بدین منظور از یک عکس که از ماهواره sentinel2 به دست آمده است استفاده شده است که true color image آن به صورت زیر است .



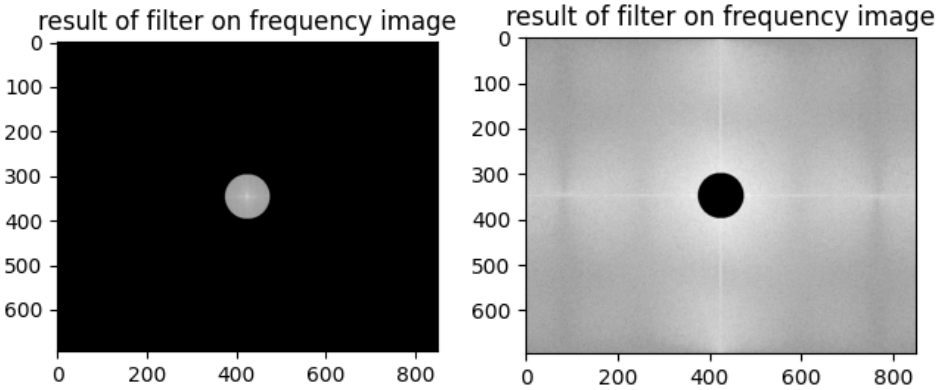
برای بردن عکس در حیطه ابتدا از fft استفاده می‌کنیم سپس بر روی آن shift اعمال می‌کنیم تا نقاط روشن در این domain در مرکز قرار گیرد .



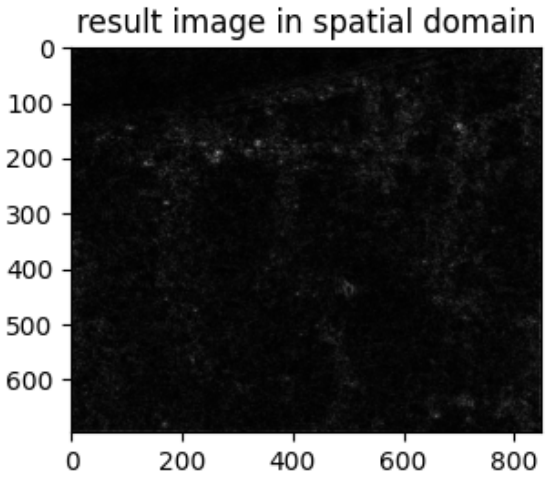
در مرحله بعد برای اعمال فیلترهای بالاگذر و پایین‌گذر ابتدا باید فیلتر ها را تشکیل دهیم که به صورت یزر می‌شود .

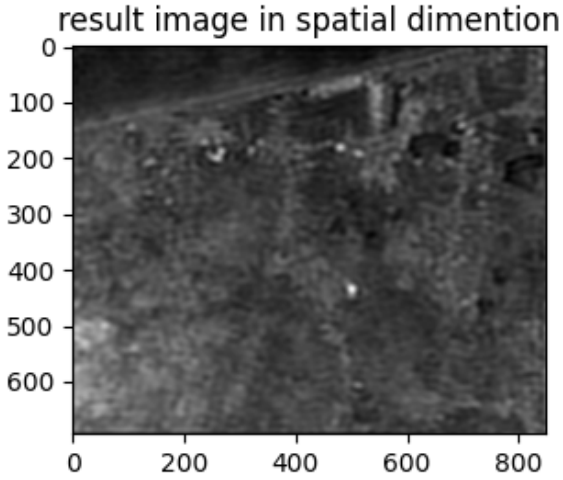


سپس این فیلتر ها را بر روی عکس در domain فرکانس اعمال می‌کنیم .

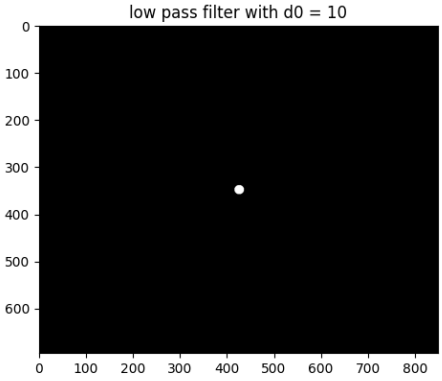


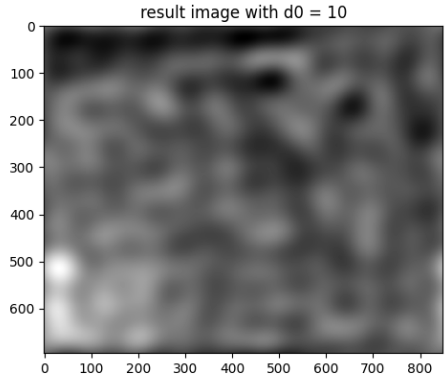
بعد از اعمال فیلترها برای مشاهده نتیجه دوباره آن را به spatial domain برمی‌گردانیم .

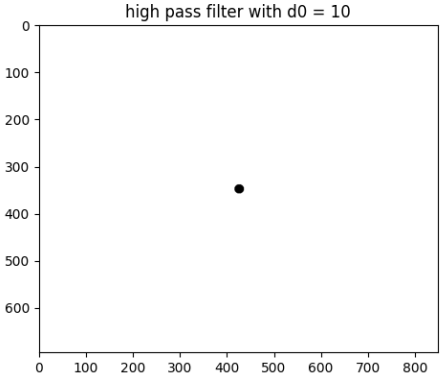


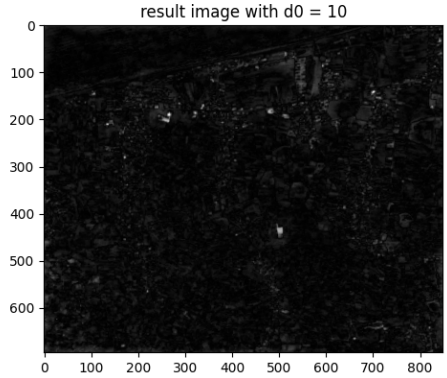


با توجه به مقادیر مختلف d0 می‌توان تفاوت در خروجی عکس را مشاهده نمود .

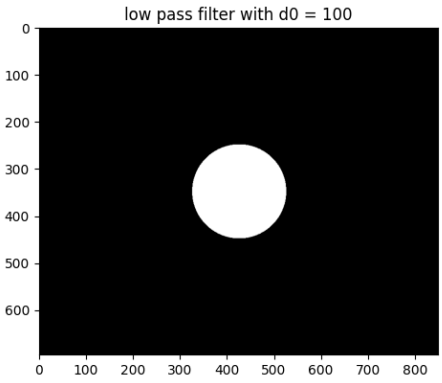


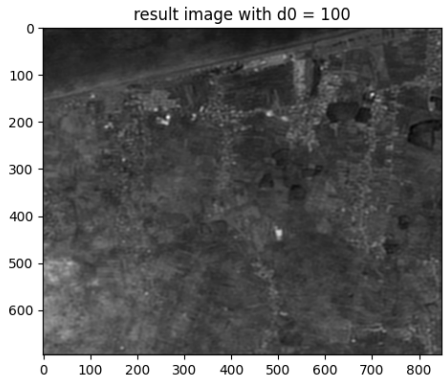


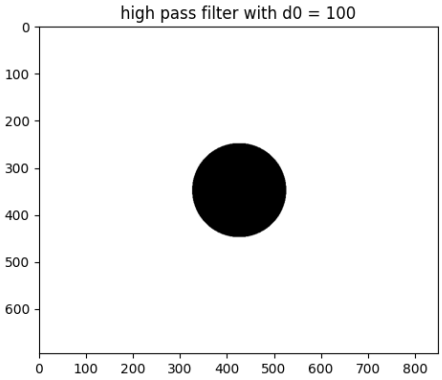


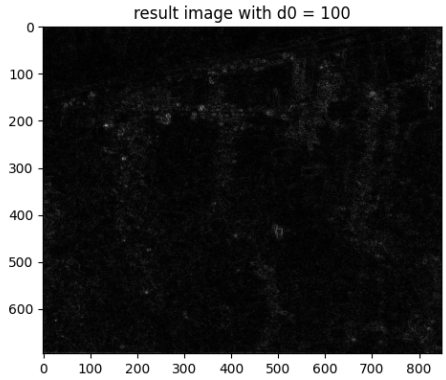


خروجی با d0=10

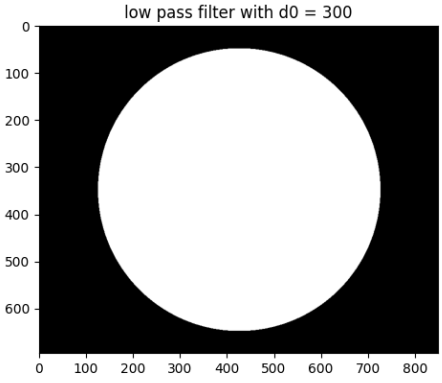


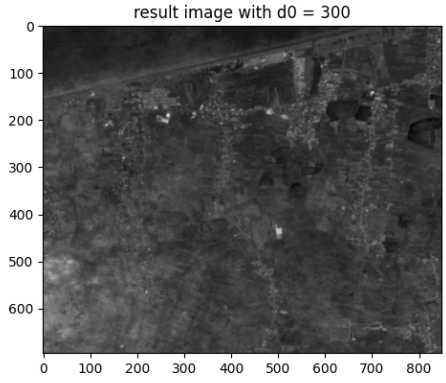


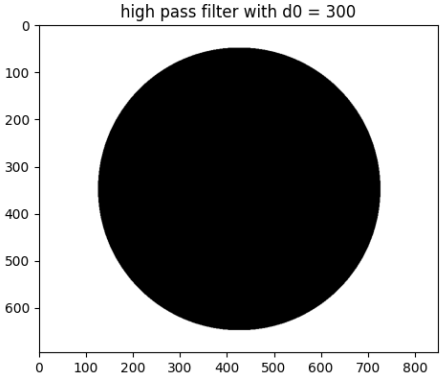


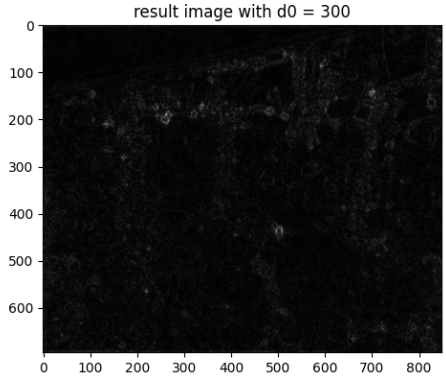


خروجی با d0=100



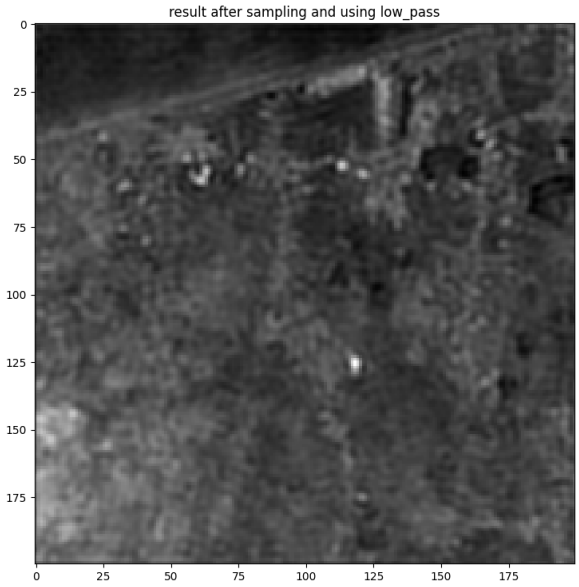


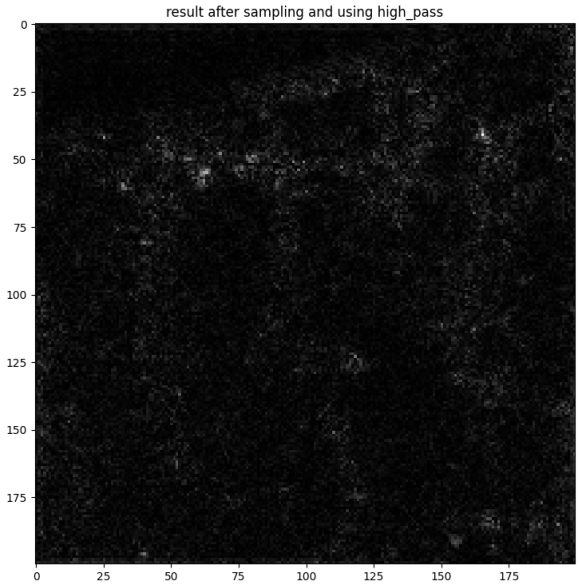




خروجی با d0=300

مشاهده می‌شود که هرچه شعاع دایره فیلتر بزرگتر می‌شود تصویر sharp تر می‌شود .






```python
import rasterio as rio
import matplotlib.pyplot as plt
import numpy as np

def UINT8(Data) :
   shape = Data.shape
   for i in range(shape[0]):
       data = Data[i , : , :]
       data = data / data.max()
       data = 255 * data
       Data[i] = data.astype(np.uint8)
   return Data


path = '/home/alireza/Desktop/seg/rectangle3.tif'
image = rio.open(path)
img = image.read()
img_vis = img[1:4 , : , :]
img_vis = UINT8(img_vis)
img_vis = img_vis.transpose(1 , 2 ,0).astype(np.uint8)
plt.imshow(img_vis)
plt.show()
```

```
def calc_2dft(input):
   ft = np.fft.fft2(input)
   return np.fft.fftshift(ft)

img = img[4, :, :]
m , n = img.shape
plt.set_cmap("gray")
ft = calc_2dft(img)
ft_sample = ft[int(m/2) - 100 : int(m/2) + 100,int(n/2) - 100 : int(n/2) + 100]

# defining ideal low_pass filter
L = np.zeros((m,n) , dtype = np.float32)
d0 = 50
for u in range(m):
   for v in range(n):
       d = np.sqrt((u - m/2)**2 + (v - n/2)**2)
       if d <= d0 :
           L[u,v] = 1
       else:
           L[u,v] = 0
H = 1 - L

LS = L[int(m/2) - 100 : int(m/2) + 100,int(n/2) - 100 : int(n/2) + 100]
HS = 1 - LS

lp_result = ft * L
hp_result = ft * H
lp_sample_res = ft_sample * LS
hp_sample_res = ft_sample * HS

#inversing low_pass filtered frequency image to spatial image
lp_inv = np.fft.ifftshift(lp_result)
hp_inv = np.fft.ifftshift(hp_result)
lp_sample_inv = np.fft.ifftshift(lp_sample_res)
hp_sample_inv = np.fft.ifftshift(hp_sample_res)


final_lpinv = np.abs(np.fft.ifft2(lp_inv))
final_hpinv = np.abs(np.fft.ifft2(hp_inv))
final_sample_lpinv = np.abs(np.fft.ifft2(lp_sample_inv))
final_sample_hpinv = np.abs(np.fft.ifft2(hp_sample_inv))

plt.figure(1)
plt.subplot(121)
plt.imshow(final_sample_lpinv)
plt.title('result after sampling and using low_pass')
plt.subplot(122)
plt.imshow(final_sample_hpinv)
plt.title('result after sampling and using high_pass')
```

```
plt.figure(2)
plt.subplot(331)
plt.imshow(img)
plt.title('image')
plt.subplot(332)
plt.imshow(np.log(abs(ft)))
plt.title('image after fft and shift')
plt.subplot(334)
plt.imshow(L)
plt.title('low pass filter with d=50')
plt.subplot(335)
plt.imshow(np.log1p(np.abs(lp_result)))
plt.title('result of filter on frequency image')
plt.subplot(336)
plt.imshow(final_lpinv)
plt.title('result image in spatial dimention')
plt.subplot(337)
plt.imshow(H)
plt.title('high pass filter with d=50')
plt.subplot(338)
plt.imshow(np.log1p(np.abs(hp_result)))
plt.title('result of filter on frequency image')
plt.subplot(339)
plt.imshow(final_hpinv)
plt.title('result image in spatial domain')

plt.show()


D0 = [10 , 100 , 300]

for i in range(len(D0)):
   d0 = D0[i]
   for u in range(m):
       for v in range(n):
           d = np.sqrt((u - m / 2) ** 2 + (v - n / 2) ** 2)
           if d <= d0:
               L[u, v] = 1
           else:
               L[u, v] = 0
   H = 1 - L
   lp_result = ft * L
   hp_result = ft * H
   lp_inv = np.fft.ifftshift(lp_result)
   hp_inv = np.fft.ifftshift(hp_result)
   final_lpinv = np.abs(np.fft.ifft2(lp_inv))
   final_hpinv = np.abs(np.fft.ifft2(hp_inv))
   plt.figure(2)
   plt.subplot(221)
   plt.imshow(L)
   plt.title(f'low pass filter with d0 = {d0}')
   plt.subplot(222)
```

```
    plt.imshow(final_lpinv)
    plt.title(f'result image with d0 = {d0}')
    plt.subplot(223)
    plt.imshow(H)
    plt.title(f'high pass filter with d0 = {d0}')
    plt.subplot(224)
    plt.imshow(final_hpinv)
    plt.title(f'result image with d0 = {d0}')

    plt.show()
```