

مبانی بینایی کامپیوتر

تمرین دهم

نیکی نزاکتی

98522094

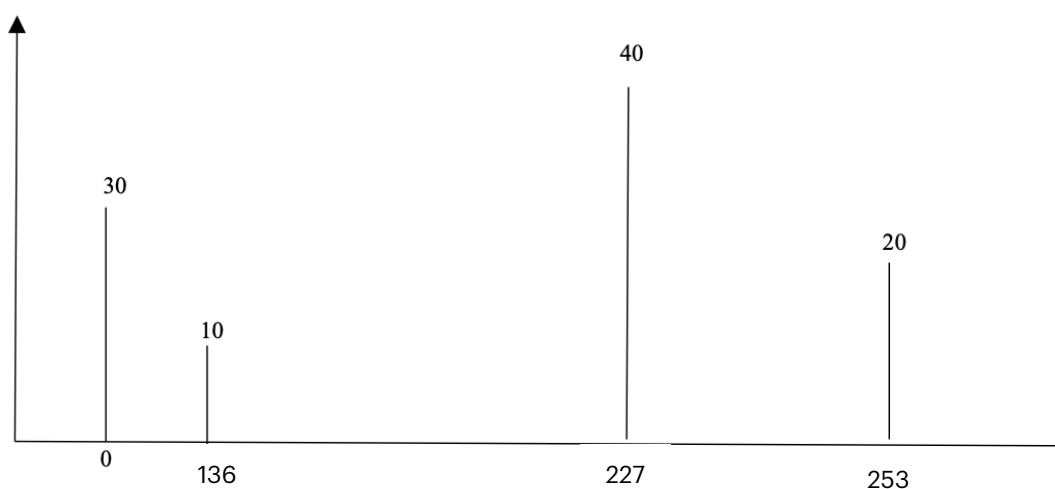
1.

نصف شدن روشنایی تصویر بر LBP بی تاثیر است چون در یک همسایگی، روشنایی همه پیکسل ها با نسبت یکسان تغییر کرده و قیاس ها یکسان می مانند. در نتیجه در تصویر اصلی LBP ها همان مقدار را خواهند داشت.

با چرخش تصویر به اندازه 270- درجه، کد های LBP به صورت زیر تغییر می کنند.

00000000 => 00000000	,	00100010 => 10001000
10001111 => 11100011	,	11110111 => 11111101

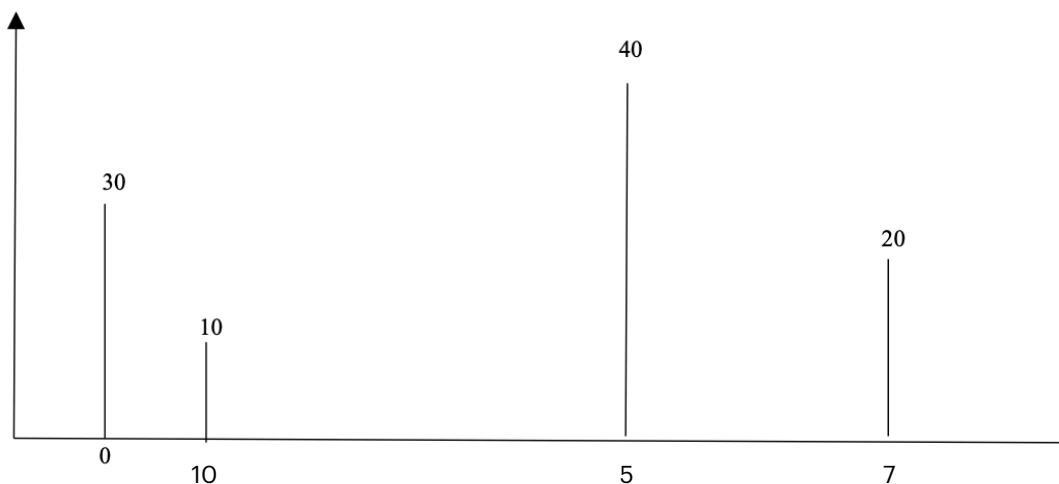
بنابراین هیستوگرام تصویر اصلی به صورت زیر خواهد بود:



در هیستوگرام یکنواخت و مستقل از چرخش ۱۰ کد داریم که نشان دهنده یکنواخت نبودن، و در موارد یکنواخت، نشان دهنده تعداد یک های موجود است.

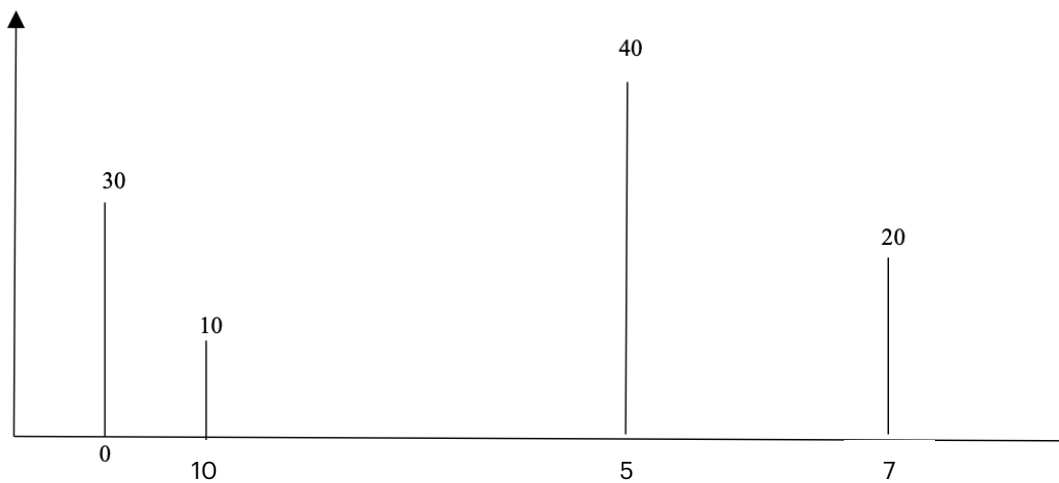
در این صورت در تصویر چرخش یافته هیستوگرام به صورت زیر خواهد بود:

00000000 => 00000000	,	00100010 => 000001010
10001111 => 00000101	,	11110111 => 00000111



به همین ترتیب هیستوگرام تصویر اصلی به صورت زیر خواهد بود:

00000000 => 00000000	,	10001000 => 000001010
11100011 => 00000101	,	11111101 => 00000111



همانطور که مشاهده می شود هیستوگرام LBP یکنواخت و مستقل از چرخش دو تصویر یکسان است که این به دلیل ماهیت LBP یکنواخت و مستقل از چرخش است که در برابر چرخش تصویر بی تغییر بوده است.

2.

الف) برای محاسبه توابع مربوط به ویژگی فشردگی، گریز از مرکز و چگالی، ابتدا تصویر را به کمک `cv2.cvtColor` به تصویر یک کاناله تبدیل کرده، با `cv2.adaptiveThreshold` باینری، و در نهایت توسط `cv2.findContours` کانتورهای تصویر را بدست میاوریم. همچنین به محیط که توسط `cv2.contourArea` و مساحت که توسط `cv2.arcLength` محاسبه می شود داریم. همچنین باید چک کنیم که کانتور های ما تعداد ۵ نقطه یا بیشتر داشته باشند تا محیط صفر نشود. برای محاسبه فشردگی از فرمول زیر استفاده می کنیم.

$$\text{compactness_score} = (4 * \text{math.pi} * \text{area}) / (\text{perimeter} ** 2)$$

برای محاسبه گریز از مرکز به Major Axis Length و minor axis length هم نیاز داریم که توسط `cv2.fitEllipse` محاسبه می شوند و طبق فرمول فشردگی داریم

$$\text{eccentricity_score} = \text{math.sqrt}(1 - (\text{MA}/\text{ma}) ** 2)$$

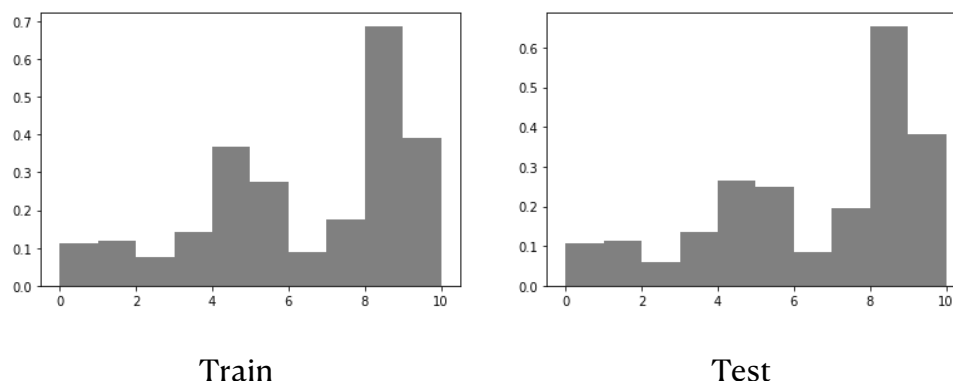
برای محاسبه چگالی از `cv2.convexHull` استفاده کرده و طبق فرمول داریم

$$\text{solidity_score} = \text{float}(\text{area})/\text{hull_area}$$

ب) برای محاسبه LBP ابتدا تصویر را یک کاناله کرده و از `local_binary_pattern` استفاده میکنیم.

$$\text{lbp} = \text{local_binary_pattern}(\text{gray}, \text{numPoints}, \text{radius}, \text{'uniform'})$$

هیستوگرام داده آموزش و تست به صورت زیر است:

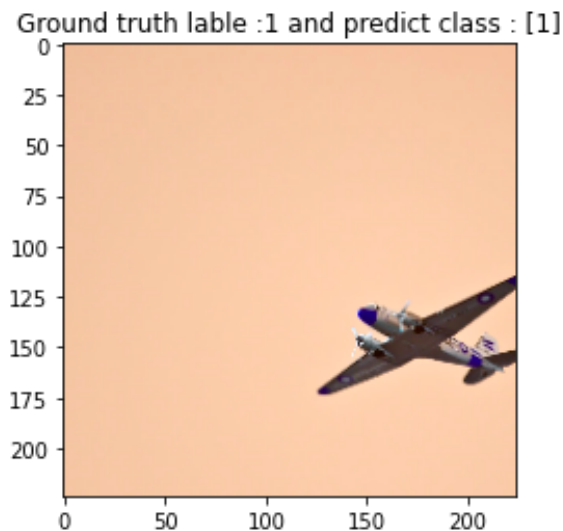


د) ماتریس خروجی تابع `get_featureMatrix` را مانند خواسته سوال از یک آرایه دو بعدی میسازیم و به همراه داده ها به مدل ورودی میدهیم.

ح) برای سنجش عملکرد مدل روی تصاویر تست، ماتریس فیچرهای تصاویر تست را بدست آورده و به مدل می دهیم.

```
test_features = get_featureMatrix(x_test)
y_pred = clf.predict(test_features)
```

خ) خروجی حاصل از عملکرد دسته بند بر روی یکی از تصاویر تست به صورت زیر است:



Resources:

https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_local_binary_pattern.html
<https://scikit-learn.org/stable/modules/svm.html>
<https://docs.opencv.org/>
<https://towardsdatascience.com/>