

توضیح دو الگوریتم sobel و kirsch

:sobel operator

برای پیاده سازی این الگوریتم از تابع `sobel.m` استفاده شده است ، که دو ورودی `image` (تصویری که قصد یافتن لبه های آن را داریم که باید با فرمت مشابه با `'pic1_section1.jpg'` وارد شود) و ورودی `plot` (که در صورت ۱ بودن هر دو تصویر اصلی و تصویر لبه یابی شده نشان داده خواهند شد و در غیر اینصورت ، تصویری مشاهده نمی شود) می گیرد و خروجی `new_pic` را می دهد که ماتریس متناظر با تصویر لبه یابی شده و فرمت `uint8` می باشد ، مراحل پیاده سازی به صورت مختصر و به شرح زیر آمده است :

۱. ابتدا تصویر را لود کرده و در صورتی که به صورت رنگی باشد ، آن را در حالت `gray` `scale` قرار می دهیم و در غیر اینصورت به همان شکل کار را ادامه می دهیم .
 ۲. `Gx` و `Gy` را مطابق با صورت پروژه تعیین کرده و با استفاده از دستور `conv2` ، طی دو بخش ابتدا `Gx` با `pic_gray` و سپس `Gy` را با `pic_gray` کانولوشن دو بعدی می گیریم و سپس با استفاده از ماتریس های حاصل از کانولوشن تصویر لبه یابی شده را به دست می آوریم ، هم چنین با توجه به اینکه دستور `conv2` طبیعتاً اندازه ی ماتریس `pic_gray` را اضافه می کند ، برای جلوگیری از این کار از دستور `'same'` در بدنه ی دستور `conv2` استفاده شده است .

۳. در مرحله ی آخر نیز در صورتی که ورودی `plot` ، یک باشد ؛ با استفاده از `subplot` هر دو تصویر اصلی و لبه یابی شده در کنار هم به نمایش در خواهند آمد .

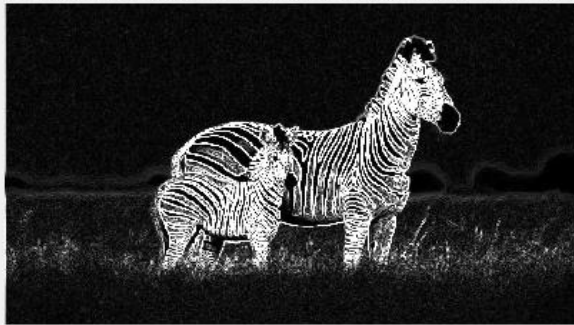
در ادامه ی این قسمت بر روی سه تصویر با نام های `pic1_section1` ، `pic2_section1` و `pic3_section1` ، این الگوریتم پیاده شده و نتایج حاصل از آن به همراه زمان انجام آن را در صفحه ی بعد مشاهده می کنیم .

the gray original image



time that sobel_algorithm for pic1_section1.jpg takes :
Elapsed time is 0.179667 seconds.

the edge detected of original image with *sobel* algorithm

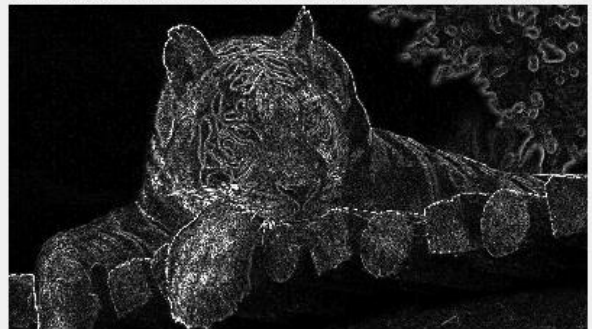


the gray original image



time that sobel_algorithm for pic2_section1.jpg takes :
Elapsed time is 0.284433 seconds.

the edge detected of original image with *sobel* algorithm



the gray original image



time that sobel_algorithm for pic3_section1.jpg takes :
Elapsed time is 0.268793 seconds.

the edge detected of original image with *sobel* algorithm



۱.۲ Kirsch Operator

برای پیاده سازی این الگوریتم از تابع `Kirsch.m` استفاده شده است ، که دو ورودی `image` (تصویری که قصد یافتن لبه های آن را داریم که باید با فرمت مشابه با `'pic1_section1.jpg'` وارد شود) و ورودی `plot` (که در صورت ۱ بودن هر دو تصویر اصلی و تصویر لبه یابی شده نشان داده خواهند شد و در غیر اینصورت ، تصویری مشاهده نمی شود) می گیرد و خروجی `new_pic` را می دهد که ماتریس متناظر با تصویر لبه یابی شده و فرمت `uint8` می باشد ، مراحل پیاده سازی به صورت مختصر و به شرح زیر آمده است :

۱. ابتدا تصویر را لود کرده و در صورتی که به صورت رنگی باشد ، آن را در حالت `gray` `scale` قرار می دهیم و در غیر اینصورت به همان شکل کار را ادامه می دهیم .
۲. سپس کرنل سه بعدی `g` را تشکیل می دهیم که ۸ طبقه داشته و هر طبقه شامل یک ماتریس ۳ در ۳ می باشد که در آن ماتریس دو بعدی مربوط به هر طبقه از چرخش ۴۵ درجه ای ماتریس طبقه ی قبلی خود به دست می آید .
۳. در مرحله ی بعد هم ماتریسی به نام `h` را در نظر گرفته که ۸ طبقه داشته و ابعاد ماتریس موجود در هر طبقه را هم برابر با ماتریس `pic_gray` در نظر می گیریم و با استفاده از دستور `conv2` هر طبقه از ماتریس `h` را با طبقه ی متناظر از ماتریس `g` کانولوشن می گیریم . و در نهایت با استفاده از دستور `max(h,[],3)` ، روی طبقه ی ماتریس `h` ماکسیمم می گیریم تا تصویر لبه یابی شده به نام `new_pic` را به دست آوریم .
۴. در مرحله ی آخر نیز در صورتی که ورودی `plot` ، یک باشد ؛ با استفاده از `subplot` هر دو تصویر اصلی و لبه یابی شده در کنار هم به نمایش در خواهند آمد .

مشابه قسمت قبل ، این الگوریتم بر روی سه تصویر مذکور پیاده شده که نتایج در صفحه ی بعد قابل مشاهده می باشد .

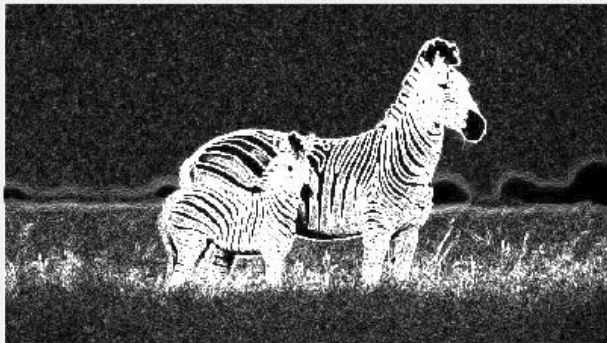
در انتخاب تصاویر سعی شده است که تصاویری در نظر گرفته شود که به صورت طبیعی پتانسیل های بیشتری برای داشتن لبه داشته باشند و یا در واقع تغییرات شدید رنگی داشته باشند .

the gray original image



time that Kirsch_algorithm for pic1_section1.jpg takes :
Elapsed time is 0.317852 seconds.

the edge detected of original image with *Kirsch* algorithm

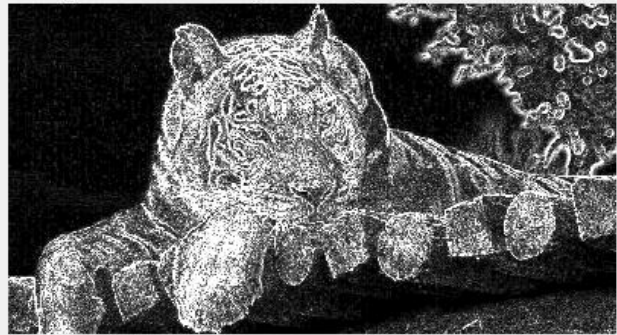


the gray original image



time that Kirsch_algorithm for pic2_section1.jpg takes :
Elapsed time is 0.590492 seconds.

the edge detected of original image with *Kirsch* algorithm

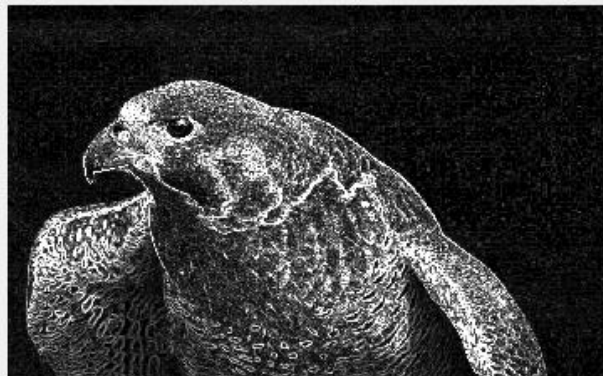


the gray original image



time that Kirsch_algorithm for pic3_section1.jpg takes :
Elapsed time is 0.547490 seconds.

the edge detected of original image with *Kirsch* algorithm



همانطور که از نتایج بر می آید ، الگوریتم Kirsch نسبت به الگوریتم sobel دارای دقت بیشتر است و هم چنین نیز زمان بیتشری برای پیاده سازی نیاز دارد .