

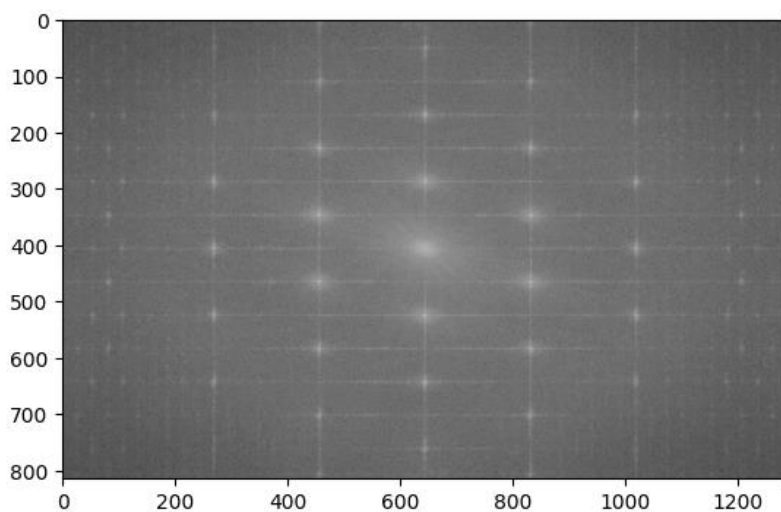
# تمرین شماره ۵

سینا اسکندری ۹۷۵۲۱۰۵۴

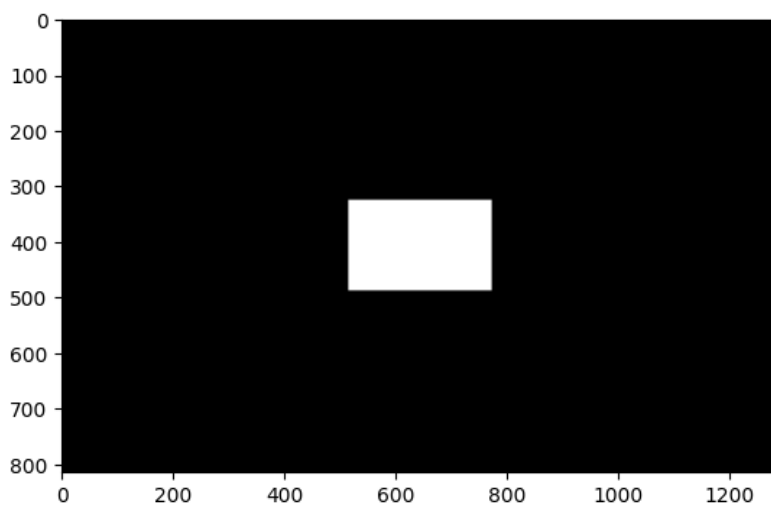
۱-

(الف)

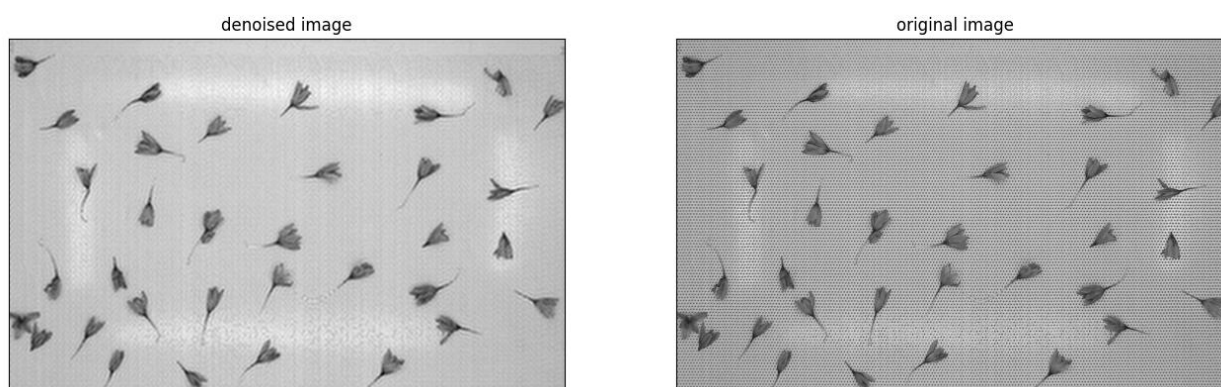
برای حذف نویز ابتدا تبدیل فوریه تصویر را رسم می کنیم.



برای حذف نویز از فیلتر زیر استفاده می کنیم.



مقایسه تصاویر حاصل شده:

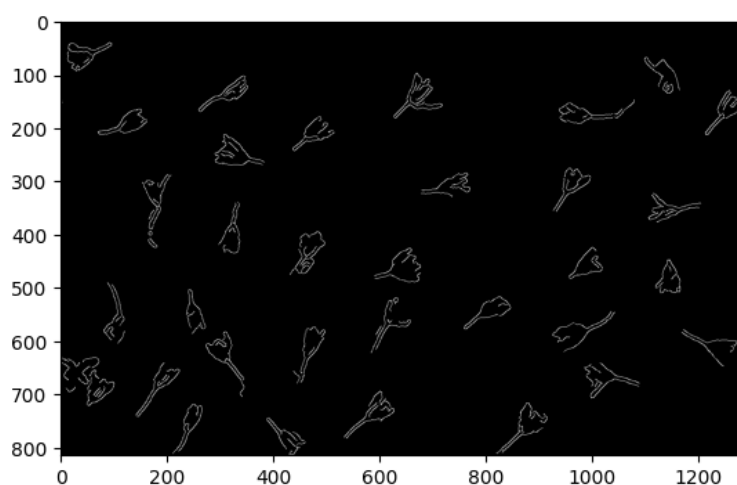


(ب)

از لبه یاب canny موجود در کتابخانه opencv استفاده می کنم. این تابع ۳ ورودی می گیرد که اولی عکس و دومی و سومی میزان سطح آستانه ای است که در مرحله آستانه گذاری دو سطحی الگوریتم canny استفاده می شود.

```
edge = cv2.Canny(denoised, 50, 120)
plt.imshow(edge, cmap='gray')
```

مقادیر ۵۰ و ۱۲۰ با امتحان کردن چندین عدد مختلف بدست آوردم که به نظر بهترین حالت است.



(ج)

با استفاده از تابع Sobel موجود در کتابخانه opencv گرادیان های تصویر در راستای x و y حساب می کنیم.

```
grad_x = cv2.Sobel(edge, cv2.CV_64F, 1, 0)
grad_y = cv2.Sobel(edge, cv2.CV_64F, 0, 1)
```

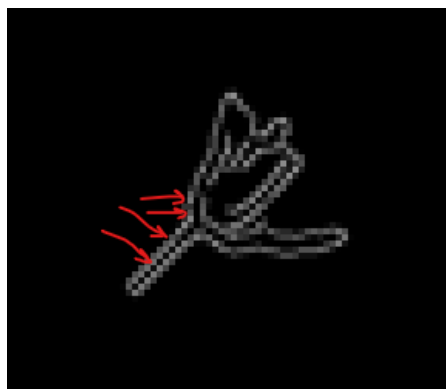
✓ 0.6s

```
dir = np.arctan2(grad_y, grad_x)
```

✓ 0.7s

(د)

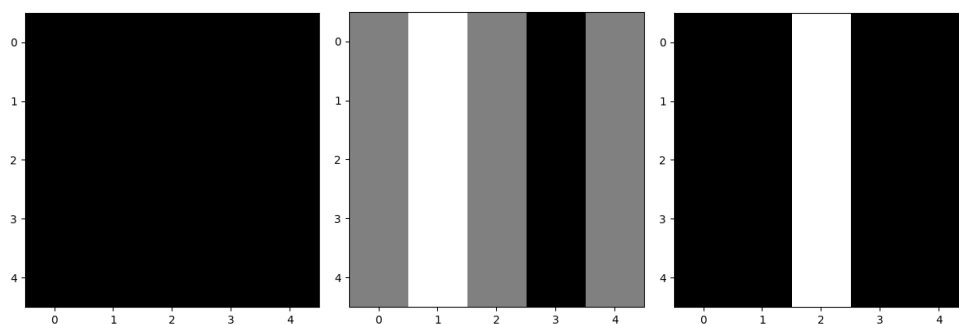
می توانیم در قسمت هایی که جهت گرادیان تغییر کرده برش بزنیم



مثلا مشخص است جهت گرادیان قسمت های بالایی با پایینی متفاوت هستند و می توان در نقطه ای که تغییر کرده برش زد.

-۳

برای ماتریس ۵ در ۵ زیر خروجی sobel در راستای x و y به ترتیب زیر می شود.



-۴

ابتدا مختصات نقاط را با دستور np.where پیدا می کنیم سپس با استفاده از فرمول صفحه ۲۲ جلسه ۸ معادله خط را بدست می آید.

```
x, y = np.where(im<100)
✓ 0.4s

m = (np.average(x) * np.average(y) - np.average(x*y)) / (np.average(x)**2 - np.average(x**2))
c = np.average(y) - m * np.average(x)
print(f'y = {m}x + {c}')
✓ 0.7s

y = 1.3709995907272559x + -202.96710625458064
```

منابع:

<https://www.geeksforgeeks.org/python-opencv-canny-function/>

[https://docs.opencv.org/4.x/d5/d0f/tutorial\\_py\\_gradients.html](https://docs.opencv.org/4.x/d5/d0f/tutorial_py_gradients.html)