

## HW10

### بابک بهکام کیا

سوال 2)

طبق راهنمایی که در داک سوالات قرار داده شده پیش می رویم.

ابتدا تابع compactness را کامل می کنیم. در مرحله اول عکس را یک کاناله می کنیم. سپس با استفاده از متود adaptive threshold تصویر را باینری می کنیم. در نهایت contour های موجود در عکس را استخراج می کنیم. بدلیل آنکه برای محاسبه area به حداقل 5 contour نیاز داریم، با یک for دنبال این مقدار می گردیم و اگر نتوانستیم پیدا بکنیم، عدد 0 را برمیگردانیم. در نهایت compactness score بر اساس فرمولی که در اسلاید ها موجود است، محاسبه می شود.

```
1 def compactness(image):
2     # cv2_imshow(image)
3     imCopy = image.copy()
4     img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
5     # cv2_imshow(img)
6     thresh = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 5, 6)
7     # cv2_imshow(thresh)
8     contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
9
10    for i in range(len(contours)):
11        if len(contours[i]) >= 5:
12            cnt = contours[i]
13            break
14    return 0
15
16    area = cv2.contourArea(cnt)
17    perimeter = cv2.arcLength(cnt, True)
18
19    if perimeter == 0:
20        return 0
21
22    compactness_score = (4 * math.pi) * area / perimeter ** 2
23    return compactness_score
```

مثالی از خروجی



برای دو تابع eccentricity و solidity هم همین کار را می کنیم.

```
1 def solidity(image):
2     # cv2.imshow(image)
3     imgCopy = image.copy()
4     img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
5     # cv2.imshow(img)
6     thresh = cv2.adaptiveThreshold(img,200,cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY,5,6)
7     # cv2.imshow(thresh)
8     contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
9
10    for i in range(len(contours)):
11        if len(contours[i]) >= 5:
12            cnt = contours[i]
13            break
14        return 0
15
16    area = cv2.contourArea(cnt)
17
18    hull = cv2.convexHull(cnt)
19    hull_area = cv2.contourArea(hull)
20
21    solidity_score = area / hull_area
22    return solidity_score
23
```

```
1 def eccentricity(image):
2     # cv2.imshow(image)
3     imgCopy = image.copy()
4     img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
5     # cv2.imshow(img)
6     thresh = cv2.adaptiveThreshold(img,200,cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY,5,6)
7     # cv2.imshow(thresh)
8     contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
9
10    for i in range(len(contours)):
11        if len(contours[i]) >= 5:
12            cnt = contours[i]
13            break
14        return 0
15
16    area = cv2.contourArea(cnt)
17
18    (x,y),(MA,ma),angle=cv2.fitEllipse(cnt)
19
20    eccentricity_score = math.sqrt(1 - (MA/ma) ** 2)
21    return eccentricity_score
```

در مرحله بعد تابع histogram\_of\_LBP را تکمیل می کنیم. ابتدا تصویر را یک کاناله می کنیم. سپس LBP تصویر را بدست می آوریم. در نهایت هیستوگرام آن را برمی گردانیم.

```
1 def histogram_of_LBP(image, numPoints, radius, eps=1e-7):
2     img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
3
4     lbp = local_binary_pattern(img, numPoints, radius, 'uniform')
5     n_bins = int(lbp.max() + 1)
6     hist_values = plt.hist(lbp.ravel(), density=True, bins=n_bins, range=(0, n_bins),
7                             facecolor='0.5')
8
9     return hist_values
```

در نهایت همه این feature ها را در یک ماتریس جمع می کنیم.

```
1 def get_featureMatrix(data):|
2     feature_matrix = []
3     for image in data:
4         features = []
5
6         features.append(np.round(compatness(image), 3))
7         features.append(np.round(eccentricity(image), 3))
8         features.append(np.round(solidity(image), 3))
9         counts = histogram_of_LBP(image, 8, 3)
10
11        for item in counts[0]:
12            features.append(item)
13
14        feature_matrix.append(features)
15
16    return feature_matrix
17
```

در دو مرحله بعد مدل svm ساده ای را train می کنیم و از آن تست می گیریم.

```
1 # model 1
2 feature_matrix_train = get_featureMatrix(x_train)
3 #determine classifier and train
4
5 clf = svm.SVC()
6 clf.fit(feature_matrix_train, y_train)

Part7

1 #test on test dataset
2 test_features = get_featureMatrix(x_test)
3 y_pred = clf.predict(test_features)
4
5 confusion_matrix(y_test, y_pred)

array([[ 0, 20],
       [ 0, 44]])
```

از نتیجه به دست آمده مشخص است که مدل همه دیتا را در یک کلاس قرار داده است.

منابع:

<https://answers.opencv.org/question/231515/arclength-and-contourarea-wrong-results>

[https://docs.opencv.org/3.4/dd/d49/tutorial\\_py\\_contour\\_features.html](https://docs.opencv.org/3.4/dd/d49/tutorial_py_contour_features.html)

<https://stackoverflow.com/questions/31475634/drawcontours-not-working-opencv-python>

<https://pythonexamples.org/python-opencv-cv2-find-contours-in-image>

<https://learnopencv.com/contour-detection-using-opencv-python-c>

[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\\_matrix.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html)

<https://stackoverflow.com/questions/20128898/get-data-points-from-a-histogram-in-python>

[https://scikit-image.org/docs/stable/auto\\_examples/features\\_detection/plot\\_local\\_binary\\_pattern.html](https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_local_binary_pattern.html)

[https://docs.opencv.org/4.x/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html)

<https://www.tutorialspoint.com/how-to-compute-the-area-and-perimeter-of-an-image-contour-using-opencv-python>

<https://www.geeksforgeeks.org/python-opencv-find-center-of-contour>

<https://stackoverflow.com/questions/49635465/error-in-ellipses-using-opencv-python>

<https://scikit-learn.org/stable/modules/svm.html>

[https://docs.opencv.org/4.x/d1/d32/tutorial\\_py\\_contour\\_properties.html](https://docs.opencv.org/4.x/d1/d32/tutorial_py_contour_properties.html)