



دانشکده مهندسی کامپیوتر

تمرین سری ۱۲ بینایی کامپیوتر

نام درس

مبانی بینایی کامپیوتر

نام دانشجو

زهرا انوریان

نام استاد درس

دکتر محمدی

پاییز ۱۳۹۹

سوالات

۱. برای موارد زیر فشردگی، صلب بودن و کشیدگی را محاسبه کنید (هر بخش ۱۰ نمره).

الف) لوزی با اندازه قطرهای a و $a\sqrt{2}$.

$$Compactness = \frac{4\pi Area}{Perimeter^2} = \frac{4\pi * a^2}{(4 * \frac{\sqrt{5}a}{2})^2} = \frac{\pi}{5}$$

$$Solidity = \frac{Area}{ConvexArea} = \frac{a^2}{a^2} = 1$$

$$Eccentricity = \sqrt{1 - \left(\frac{MinorAxisLength}{MajorAxisLength}\right)^2} = \sqrt{1 - \left(\frac{a}{2a}\right)^2} = \frac{\sqrt{3}}{2}$$

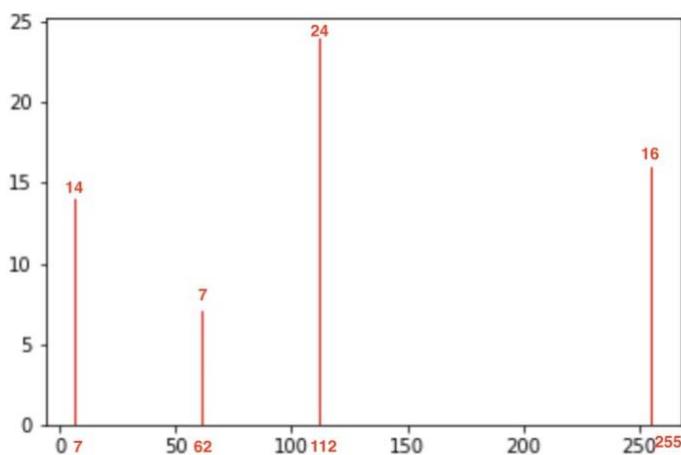
ب) مستطیل با اندازه اضلاع a و $a\sqrt{2}$.

$$Compactness = \frac{4\pi Area}{Perimeter^2} = \frac{4\pi * 2a^2}{36a^2} = \frac{2\pi}{9}$$

$$Solidity = \frac{Area}{ConvexArea} = \frac{2a^2}{2a^2} = 1$$

$$Eccentricity = \sqrt{1 - \left(\frac{MinorAxisLength}{MajorAxisLength}\right)^2} = \sqrt{1 - \left(\frac{a}{2a}\right)^2} = \frac{\sqrt{3}}{2}$$

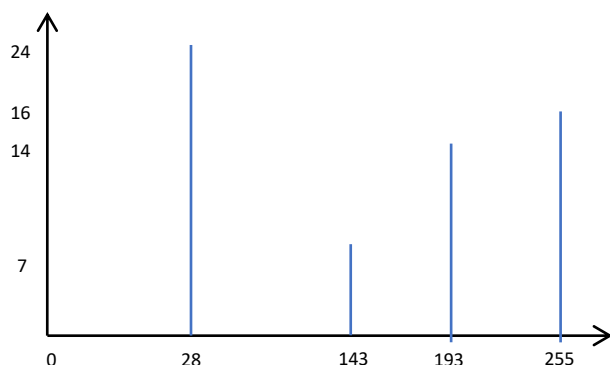
۲. هیستوگرام کد LBP_8^1 یک تصویر در نمودار زیر آورده شده است. اگر تصویر را ۹۰ درجه ساعتگرد بچرخانیم و شدت روشنایی را دو برابر کنیم، هیستوگرام نهایی را بدست آورید (۱۰ نمره).



پاسخ: تغییر شدت روشنایی که تاثیری در هیستوگرام LBP_8^1 ندارد اما با چرخش ۹۰ درجه‌ای ساعتگرد تصویر باعث می‌شود که کدهای باینری حاصل از LBP_8^1 دو بیت به سمت راست شیفت یابد پس داریم:

$7 = 00000111 \rightarrow \text{shifted to right: } 11000001 = 193$
 $62 = 00111110 \rightarrow \text{shifted to right: } 10001111 = 143$
 $112 = 011100000 \rightarrow \text{shifted to right: } 00011100 = 28$
 $255 = 11111111 \rightarrow \text{shifted to right: } 11111111 = 255$

هیستوگرام تغییر یافته:

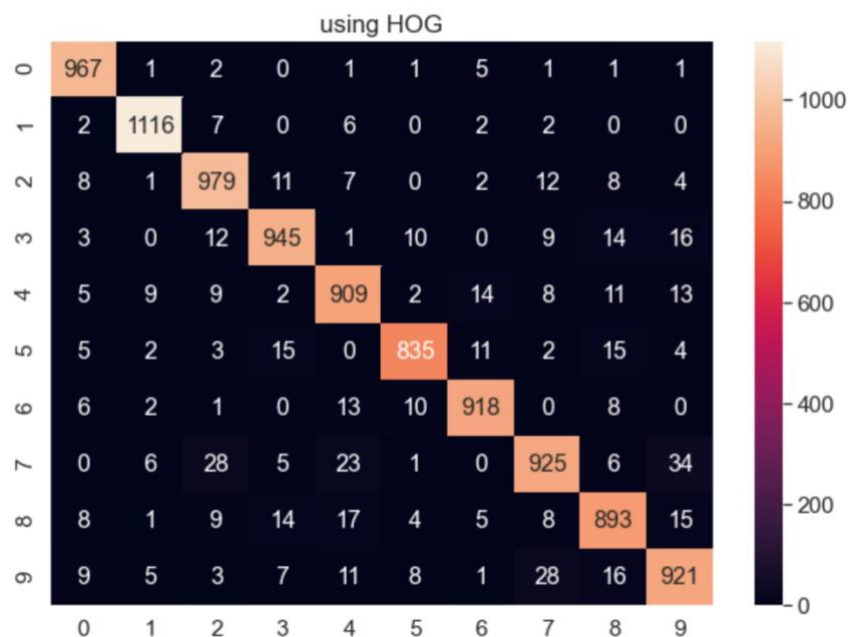


۳. مجموعه داده MNIST به فایل تمرین پیوست شده است. با توجه به موارد زیر در هر قسمت طبقه‌بندی خواسته شده را انجام دهید. هر قسمت از دو فاز آموزش و ارزیابی تشکیل شده است. در فاز آموزش، از داده آموزش برای آموزش دادن طبقه‌بند استفاده کنید. در فاز تست از داده تست استفاده کرده و نتیجه طبقه‌بندی را با برچسب‌های موجود مقایسه کرده و طبقه‌بند را ارزیابی کنید. برای این ارزیابی از معیارهایی مانند دقت و ماتریس درهم‌ریختگی (confusion) استفاده کنید (هر بخش ۱۰ نمره).

الف) با استفاده از HOG از تصاویر ویژگی استخراج کنید و با طبقه‌بند SVM تصاویر را طبقه‌بندی کنید (برای راهنمایی به [لینک ۱](#) و [لینک ۲](#) مراجعه کنید).

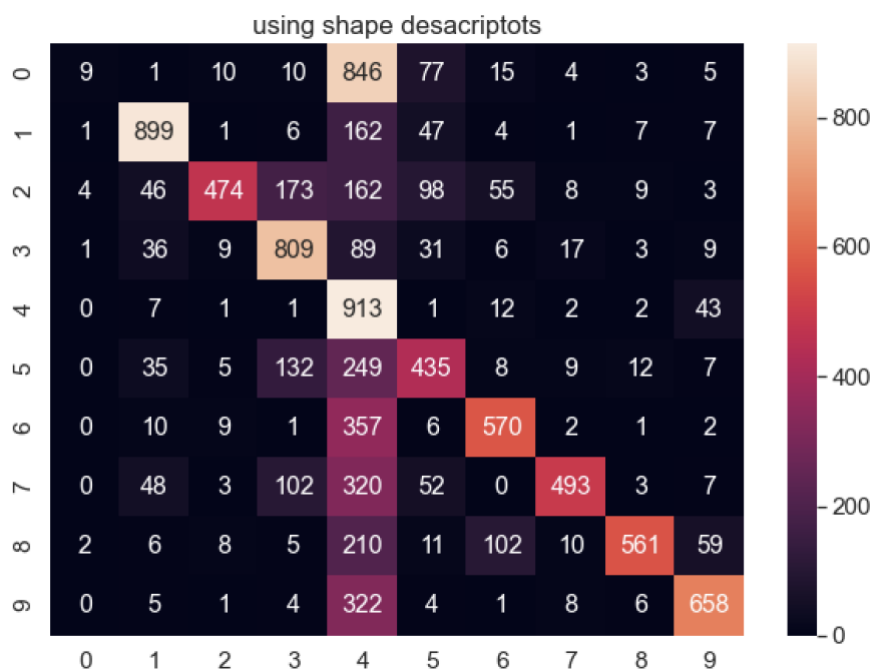
پاسخ: ابتدا با استفاده از تابع `hog()`، برای هر تصویر فیچرهایش را استخراج می‌کنیم و سپس با استفاده از طبقه‌بند SVM فیچرهای استخراج شده را با لیبل‌های داده شده `fit` می‌کنیم. حال برای تست آن باید فیچرهای `hog` تصاویر تست را نیز استخراج کنیم و سپس به کمک تابع `predict()` طبقه‌بند SVM لیبل‌های پیش‌بینی شده را بدست می‌آوریم و در نهایت به دقت ۹۴.۰۸ می‌رسیم.

Accuracy for using HOG = 94.08



ب) با استفاده از توصیفگرهای شکل و با طبقه‌بند SVM تصاویر را طبقه‌بندی کنید.

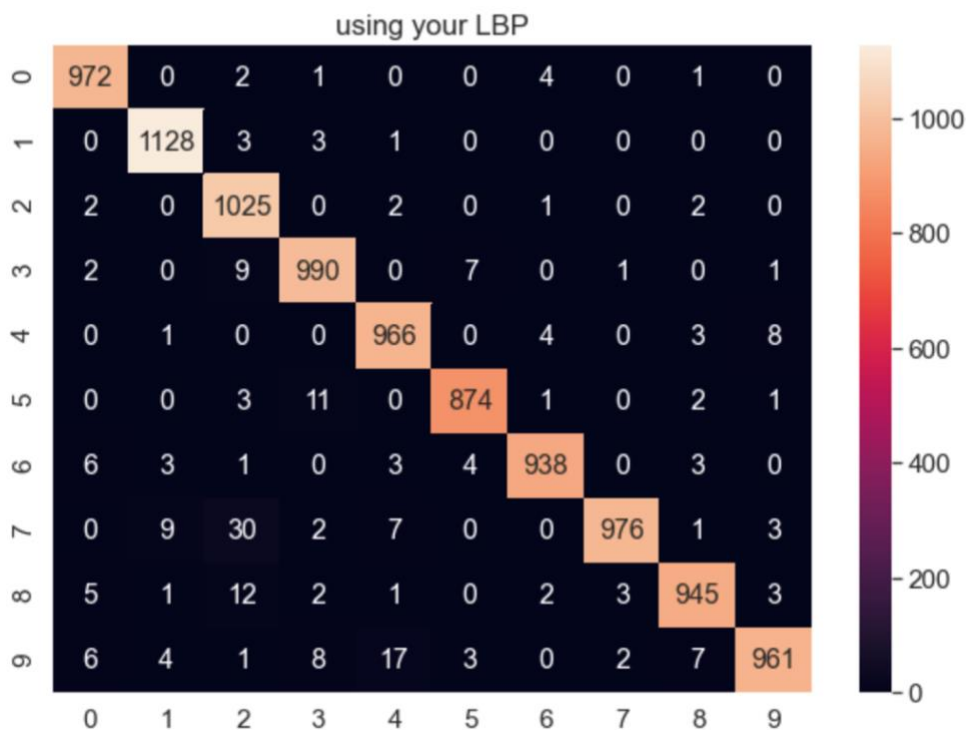
پاسخ: ابتدا برای تصاویر ورودی ویژگی moments را با استفاده از تابع `moments()` بدست می‌آوریم و سپس آن‌ها را به عنوان ورودی تابع `fit()` طبقه‌بند می‌دهیم و برای تصاویر تست نیز همین کار را انجام می‌دهیم و ویژگی‌های بدست آمده را به عنوان ورودی به تابع `predict()` می‌دهیم و لیبل‌های پیش‌بینی شده را بدست می‌آوریم و در نهایت به درصد دقت ۵۸.۸ می‌رسیم.



ج) توصیفگر بافت LBP_8^1 را پیاده‌سازی کنید (از توابع موجود استفاده نکنید) و با استفاده از این توصیفگر و با استفاده از طبقه‌بند SVM تصاویر را طبقه‌بندی کنید.

پاسخ: برای پیاده‌سازی توصیفگر بافت LBP_8^1 ابتدا به اندازه یک واحد به تصویر border می‌دهیم سپس یک پنجره به ابعاد 3×3 روی پیکسل‌های تصویر می‌لغزانیم و با پیکسل مورد نظر چک می‌کنیم که اگر بزرگتر مساوی آن بود مقدار ۱ و در غیر این صورت مقدار صفر را جایگزین کند سپس به صورت ساعتگرد ارقام بدست آمده از پنجره را در کنار هم قرار می‌دهیم که تولید یک عدد باینری می‌کنند. حال معادل آن در مبنای ۲ را بدست می‌آوریم و جای پیکسل مورد نظر در آرایه‌ی خروجی قرار می‌دهیم. حال برای بدست آوردن فیچرهای بافتی تصاویر ورودی ابتدا تصویر را به بخش‌های 7×7 تقسیم می‌کنیم و LBP_8^1 را برای هر بخش محاسبه می‌کنیم و سپس هیستوگرام آن را بدست می‌آوریم و در نهایت هیستوگرام‌های بخش‌های تصویر را در کنار هم قرار داده و به عنوان ورودی تابع $fit()$ طبقه‌بند SVM می‌دهیم و برای تصاویر تست نیز ابتدا فیچرهای بافتی آن را بدست آورده و سپس به تابع $predict()$ طبقه‌بند SVM می‌دهیم و لیبل‌های پیش‌بینی شده را بدست می‌آوریم و در نهایت به دقت ۹۷.۷۵ می‌رسیم.

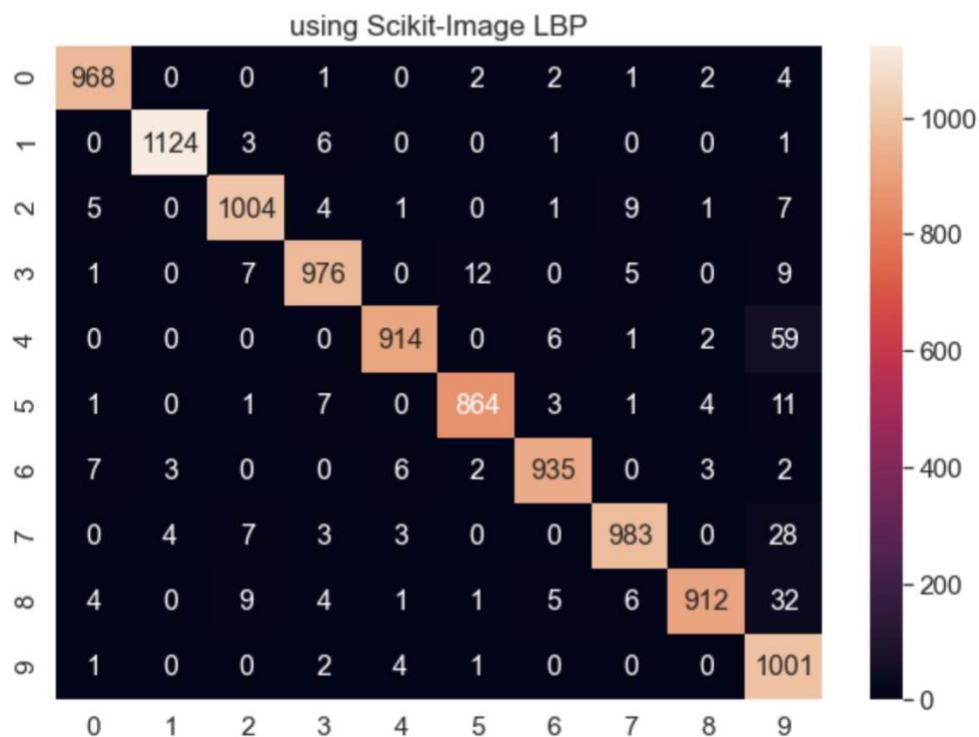
time for your LBP: 1315.770961 s
Accuracy for using your LBP = 97.75



د) به کمک تابع `skimage.feature.local_binary_pattern`، LBP_8^1 را استخراج کرده و با طبقه‌بند SVM تصاویر را طبقه‌بندی کنید. نتیجه و زمان اجرا این کد موجود را با کدی که در قسمت قبل توسعه دادید، مقایسه کنید (برای راهنمایی به [لینک](#) مراجعه کنید) (برای استفاده از طبقه‌بند گفته شده، از لینک [SVM](#) می‌توانید استفاده کنید).

پاسخ: با استفاده از تابع `local_binary_pattern()` ابتدا مقدار LBP تصویر را حساب کرده سپس با استفاده از تابع `histogram()` تصاویر را بدست می‌آوریم و در نهایت آن را به عنوان ورودی به تابع `fit()` طبقه‌بند SVM می‌دهیم. حال برای تست نیز ابتدا باید برای تصاویر تست نیز مقدار LBP را بدست آورده و هیستوگرام آن‌ها را به عنوان ورودی تابع `predict()` می‌دهیم و لیبل‌های پیش‌بینی شده را بدست می‌آوریم و در نهایت به درصد دقت ۹۶.۸۱ می‌رسیم و همچنین با توجه به زمان محاسبه شده، الگوریتم `local_binary_pattern()` سریع‌تر از الگوریتم پیاده‌سازی شده عمل می‌کند.

```
time for Scikit-Image LBP: 399.871784 s
Accuracy for using Scikit-Image LBP = 96.81
```



۴. با استفاده از ویژگی‌های هندسی (مثلاً استفاده از کانتور) یک طبقه‌بند برای تصاویر موجود در پوشه `images` بنویسید که دو کلاس داده را دسته‌بندی کند. در این سوال با استفاده از `OpenCV` و کانتورها ویژگی هندسی مناسبی برای جداسازی این دو دسته از یکدیگر پیدا کنید و طبقه‌بند را پیاده‌سازی کنید (ویژگی‌های مورد استفاده باید نسبت به بزرگنمایی تصویر ایمن باشد و از طبقه‌بند آماده استفاده نکنید)

(برای راهنمایی بیشتر به دو لینک زیر مراجعه کنید) (۳۰ نمره).

[لینک ۱](#) [لینک ۲](#)

پاسخ: برای بدست آوردن کانتور مناسب ابتدا باید تصویر را هموار کنیم و سپس HSV تصویر را محاسبه می‌کنیم و روی آن مناطق غیر سبز رنگ را حذف می‌کنیم سپس تصویر بدست آمده را خاکستری کرده و آن را باینری می‌کنیم و با استفاده از تابع `findContours()`، کانتورهای تصویر را بدست می‌آوریم و برای آن‌ها ویژگی گریز از مرکز را محاسبه می‌کنیم با گذاشتن شرطی روی مقادیر گریز از مرکز (بیشتر از ۰.۴ را در دسته‌ی سیب بگذارد)، آن‌ها را طبقه‌بندی می‌کنیم.

e: 0.7716147127062696

e: 0.776727742505656

e: 0.7354417923030598

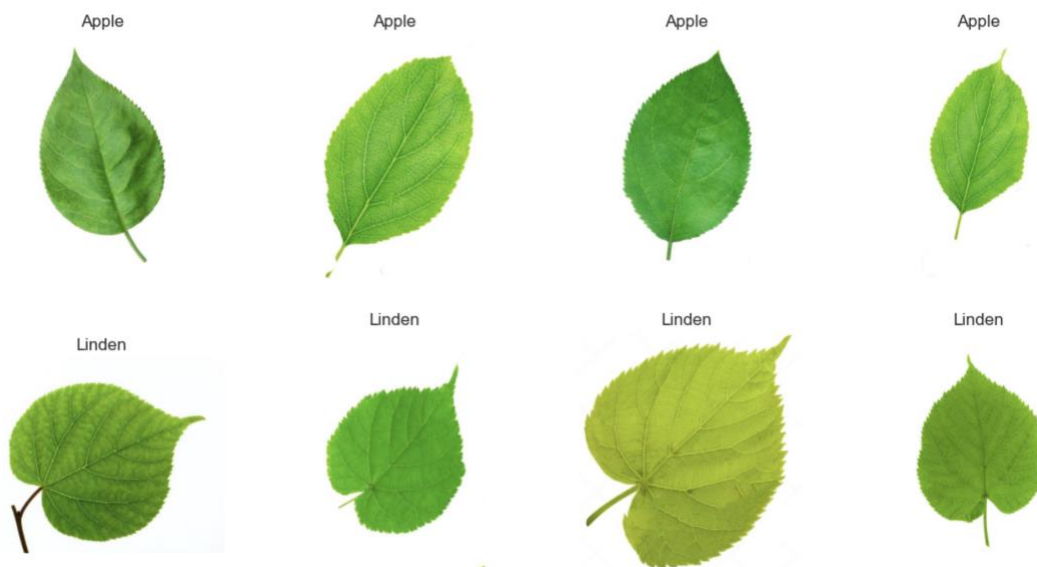
e: 0.8197018213072973

e: 0.5034133685437017

e: 0.5890728898833614

e: 0.6995199098239013

e: 0.5123522678318825



• موفق باشید.