***1:***

**قسمت الف**

*Noise reduction*

الگوریتم های آشکار سازی لبه حساس به نویز است و اگر تصویر دارای نویز بالا باشد ما خروجی مطلوب نخواهد بود.

*Gradient calculation*

لبه یک تصویر ممکن است در راستاهای مختلفی باشد، در نتیجه الگوریتم Canny از 4 فیلتر برای تشخیص لبه های افقی، عمودی و قطری در تصویر هموار شده استفاده می کند. عملگر لبه یاب ( برای مثال prewitt,sobel,Roberts) مشتق اول در راستای افقی و عمودی بدست می دهد.

*Non-maximum suppression*

حذف نقاط غیر بیشینه یک روش برای نازک کردن لبه است. حذف نقاط غیر بیشینه برای

بزرگترین لبه استفاده می شود. پس از محاسبات گرادیان، لبه بدست آمده از بزرگترین لبه یافتن

مقدار گرادیان همچنان به طور کامل تار است. تنها باید یک پاسخ دقیق به لبه وجود داشته

باشد. به این ترتیب حذف نقاط غیر بیشینه می تواند به حذف همه مقادیر گرادیان (با صفر

کردن آن ها) به جز بیشینه های محلی که نشان دهنده مکان هایی با سریعترین تغییر در مقدار

روشنایی هستند کمک کند.

*Double threshold*

پس از اعمال حذف غیر بیشینه، پیکسل های لبه باقی مانده، ارائه دقیقتری از لبه های واقعی

تصویر دارند. اگرچه بعضی از پیکسل لبه ها به علت نویز و تغییرات رنگ باقی می مانند. برای در

نظر گرفتن این پاسخ های غیر واقعی، فیلتر کردن پیکسل های لبه با مقدار گرادیان ضعیف و

نگه داشتن پیکسل های لبه با مقدار گرادیان زیاد ضروری است.

*Edge Tracking by Hysteresis*

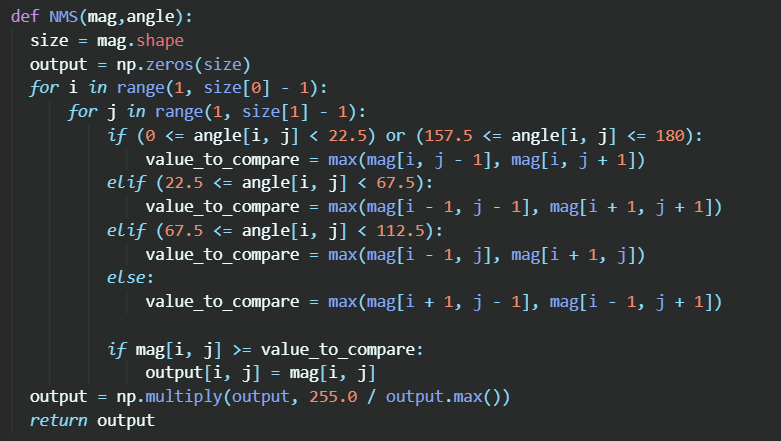
تا اینجا پیکسل های لبه قوی قطعاً باید در تصویر لبه نهایی جای بگیرند، زیرا آن ها از لبه های

واقعی در تصویر بدست آمده اند. اگر چه بحث هایی در مورد پیکسل های لبه ضعیف وجود دارد،

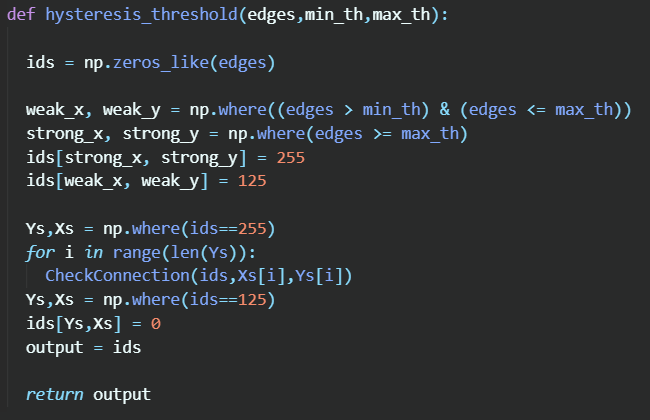
زیرا این پیکسل ها می تواند هم از لبه واقعی و هم از تغییرات رنگ/نویز بدست آیند. برای

رسیدن به نتیجه دقیق، لبه های ضعیف ناشی از تغییرات رنگ یا نویز باید حذف شوند

***قسمت ب:***



این قسمت برای NMS است



این قسمت برای threshold است.

نقاط بین Min , Max برابر 125 خواهد بود.نقاط بیشتر از Max برابر 255 یا سفید خواهد بود.

***قسمت ج:***

تصویر ورودی:



کف:0 و سقف:50



کف:0 و سقف:100



کف:0 و سقف:20



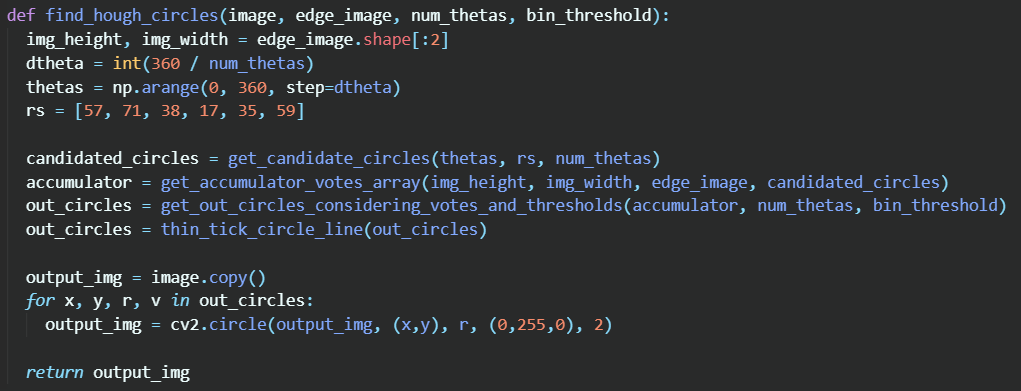
هر چه maximum threshold را بالاتر برده و فاصله ی بین max , min را بیشتر میکنیم تصاویر با جزییات کمتر و آبجکت های مهمتر خواهیم داشت.

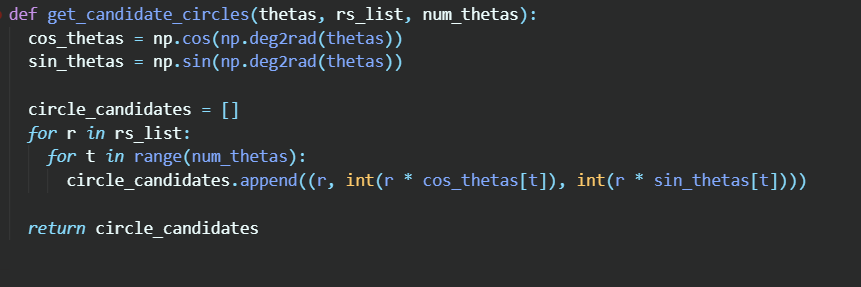
کف:50 و سقف:100



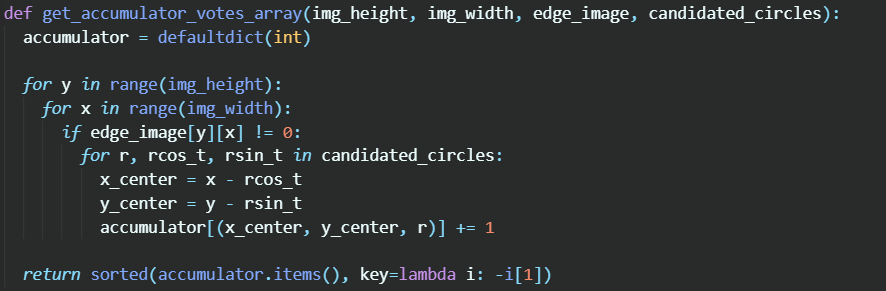
زیاد کردن کف نیز باعث کمتر شدن جزییات خواهد شد.

***2:***

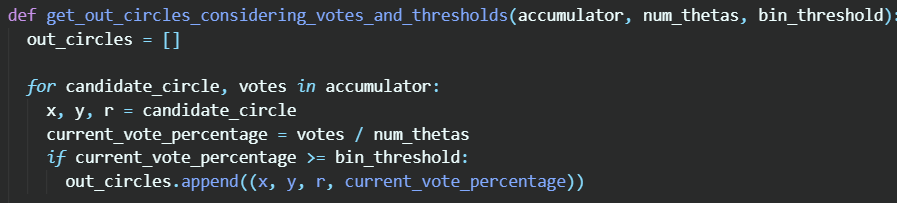


ابتدا طول و عرض تصاویر را گرفته و سپس زوایا را با توجه به تعداد زوایایی که ورودی داده ایم به دست می اوریم و سپس rs ها را به صورت ثابت مقداری که درون سوال گفته شده است قرار می دهیم. سپس بعد آن در فانکشن get\_candidate\_circles همه دایره های ممکن را تشکیل می دهیم.

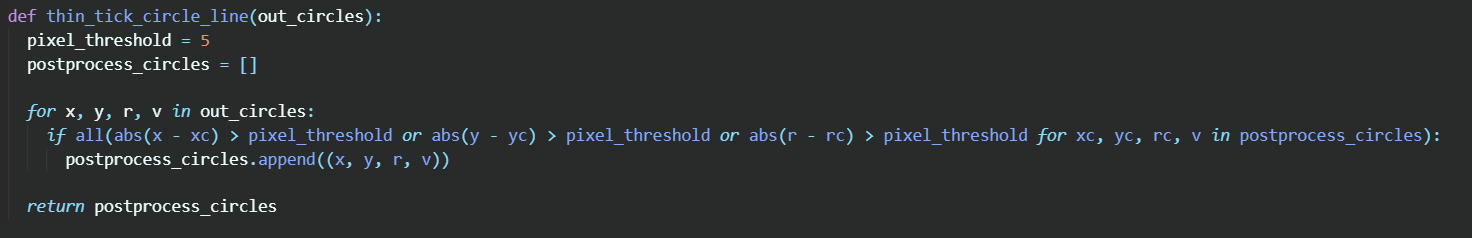
در فانکشن get\_accumulator\_votes\_array برای همه نقاطی که رنگی هستند امتیازی با توجه به اینکه درون دایره هستند یا نیستند به دست می آوریم.



در تابع get\_out\_circles\_considering\_votes\_and\_thresholds با توجه بهthreshold های تعریفی و امتیاز هایی که به دست آوردیم دایره های نهایی را به دست می اوریم.



و در آخرین فانکشن این فرایند با توجه به اشکالی که دیدیم و اینکه بعضا دایره ها کمی ضخیم تر از حدی که باید می شدند دایره ها را با threshold پنج پیکسل کمی نازک تر کردیم.



تصاویر خروجی:

















***3:***

با استفاده از کتابخانه اقدام به مقداردهی threshold ها میکنیم که در تصویر زیر مشخص است.



هیستوگرام تصویر:



دو نمونه تصویر که مقادیر 110 , 100 را برای global thresholding نشان می دهد.تصویر اول 100 و تصویر دوم 110 است.مشخص است که مقدار 100 بهتر است.



