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.

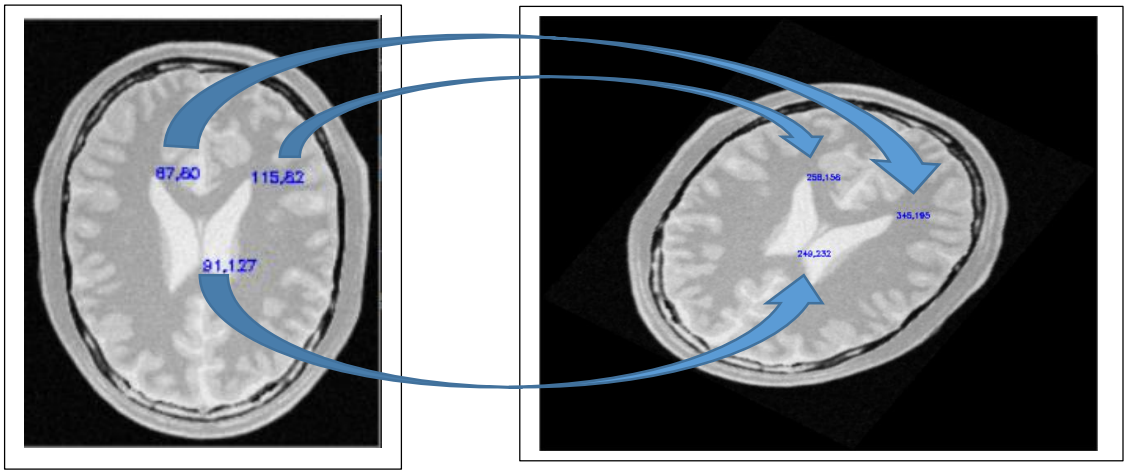
که این تابع به شکل زیر است :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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 .



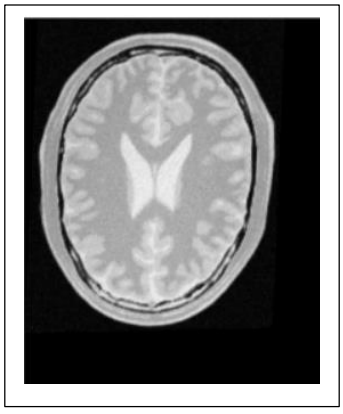
ما در این مسئله تابع تبدیل تصویردوم به تصویر اول را به دست آوردیم

```
[[-3.93442623e-01 -3.72175454e-01  2.75311918e+02]
 [-2.67759563e-01  5.75542756e-01  6.01462118e+01]]
```

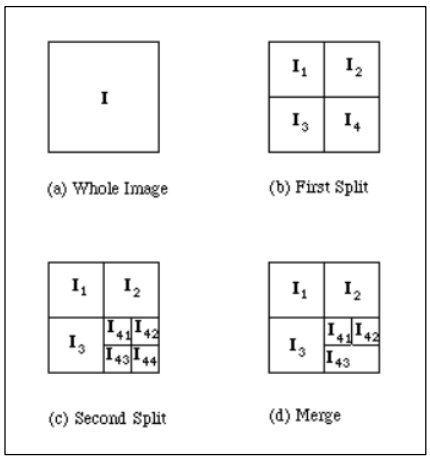
ینی تصویری که کج شده بود را به تصویر صاف .

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 .



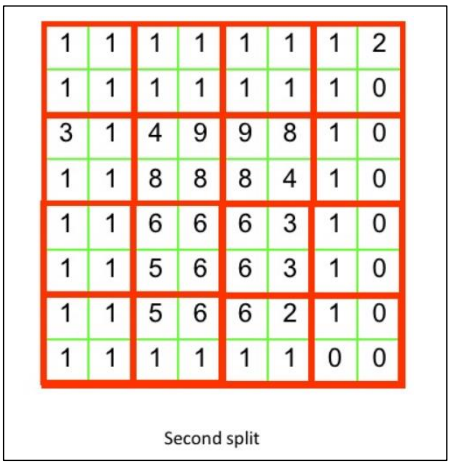
حال میخواهیم این الگوریتم را به صورت واضحتر با مثالی ببینیم :

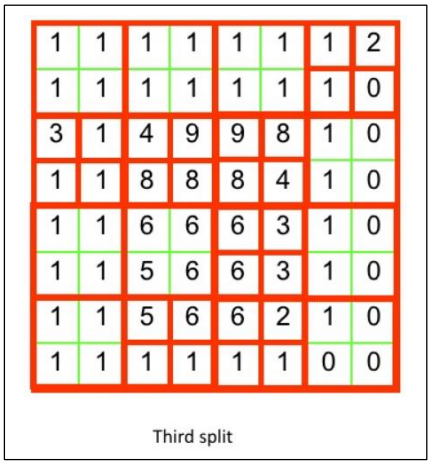
1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

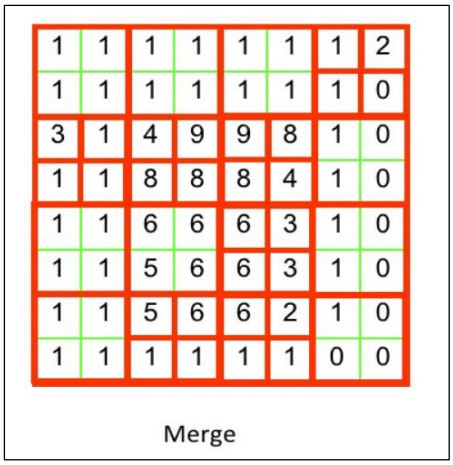
Sample image

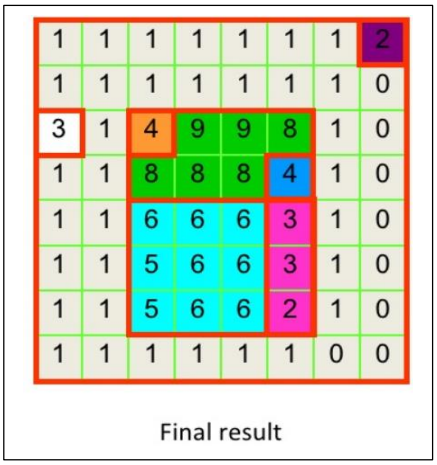
1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

First split


1 1 1 1 1 1 1 2
1 1 1 1 1 1 1 0
3 1 4 9 9 8 1 0
1 1 8 8 8 4 1 0
1 1 6 6 6 3 1 0
1 1 5 6 6 3 1 0
1 1 5 6 6 2 1 0
1 1 1 1 1 1 0 0
Second split


1 1 1 1 1 1 1 2
1 1 1 1 1 1 1 0
3 1 4 9 9 8 1 0
1 1 8 8 8 4 1 0
1 1 6 6 6 3 1 0
1 1 5 6 6 3 1 0
1 1 5 6 6 2 1 0
1 1 1 1 1 1 0 0
Third split


1 1 1 1 1 1 1 2
1 1 1 1 1 1 1 0
3 1 4 9 9 8 1 0
1 1 8 8 8 4 1 0
1 1 6 6 6 3 1 0
1 1 5 6 6 3 1 0
1 1 5 6 6 2 1 0
1 1 1 1 1 1 0 0
Merge


1 1 1 1 1 1 1 2
1 1 1 1 1 1 1 0
3 1 4 9 9 8 1 0
1 1 8 8 8 4 1 0
1 1 6 6 6 3 1 0
1 1 5 6 6 3 1 0
1 1 5 6 6 2 1 0
1 1 1 1 1 1 0 0
Final result

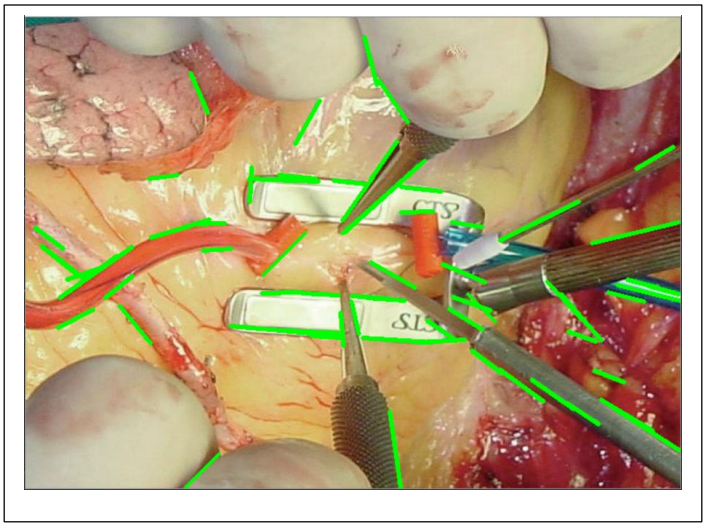
سوال 3 : در این سوال از ما خواسته شده تا با روش Hough transformation اجزای خواسته شده در تصاویر را جداسازی کنیم.

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 .

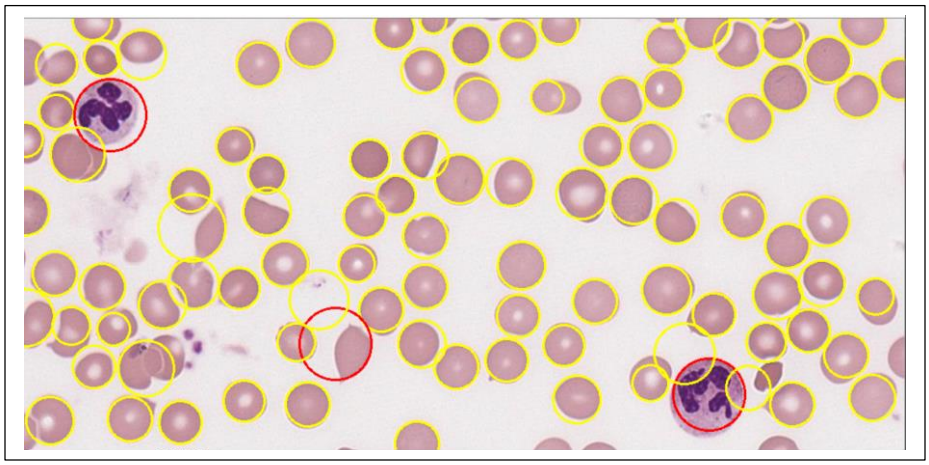
حالا پس از اعمال تابع هاف شعاع دایره هایی که تشخیص داده است را چاپ میکنیم :

```
[28, 24, 27, 28, 26, 25, 23, 29, 27, 27, 44, 24, 25, 30, 24, 24, 30, 27, 27, 29, 30, 29, 29, 26, 27, 26, 29, 27, 27, 27, 26, 29, 26, 27, 26, 2
7, 27, 27, 27, 29, 25, 31, 44, 28, 22, 25, 22, 29, 28, 29, 24, 30, 21, 27, 26, 23, 20, 30, 28, 28, 27, 26, 26, 28, 29, 28, 22, 22, 27, 22, 30,
44, 33, 29, 26, 26, 25, 33, 29, 23, 23, 34, 24, 20, 27, 25, 35, 28, 31, 41, 24, 37, 21, 16, 36, 27]
```

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 .

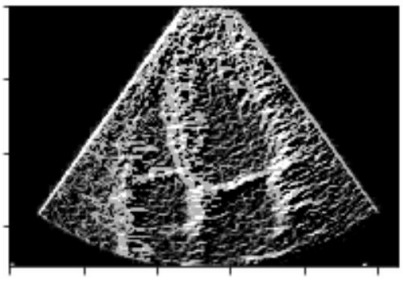
تصویر اصلی

ابتدا فیلتر sobel را اثر میدهیم . باید دقت کنیم که فیلتر sobel را رویGaussian اعمال میکنیم و باید در جهت x و sobel y را محسابه کرده و سپس برای مشخص شدن لبه های عمودی و افقی $\sqrt{(G_x^2 + G_y^2)}$ را محسابه میکنیم .

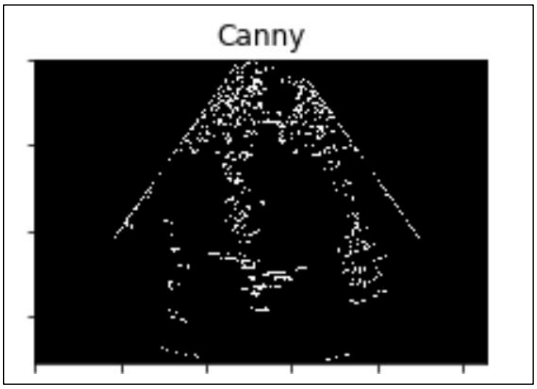
$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sobel Mask

Sobel



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 :

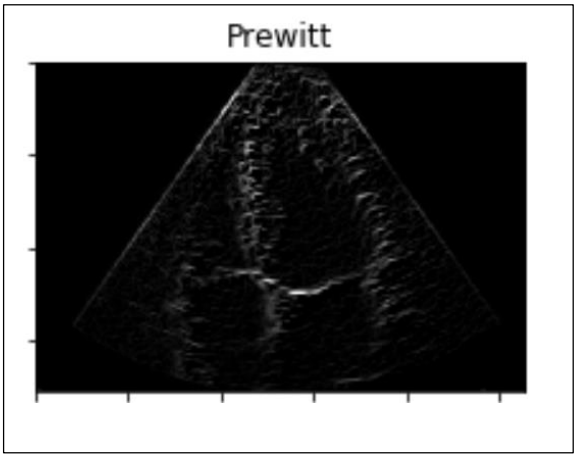
$$\begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix}$$

Prewitt-x

$$\begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Prewitt-y

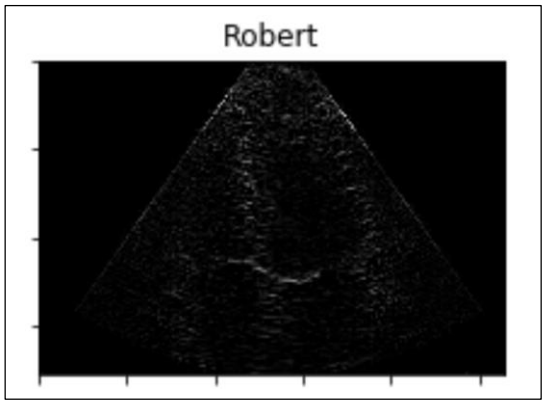
برای مشخص شدن لبه های عمودی و افقی هر دوباهم نیز مانند سوبل $\sqrt{(P_x^2 + P_y^2)}$



فیلتر بعدی، فیلتر Roberts است .

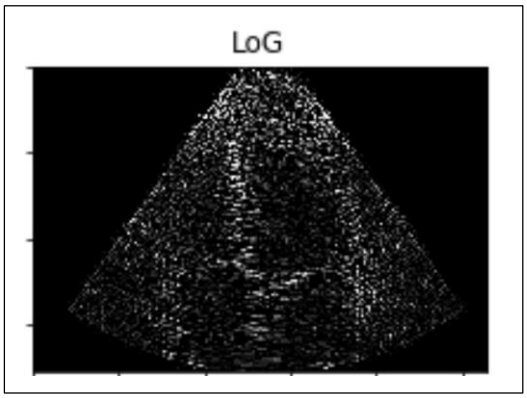
این الگوریتم به نویز حساسیت زیادی دارد وپیکسل های کمتری را برای تقریب گرادیان بکار می برد،درضمن نسبت به الگوریتم canny هم قدرت کمتری دارد.

$$G_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}; \quad G_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$



و در آخر فیلتر(LoG(Laplacian of gaussian 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 .

$$\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$



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