



دانشکده مهندسی کامپیوتر

تمرین سری ۱۰ بنیایی کامپیوتر

نام درس

مبانی بنیایی کامپیوتر

نام دانشجو

زهرا انوریان

نام استاد درس

دکتر محمدی

پاییز ۱۳۹۹

سوالات

۱. ✎ در کلاس روش‌های ناحیه‌بندی تصاویر را یاد گرفتید. یکی از کاربردهای ناحیه‌بندی تصویر، پیدا کردن شکل اشیا است که در سوال ۳ بررسی شده است. با تحقیق، یکی دیگر از کاربردهای عملی ناحیه‌بندی تصویر را پیدا کنید و توضیح دهید. (۱۰ نمره)

پاسخ: در پردازش تصویرهای دیجیتال و بینایی کامپیوتر، ناحیه‌بندی تصویر به تقسیم‌بندی تصویر دیجیتال به چندین ناحیه گفته می‌شود. هدف اصلی ناحیه‌بندی، ساده‌سازی و یا تغییر نمایش تصویر به چیزی است که معنادارتر و تجزیه و تحلیل آن ساده‌تر باشد. ناحیه‌بندی تصویر معمولاً برای تعیین مکان اشیا و کانتورها (مرزها) در تصویر می‌باشد. به طور دقیق‌تر، ناحیه‌بندی تصویر فرآیند اختصاص دادن لیبل به هر پیکسل در تصویر به گونه‌ای که پیکسل‌هایی با همان لیبل در تصویر دارای ویژگی خاصی می‌باشند. نتیجه‌ی ناحیه‌بندی مجموعه‌ای از نواحی‌ایست که به طور کلی تمام تصویر را می‌پوشانند و یا مجموعه‌ای از کانتورهای استخراج شده از تصویر است. زمانی که ناحیه‌بندی را روی یک سری تصویر (به طور مثال در تصاویر پزشکی) اعمال می‌کنیم، از کانتورهای بدست آمده‌ی آن‌ها می‌توان برای ایجاد بازسازی‌های ۳ بعدی با کمک الگوریتم‌های درونی‌یابی مانند marching cubes استفاده کرد و یا در تشخیص پلاک‌های ماشین‌ها و یا در تشخیص صورت و همانطور که گفته شد در عکس‌های پزشکی مانند تشخیص تومورهای موجود در بدن کاربرد دارد.

لینک:

https://en.wikipedia.org/wiki/Image_segmentation

۲. عملگر سایش و افزایش را با توجه به عنصر ساختاری زیر بر روی تصویر زیر انجام دهید. (۱۵ نمره)

	۱	۰	۱	۰	۰	۱	۱	۱		<table><tr><td>۰</td><td>۱</td><td>۰</td></tr><tr><td>۱</td><td>*۱</td><td>۱</td></tr><tr><td>۰</td><td>۱</td><td>۰</td></tr></table>	۰	۱	۰	۱	*۱	۱	۰	۱	۰
۰	۱	۰																	
۱	*۱	۱																	
۰	۱	۰																	
	۱	۰	۰	۱	۱	۰	۰	۱											
	۰	۰	۰	۰	۰	۰	۱	۱											
	۰	۱	۰	۰	۰	۱	۱	۱											
	۰	۰	۱	۰	۰	۰	۱	۰											
	۰	۱	۱	۰	۱	۰	۰	۰											
	۰	۰	۰	۰	۱	۰	۰	۰											
	۰	۰	۰	۱	۰	۱	۰	۰											
تصویر										عنصر ساختاری									

پاسخ:

عملگر گسترش: برای انجام عملگر گسترش با توجه به فرمول زیر عمل می‌کنیم. ابتدا از zero- padding استفاده می‌کنیم و سپس عملگر گسترش را به تصویر اعمال می‌کنیم و در نهایت تصویر بدست آمده را بدون padding نمایش می‌دهیم.

$$A \oplus B = \{(B)_Z \cap A \neq \emptyset\}$$

۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۰	۱	۰	۱	۰	۰	۱	۱	۱	۰
۰	۱	۰	۰	۱	۱	۰	۰	۱	۰
۰	۰	۰	۰	۰	۰	۰	۱	۱	۰
۰	۰	۱	۰	۰	۰	۱	۱	۱	۰
۰	۰	۰	۱	۰	۰	۰	۱	۰	۰
۰	۰	۱	۱	۰	۱	۰	۰	۰	۰
۰	۰	۰	۰	۰	۱	۰	۰	۰	۰
۰	۰	۰	۰	۱	۰	۱	۰	۰	۰
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰

➔

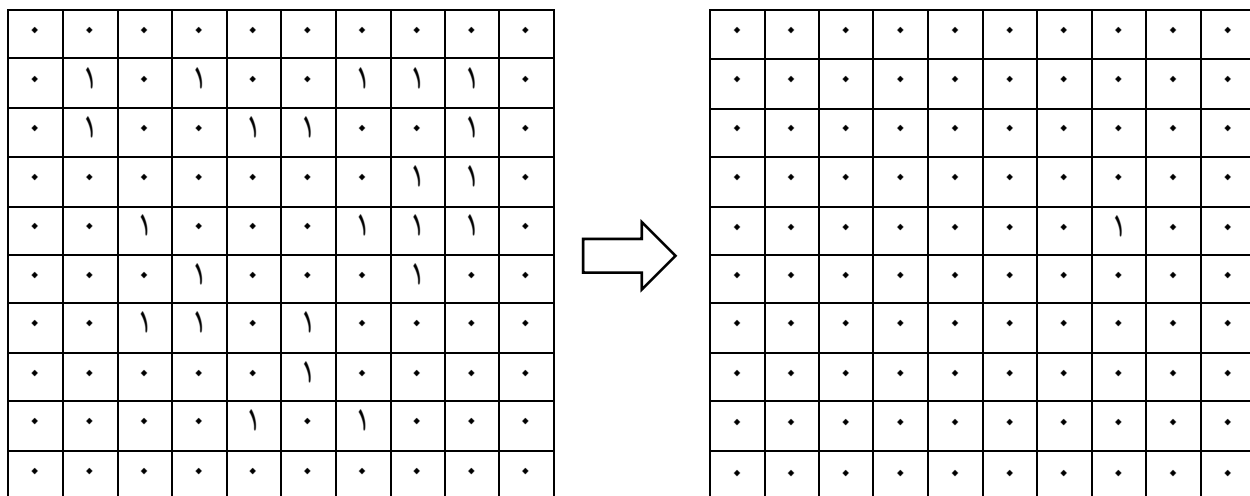
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۰	۱	۱	۱	۱	۱	۱	۱	۱	۰
۰	۱	۱	۱	۱	۱	۱	۱	۱	۰
۰	۱	۱	۰	۱	۱	۱	۱	۱	۰
۰	۱	۱	۱	۱	۰	۱	۱	۱	۰
۰	۱	۱	۱	۱	۱	۱	۱	۱	۰
۰	۰	۱	۱	۱	۱	۱	۱	۱	۰
۰	۱	۱	۱	۱	۱	۱	۱	۱	۰
۰	۰	۱	۱	۱	۱	۱	۱	۱	۰
۰	۰	۰	۱	۱	۱	۱	۱	۱	۰
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰

نتیجه نهایی:
تصویر گسترش یافته

۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۰	۱	۱	۱	۱	۱
۱	۱	۱	۰	۱	۱	۱	۱
۰	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۰
۰	۱	۱	۱	۱	۱	۰	۰
۰	۰	۱	۱	۱	۱	۱	۰

عملگر سایش: برای انجام عملگر سایش باتوجه به فرمول زیر عمل می‌کنیم. ابتدا از zero-padding استفاده می‌کنیم و سپس عملگر سایش را به تصویر اعمال می‌کنیم و در نهایت تصویر بدست آمده را بدون padding نمایش می‌دهیم.

$$A \ominus B = \{z | (B)_z \subseteq A\}$$



1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

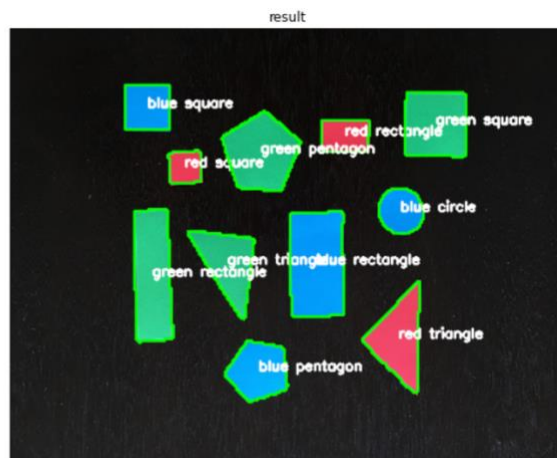
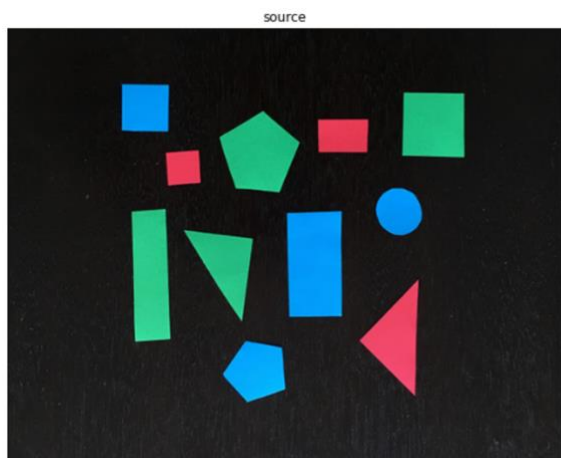
نتیجه نهایی:
تصویر سایش یافته

۳. با استفاده از لینک‌های زیر برنامه‌ای بنویسید که شکل و رنگ اشیاء ورودی را پیدا کند. (۳۵ نمره)

[لینک ۱](#) [لینک ۲](#) [لینک ۳](#)

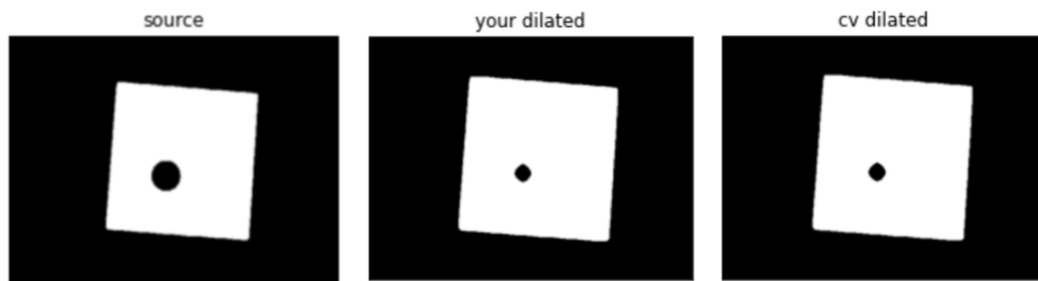
پاسخ: پیاده‌سازی این برنامه دارای سه مرحله است. ابتدا باید مرکز اشکال را بدست آوریم سپس اشکال مختلف را تشخیص بدهیم و سپس هر کدام را برحسب رنگ و نوعشان لیبل بزنیم.

حال برای بدست آوردن مرکز اشکال ابتدا شکل را هموار می‌کنیم که اگر نویزی دارد، برطرف شود و سپس آن را به طیف خاکستری می‌بریم و با استفاده از آستانه‌گذاری تصویر را باینری کرده و با استفاده از تابع `cv2.findContours` کانتور و یا مرزهای موجود در تصویر را بدست می‌آوریم و سپس برای بدست آوردن مراکز این کانتورها از `moment` استفاده می‌کنیم. اطلاعاتی که می‌توان از آن بدست آورد ناحیه‌ی شیء، مرکز ناحیه، جهت و... است. حال برای مرحله ۲ یعنی بدست آوردن شکل نواحی موجود در تصویر، از تابع `cv2.approxPolyDP` استفاده می‌کنیم که در واقع الگوریتمی برای کاهش تعداد نقاط منحنی است و برای همین اصطلاح تقریب به آن داده شده است و خروجی آن نقاطی است که زیرمجموعه‌ی نقاط منحنی اصلی است و سپس تعداد منحنی‌های هر ناحیه را با مقادیر ۳، ۴، ۵ مقایسه می‌کنیم که شکل ناحیه را بدست آوریم (به طور مثال اگر ۳ منحنی داشتیم آن ناحیه یک مثلث است). حال برای مرحله ۳ یعنی بدست آوردن رنگ ناحیه و لیبل زدن آن، ابتدا تصویر را به سطوح رنگی Lab می‌بریم که درواقع L همان میزان روشنایی از مشکی (۰) تا سفید (۱۰۰) و a از قرمز (-) تا سبز (+) و b از آبی (-) تا زرد (+) را نشان می‌دهد. حال ابتدا یک ماسک برای کانتورها بدست می‌آوریم و سپس میانگین مقدار Lab را برای منطقه‌ی ماسک شده محاسبه می‌کنیم و در نهایت فاصله‌ی اقلیدسی میان میانگین حساب شده و رنگ‌های بدست آمده از Lab می‌گیریم و آن رنگی که کمترین فاصله را داشته باشد به عنوان رنگ آن ناحیه انتخاب می‌کنیم و به شکل لیبل می‌زنیم.



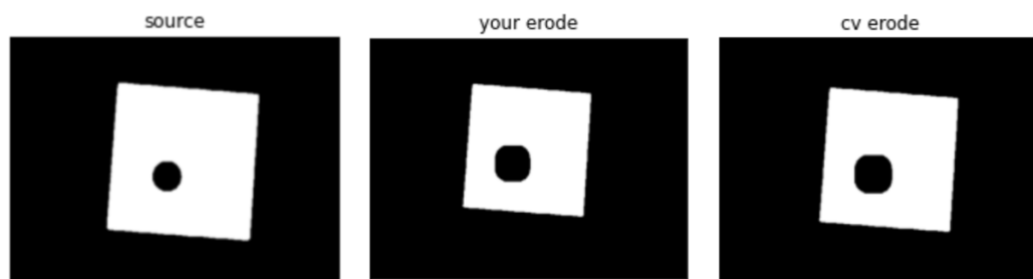
۴. الف) بدون استفاده از توابع OpenCV برنامه‌ای بنویسید که عملگر گسترش را با استفاده از عنصر ساختاری مربعی به طول دلخواه انجام دهد و آن را بر روی تصویر q4.jpg اعمال کنید. خروجی را بررسی کرده و از لحاظ سرعت با کد معادل OpenCV مقایسه کنید. (۲۰ نمره)

پاسخ: برای پیاده‌سازی عملگر گسترش، ابتدا یک عنصر ساختاری 51x51 در نظر می‌گیریم و anchor آن را در وسط آن در نظر می‌گیریم سپس بر روی تصویر می‌لغزانیم. به طوری که هر پیکسل بر روی anchor عنصر ساختاری قرار گیرد و چک می‌کنیم که اگر مقدار آن پیکسل برابر ۲۵۵ بود آن پیکسل از تصویر را به اندازه‌ی عنصر ساختاری ۲۵۵ کنیم. در واقع آن را گسترش می‌دهیم که این عمل باعث کوچک شدن دایره‌ی مشکی داخل مربع سفید و بزرگ شدن مربع سفید می‌شود. زمان = ۵.۵۰۱۷۱۲ ثانیه
لازم به ذکر است که عملگر گسترش را می‌توان با استفاده از عملگر کانولوشن نیز پیاده‌سازی کرد زیرا همان کار را انجام می‌دهد. ابتدا کرنل (عنصر ساختاری) را ۱۸۰ درجه می‌چرخاند و بعد در تصویر ضرب می‌کند (اشتراک می‌گیرد). زمان = ۰.۰۵۵۶۶۲ ثانیه



ب) بدون استفاده از توابع OpenCV برنامه‌ای بنویسید که عملگر کاهش را با استفاده از عنصر ساختاری مربعی به طول دلخواه انجام دهد و آن را بر روی تصویر q4.jpg اعمال کنید. خروجی را بررسی کرده و از لحاظ سرعت با کد معادل OpenCV مقایسه کنید. (۲۰ نمره)

پاسخ: برای پیاده‌سازی عملگر سایش، ابتدا یک عنصر ساختاری 51x51 در نظر می‌گیریم و anchor آن را در وسط آن در نظر می‌گیریم سپس بر روی تصویر می‌لغزانیم. به طوری که هر پیکسل بر روی anchor عنصر ساختاری قرار گیرد و چک می‌کنیم که اگر عنصر ساختاری به کلی در آن پیکسل از تصویر برابر بود، مقدار آن پیکسل از تصویر را ۲۵۵ و در غیر این صورت ۰ در نظر می‌گیریم. در واقع آن را سایش می‌دهیم که این عمل باعث بزرگ شدن دایره‌ی مشکی داخل مربع سفید و کوچک شدن مربع سفید می‌شود.



و همچنین با مقایسه‌ی مدت زمان اجرای توابع، متوجه می‌شویم که تابع opencv برای هر دو عملگر سریعتر از تابع پیاده‌سازی شده‌ی بنده می‌باشد.

time for your_dilate: 0.055662 s	time for your_dilate: 5.501712 s
time for cv_dilate: 0.003071 s	time for cv_dilate: 0.002555 s
time for your_erode: 15.566228 s	time for your_erode: 17.187223 s
time for cv_erode: 0.002764 s	time for cv_erode: 0.002511 s