به نام خدا

تمرین سری پنجم پردازش تصویر

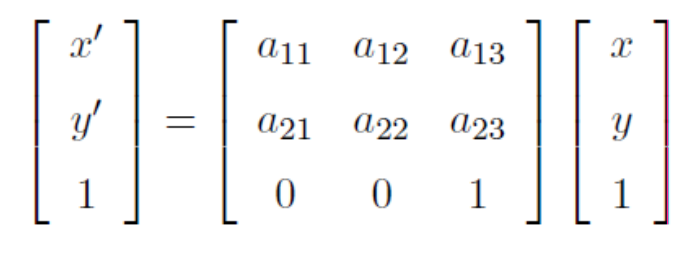
نام استاد : دکتر آذرنوش

امیرحسین شریفی صدر 9733044

سوال 1 : در این سوال قصد داریم که تصویری را روی تصویری دیگر به وسیلهfeature base registration رجسیتر کنیم .

بدین صورت که میدانیم 2 تصویر ما توسط تابع تبدیلی به هم مرتبط هستند . که این تابع تبدیلAffine است.

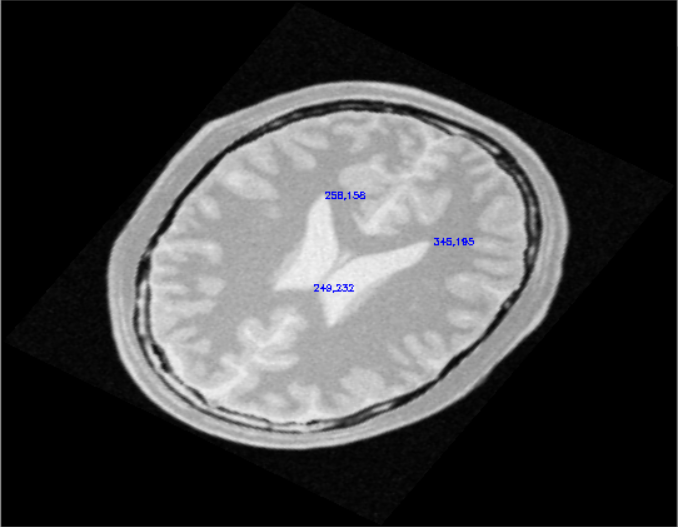
که این تابع به شکل زیر است :

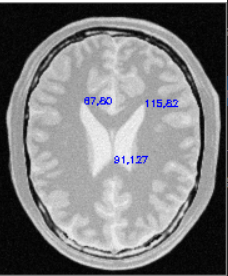


پس ما باید درایه های این ماتریس تبدیل را به دست آوریم .

6 مجهول داریم . برای رسیدن به این 6 مجهول ما احتیاج داریم که به صورت اتومایتک یا دستی 3 نقطه از تصویر را انتخاب کرده و نگاشت تقریبی آن ها در تصویر دوم را نیز مشخص کنیم (روش point to point) و سپس مختصات این نقاط را در تابع cv.getAffineTransform در پایتون گذاشته تا ما را به 6 درایه ماتریس تبدیلمان برساند .

این 3 نقطه را در تصویر اول به با کلیک انتخاب میکنیم و سپس space را میزنیم تا تصویر دوم نمایش داده شود، سپس 3 نقطه متناظر را در تصویر دوم مشخص کرده و به تابع میدهیم .





ما در این مسئله تابع تبدیل تصویردوم به تصویر اول را به دست آوردیم ینی تصویری که کج شده بود را به تصویر صاف .

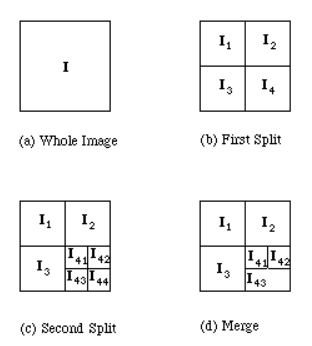


حال میخواهیم برای اطمینان از روش خود تابع تبدیل به دست آمده را روی تصویر دوم تاثیر بدهیم و ببینیم که آیا به تصویر اول میرسیم یا نه . این کار را در پایتون به با دستور cv.warpAffine انجام میدهیم.

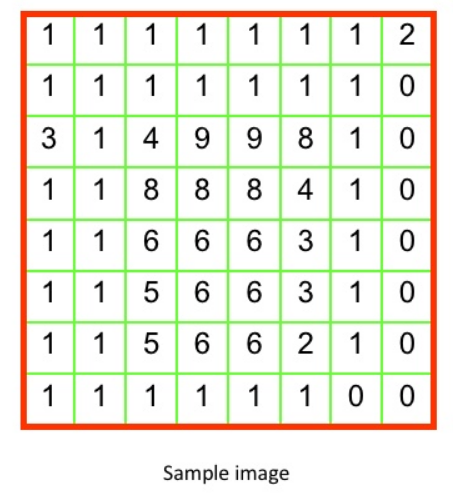


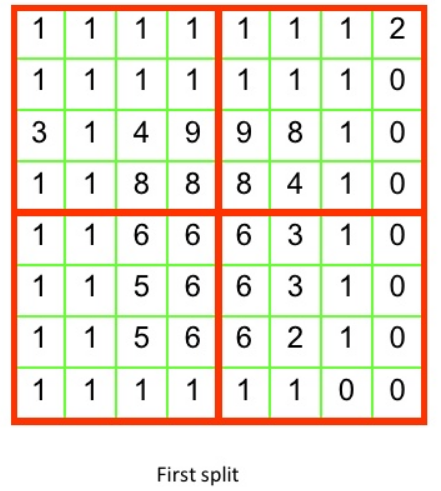
نتیجه را مشهاده میکنید :

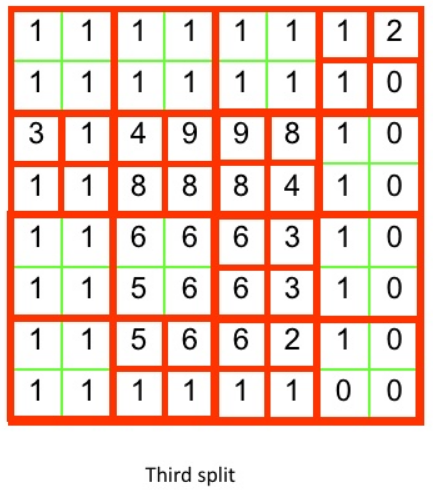
سوال 2 : به صورت کلی الگوریتم Split and Merge بدین صورت است که ابتدا تصویر را به 4 بخش مساوی تقسیم میکنیم حال در این بخش ها چک میکنیم که ایا تفاوتی بی ناجزای آن بخش وجود دارد یا نه . اگر تفاوتی وجود نداشت به همان شکل باقی میماند . اگر تفاوت وجود داشت دوباره آن بخش را به 4 قسمت تقسیم میکنیم و این روند را ادامه مدهیم . حال نوبت merge کردن اجزا هست . بدین صورت که پس از تقسیم بندی حال بخش هایی که کنار هم هستند و شباهت زیادی به دارند رو ترکیب کرده و یک بخش در نظر میگیریم .

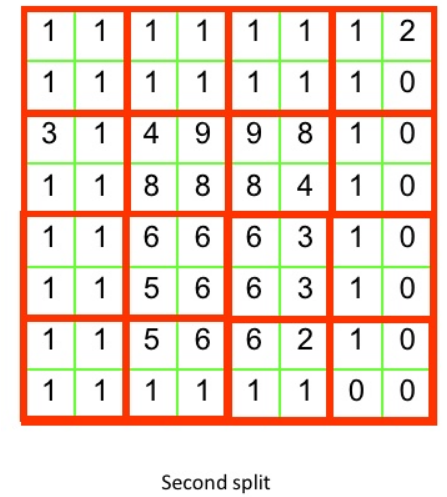


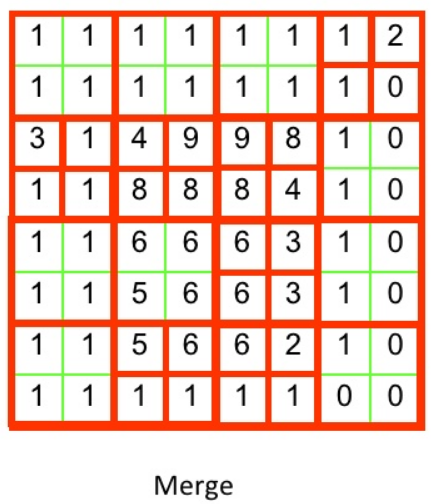
حال میخواهیم این الگوریتم را به صورت واضحتر با مثالی ببینیم :

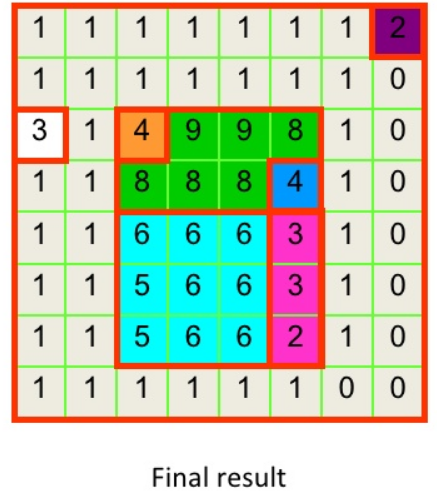












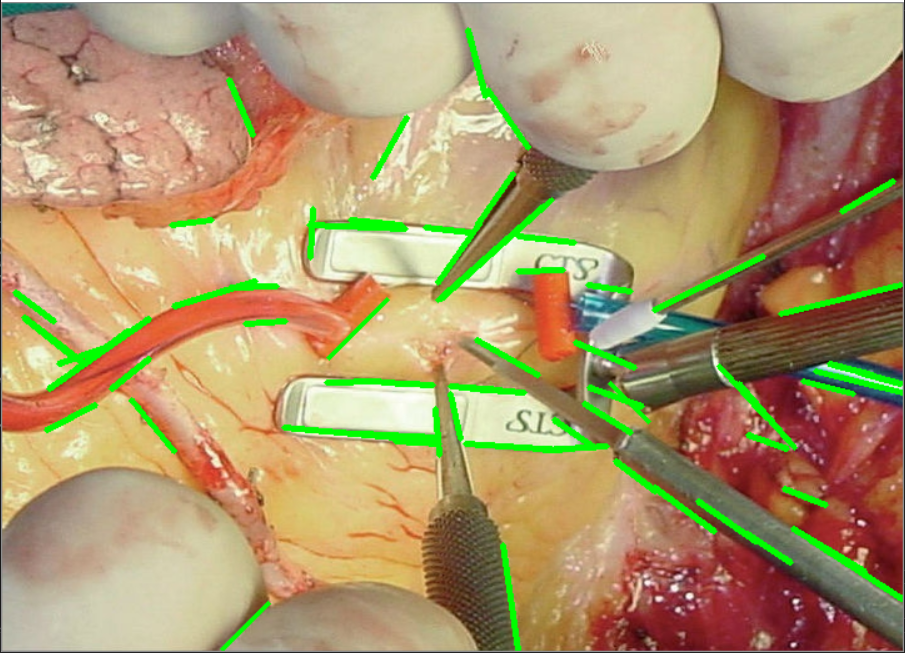
سوال 3 : در این سوال از ما خواسته شده تا با روش Hough transformation اجزای خواسته شده در تصاویر را جداسازی کنیم.

این تبدیل ابتدا لبه ها را تشخیص میدهد و سپس آن هارا انتخاب میکند.

در بخش الف باید به شیوه جداسازی خطوط توسط تبدیل هاف اشیای جراحی موجود در تصویر را مشخص کنیم .

ابتدا لبه های موجود در تصویر را توسط فیلتر هایی مانند بلور و canny استخراج و ترشولد میکنیم سپس با استفاده از تابع آماده cv.HoughLinesP و امتحان مقادیر مختلف برای پارامترهای آن سعی میکینم که به بهترین نتیجه در استخراج لوازم جراحی برسیم .

تبدیل HoughLineP یک نوع تبدیل احتمالاتی است .



همان طور که مشاهده میشود با

تقریب خوبی لوازم جراحی مشخص

شده اند و لی کمی خطا نیز وجود دارد

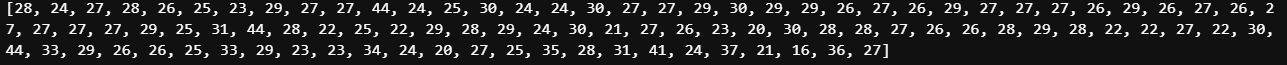
که میتوانست کمتر نیز شود ولی این

تصویر نتیجه تلاش بنده است .

در بخش ب سوال از ما خواسته شده تا گلبول های سفید و قرمز درون تصویر را جدا سازی کنیم .

توسط تابع آماده cv.HoughCircles و پارامتر هایی که دارد مقادیری را به آن میدهیم تا بتواند دایره های موجود در تصویر که همان گلبول ها هستند را با تقریب خوبی تشخیص دهد . و البته قبلش سعی میکنیم که با blur کردن تصویر لبه ها را صاف تر کنیم .

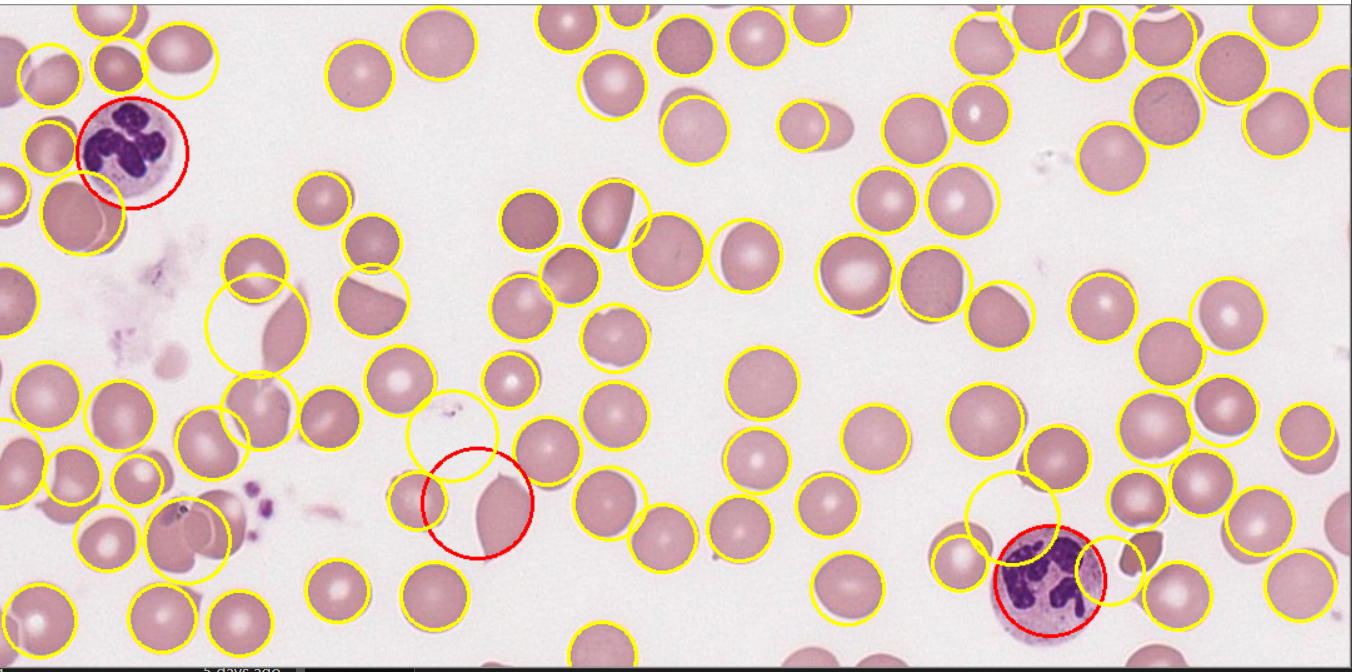
حالا پس از اعمال تابع هاف شعاع دایره هایی که تشخیص داده است را چاپ میکنیم :



در تابع cv.HoughCircles حداقل شعاعی که مشخص کند را برابر با 15 و حداکثر شعاعی که تشخیص دهد را برابر با 45 قرار داده بودیم و در خروجی میبنیم که 4 خروجی با شعاع بزرگتر از 40 قرار دارد که حدس میزنیم دو تا از آن ها گلبول های سفید درون تصویر باشد .

پس مقداری مثلا برابر با شعاع 42 انتخاب میکنیم که اگر چیزی بزگتر از آن بود را گلبول سفید تشخیص دهد و بقیه شعاع ها را گلبول قرمز .

که البته میدانیم کاملا دقیق نیست و خطا نیز دارد ولی به پاسخ دلخواه ما بسیار نزدیک است .



میبینیم که 1 شکل را

به اشتباه گلبول سفید

تشخیص داده و 2

گلبول قرمز را نیز

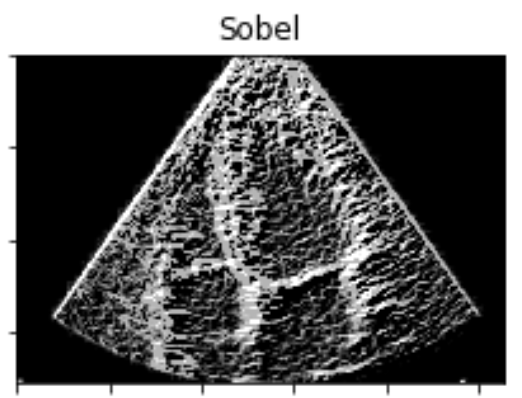
اضافی تشخیص داده .

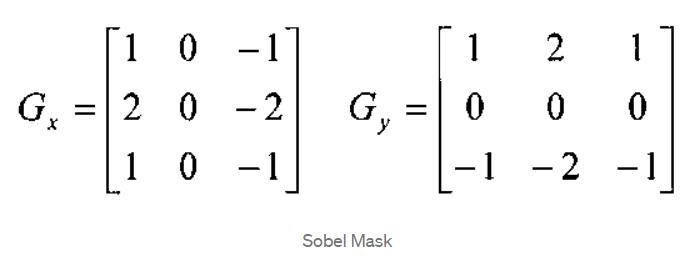
سوال 4 : در این سوال ما فیلترهای مختلف برای استخراج لبه ها را بر روی تصویری اعمال میکنیم و نتایج حاصل را بررسی میکنیم .

تصویر اصلی

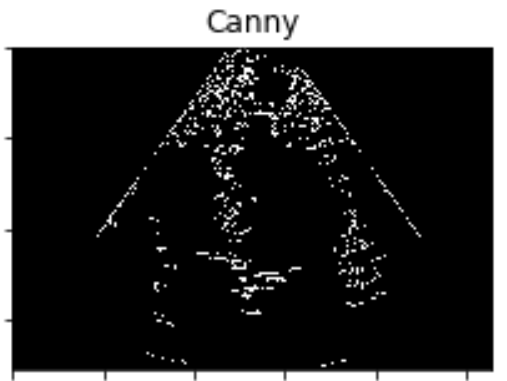


ابتدا فیلتر sobel را اثر میدهیم . باید دقت کنیم که فیلتر sobel را روی Gaussian اعمال میکنیم و باید در جهت x و y sobel را محسابه کرده و سپس برای مشخص شدن لبه های عمودی و افقی  
 را محسابه میکنیم .

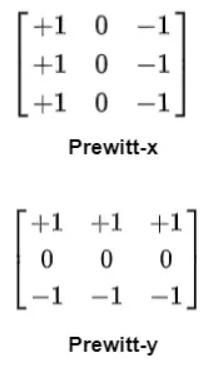




فیلتر بعدی Canny است که از بهترین الگوریتم های لبه یابی است که با استفاده از کم کردن نویز و ترشولد کردن با استفاده از دو آستانه بالا و پایین این کار را انجام میدهد .

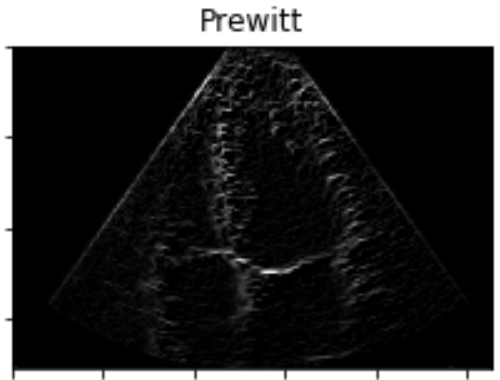


فیلتر بعدی Prewitt است . این ماتریس را روی تصویر اعمال میکند :



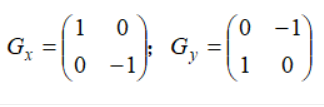
برای مشخص شدن لبه های عمودی و افقی

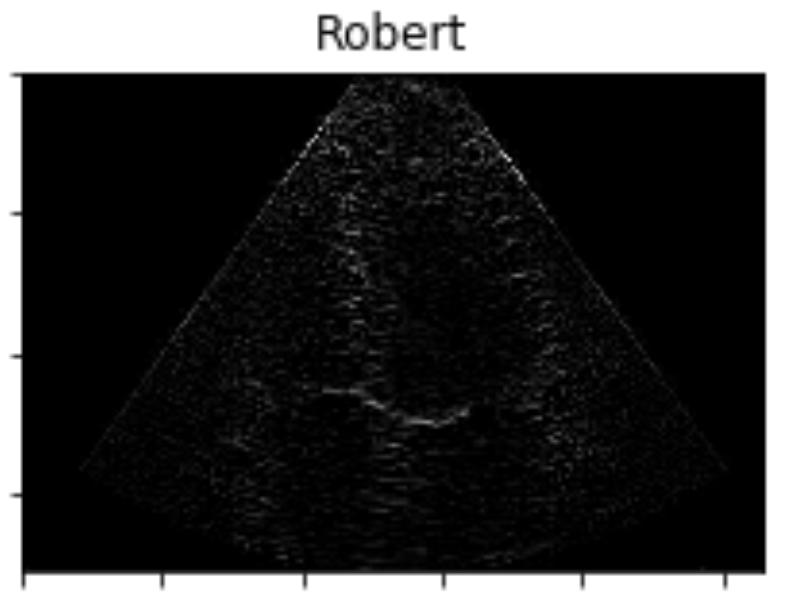
هر دوباهم نیز مانند سوبل



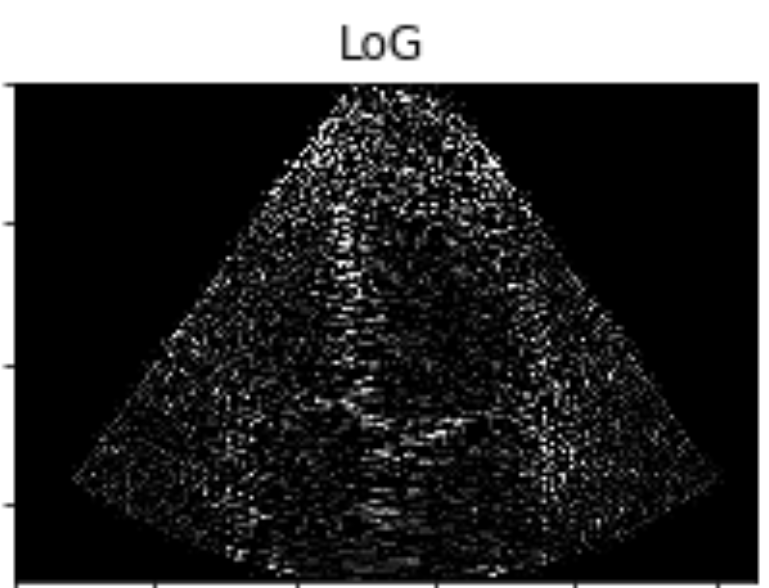
فیلتر بعدی، فیلتر Roberts است .

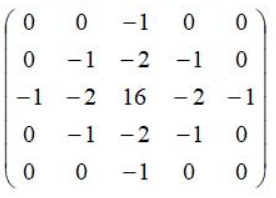
این الگوریتم به نویز حساسیت زیادی دارد وپیکسل های کمتری را برای تقریب گرادیان بکار می برد،درضمن نسبت به الگوریتم canny هم قدرت کمتری دارد.





و در آخر فیلترLaplacian of gaussian) LoG( را بررسی خواهیم کرد . لاپلاس یک تصویر، مناطق تغییرات شدت سریع را نشان می دهد و بنابراین اغلب برای تشخیص لبه استفاده می شود.ماتریس پایین را روی تصویر اعمال میکنیم .





از cv.filter2D برای اعمال این ماتریس ها و فیلترها استفاده میکنیم.