به نام خدا

امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

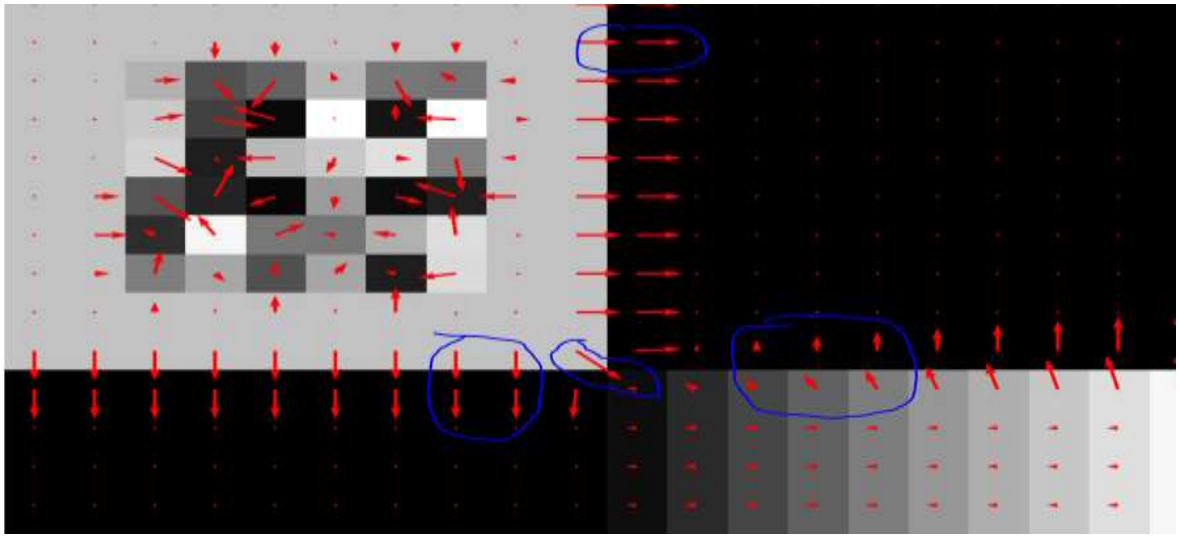
توضیحات تمرین ۳ – بینایی کامپیوتر

۱)

a) بردار گرادیان برای تصویر که به صورت دو بعدی میباشد طبق جزوه به صورت زیر است:

$$\nabla f(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

b) بردار گرادیان که به وسیله مشتق محاسبه میشود نمایانگر جهت تغییرات شدت روشنایی در تصویر میباشد.(جهت تغییرات را هم به صورت افقی هم به صورت عمودی و به صورت برآیند این دو نیز میتوانیم مشاهده کنیم). تصویر زیر میتواند نمایانگر این موضوع میباشد:



از فواید این بردار در پیدا کردن لبه ها و خط ها میتوان اشاره کرد.

c) اندازه گرادیان طبق رابطه زیر محاسبه میشود:

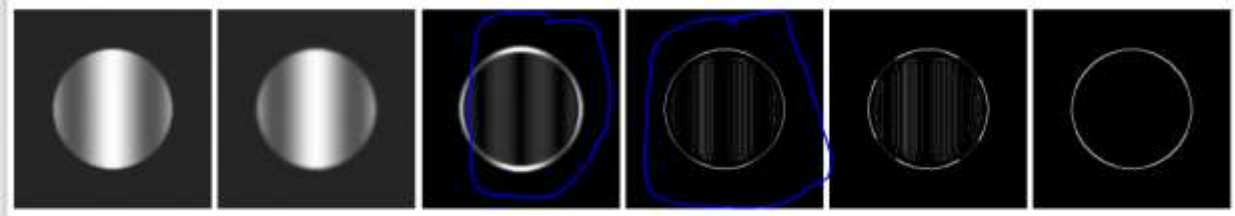
• اندازه گرادیان $$M(x,y) = \|\nabla f\| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

d) جهت گرادیان طبق رابطه زیر محاسبه میشود:

• جهت گرادیان $$\alpha(x,y) = \text{dir}(\nabla f) = \text{atan2}(g_y, g_x)$$

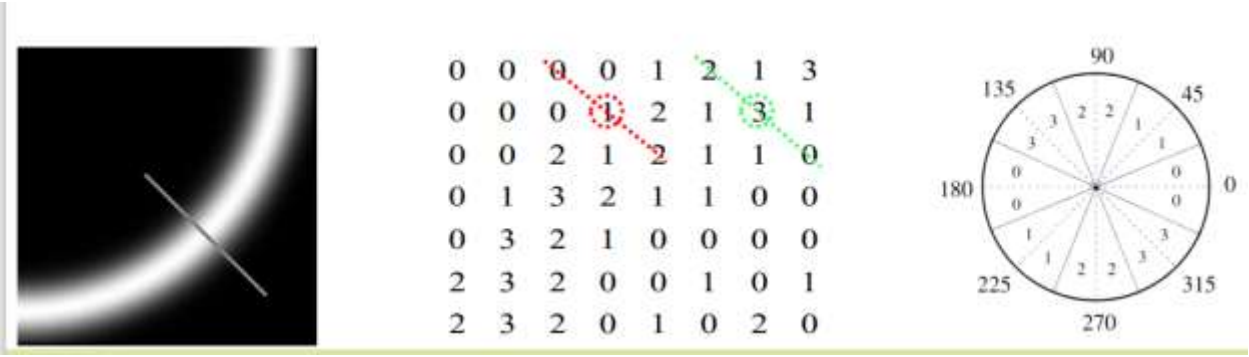
e) یکی از مراحل اصلی برای اجرای لبه یاب canny محاسبه بردار گرادیان است که نقش آن یافتن جهت تغییرات است که در پیدا کردن لبه برای ما موثر است.البته مشکلی نیز دارد که به برخی لبه ها بیشتر از یک پیکسل اختصاص داده میشود. که در مراحل بعد الگوریتم آنرا اصلاح میکنیم.

در مرحله بعد که حذف مقادیر غیر بیشینه است یک پنجره در نظر میگیریم و بر اساس جهت و زاویه گرادیان یک پیکسل را در نظر میگیریم که اگر بر اساس جهت گرادیان بیشینه است لبه باشد اگر نه به نوعی از لبه بودن حذفش میکنیم. در نتیجه بردار گرادیان در این مرحله نیز مفید بوده است.



به طور کلی در این دو مرحله مفید بوده است.

طرز کار آن در حذف مقادیر غیر بیشینه به صورت زیر است:



f) از ضعف های عملگر لاپلاسین در مواجه با نویز میتوان اشاره کرد که اثرات نویز را بیشتر میکند و در تشخیص لبه ناتوان میشود. در دو عملگر ذکر شده دیگر ابتدا تصویر را هموار میکنیم تا اثرات نویز کمتر شود.

عملگر لاپلاسین در رابطه با تغییرات شدت روشنایی اطلاعی به ما نمیدهد که در بحث تشخیص لبه مسئله مهمی میباشد. زیرا از یک شکل به شکل دیگری میرویم و نقطه ای که تغییرات رخ میدهد لبه است که عملگر لاپلاسین قابلیت تشخیص آنرا ندارد و نمیتواند در یافتن لبه عملکرد مناسبی مثل سایر عملگر های معرفی شده داشته باشد.

دقت این عملگر در تشخیص لبه بالا نیست یعنی میتواند مثلا خط هایی را لبه تشخیص دهد که نیستند.

۲) در رابطه با کد این دو بخش بر اساس فرمول ها پیش رفتیم و در تابع اول فاز و دامنه را محاسبه کردیم. در تابع دوم فاز و دامنه هر دو تابع را به دست آوردیم و تصاویر جدید را با استفاده از مقدار magnitude و فاز ساختیم و فوریه معکوس اعمال کردیم تا تصویر با دامنه و فاز تصویر دیگر ساخته شود.
همانطور که در خروجی مشخص است تصویر دچار نویز شده و قسمت های از بدن یوزپلنگ و گورخر در تصاویر جدید به خوبی مشخص نیست.
در اینجا نتیجه میگیریم با عوض کردن فاز دو تصویر مقدار زیادی از مشخصه های آن تصاویر نیز جا به جا میشوند و هنگامی که بحث امنه نیز پیش می آید با تغییر فاز منطقا به تصویر جدیدی میرسیم.دامنه بیشتر نویز تصویر را زیاد کرده است.

۳)

a) علت اینکه از تبدیل فوریه استفاده میکنیم این است که در این تصویر نویز های متناوب داریم و از خصوصیات این تبدیل است که این نوع نویز ها را به خوبی تشخیص دهد.

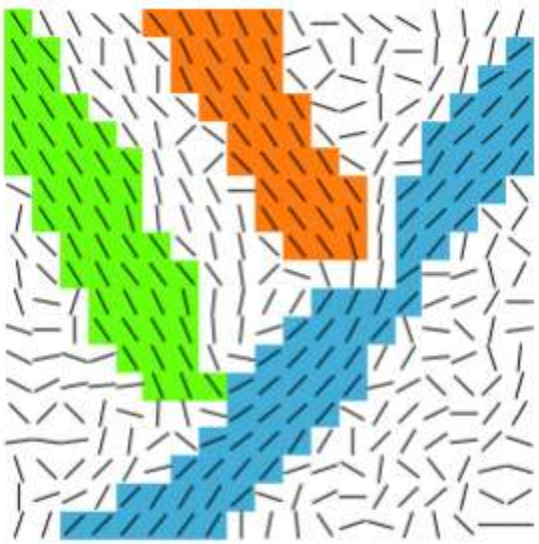
ابتدا تبدیل فوریه میگیریم و در مرحله بعد آنرا به مرکز شیفت میدهیم. در مرحله بعد از فیلتر گوسی استفاده میکنیم(در کد مشخص است که ابتدا کرنل خود را مشخص کردیم و سپس فرمول گاوسی را نیز اعمال کردیم و برای سیگما نیز مقدار نسبتا بزرگی را انتخاب کردیم تا اثرات نویز را تا حد امکان کم کند) و سپس از تبدیل فوریه و عکس کردن آن تصویر جدید را که بعد از اعمال فیلتر گاوسی است بسازیم و نمایش دهیم.

b) در بخش بعدی با استفاده از تابع آماده canny به اعمال این لبه یاب پرداختیم.نکاتی که در رابطه با کد این مورد وجود دارد ابتدا cast کردن تصویر

denoised (یا نویز زدایی شده) به unit8 میباشد(در کد قسمت قبل از normalization استفاده کردیم) که مبنای کار کردن این تابع میباشد. مقدار ۱۰۰ و ۳۵۰ با استفاده از آزمون و خطا به دست آمدند.اصول کار کردن با این threshold ها به این صورت است که مقادیر بالاتر از ۳۵۰ لبه هستند و برای مقادیر بین این دو در صورت همسایگی با یک لبه جز لبه حساب میکنیم و مقادیر کمتر از ۱۰۰ را نیز حذف میکنیم. پارامتر آخر برای هموارسازی به کار میرود.(این مورد نیز با آزمون و خطا به دست آمد بدون این پارامتر تصویر خروجی یک تصویر کاملا سیاه میباشد)

c) در این مرحله گرادیان در جهات x,y را با استفاده از عملگر sobel محاسبه کردیم سپس مقدار arctan2 را محاسبه میکنیم و پلات آنرا رسم میکنیم.

d) جهت نقاطی که ساقه زعفرون را تشکیل میدهند در یک راستا میباشند و برای تشخیص این مورد اگر از LSD استفاده کنیم میتوانیم به نتیجه خوبی برسیم.گل زعفرون به دلیل شکل منحنی طوری که دارد هم میتواند جهت گرادیان را در آن پیدا کنیم و میتوانیم دو نوع شکل را طبقه بندی کنیم و در بین آنها ساقه و گل زعفرون را از هم تفتیش کنیم.



این شکل مثلا میتواند مناسب باشد.ساقه های زعفرون هر یک در جهات مختلف هستند و به صورت خطی شکل نیز میباشند(بر عکس گل زعفرون که حالت انحنا دارد). و همچنین بین صفحه بک گراند و ساقه زعفران نیز چون تغییر شدت روشنایی داریم در نتیجه ساقه را میتوانیم به عنوان لبه نیز در نظر بگیریم در نتیجه این الگوریتم میتواند عملکرد نسبی مناسبی داشته باشد.

راه دیگری نیز وجود دارد که کد آنرا میتوانید در آخرین سِل مشاهده کنید.

۴)

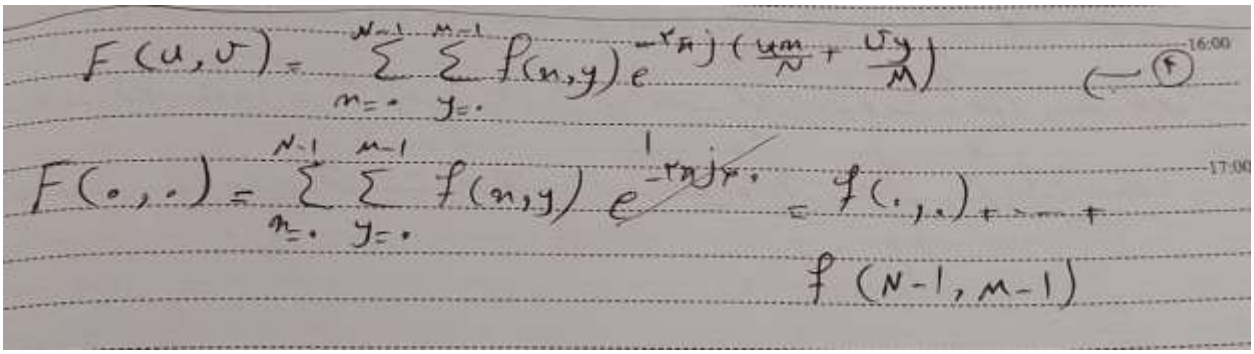
الف) کاربرد اول تبدیل فوریه در تصاویری است که نویز آنها متناوب است و میتواند این نوع نویز ها را به خوبی شناسایی کند. علت آن این است که تبدیل فوریه عموما با فرکانس و به تبع آن با دور تناوب درگیر است در نتیجه نویز های متناوب که با دور تناوب مشخصی تکرار میشوند به راحتی شناسایی و در مرحله بعد حذف میشوند.

کاربرد دوم این روش یافتن خطوط مهم و لبه ها و گوشه های تصویر و ... میباشد. چون این روش در حوزه فرکانس کار میکند و این نقاط دارای فرکانس های بالاتری میباشند در نتیجه میتواند این نقاط را کشف کند.

در بسیاری از برنامه‌های پردازش تصویری، مانند تصویربرداری ماهواره‌ای، زمین‌شناسی، یا تصویربرداری فضایی، تحلیل فوریه جهت تشخیص و تحلیل الگوهای مختلف اهمیت دارد. برای مثال، در تصویربرداری ماهواره‌ای، تحلیل فوریه می‌تواند به تشخیص الگوهای مختلف از منابع طبیعی یا مصنوعی بر روی زمین کمک کند. این الگوها ممکن است

اطلاعاتی در مورد جنس خاک، پوشش گیاهی، یا ساختار جغرافیایی محل مورد نظر ارائه دهند که با تحلیل فوریه به صورت موثر تشخیص داده می‌شوند.

ب)



۵) این سوال چون مشخصا خواسته نشده بود که سوال و روش خود را توضیح دهیم به صورت خلاصه در داک توضیح دادم و به صورت کامل تر با کامنت در کد موجود است.

در قسمت اول باید تابع convolve را به دست آوریم که به صورت زیر محاسبه کردیم:

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots + w(0,0)f(x,y) + \ldots + w(1,1)f(x+1,y+1)$$

$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

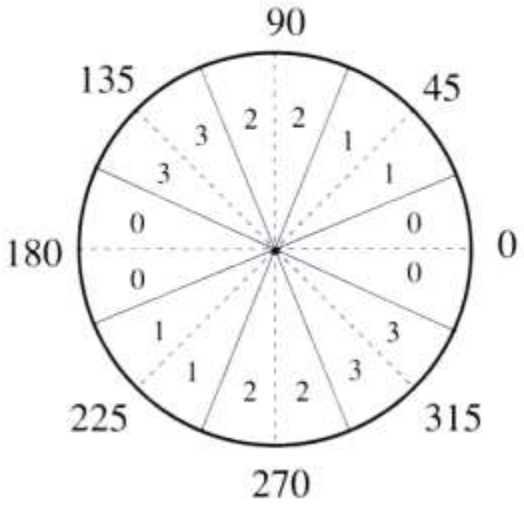
در قسمت باید کرنل گاوسی را در نظر بگیریم که ابتدا پنجره ای به اندازه مدنظر باز مدنظر قرار میدهیم و سپس فرمول زیر که از جزوه است را برای آن میگذاریم:

$$G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}} = Ke^{-\frac{r^2}{2\sigma^2}}$$

در قسمت بعدی که باید مشتق را برحسب x,y به صورت جداگانه حساب کنیم کرنل هایی را مدنظر قرار میدهیم و مقادیر آنها را بین 0.5- تا 0.5 قرار میدهیم و سپس کانولوشن را انجام میدهیم.

در قسمت بعدی که محاسبه خود تابع گرادیان بود از مشتق افقی و عمودی استفاده کردیم و اندازه و جهت گرادیان را به دست آوردیم

در قسمت بعدی که حذف مقادیر غیر بیشینه بود، زوایا را مانند اسلاید در نظر گرفتیم:



و همسایه ها را برای در نظر گرفتن بیشترین مقدار بر این اساس در نظر گرفتیم.

در قسمت بعدی آستانه گذاری دو سطحی را پیاده سازی کردیم.
روش کار آن به این صورت است که اگر از بیشینه threshold بیشتر باشد لبه میگیریم اگر بین کمینه و بیشینه بود و در همسایگی لبه بود آنرا لبه بگیرد و در غیر

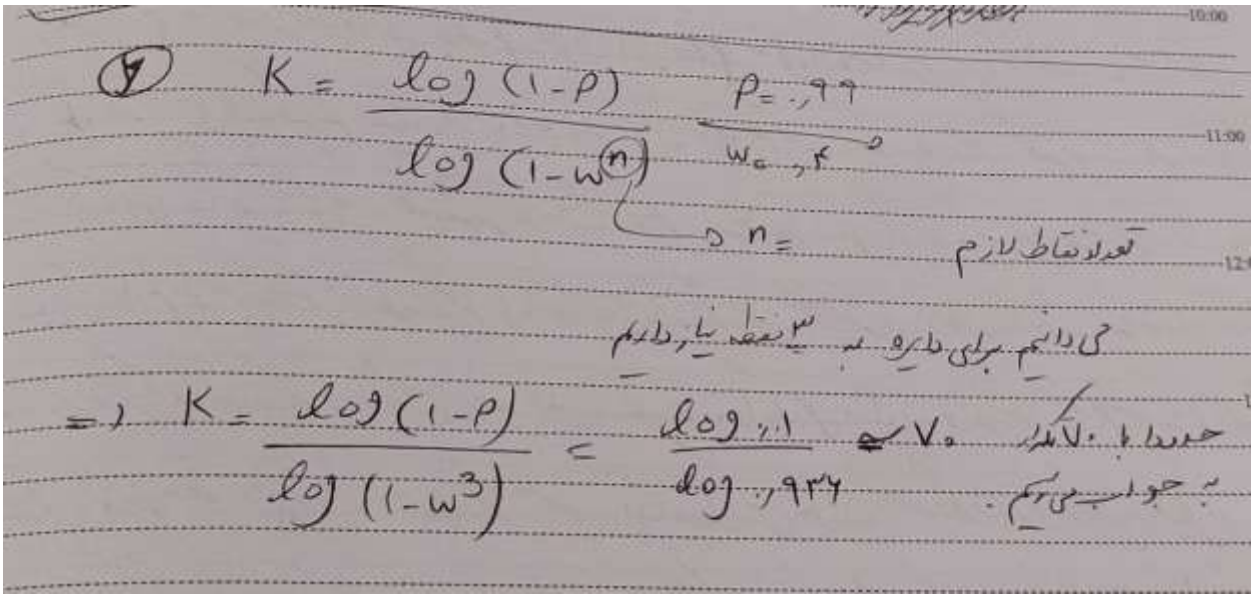
اینطورت خیر.

در قسمت بعدی پیاده سازی همسایه ها را انجام دادیم.

در قسمت بعدی با استفاده از الگوریتم bfs نقاطی که به نقاط لبه متصل هستند و بین دو مقدار threshold هستند را لبه میگیریم و به سراغ همسایه های لبه های جدید میرویم و به همین صورت ادامه میابد.

در مرحله آخر مراحل پیاده سازی لبه یاب canny را با توجه به پیاده سازی هایی که در مرحله قبل داشتیم انجام میدهیم.

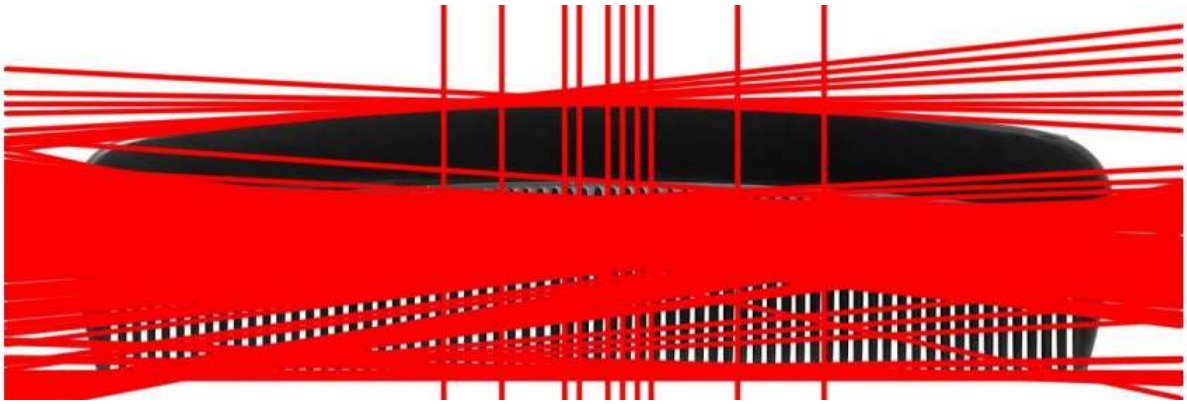
۶)



۷)

الف) در LSD ما ۴ پارامتر داریم که شامل x,y نقاط شروع و پایان میشود ولی در Hough ۲ پارامتر داریم.

استفاده از جهت گرادیان کمی متفاوت است. در LSD به این صورت است که از این جهت برای پیدا کردن پیکسل های هم راستا استفاده میشود و با پیدا کردن این پیکسل ها این خط را رسم میکند. در Hough برای کنترل جهت تغییرات است که به طور مثال خطوطی که مدنظر ما به عنوان لبه نیستند کنترل شوند.



همچنین LSD برای یافتن نقاط ابتدا و انتها است ولی Hough به این صورت نیست از این جهت از نظر کاربرد نیز با هم متفاوتند.

در رابطه با دقت آنها نیز باید گفت LSD دقت بالایی دارد و اگر تصویر نویزی باشد یا کیفیت در بعضی نقاط خوب نباشد کماکان میتواند تشخیص دهد اما در رابطه با hough به اینگونه نیست؛ مانند مثال زیر:

که مشخص است در hough با کیفیت و دقت LSD مشخص نیست.

ب)

ابتدا تصویر را به فضای grayscale بردیم سپس از تابع hough circle استفاده کردیم. در رابطه با hough گفتیم که بر اساس رای گیری مشخص میشود که یک نقطه لبه هست یا خیر و برای کنترل کردن این موضوع از param1,2 در تابع استفاده کردیم که بیشینه وکمینه رای مورد نیاز برای در نظر گرفتن یک نقطه به

عنوان لبه است. در قسمت radius ها هم شعاع مدنظر برای دایره های تصویر را در نظر گرفتیم که در تصویر دو دایره بیشتر نداشتیم و با آزمون و خطا دایره بزرگتر دارای شعاع کمتر از ۴۰ میباشد در نتیجه مقدار max radius را برابر ۴۰ گذاشتیم
سپس بر اساس شعاع و مرکز این دایره ها را شناسایی کردیم. برای اینکه این دایره ها را دیگر در تصویر نبینیم رنگ این دایره ها را به رنگ بکگراند یعنی سیاه تغییر دادیم.

پ) تقریبا تمام مراحل قبل را تکرار میکنیم با این تفاوت که از توابع ذکر شده در سوال و برای کشف کردن خطوط لبه استفاده کرده ایم که در خروجی این سِل مشخص هستند.

امتیازی : از تابع دیگری استفاده کردیم که این تابع خطوط لبه را به طور کامل و دقیق تشخیص نمیدهد.تفاوت آن با تابع سِل قبل مانند این اسلاید ها میباشد:

$$\rho = x\cos\theta + y\sin\theta$$

$$H(\rho,\theta) = H(\rho,\theta) + 1$$

در اینجا زاویه را که در الگوریتم هاف دخیل کردیم بهتر عمل کرد مانند کاری که در houghlinesp انجام میشود ولی اگر انجام دهیم الگوریتم کلاسیک و بهبود نیافته هاف است که خطوط نامربوط را نیز مشخص میکند.

در رابطه با خود تابع houghline مقدار ۵۰ که طبق پارامتر های این تابع threshold میباشد ؛ نقش تعیین کننده ای در خطوط دارد اگر مقدار آنرا پایین بیاوریم خطوط نامنظم کمتری میبنیم.

ت)

مانند مراحل قبل عمل میکنیم ابتدا از فیلتر گاوسی برای هموار سازی نیز استفاده میکنیم سپس از canny استفاده میکنیم.سپس از توابع find contours , approx. poly dp که در متن سوال ذکر شده استفاده میکنیم و اسم تمام شکل ها را در کد مشخص کرده ام. و در نهایت plot را رسم میکنیم.
نحوه تشخیص اشکال بر اساس تعداد گوشه ها و رئوس است که پیدا کردیم و بر این اساس اسم گذاری کرده ایم.

۸) در این سوال فقط قسمت هایی که پیاده سازی داشته را توضیح دادم.

ابتدا توابعی که در سوال ۵ داشتیم را در یک فایل پایتون قرار میدهیم و از آنها استفاده میکنیم. و سپس از canny استفاده میکنیم.

قسمت بعدی پیاده سازی مربوط به الگوریتم هاف بود که از این قسمت از اسلاید و فرمول ها در حل کردن آن استفاده کرده ام:

$$\rho = x \cos\theta + y \sin\theta$$

$$H(\rho, \theta) = H(\rho, \theta) + 1$$