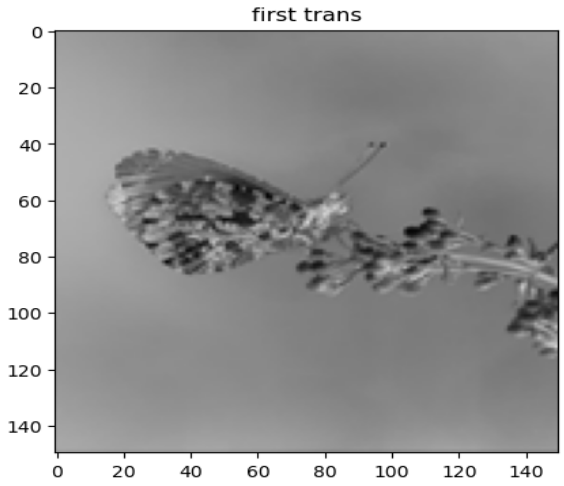
گزارش کار تمرین 1
املین غازاریان 9923056

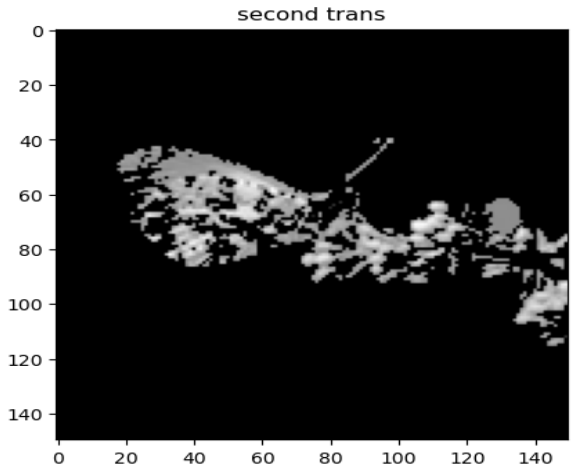
سوال 1:
توضیح contrast stretching:
یک تکنیک ساده برای افزایش کنتراست یک تصویر دیجیتال (با نگاشت مقادیر پیکسل در محدوده وسیع تری) است که می تواند دید جزئیات و ویژگی های یک تصویر را بهبود بخشد، به خصوص زمانی که تصویر اصلی دارای کنتراست کم یا نور ضعیف است.به بیان دیگر contrast stretching یک محدوده intensity موجود در تصویر را به محدوده intensity دیگری ترسیم می کند که برای افزایش دامنه دینامیکی سطوح خاکستری در تصویر استفاده می شود.
یک روش دیگری برای افزایش کنتراست نیز وجود دارد که به آن histogram equalization می گویند.در این روش، شکل کلی هیستوگرام تغییر می کند، در حالی که در contrast stretching، شکل کلی هیستوگرام ثابت می ماند.
حال برای انجام contrast stretching نیاز است اول محدوده مقدار پیکسل ها تعیین شود که می دانیم برای تصویر gray scale بین 0 تا 255 است

نمودار یک:



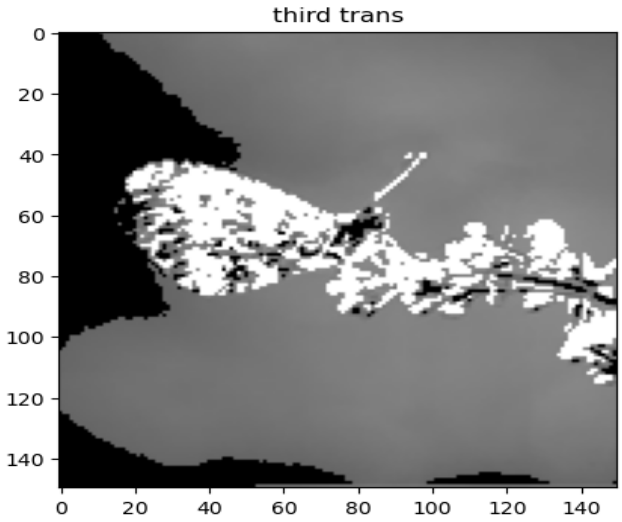
از این نمودار می توان فهمید که دارد intensity پیکسل ها را قرینه می کند یعنی در واقع جاهایی که روشن ترین است تیره ترین می شود و برعکس.که برای نمایش تصویر stretch شده فقط کافیست تصویر gray scale را از حداکثر مقدار پیکسل ها یعنی 255 کم کنیم تا تمام مقادیر پیکسل ها قرینه شوند.

نمودار دوم:



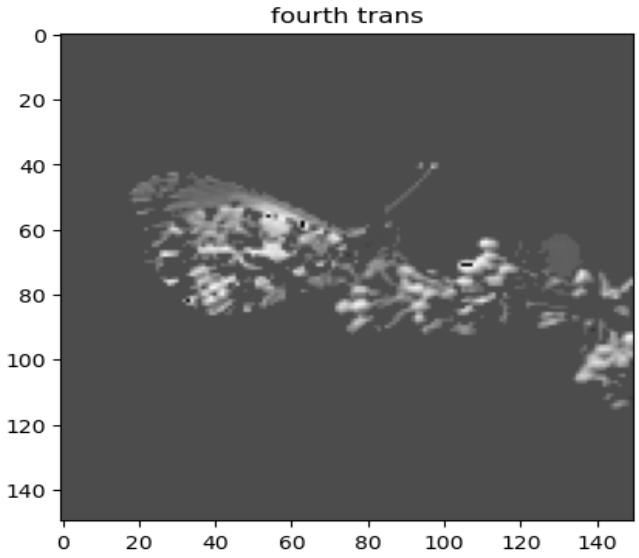
از این نمودار می توان فهمید که پیکسل هایی که شدت روشنایی آن ها در بازه ی بین 0.2x225 و 0.55x255 است به رنگ مشکی ببینیم ولی آن قسمت هایی که نزدیک سفید یا نزدیک مشکی هستند را بدون تغییر ببینیم

نمودار سوم:



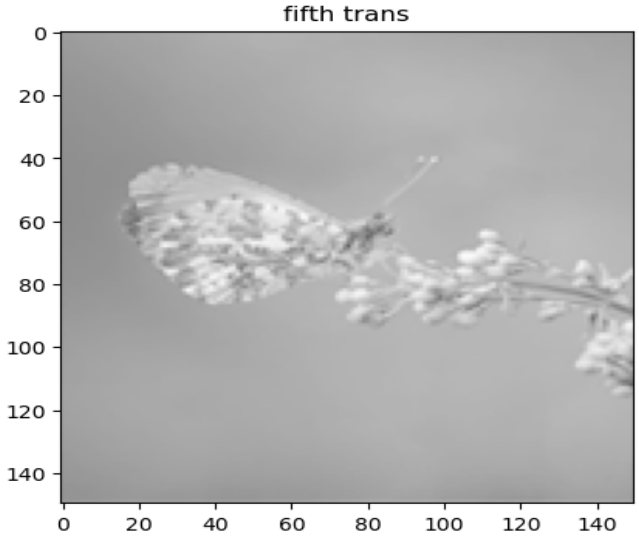
از این نمودار می توان فهمید که پیکسل هایی که شدت روشنایی ان ها در بازه ی بین 0.4x225 و 0.55x255 است تغییری در شدت روشنایی نداشته باشند ولی قسمت هایی که نزدیک سفید است مشکی و قسمت هایی که نزدیک مشکی است سفید ببینیم

نمودار چهارم:



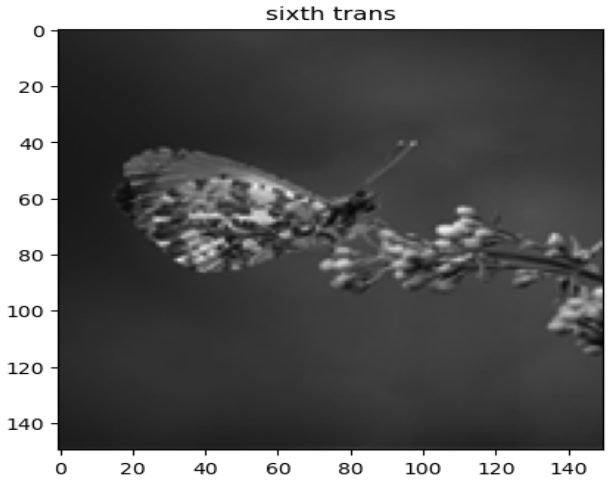
از این نمودار می توان فهمید که آن قسمت هایی که شدت روشنایی کمی دارند کمی روشن تر شود و قسمت هایی که شدت روشنایی زیادی دارند کمی تیره تر شوند و در بازه ی وسط شدت روشنایی برابر 0.3 باشد

نمودار پنجم:



برای stretch کردن دو نمودار آخر باید اول روی یک اسکیل کنیم (یعنی 255 را روی یک اسکیل بکنیم) که می دانیم اگر از اعداد بین صفر و یک جذر بگیریم حاصل از خود عدد بزرگتر می شود پس باید شدت روشنایی به صورت ناگهانی(غیر خطی) و بسیار زیادی افزایش یابد (در نزدیکی intensity صفر) پس یعنی تصویر روشن تر می شود

نمودار ششم:



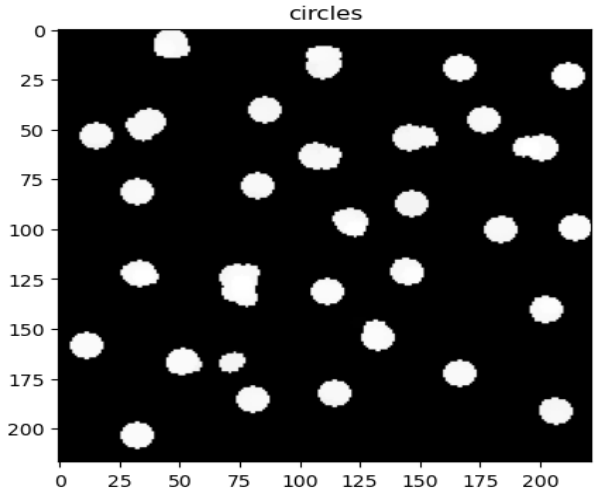
می دانیم اگر اعداد بین صفر و یک را به توان 2 برسانیم حاصل از خود عدد کوچکتر می شود یعنی به شدت روشنایی کمتر میل می کند پس تصویر تیره تر می شود.

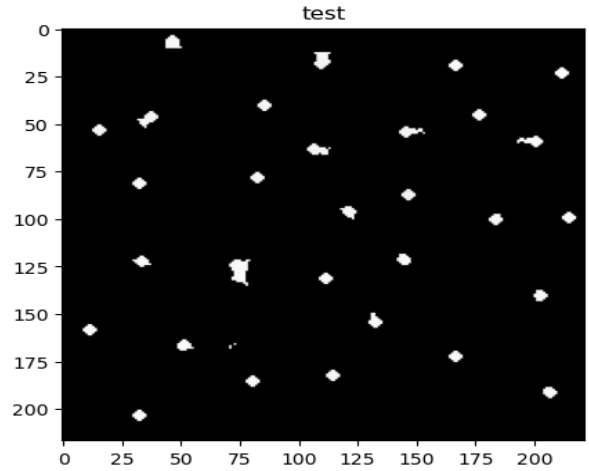
سوال 2:

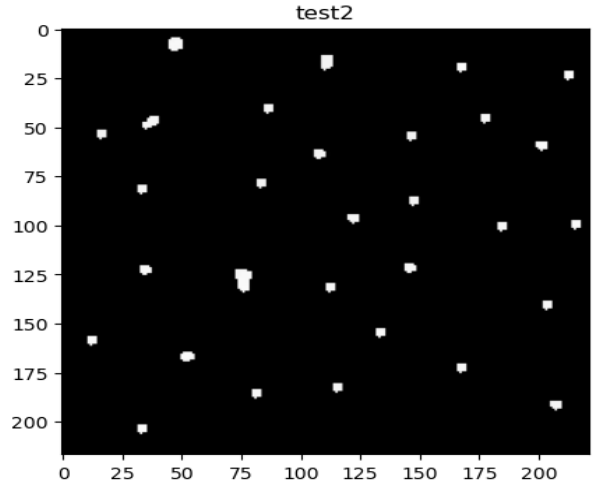
در تصویری که داریم تغییر رنگ آن چنانیی نداریم در نتیجه وقتی رنگ ها را از تصویر جدا کنیم تفاوت خاصی بین تصاویر احساس نخواهیم کرد . می دانیم تبدیلات linear interpolation باعث وضوح تصویر می شوند و از آن جایی که تصویر ما به اندازه کافی واضح است برای همین بعد اعمال این تبدیلات تغییر آن چنانی مشاهده نخواهیم کرد به همین دلیل در تابع linear interpolation مقدار fx و fy را کمتر از یک قرار می دهیم تا وضوح تصویر بد شود و بتوانیم تغییری مشاهده کنیم (زیرا اگر عددی بیشتر از یک بدیم وضوح تصویر را بیشتر خواهد کرد که تغییری مشاهده نمی کنیم). در نهایت تصاویر به وجود آمده برای هر سه رنگ را merge می کنیم و نمودار هیستوگرام آن را نمایش می دهیم.

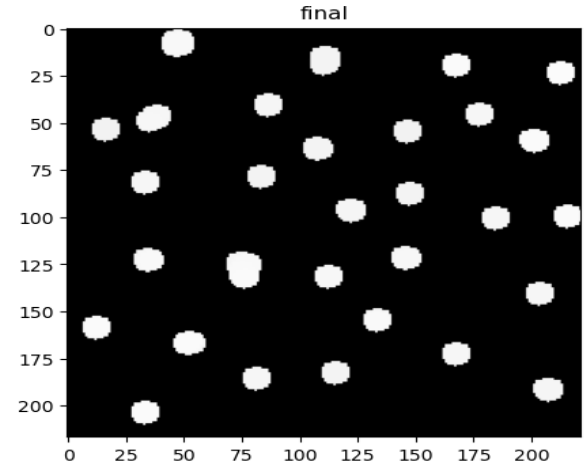
سوال 3:

جدا کردن دایره ها از تصویر:
برای جدا کردن دایره ها از تصویر اول باید یک کرنل دایره ای با شعاع 9 تعریف می کنیم(مقدار شعاع دایره از آزمون و خطا بدست آمده) سپس با استفاده از opening که یک morphological operator است که با توجه به سایز کرنل اجسامی که با سایز کرنل همخوانی ندارد حذف می کند پس مشاهده می کنیم که خطوط را از تصویر حذف می کند ولی می توان دید که با یک خطایی این عمل حذف انجام شده ، بعضی از تقاطع دایره ها با خط ها را به عنوان دایره تشخیص داده است پس برای حذف این دایره های اضافی از erode استفاده می کنیم که می دانیم erosion با توجه به شکل و سایز کرنل مرز تصاویر را حذف می کند و باعث کاهش سایز اجسام تصویر می شود . حال با استفاده از erode آن نواحی سایزشان بسیار کمتر می شود و مانند نقطه هایی به نمایش می آیند حال دوباره از opening استفاده می کنیم که در واقع opening نویز را حذف می کند که منظور از نویز در اینجا همان نقاطه کوچکی است که توسط erode کردن ایجاد شده. سپس dilate می کنیم سایز دایره ها به حالت عادی خود بر می گردد .(ترتیب شکل ها از چپ به راست)

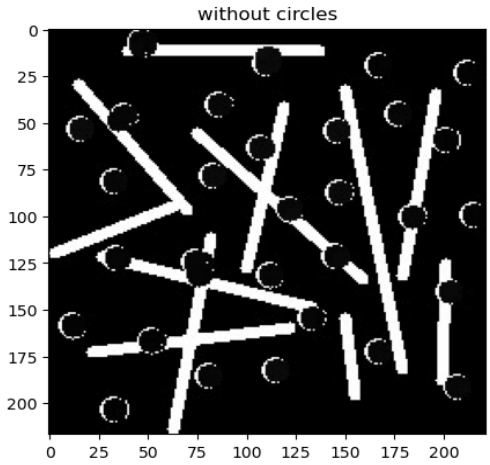


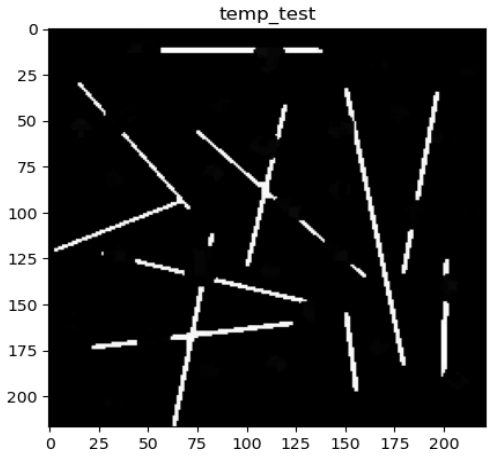


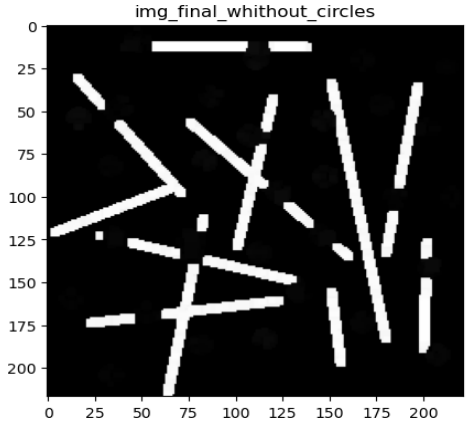




حال اگر بخواهیم دایره ها را از تصویر حذف کنیم و محل تقاطع آن ها با خطوط مشخص باشد باید تصویر خروجی قسمت قبل را از تصویر اولی (همان تصویر ورودی) کم کنیم که این باعث می شود دایره ها از تصویر حذف شوند ولی همراه با خطایی. برای حذف قسمت های باقی مانده از دایره ها از erode استفاده می کنیم . ابتدا یک کرنل مربعی با ضلع 4 تعریف می کنیم سپس با این کرنل erode می کنیم و آن قسمت های باقی مانده حذف می شوند سپس با dilation سایز خطوط را با حالت عادی برمی گردانیم (در erode کردن سایز کوچک می شود).

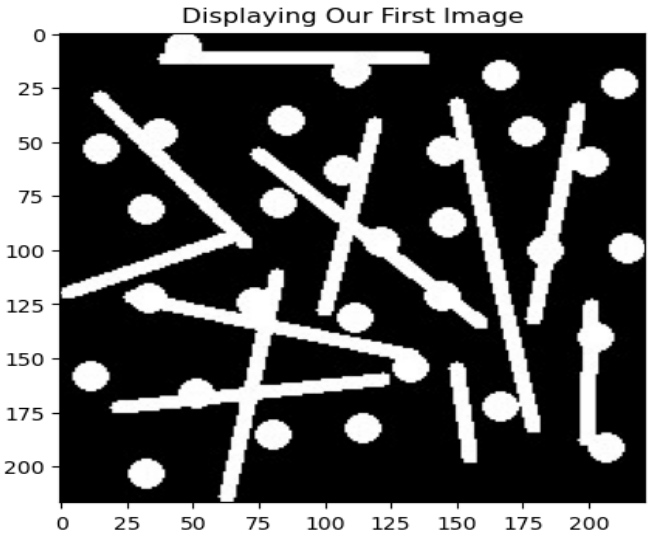


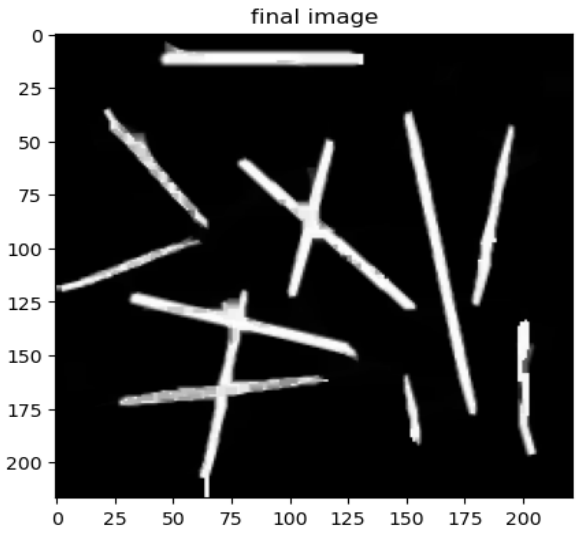




جدا کردن خطوط:

برای تشخیص خطوط روشی که وجود دارد این است که خطوط عمودی و افقی را از تصویر اصلی تشخیص دهیم ، این طور که 2 تا کرنل بنویسیم به سایز 1 در 20 تا خطوط افقی را تشخیص دهد و یک کرنل دیگر بنویسیم به سایز 20 در 1 تا خطوط عمودی را تشخیص دهیم و آنها را در تصاویر جداگانه ذخیره کنیم. حال برای تشخیص خطوطی که عمودی یا افقی نیستند باید تصویر اصلی را بچرخانیم با زاویه ی ثابتی تا خطوط عمودی و افقی ایجاد شود و آن دو کرنل را که برای خطوط افقی و عمودی نوشتیم هر بار پیاده کنیم سپس با هر بار چرخاندن و ذخیره خطوط عمودی و افقی در تصاویر جداگانه باید حواسمان باشد تا همان درجه که چرخواندیم همان درجه تصویر را برگردانیم سپس دوباره ذخیره کنیم. در نهایت همه این تصاویر ایجاد شده در هر بار چرخاندن را باهم جمع میکنیم .سپس از closing استفاده می کنیم تا قسمت های خالی یا کم رنگ خطوط را پر کند .
برای انتخاب سایز کرنل ها از آزمون و خطا استفاده می کنیم تا همه ی خطوط تشخیص داده شوند (در بعضی از rotation ها سایز کرنل متفاوت از بقیه است زیرا در آن زاویه خطوط به خوبی تشخیص داده نمی شده اند و مجبور به تغییر سایز کرنل شده ام)





سوال 4:

بررسی smoothing filter های متفاوت بر روی الگوریتم های لبه یابی:

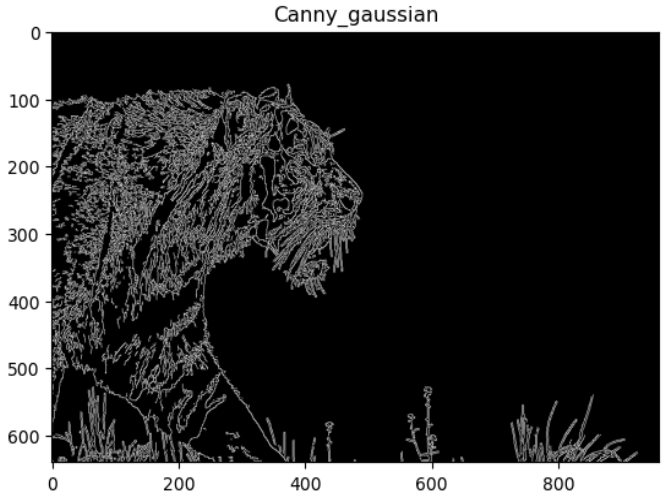
دلیل استفاده از این فیلترها: از این فیلترها استفاده استفاده می کنیم تا نویز را کاهش دهیم. زیرا در لبه یابی مشتقات عددی چگالی پیکسل ها را محاسبه می کنیم که این باعث می شود لبه های نویزی به وجود آید. Blur کردن تغییرات چگالی نزدیک لبه ها را کمتر می کند و باعث می شود لبه غالب را آسان تر تشخیص دهیم.

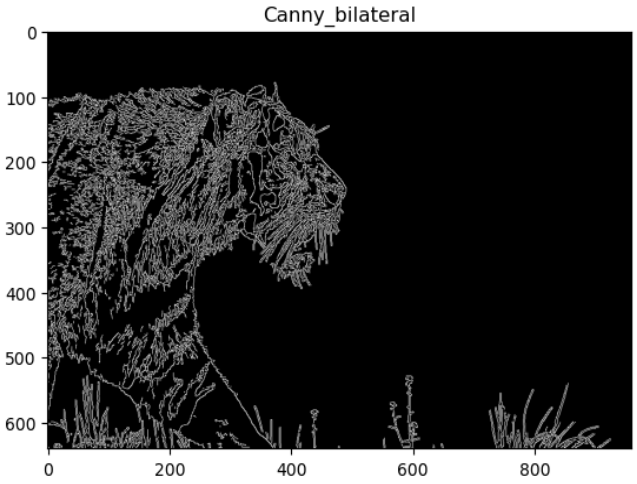
اولی ← median filter: چگالی pixel های را که دور pixel مرکزی هستند(در یک کرنل nxn) محاسبه می کند و سپس چگالی پیکسل مرکزی را جایگذاری می کند.

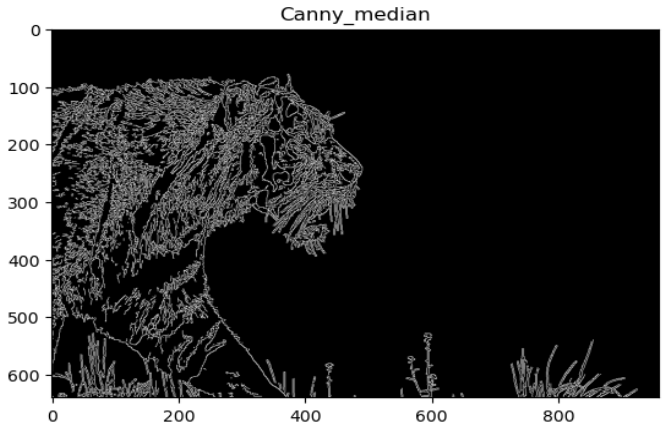
دومی ← gaussian filter: فیلتر گوسی میانگین وزنی پیکسل‌های اطراف را در بر می‌گیرد و یک پارامتر سیگما دارد. فیلتر گوسی لبه‌های یک تصویر را محو می‌کند (مانند فیلتر میانگین)، عملکرد بهتری در حفظ لبه‌ها نسبت به فیلترهای میانگین با اندازه مشابه دارد.

سومی← bilateral filter: هر پسکل را با میانگین وزنی شدت پیکسل ها مجاور جایگزین می کند که برای کاهش نویز و حفظ لبه می شود.

ابتدا روش canny را بررسی می کنیم:

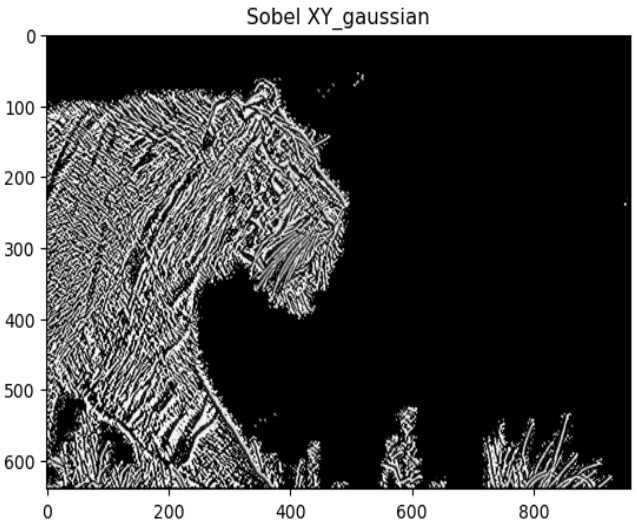


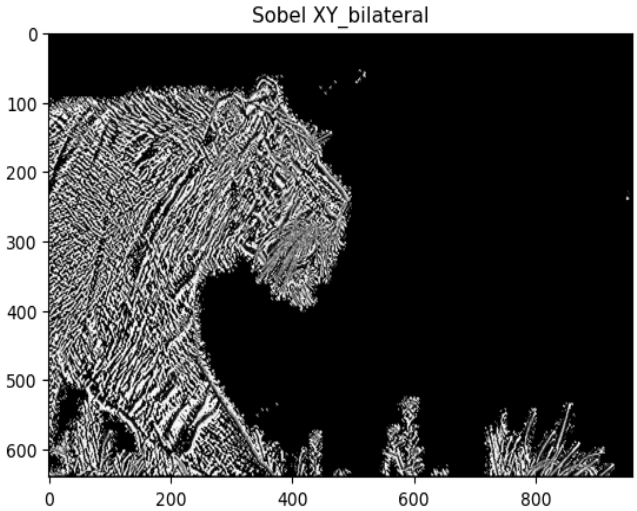


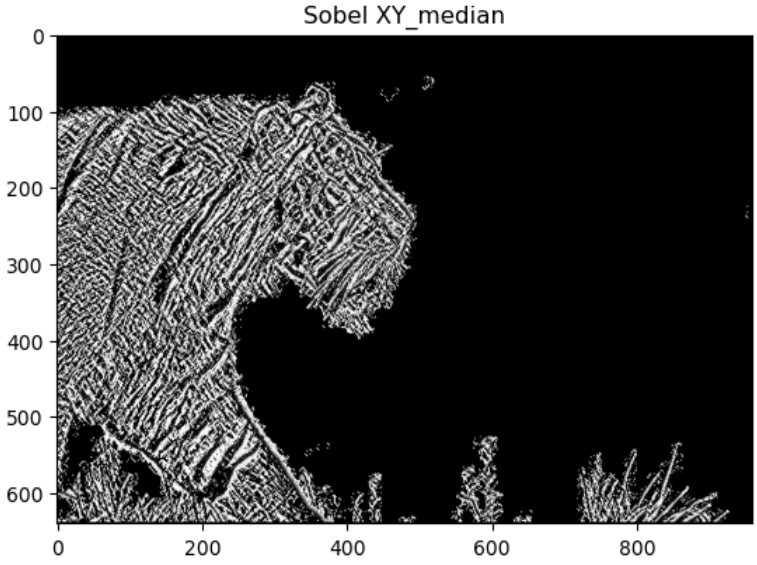


با تغییر سایز کرنل می توان مشاهده کرد که هر چه سایز کرنل کمتر باشد این روش بهتر جواب می دهد و لبه یابی بهتر صورت می گیرد

روش sobel:







می توان مشاهده کرد با فیلتر bilateral ضخامت لبه های بیشتر و دقت بیشتری در لبه ها وجود دارد.