

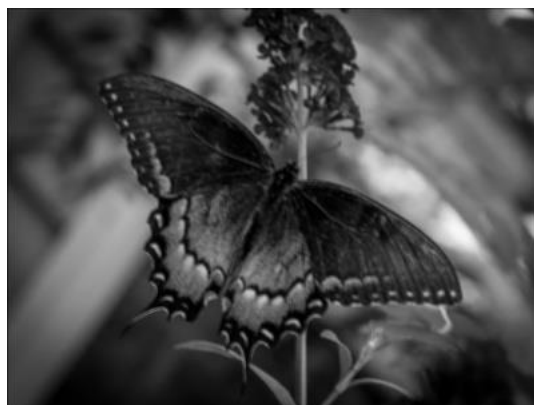
ابتدا تصویر مورد نظر را با استفاده از دستور `cv2.imread` می‌خوانیم. سپس با دستور `cv2.resize` سایز آن را به  $300 \times 400$  تبدیل می‌کنیم.

از یک فیلتر میانگین‌گیر نرمالیزه شده  $3 \times 3$  به عنوان فیلتر پایین‌گذر استفاده می‌کنیم. سپس با استفاده از دو حلقه `for` تو در تو و پرمایش تصویر، خروجی‌های حاصل از اعمال فیلتر پایین‌گذر و فیلترهای لبه‌یاب لبه‌های عمودی و افقی را بدست می‌آوریم.

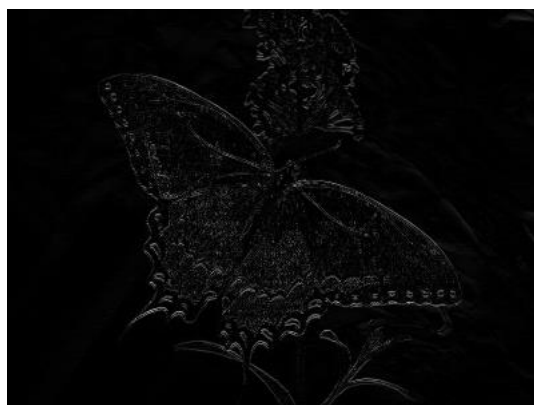
از آن جایی که در تصویر حاصل از فیلتر کردن تصویر اصلی به وسیله فیلترهای لبه‌یاب لبه‌های عمودی و افقی، هدف پیدا کردن لبه‌های عمودی و افقی است و نباید تفاوتی بین لبه‌های حاصل از رفتن از نقاط تاریک به روشن و لبه‌های حاصل از رفتن از نقاط روشن به تاریک وجود داشته باشد، با استفاده از دستور `numpy.absolute`، قدر مطلق تصویر خروجی را محاسبه می‌کنیم.

از آن جایی که تصویر حاصل از فیلتر کردن به وسیله فیلتر پایین‌گذر فقط حاوی فرکانس‌های پایین است با کم کردن آن از تصویر اصلی، تصویر بالاگذر شده را بدست می‌آوریم. اما چون تصویر حاصل دارای مقادیر منفی است، برای نمایش آن مینیمم مقدار تصویر را به صفر، ماکسیمم مقدار آن را به 255 و بقیه مقادیر را به مقادیر متناظر منتقل می‌کنیم.

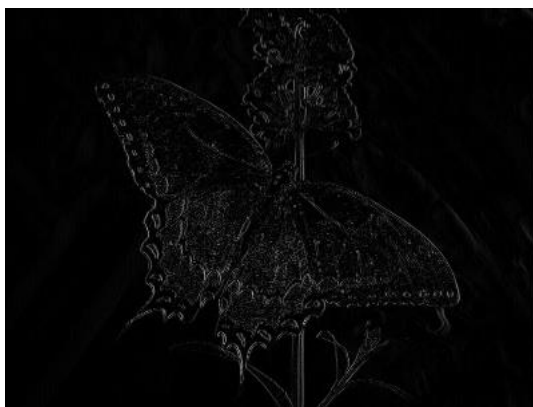
فیلتر پایین‌گذر:



فیلتر لبه‌یاب لبه عمودی:



فیلتر لب‌یاب لبه افقی:



تصویر بالاگذر شده:



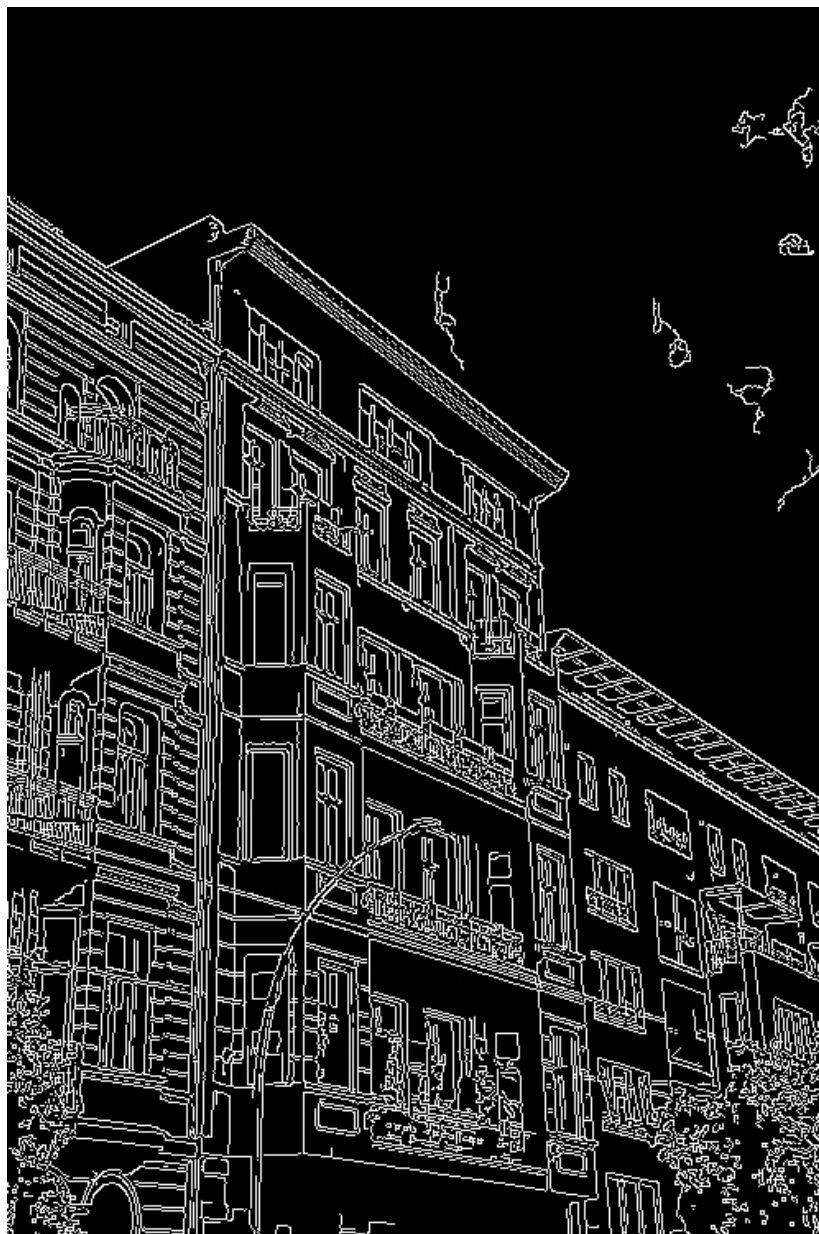
سوال 4:

با استفاده از دستورهای `cv2.Sobel` و `cv2.threshold` عملگر لبه‌یاب Sobel را پیاده‌سازی می‌کنیم.

ابتدا با استفاده از دستور `cv2.Sobel` تصاویر اصلی را با کرنل‌های 3 در 3 در دو راستای  $x$  و  $y$  فیلتر می‌کنیم. سپس برای هر پیکسل به طور مجزا اندازه بردار حاصل را بدست می‌آوریم. در انتها با استفاده از دستور `cv2.threshold` پیکسل‌های لبه را مشخص می‌کنیم.



با استفاده از دستور cv2.Canny عملگر لبه‌یاب Canny را پیاده‌سازی می‌کنیم.



با استفاده از دستورهای `cv2.GaussianBlur`، `cv2.laplacian` و `cv2.threshold` عملگر لبه‌یاب LoG را پیاده‌سازی می‌کنیم.

ابتدا با استفاده از دستور `cv2.GaussianBlur` تصاویر اصلی را با کرنل 3 در 3 فیلتر می‌کنیم. سپس با استفاده از دستور `cv2.laplacian` شیب عبور از صفر را برای هر پیکسل محاسبه می‌کنیم. در انتها با استفاده از دستور `cv2.threshold` روی قدر مطلق خروجی قسمت قبل، پیکسل‌های لبه را مشخص می‌کنیم.



مشخص‌ترین نتیجه‌ای که می‌توان مشاهده کرد این است که نازک‌ترین لبه‌ها مربوط به روش Canny است. برتری دیگر این روش پیوستگی بیشتر در لبه‌ها نسبت به دو روش دیگر است. در واقع تکه تکه بودن لبه‌ها در این روش کمتر است. برتری روش LoG در دقت بین پیکسلی لبه‌ها را تشخیص می‌دهد. البته شایان ذکر است که روش Sobel هم کیفیت بسیار خوبی دارد.

یک روش مناسب برای افزایش کیفیت تشخیص لبه‌ها و حذف نویز استفاده از فیلترهای پایین‌گذر به خصوص فیلتر گوسی قبل از شروع عملیات لبه‌یابی است. روش دیگر استفاده از عملگرهای morphological نظیر فیلترهای opening و closing است.

سوال 6:

پارامتر minDistBetweenBlobs حداقل فاصله دو لکه از یکدیگر را بر حسب پیکسل تعیین می‌کند.

برای یافتن لکه از دستوره‌های cv2.SimpleBlobDetector استفاده می‌کنیم. این توابع پارامترهای متنوعی دارند که روش یافتن لکه و نحوه تصمیم‌گیری را مشخص می‌کنند. در ادامه به توضیح این پارامترها می‌پردازیم:

فیلتر بر اساس اندازه:

در صورت یک بودن filterByArea لکه‌ها بر اساس مساحت لکه بر حسب پیکسل، فیلتر می‌شوند. برای این کار از دو مقدار minArea و maxArea استفاده می‌شود و لکه‌هایی که مساحت آن‌ها بین این دو مقدار باشد، تایید می‌شوند. این فیلتر روی لکه‌های ردیف اول تاثیر دارد.

فیلتر بر اساس شدت:

در حالت کلی فقط لکه‌هایی که intensity آن‌ها بین minThreshold و maxThreshold باشد، تایید می‌شوند. هر چه thresholdStep کوچک‌تر باشد، دقت بررسی بیشتر خواهد بود. این فیلتر روی لکه‌های ردیف دوم تاثیر دارد.

فیلتر بر حسب شکل:

در صورت یک بودن filterByCircularity لکه‌ها بر اساس میزان دایروی بودن شکل لکه، فیلتر می‌شوند. برای این کار از دو مقدار minCircularity و maxCircularity استفاده می‌شود و لکه‌هایی که میزان دایروی بودن آن‌ها بین این دو مقدار باشد، تایید می‌شوند.

برای محاسبه میزان دایروی بودن یه شکل از فرمول زیر استفاده می‌شود:

$$\frac{4\pi(Area)}{(perimeter)^2}$$

این فیلتر روی لکه‌های ردیف سوم تاثیر دارد.

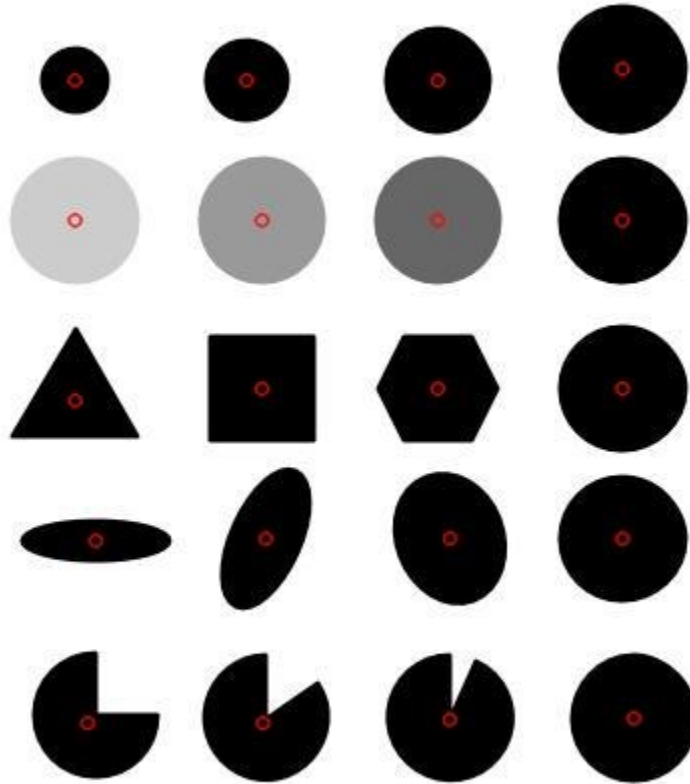
در صورت یک بودن filterByInertia لکه‌ها بر اساس میزان کشیدگی شکل لکه، فیلتر می‌شوند. برای این کار از دو مقدار minInertiaRatio و maxInertiaRatio استفاده می‌شود و لکه‌هایی که میزان کشیدگی آن‌ها بین این دو مقدار باشد، تایید می‌شوند. این فیلتر روی لکه‌های ردیف چهارم تاثیر دارد.

در صورت یک بودن filterByConvexity لکه‌ها بر اساس میزان لختی شکل لکه، فیلتر می‌شوند. برای این کار از دو مقدار minConvexity و maxConvexity استفاده می‌شود و لکه‌هایی که میزان لختی آن‌ها بین این دو مقدار باشد، تایید می‌شوند.

برای محاسبه میزان لختی بودن یه شکل از فرمول زیر استفاده می‌شود:

$$\frac{\text{Area of the bolb}}{\text{Area of the bolb's convex hull}}$$

این فیلتر روی لکه‌های ردیف پنجم تاثیر دارد.



(ب) سوال 3:

ابتدا با استفاده از سوال 1 ویدئویی ذخیره می‌کنیم.

مشخص‌ترین نتیجه‌ای که می‌توان مشاهده کرد این است که نازک‌ترین لبه‌ها مربوط به روش Canny است. برتری دیگر این روش پیوستگی بیشتر در لبه‌ها نسبت به دو روش دیگر است. در واقع تکه تکه بودن لبه‌ها در این روش کمتر است. ضخامت لبه‌ها در دو روش دیگر تقریباً برابر است. اما در روش Sobel نسبت به روش Prewitt، لبه‌ها پرتز هستند. همچنین نویز روش Prewitt از دو روش دیگر بیشتر است و این روش نسبت به تغییر مقدار threshold حساستر است. در حالتی که در ابتدا فیلترینگ گوسی انجام می‌شود در هر سه روش نویز به طور چشمگیری کاهش می‌یابد.

