



باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

۲۵۷۴۲ گروه ۴ - سیگنال‌ها و سیستم‌ها - بهار ۱۳۹۸ - ۹۹

پروژه امتیازی

موعد تحویل: دوشنبه ۳۰ تیر ۱۳۹۹، ساعت ۲۳:۵۵

نحوه‌ی تحویل:

- گزارش تمرین خود را در قالب یک فایل pdf. تحویل دهید. در گزارش لازم است تمامی خروجی‌ها و نتایج نهایی، پرسش‌های متن تمرین، و توضیح مختصری از فرآیند حل مسأله‌ی خود در هر قسمت را ذکر کنید.
- کد کامل تمرین را در قالب یک فایل m. تحویل دهید. لازم است بخش‌های مختلف تمرین در section‌های مختلف تفکیک شوند و کد تحویلی منظم و دارای کامنت‌گذاری مناسب باشد. بدیهی است آپلود کردن کدی که به درستی اجرا نشود، به منزله‌ی فاقد اعتبار بودن نتایج گزارش شده نیز می‌باشد.
- توابعی را که (در صورت لزوم) نوشته‌اید، در قالب فایل‌های m. در کنار فایل‌های گزارش و کد اصلی تمرین، ضمیمه کنید.
- مجموعه‌ی تمامی فایل‌ها (گزارش، کد اصلی، توابع، و خروجی‌های دیگر در صورت لزوم) را در قالب یک فایل zip/.rar. ذخیره کرده و از طریق سامانه‌ی CW تحویل دهید.
- نام‌گذاری فایل‌های تحویلی را به صورت HW03_StudentNumber.pdf/.m/.zip/.rar انجام دهید.

معیار نمره دهی:

- ساختار مرتب و حرفه‌ای گزارش
- استفاده از توابع و الگوریتم‌های مناسب
- پاسخ به سؤالات تئوری و توضیح روش‌های مطلوب سوال
- کد و گزارش خروجی کد برای خواسته‌های مسأله

نکات تکمیلی:

- همواره در تمامی تمارین و پروژه‌ها، تا سقف ۱۰٪ نمره اضافه برای قسمت‌های امتیازی و نیز هر گونه روش‌های ابتکاری و فرادرس در نظر گرفته می‌شود و سقف نمره‌ی قابل کسب معادل با ۱۱۰/۱۰۰ می‌باشد.
- شرافت انسانی ارزشی به مراتب بالاتر از تعلقات دنیوی دارد. رونویسی تمارین، زیر پا گذاشتن شرافت خویشتن است؛ به کسانی که شرافتشان را زیر پا می‌گذارند هیچ نمره‌ای تعلق نمی‌گیرد.

۱ الگوریتم‌های یافتن لبه در تصاویر دیجیتال

در این بخش به بررسی دو روش معروف برای یافتن لبه‌های موجود در یک تصویر دیجیتال می‌پردازیم.

۱.۱ Sobel Operator

عملگر Sobel-Feldman، که در سال ۱۹۶۸ توسط **Irwin Sobel** و **Garry Feldman** معرفی شد، یک روش کلاسیک برای تشخیص لبه‌های یک تصویر است. این عملگر، یک عملگر مشتق‌گیری است که تقریبی از گرادیان شدت نور تصویر را در هر نقطه ارائه می‌کند. قدرت این روش در این است که از نظر محاسباتی بسیار ساده است و به این دلیل تا کنون نیز مورد استفاده قرار می‌گیرد.

این عملگر از دو کرنل 3×3 استفاده کرده و دو تصویر زیر را تولید می‌کند.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \quad (۱)$$

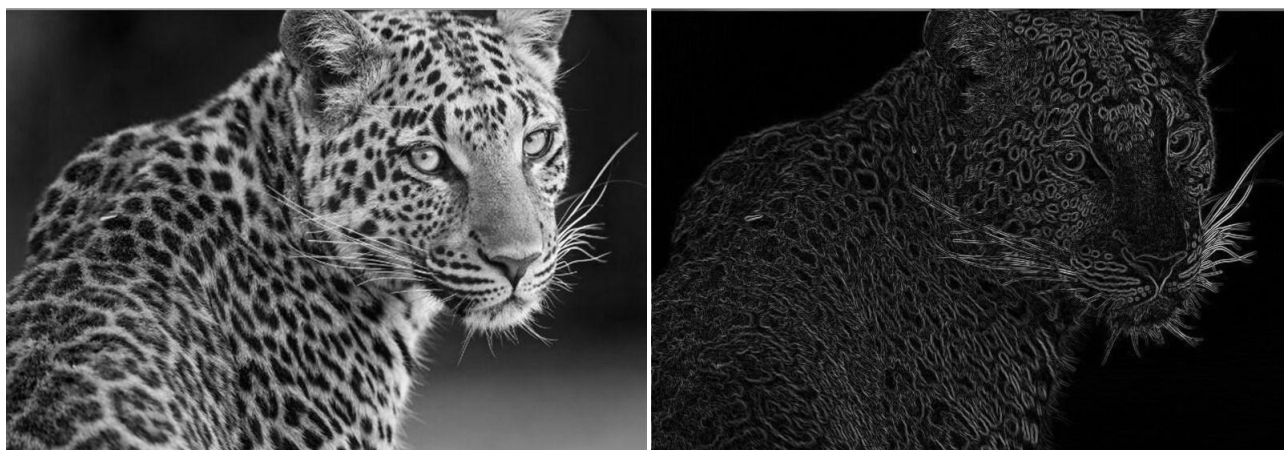
و

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & +2 & +1 \end{bmatrix} * A \quad (۲)$$

که در آن ماتریس A سیگنال دو بعدی تصویر دیجیتال ماست. دو تصویر فوق، تقریبی از مشتق تصویر در جهت عمودی و افقی هستند. از تصویر

$$G = \sqrt{G_x^2 + G_y^2}$$

استفاده خواهیم کرد. برای مشاهده‌ی مثالی از عملگر این عملگر، به شکل (۱) توجه کنید.



شکل ۱: نمونه‌ای از اعمال عملگر Sobel

۲.۱ Kirsch Operator

در این روش، همانند روش Sobel عمل کرده اما از تعداد بیشتری کرنل استفاده خواهیم کرد. در واقع یک کرنل را در نظر گرفته و با چرخاندن آن به اندازه‌ی مضارب ۴۵ درجه، به یک کرنل به ازای هر یک از ۸ جهت قطب‌نما می‌رسیم. تصویر خروجی این الگوریتم برابر

$$h[n, m] = \max_{z \in \{1, 2, \dots, 8\}} (g^{(z)} * A)[n, m]$$

است که در آن

$$g^{(1)} = \begin{bmatrix} +5 & +5 & +5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, g^{(2)} = \begin{bmatrix} +5 & +5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, g^{(3)} = \begin{bmatrix} +5 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & -3 & -3 \end{bmatrix}, g^{(4)} = \begin{bmatrix} -3 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & +5 & -3 \end{bmatrix}, \dots$$

شکل (۲) عملکرد این روش را بر یک تصویر نمایش می‌دهد.



شکل ۲: نمونه‌ای از اعمال عملگر Kirsch

بعد از پیاده‌سازی هر دو الگوریتم، و با انتخاب تصاویر مناسب، عملکرد این دو الگوریتم را از نظر زمان اجرا و کیفیت مقایسه کنید. در انتخاب تصویرهای خود برای مقایسه این دو الگوریتم، چه ویژگی‌هایی را مد نظر داشتید؟

۲ الگوریتمی برای شمارش دایره در تصویر

الف) در این قسمت الگوریتمی ارایه کنید که تعداد اجسام گرد و در واقع دایره‌های موجود در یک تصویر را به شما بدهد (انتظار می‌رود الگوریتم مورد نظر را به صورت خلاصه در گزارش توضیح دهید). الگوریتم خود را بدون استفاده از دستورات پیش فرض موجود در MATLAB پیاده‌سازی کرده و آن را بر تصویر circles.jpg که در ضمیمه‌ی این تمرین قرار دارد، آزمایش کنید. دو تصویر دیگر نیز به دلخواه از اینترنت انتخاب نموده و نتیجه اعمال آن الگوریتم بر آن‌ها را در گزارش خود بیاورید.

راهنمایی: می‌توانید کمی در مورد Hough Transform مطالعه کنید. ممکن است در این روش برای رفع مشکل شمارش چند تایی دایره‌های زخمیم و یا تداخل دو دایره نیاز به خوشه‌بندی داشته باشید!

ب) اکنون یک بار دیگر به کمک دستورات موجود در متلب کدی بنویسید که تعداد دایره‌های موجود در یک تصویر را بشمارد. نتیجه حاصل از این روش را نیز در گزارش خود بیاورید.

۳ حذف نویز

لازم است تصاویری که در این بخش از آن‌ها استفاده می‌کنید تصاویر رنگی باشند و تصاویر سیاه و سفید قابل قبول نیستند!

الف) شناخت انواع نویز و همچنین عوامل ایجاد کننده آن‌ها و روش‌های حذف هر یک بسیار مهم هستند، به همین جهت در این قسمت از پروژه، به این مباحث می‌پردازیم.

در ابتدا در مورد هر یک از چهار نوع نویز زیر مطالعه کرده و خلاصه‌ای از مطالعات خود را در گزارش خود ذکر نمایید. (منبع ذکر شود و نیازی به توضیحات طولانی و با جزئیات نیست)

- (a) Salt & Pepper
- (b) Gaussian
- (c) Poisson
- (d) Speckle

به دلخواه خود، یک تصویر انتخاب کرده و هر بار به آن یکی از چهار نویز بالا را اضافه کنید. توجه داشته باشید که توان نویز آن قدر زیاد نباشد که تصویر قابل شناسایی نبوده و آنقدر نیز کم نباشد که نویز مشهود نباشد.

(ب) در مورد فیلتر میانه‌گیر Median Filter و فیلتر گوسی Gaussian Filter به دقت مطالعه کرده و در مورد هر یک به طور مختصر در گزارش کار خود توضیح دهید.

(ج) در این بخش قصد داریم فیلتر گوسی و میانه‌گیر را پیاده‌سازی کنیم و عملکرد هر یک را برای حذف انواع مختلف نویز با هم مقایسه کنیم.

۱. تابعی در متلب بنویسید که با گرفتن عکس، سایز کرنل و انحراف معیار، فیلتر گوسی یا Gaussian Filter را روی تصویر ورودی اعمال کند و در خروجی تصویر فیلترشده را به ما بدهد. لازم به ذکر است که:

در این قسمت اجازتی استفاده از توابع آماده متلب را ندارید.

راهنمایی: هنگام پیاده‌سازی ممکن است در لبه‌ها به مشکل برخوردید، اما می‌توانید با اضافه کردن صفر به کناره‌های تصویر مشکل خود را حل کنید!

۲. تابعی بنویسید که با دریافت سایز کرنل، فیلتر میانه‌گیر یا Median Filter را بر تصویر ورودی اعمال کند و تصویر فیلترشده را بازگرداند.

۳. در این بخش، دو تابعی که در بخش‌های قبل نوشته‌اید را بر عکس‌های آلوده به هر یک از چهار نوع نویز معرفی شده، اعمال کرده و تصویر خروجی هر یک از آن‌ها را بازگرداند. برای هر عکس، تلاش کنید بهترین پارامترهای فیلتر را انتخاب کنید. تصاویر آلوده به نویز را به همراه تصاویر حاصل از عبور آن‌ها از فیلترها در گزارش کار خود بیاورید.

۴. (امتیازی) با جست‌وجو در اینترنت سعی کنید که روش‌های دیگری برای فیلترکردن تصاویر پیدا کنید و در صورت امکان آن‌ها را پیاده‌سازی کنید و منطق ریاضی آن‌ها را نیز حتما در گزارش کار خود بیاورید.

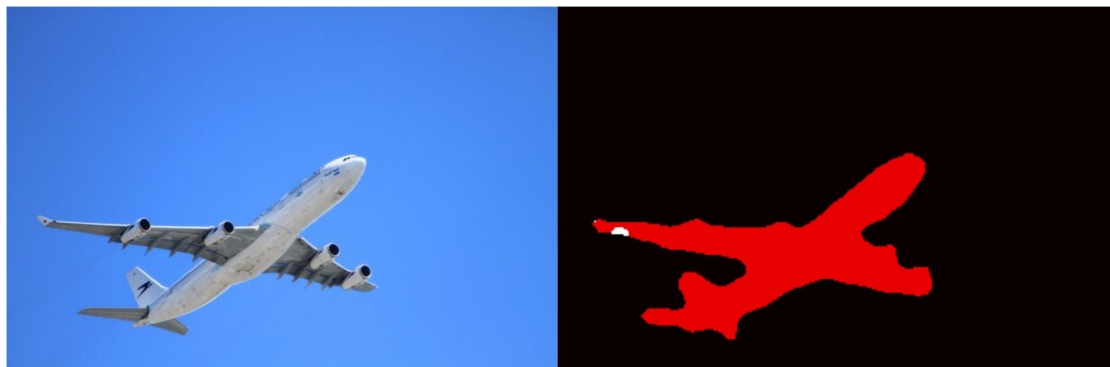
نمره اتخاذ شده از این بخش وابسته به تعداد روش‌ها و به علاوه سختی پیاده‌سازی آن‌هاست.

۴ Image Segmentation

توضیحات اولیه: در این پروژه قصد داریم به مسئله‌ی بخش‌بندی تصویر یا Image Segmentation بپردازیم که در حال حاضر از مسائل داغ حوزه‌ی بینایی ماشین است. در ابتدا یک روش برای این کار به صورت اجمالی معرفی می‌شود. شما باید روش مطرح شده را پیاده‌سازی کنید و همچنین در وهله بعدی از میان دو روش گفته شده، یک روش را انتخاب کرده و با جست‌وجو در اینترنت آن را توضیح داده و پیاده کنید. در تمام طول این پروژه توجه داشته باشید که لزومی ندارد الگوریتم‌های زیر را بر روی خود تصاویر داده‌شده اجرا کنید بلکه ممکن است بعد از یک پیش‌پردازش (و کارهایی مشابه بخش اولیه پروژه) بتوانید به نتایج بهتری برسید!

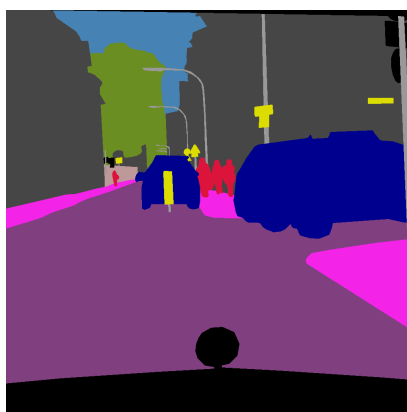
در بینایی ماشین، بخش‌بندی تصویر، به فرآیند قطعه‌بندی کردن یک تصویر دیجیتال به چند بخش (مجموعه‌ای از پیکسل‌ها) گفته می‌شود. هدف بخش‌بندی، ساده‌سازی یا تغییر در نمایش یک تصویر به چیزی است که هم معنی‌دارتر و هم برای آنالیز آسان‌تر است.

بخش‌بندی تصویر معمولاً برای پیدا کردن محل اشیا موردنظر و مرزها (خطوط، منحنی‌ها و غیره) در تصویر استفاده می‌شود. به عبارت دقیق‌تر، بخش‌بندی تصویر، به فرآیندی گفته می‌شود که در آن، به هر پیکسل، برچسبی اختصاص داده می‌شود، به طوری که پیکسل‌هایی با برچسب یکسان، ویژگی‌های مشابهی دارند. یک مثال از بخش‌بندی، فرآیند تشخیص خیابان، پیاده‌رو، تابلوهای راهنمایی رانندگی، عابر پیاده، خودرو و ... توسط یک ماشین خودران است. یک مثال ساده از بخش‌بندی را در شکل ۳ مشاهده می‌کنید.



شکل ۳: مثالی از بخش‌بندی یک تصویر

شکل ۴: مثالی از بخش‌بندی یک خیابان است که بر خلاف شکل مثال فوق، شامل بیش از دو بخش است.



شکل ۴: مثالی از بخش‌بندی یک تصویر پیچیده

در این تمرین ما تنها به بخش‌بندی تصویر به دو بخش می‌پردازیم.

کاری که از شما می‌خواهیم ساده است! یک تصویر آسمان و هواپیما به شما داده می‌شود که به صورت ذاتی دارای دو بخش است (مثل تصویر ۳) و از شما خواسته می‌شود آن تصویر را به دو بخش آسمان و هواپیما تقسیم کنید و خروجی‌ای مشابه شکل ۳ به ما بدهید.

۱.۴ روش اول: استفاده از K-Means

الگوریتم K-Means الگوریتمی معروف برای تقسیم داده‌هایی در فضای اقلیدسی به k خوشه است. این الگوریتم قصد دارد تا با گرفتن مجموعه‌ای از نقاط مانند $\{x_1, x_2, \dots, x_n\}$ ، که هر کدام از آنها، یک بردار هستند، آن‌ها را به k خوشه‌ی $S = \{S_1, S_2, \dots, S_k\}$ تقسیم کند به نحوی که عبارت زیر در آن کمینه شود:

$$\sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

که در آن، μ_i برابر میانگین نقاطی است که در دسته‌ی i قرار گرفته‌اند. در واقع در حال حل مسئله‌ی زیر هستیم:

$$S^* = \operatorname{argmin}_{S=\{S_1, S_2, \dots, S_m\}} \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

که S^* خوشه‌بندی بهینه‌ی مد نظر ماست.

این مسئله، یک مسئله‌ی بهینه‌سازی ترکیبیاتی سخت است و در واقع NP-Hard است و تا کنون هیچ الگوریتم بهینه‌ای برای حل آن یافت نشده است، بنابراین برای حل این مسئله به سمت الگوریتم‌های تقریبی و روش‌های سرانگشتی می‌رویم.

- به صورت شهودی، بیان کنید که چرا علاقه به کمینه کردن عبارت بالا داریم.
- با جست‌وجو، یک روش معروف برای به دست آوردن یک جواب معقول برای این مسئله را معرفی کنید. روش را به طور کامل توضیح دهید. توجه داشته باشید که نحوه‌ی انتخاب اولیه‌ی مرکز خوشه‌ها و همچنین شرط اتمام الگوریتم را نیز در گزارش خود ذکر کنید.
- بیان کنید که چگونه می‌توان از K-Means برای حل مسئله‌ی مطلوب ما استفاده کرد.
- با پیاده‌سازی الگوریتمی که یافته‌اید، تصویر را به دو بخش تقسیم کنید و چند نمونه از خروجی مطلوب را در گزارش کار خود بیاورید.

۲.۴ روش دوم: انتخابی!

از بین دو الگوریتم فورد-فالکرسون و الگوریتم Otsu یکی را به دلخواه انتخاب کرده و به طور دقیق آن را در گزارش کار خود توضیح دهید. سپس این الگوریتم را در متلب پیاده‌سازی کرده و چند نمونه از خروجی مطلوب را در گزارش کار خود بیاورید.

۳.۴ مقایسه‌ی روش‌ها

به طور مختصر، دو روش فوق را با هم مقایسه کنید و مزایا و معایب هر یک را ذکر کنید. توجه کنید که این بخش از اهمیت زیادی برخوردار است پس آن را با حداکثر دقت انجام دهید.