

بسمه تعالی



دانشکده مهندسی کامپیوتر

بینایی کامپیوتر

نام استاد: دکتر محمدی

تمرین پنجم

نام دانشجو: آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

آبان ۱۴۰۱

فهرست

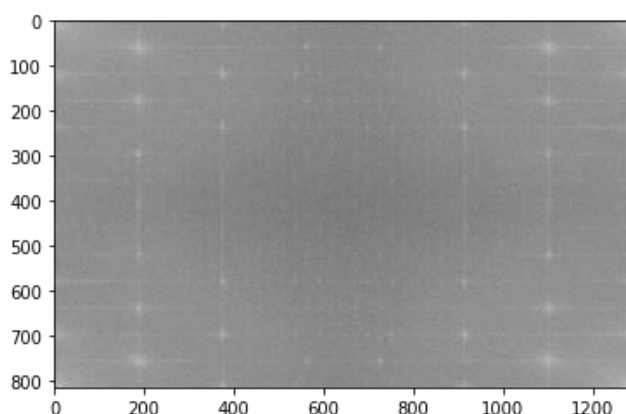
۳	پاسخ سوال اول
۳	الف)
۴	ب)
۵	ج)
۵	د)
۵	منابع
۶	پاسخ سوال دوم
۷	پاسخ سوال سوم
۸	منابع
۹	پاسخ سوال چهارم
۹	منابع

پاسخ سوال اول

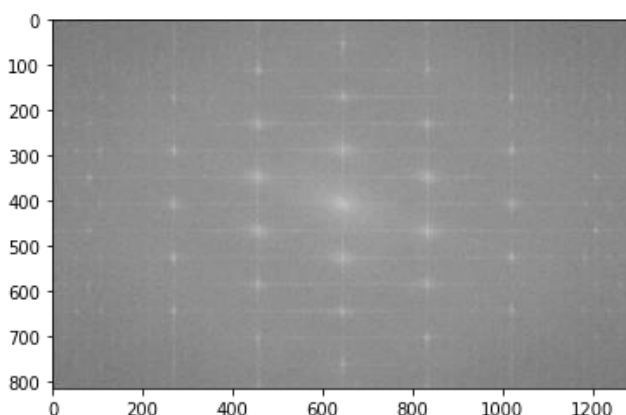
پیاده سازی این سوال در بخش Q1 در نوتبوک HW5.ipynb ضمیمه شده است.

الف)

ابتدا تبدیل فوریه تصویر را به دست می آوریم. بخش حقیقی آن را در زیر میبینید.

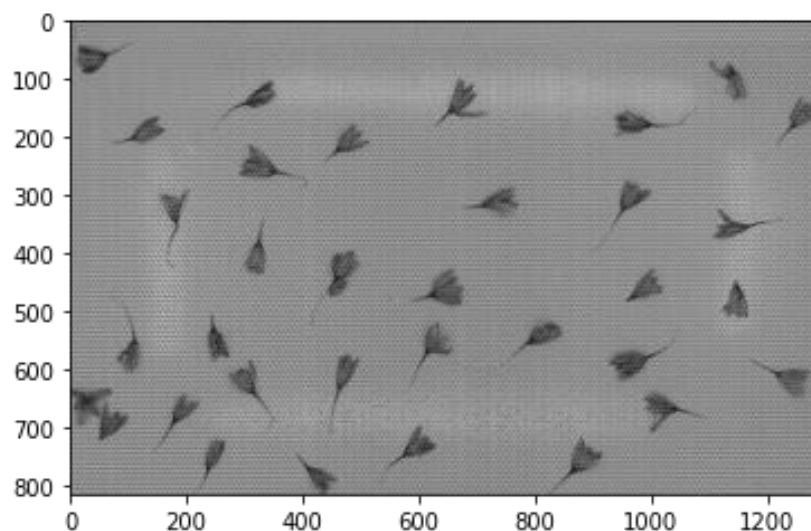


بعد آن را شیف्ट می‌دهیم به نحوی که روشن ترین نقطه یعنی $(0,0)$ در مرکز باشد. بخش حقیقی را در زیر میبینید.



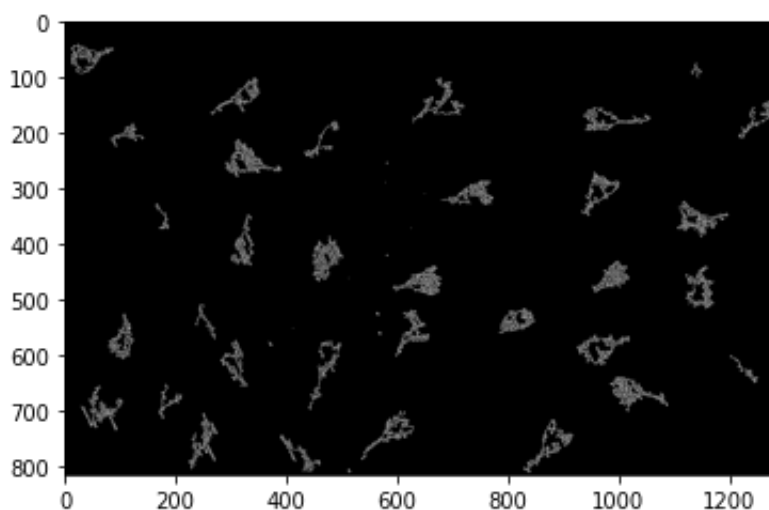
حالا چون نویز متناوب داریم، در دامنه فرکانسی میتوانیم برخی ضرایب خاص فوریه را صفر کنیم و این نویز کاهش پیدا کند، بدون این که جزئیات عکس دچار تغییر گسترده ای بشود. که ۲۰ نقطه روشن عکس را انتخاب کرده (آن هایی که مقدار بیش از ۱۴.۵ دارند، که این عدد را با آزمون و خطا به دست آوردم) و آن ضرایب حقیقی فوریه آن ها را صفر میکنیم.

بعد عکس تبدیل فوریه را با ضرایب جدید محاسبه میکنیم و به این شکل میرسیم که نویز کمتری دارد:



(ب)

از پیاده سازی کتابخانه OpenCV استفاده میکنیم که باید دو ورودی **threshold** به آن بدهیم و بقیه مراحل ها را خودش انجام می دهد. با قرار دادن دو عدد ۱۰۰ و ۶۰۰ به عکس زیر میرسیم که به خوبی گل ها را یافته است:



ترشلد دوم باید عدد زیادی باشد که گرادیان های بزرگ فقط باقی بماند و اثرات نویز زمینه که هنوز کامل از بین نرفته است، از ترشلد دوم بزرگتر نباشند. ترشلد اول باید عددی باشد که همه ی لبه گل ها از آن بزرگتر است و ۱۰۰ عدد مناسبی است که نسبتا اندازه گرادیان کوچک است.

ج)

با توابع `numpy` به راحتی پیاده سازی کردم و جهت گرادیان تک تک نقاط را میدانیم. که تقریباً ۴ درصد نقاط گرادیان غیر صفر داشتند، چون عمده عکس حاصل از `candy` را نقاط سیاه ثابت تشکیل میدادند و گرادیان در نقاط مشابه با مجاور برابر صفر است.

د)

چون قاعدتا جهت برش باید عمود بر جهت اتصال ساقه و گلبرگ باشد، در این نقاط `arctan` را محاسبه میکنیم و زاویه گلبرگ در تصویر را حساب میکنیم. سپس چون خروجی `arctan` بین $-\pi$ تا π است، آن را به درجه `map` میکنیم. و جهت تیغه را با رابطه $90-x$ حساب میکنیم که x همان زاویه گلبرگ و ساقه در تصویر است.

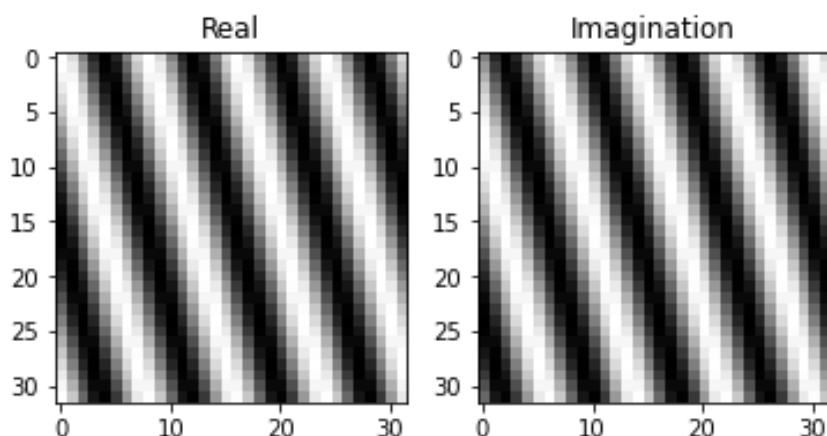
منابع

- <https://stackoverflow.com/questions/59179262/undo-np-fft-fft2-to-get-the-original-image>
- <https://stackoverflow.com/questions/49732726/how-to-compute-the-gradients-of-image-using-python>

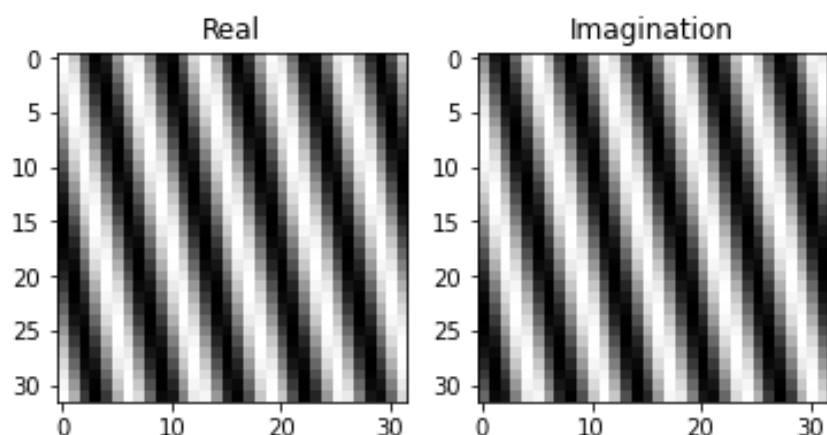
پاسخ سوال دوم

پیاده سازی این سوال در بخش Q2 در نوتبوک HW5.ipynb ضمیمه شده است.

ابتدا مطابق شکل داده شده، ماتریسی 32×32 میسازیم که فقط نقطه $[31,4]$ آن سفید است و مابقی سیاه هستند. میبینیم که طیف فرکانسی آن که با numpy محاسبه میکنیم، در دو بخش حقیقی و موهومی به صورت زیر است:



حالا آن نقطه را هم سیاه میکنیم و فقط نقطه $[31,5]$ را سفید میکنیم (انگار شیفت به راست داده ایم). و به این خروجی ها برای طیف فرکانسی میرسیم:



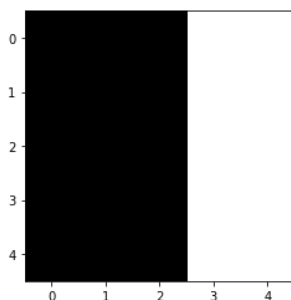
که نشان می دهد، طیف فرکانسی به مکان وابسته است و بر اساس مکان اشیا موجود در تصاویر به دست می آید. و با زیاد شدن عرض نقطه سفید، تعداد سیاه و سفیدها در این شکل افزایش می یابد. و این موضوع در بخش موهومی و حقیقی هر دو صادق است.

توضیح: در این سوال، طیف فرکانسی $(0,0)$ را به مرکز نیاوردم چون هدف مقایسه آن موضوع نبود.

پاسخ سوال سوم

پیاده سازی این سوال در بخش Q3 در نوتبوک HW5.ipynb ضمیمه شده است.

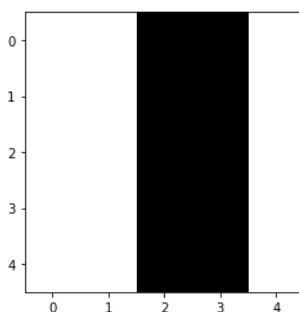
ابتدا تصویری 5×5 ایجاد میکنیم که نیمی از آن سفید است. چون هنگام نمایش باید بزرگتر از ۵ پیکسل باشد، scale میکند و ضخیم به نظر میرسد.



می دانیم برای پیدا کردن لبه ی عمودی، باید مشتق عمودی بگیریم. فیلتر sobel برای مشتق افقی به این صورت است:

-1	0	+1
-2	0	+2
-1	0	+1

با کدی که در سوال سوم تمرین سری قبل زدیم، این فیلتر را بر روی تصویر اول که ماتریسی 5×5 بود اعمال میکنیم و به این تصویر میرسیم:



میتوانیم مقادیر ماتریسها را هم ببینیم. که مشخص است قسمت لبه تشخیص داده شده است. شکل سمت چپ پیش از اعمال فیلتر و سمت راست پس از آن است:

```
[[ 0.  0.  0. 255. 255.]
 [ 0.  0.  0. 255. 255.]
 [ 0.  0.  0. 255. 255.]
 [ 0.  0.  0. 255. 255.]
 [ 0.  0.  0. 255. 255.]]
```

```
[[ 0.  0. -1020. -1020.  0.]
 [ 0.  0. -1020. -1020.  0.]
 [ 0.  0. -1020. -1020.  0.]
 [ 0.  0. -1020. -1020.  0.]
 [ 0.  0. -1020. -1020.  0.]]
```

منابع

- اسلایدهای جلسه هشتم کلاس
- <https://stackoverflow.com/questions/49732726/how-to-compute-the-gradients-of-image-using-python>

پاسخ سوال چهارم

پیاده سازی این سوال در بخش Q4 در نوتبوک HW5.ipynb ضمیمه شده است.

ابتدا تصویر را در حالت grayscale میخوانیم. سپس میبینیم که تمام تصویر از مقادیر ۹۸ یا ۲۵۵ است که ۹۸ ها نمایانگر خط ما هستند. (البته در دنیای واقعی احتمالا باید این را با فیلترهای پیدا کننده لبه بباییم.) پس تمام x و y های این نقاط را در لیست های جدا میریزیم. و بعد به راحتی با میانگین در `numpy`، طبق این فرمول های جزوه، m و c خط را محاسبه میکنیم:

$$m = \frac{\bar{x}\bar{y} - \overline{xy}}{\bar{x}^2 - \overline{x^2}}$$

$$c = \bar{y} - m\bar{x}$$

که به دست می آید:

$$m = 1.37, \quad c = -202.97$$

در نتیجه معادله خط برابر است با:

$$Y = 1.37x - 202.97$$

منابع

- ویدئو جلسه هشتم کلاس