

*پردازش تصاویر دیجیتال*

گزارش تمرین چهارم : آشنایی با پردازش تصاویر در حوزه فرکانس

**نام استاد درس :** دکتر حامد آذرنوش

**نام و شماره دانشجویی دانشجو :** سپهر کلانکی - ۹۸۳۳۰۵۸

**تاریخ تحویل تمرین :** ۲۴/۹/۱۴۰۱

**فهرست مطالب**

|  |  |
| --- | --- |
| **عنوان** | **صفحه** |

[سوال اول 1](#_Toc121004805)

[سوال دوم 2](#_Toc121004806)

[آ) 2](#_Toc121004807)

[ب) 2](#_Toc121004808)

[ج) 3](#_Toc121004809)

[سوال سوم 4](#_Toc121004810)

[آ) 4](#_Toc121004811)

[ب) 4](#_Toc121004812)

[ج) 5](#_Toc121004813)

[د) 5](#_Toc121004814)

[سوال چهارم 7](#_Toc121004815)

[آ) 7](#_Toc121004816)

[ب) 8](#_Toc121004817)

[ج) 9](#_Toc121004818)

[د) 10](#_Toc121004819)

[ه) 11](#_Toc121004820)

# سوال اول

# سوال دوم

## آ)

فیلتر میانه نویز های نمک و فلفل (ضربه‌ای یا ایمپالسی )را حذف میکند و فیلتر میانگین، هم نویز های پله‌ای

# سوال سوم

## ۱)

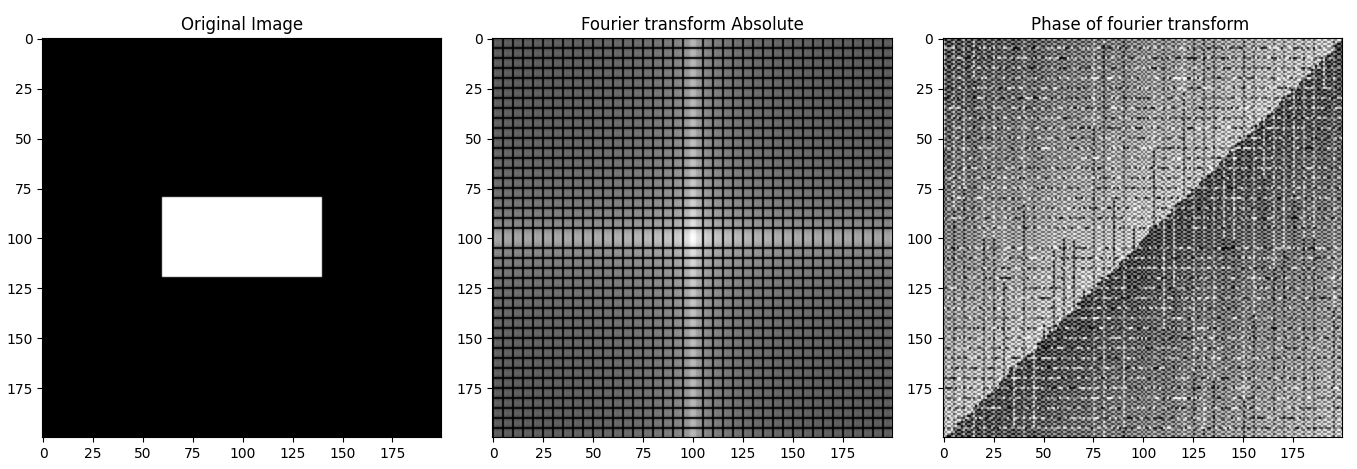
مستطیل با ابعاد ۴۰ در ۸۰ ساختم.

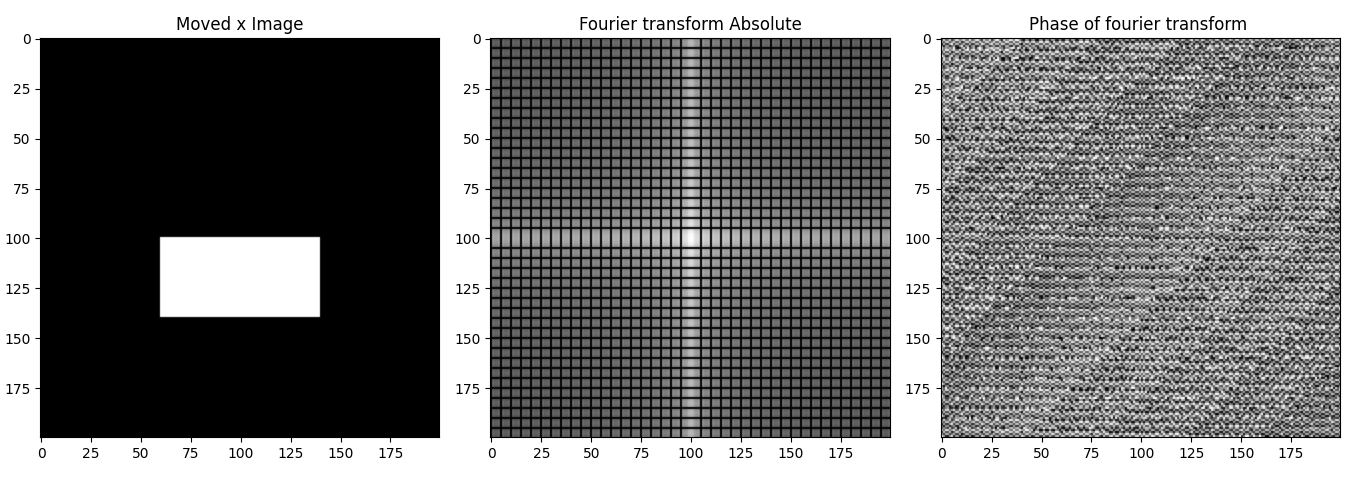
## ۲)

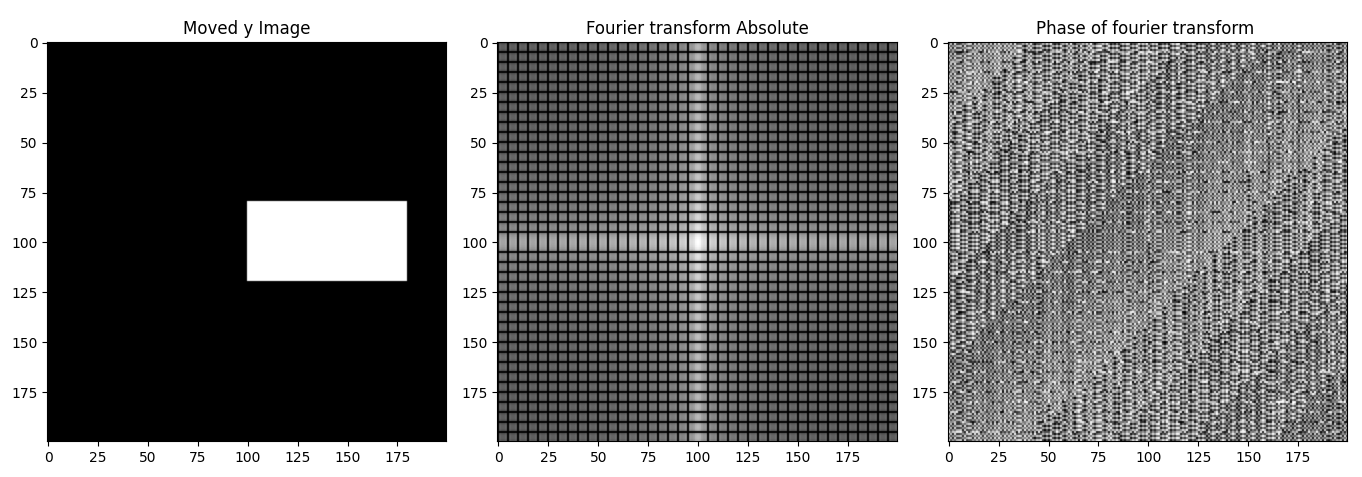
در هنگام استفاده از تابع wrap affine از Open CV فقط باید به این نکته توجه داشته باشیم که جهت مختصات x و y در numpy و Open CV باهم فرق دارند!

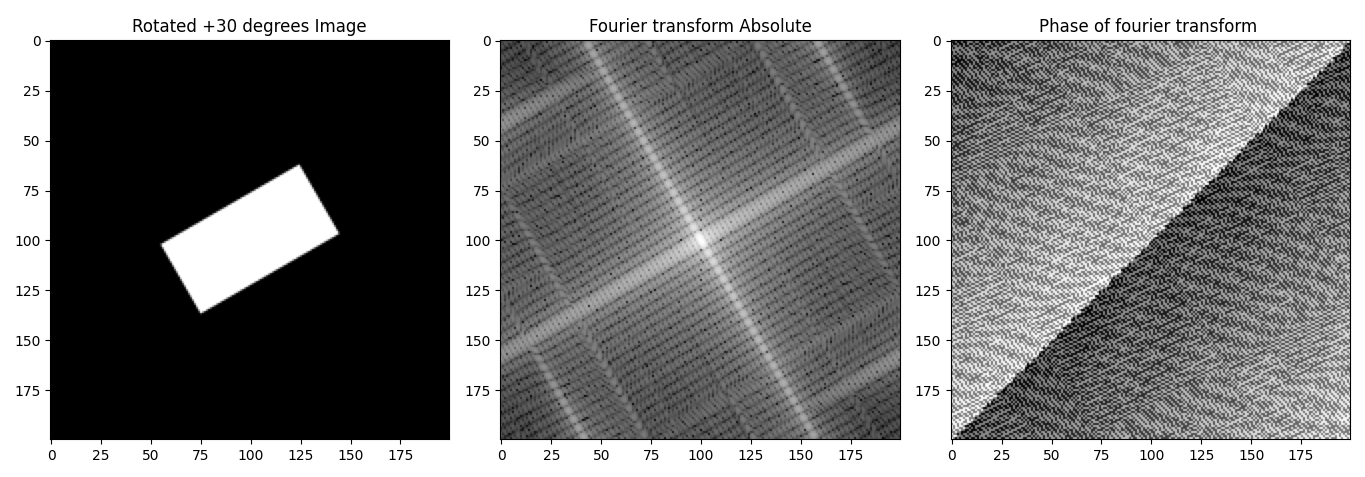
حالا اینجا چون تصویر اصلیمون یک تصویر مربعی ۲۰۰ در ۲۰۰ بود تفاوتی ایجاد نمیکرد.

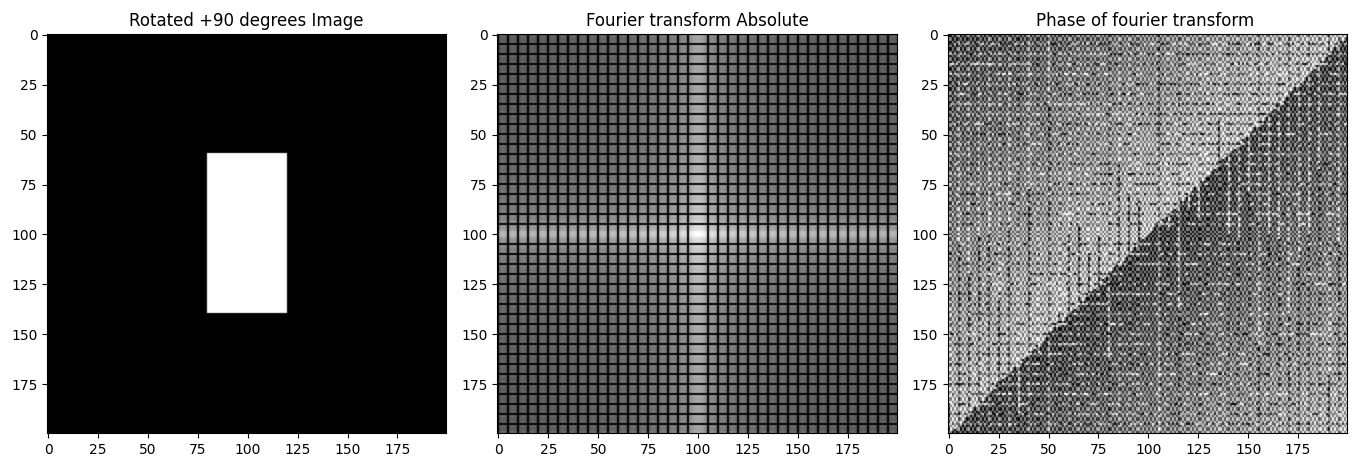
## ۳)









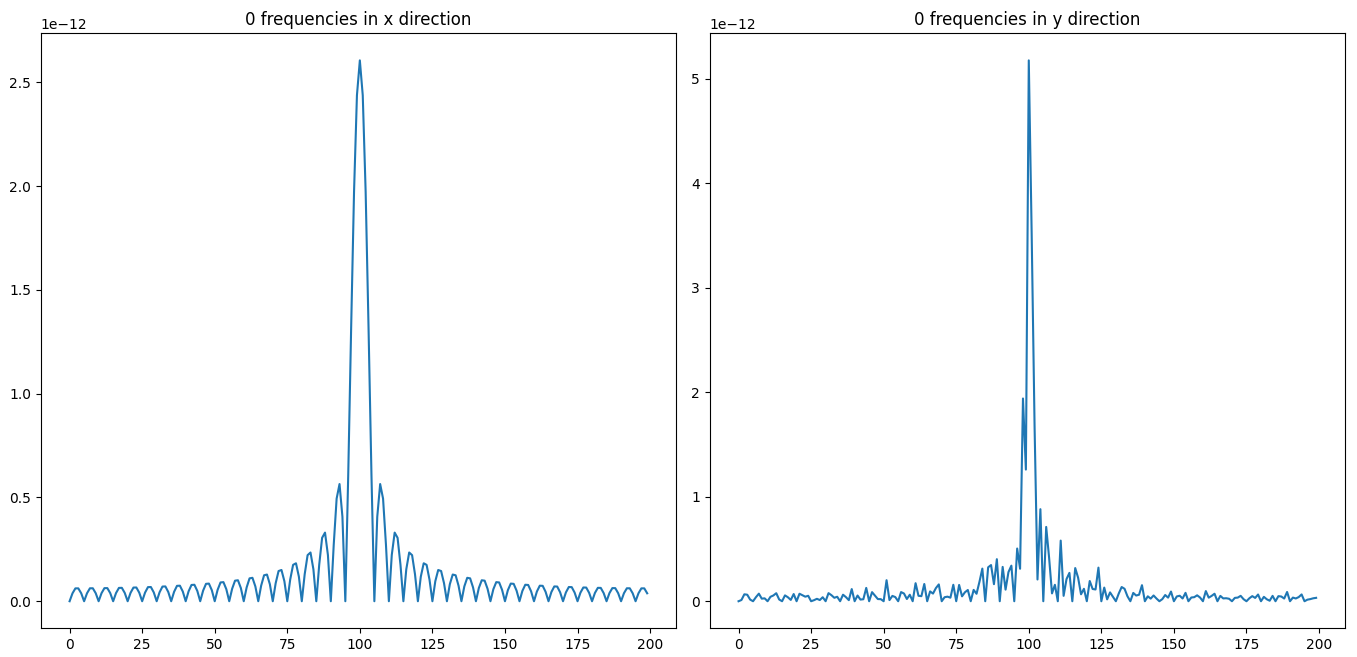


با جابه جا کردن تصویر، اندازه تبدیل فوریه فرقی نمیکند و با چرخاندن تصویر اصلی، تبدیل فوریه هم به همان اندازه و در همان جهت دوران پیدا می‌کند.

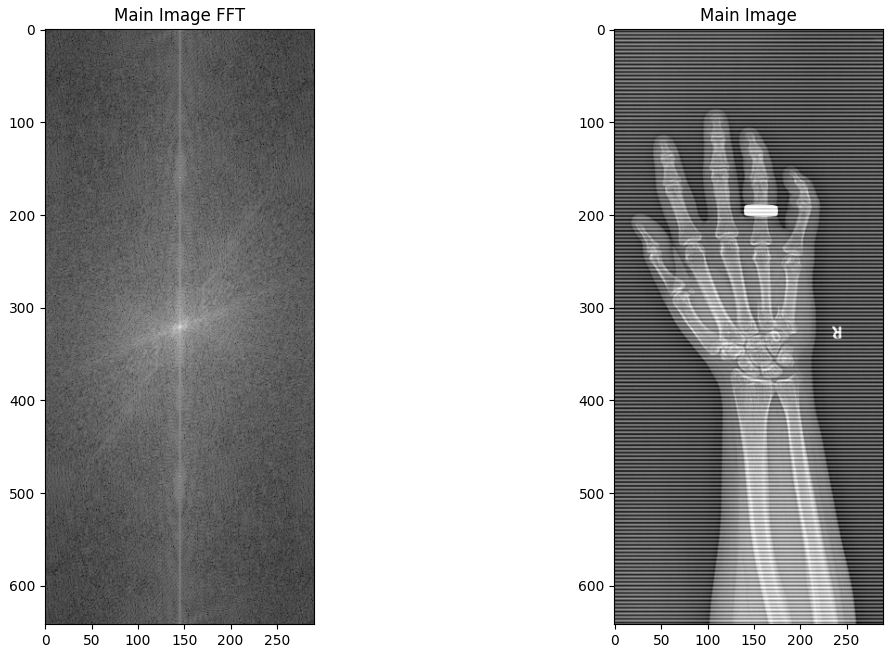
## ۴)



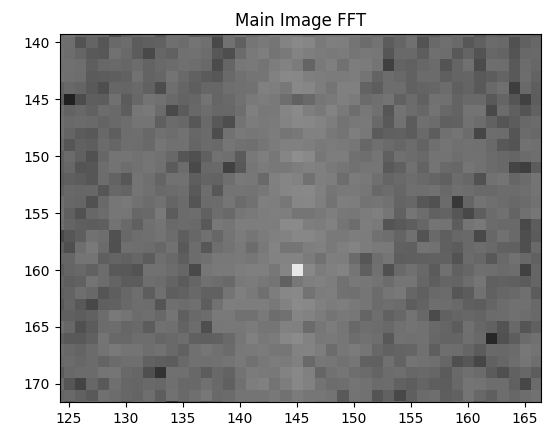
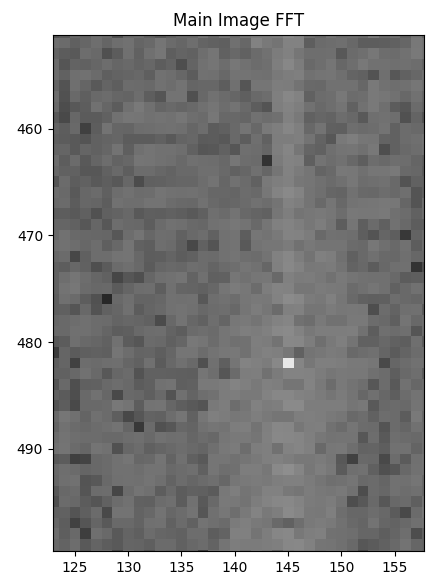
به این ترتیب بردار های فرکانس های صفر در جهت x و y را استخراج کردم.



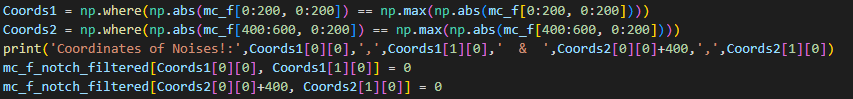
## ۵ و ۶ و ۷)



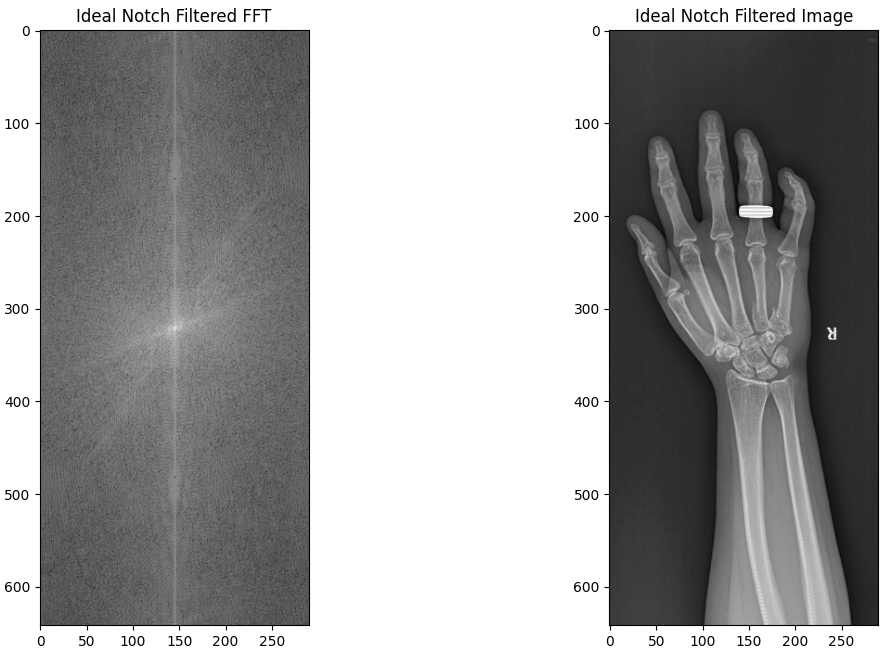
در تبدیل فوریه‌ی تصویر در محدوده‌ی پیکسل های 100 تا 200 در جهتx و 120 تا 170 دو نقطه‌ی سفید وجود دارد که نویز هستند:



به کمک این دستور مختصات این نقاط سفید را پیدا کردم و صفرشان کردم:



نتایح خروج:



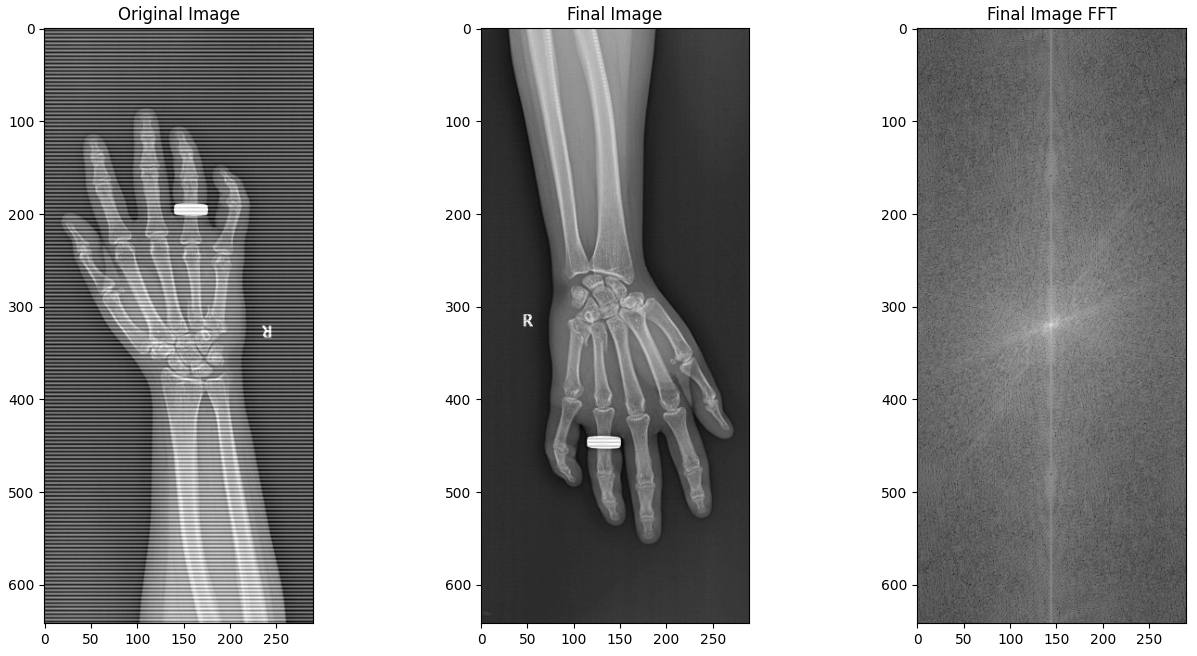
میبینیم که دقیقا زدیم تو خال و نویز دوره ای کللن حذف شدن!

## ۸ و ۹ و ۱۰)

هدف اینه تصویر در حوزه مکان ۱۸۰ درجه بچرخد ولی با اعمال تبدیل بروی حوزه فرکانسش!

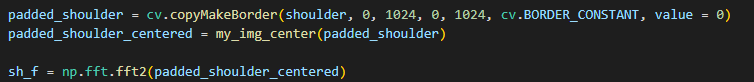
در بخش قبلی همین سوال دیدیم که معادل چرخیدن در مکان و فرکانس باهم برابرن.

لذا تصویر را میبریم به حوزه فرکانس، ۱۸۰ درجه میچرخانیم (به کمک دستور np.flip) سپس فوریه معکوس گرفته و به حوزه مکان بر میگردیم:



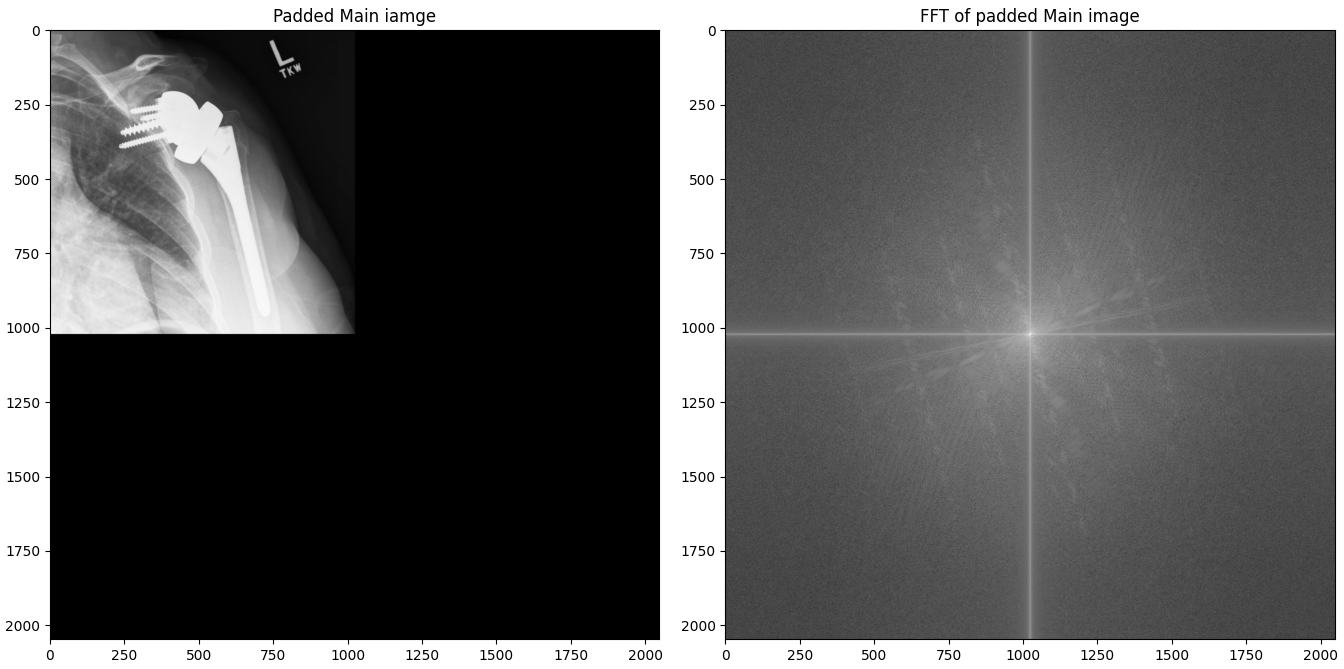
# سوال چهارم

## ۱ و ۲)



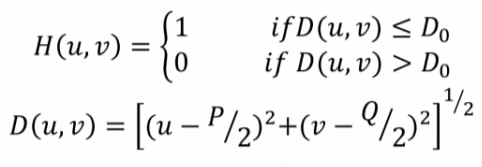
ابتدا تصویر را پد کردم سپس سنتر کردم و از تصویر و پد و سپس سنتر شده، تبدیل فوریه گرفتم.

خروجی:

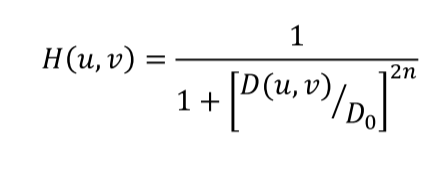


## ۳)

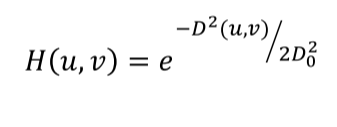
به کمک این روابط توابع فیلتر‌هارا نوشتم. همه فیلتر ها یک تصویر و یک فرکانس قطع به عنوان ورودی میگیرند و فیلتر باتر ورث یک ورودی اضافه به نام درجه فیلتر هم میگیرد که من آنرا ۲ در نظر گرفتم.



فیلتر ایده‌آل:



فیلتر باتر ورث:

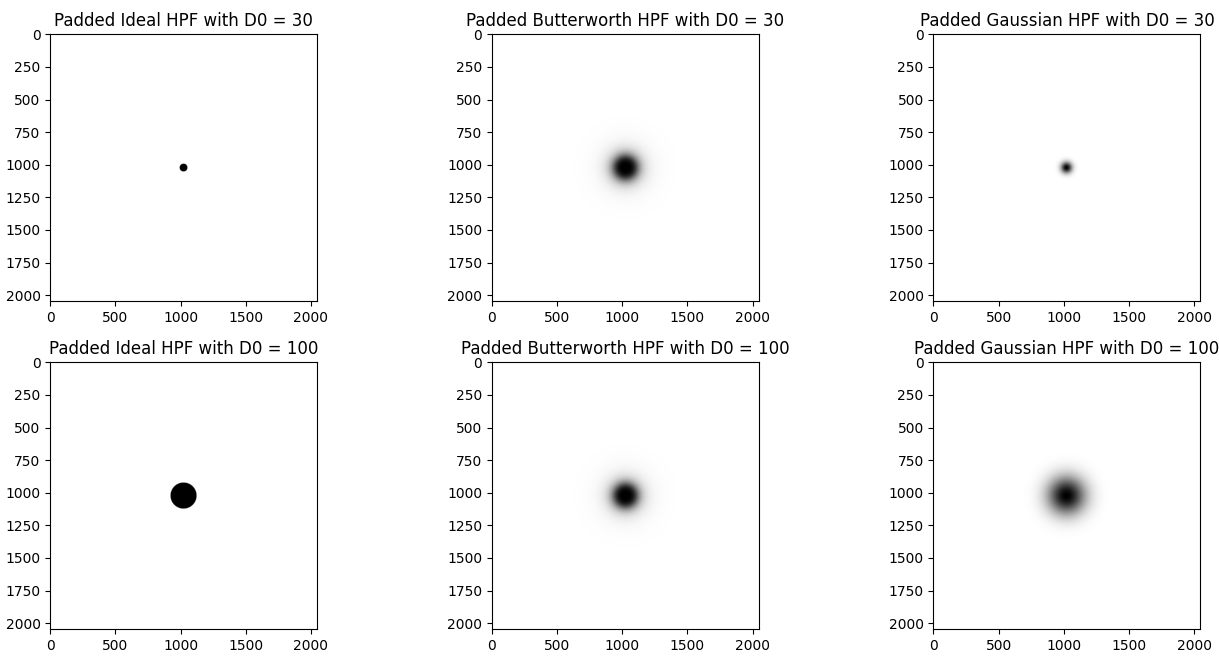


فیلتر گاوسین:

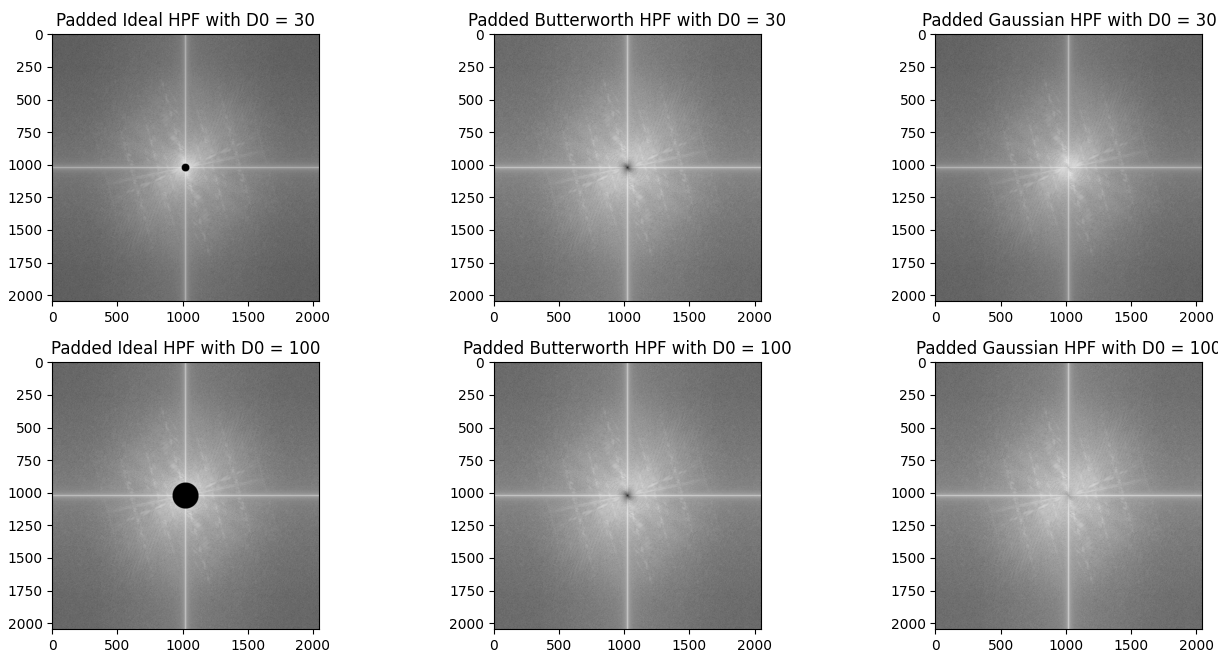
نکته‌ی مهم این است که این فیلتر‌ها پایین‌گذرند و برای بالاگذر شدنشان باید آن‌هارا از یک کم کنیم (به همین سادگی)

## ۴)

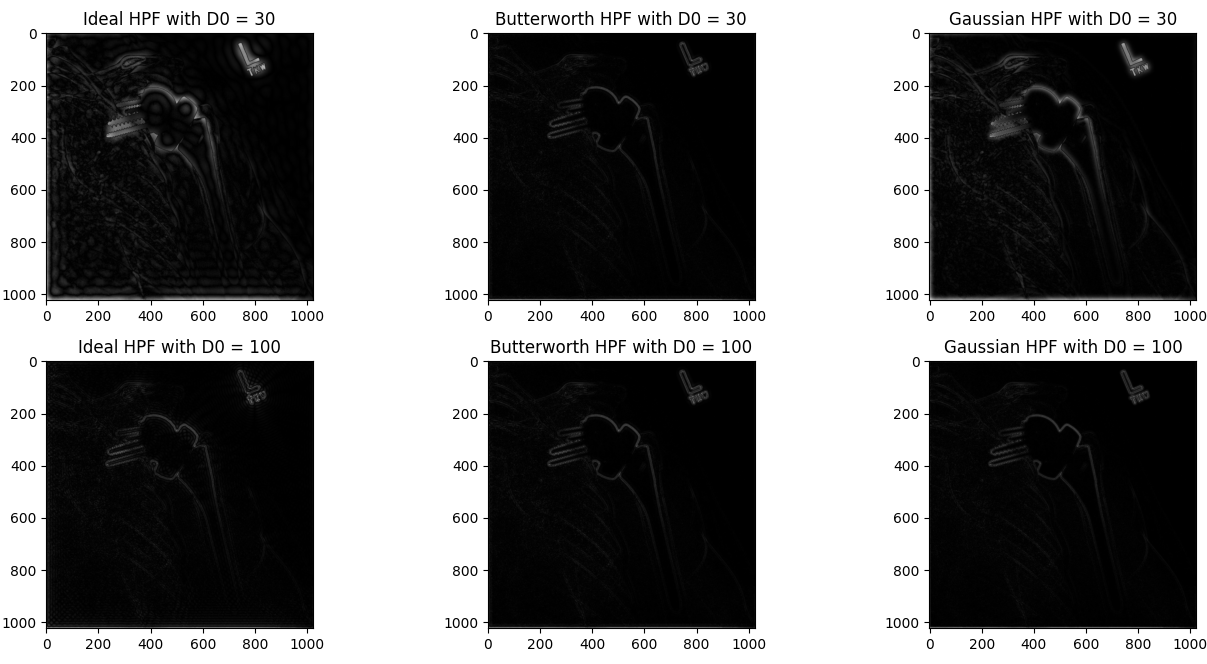
پاسخ فیلتر ها:



حاصل ضرب فیلتر‌ها در تبدیل فوریه‌ی تصویر پد شده:



تصاویر خروجی (از حالت پد در آوردمشون):



فیلتر گاوسین با شعاع ۱۰۰ و باترورث با شعاع ۳۰ و همچنین ۱۰۰ به خوبی توانستند آیتم پیچ و حرف L را در تصویر نمایش دهند.