4/9/2019

DIGITAL IMAGE PROCESSING

ASSIGNMENT # 1

MUHAMMAD HASHIR SHOAIB
MISHAAL SAFEER
SAMEEN ABBAS
FOHA KHALID

Assignment # 1

(CLO1 -> PLO1)

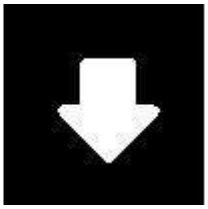
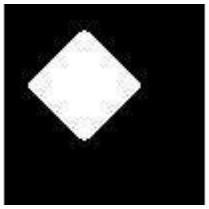
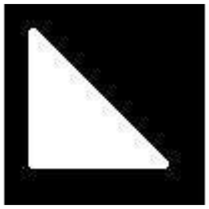
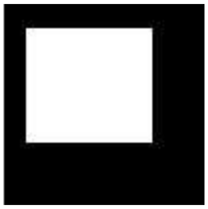
Digital Image Processing

Objects Analysis Based on Connected component labeling

Submission Deadline: 7th Apr 2019

Note: Students should score 40% in OBE specific questions to ensure their accumulated scores towards respective PLOs are above 40%

Connected component analysis is used for detailed study of different objects given in binary image. It can also be used to extract some fruitful information from corresponding color or gray image for same binary image. Similar objects can also be grouped using connected component analysis **Use** your knowledge and **perform** the following operations on these images.

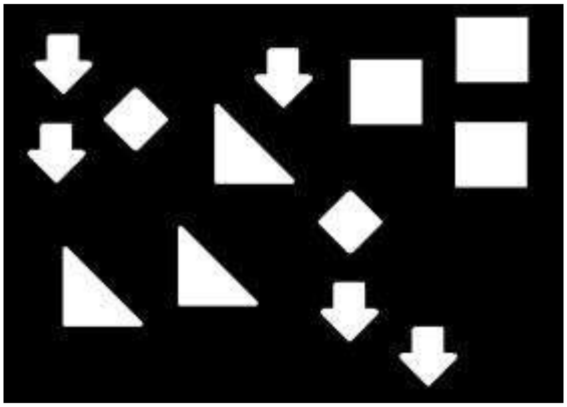
a. Convert given images into binary images

b. Find following features for each type of objects using blob analysis. You can use building functions but should know the basics.

i. 'Eccentricity' ; ii. Ratio1 = 'MinorAxisLength' / 'MajorAxisLength' ; iii. Ratio2 = 'Perimeter' / 'Area' ;

c. For all four types of objects, save these attributes in a CSV file like following table and also assign Label to each type of object

Eccentricity	Raito1	Raito2
.	.	.
.	.	.
.	.	.
.	.	.



Label
1
2
3
4

Use the above calculated features stored in a **CSV** file. Now you have train your system to take a decision after providing 3 features (a **Feature Vector**) for 4 objects with a given unique label for each corresponding object. Each row in above table will be called as a **Feature Vector or observation** about an object and each column is a particular **feature.** Perform the following task below:

a. Use the image above and compute the feature vector for each **new** object one by one present in the image.
b. Use the **Euclidian distance** formula and calculate the distance between new **feature vector** and the **feature vectors** calculated previously for each objects.
c. Assign a label of that feature vector to this **new object** for which it gets smallest distance value.
d. Display the count of total # of objects of each category with object and comment whether your system have assign an accurate label to each new object.
e. Create a new image similar to above image having each pixel of each object assigned with a corresponding label assigned to it and show distinct object with different color.

Code:

```
class Blob:
    # Area of a BLOB is the number of pixels the BLOB consists of.
    label = 0

    def __init__(self):
        Blob.label = Blob.label + 1
        self.eccentricity = 0
        self.ratio1 = 0  # MinorAxisLength / MajorAxisLength
        self.ratio2 = 0  # Perimeter / Area
        self.Label = Blob.label

    def extractfeature(self, image):
        res = image[:]
        ret, thresh = cv.threshold(res, 127, 255, 0)
        contours, hierarchy = cv.findContours(thresh, cv.RETR_TREE,
cv.CHAIN_APPROX_NONE )  #(thresh, 1, 2)
        cnt = contours[0]
        area = cv.contourArea(cnt)
        perimeter = cv.arcLength(cnt, True)
        self.ratio2 = perimeter/area

        ellipse = cv.fitEllipse(cnt)
        (x, y), (Major, minor), angle = ellipse
        MA = max(Major, minor)
        ma = min(Major, minor)
        self.ratio1 = MA / ma
        a = MA/2
        b = ma/2
        c = m.sqrt(m.pow(a,2) - m.pow(b,2))
        e = c/a
        self.eccentricity = e

        return [area , perimeter, MA, ma, e]

    def features(self):
        return [self.eccentricity, self.ratio1, self.ratio2, self.Label]

    def findobjects(self, image, features):
        labels = []
        feat = np.array(features, float)
        res = image[:]
        ret, thresh = cv.threshold(res, 127, 255, 0)
        contours, hierarchy = cv.findContours(thresh, cv.RETR_TREE,
cv.CHAIN_APPROX_NONE)  # (thresh, 1, 2)
        labels = []
        for cnt in contours:
            area = cv.contourArea(cnt)
            perimeter = cv.arcLength(cnt, True)
            self.ratio2 = perimeter / area

            ellipse = cv.fitEllipse(cnt)
            (x, y), (Major, minor), angle = ellipse
            MA = max(Major, minor)
            ma = min(Major, minor)
            self.ratio1 = MA / ma

            a = MA / 2
            b = ma / 2
            c = m.sqrt(m.pow(a, 2) - m.pow(b, 2))
            e = c / a
            self.eccentricity = e

            distancelist = []
            for i in feat:
                data = m.sqrt(m.pow(i[0] - self.eccentricity, 2) + m.pow(i[1] -
self.ratio1, 2) + m.pow(i[2] - self.ratio1, 2))
                distancelist.append(data)
            # print(distancelist)
            # print(distancelist.index(min(distancelist)) + 1)
            labels.append(distancelist.index(min(distancelist)) + 1)
            distancelist.clear()
        return labels
    # WHAT IS BLOB
    # A Blob is a group of connected pixels in an image that share some common
property ( E.g grayscale value ).
    # LINK: https://www.learnopencv.com/blob-detection-using-opencv-python-c/
    #       http://what-when-how.com/introduction-to-video-and-image-processing/blob-
analysis-introduction-to-video-and-image-processing-part-1/
    #       https://www.mathsisfun.com/geometry/eccentricity.html
    #       https://docs.opencv.org/3.0-
beta/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=conn
ectedcomponents#connectedcomponents
    #       https://www.tutorialspoint.com/python3/python_classes_objects.htm
    #       https://docs.opencv.org/3.1.0/dd/d49/tutorial_py_contour_features.html
    #       https://opencv-python-
tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_prope
rties/py_contour_properties.html
    #       https://stackoverflow.com/questions/39486869/how-to-fit-an-ellipse-
contour-with-4-points
arrytemp1 = cv.imread("temp1.png", 0)
arrytemp2 = cv.imread("temp2.png", 0)
arrytemp3 = cv.imread("temp3.png", 0)
arrytemp4 = cv.imread("temp4.png", 0)
assignment = cv.imread("assignment1.png", 0)

obj1 = b.Blob()
obj2 = b.Blob()
obj3 = b.Blob()
obj4 = b.Blob()
data1 = obj1.extractfeature(arrytemp1)
data2 = obj2.extractfeature(arrytemp2)
data3 = obj3.extractfeature(arrytemp3)
data4 = obj4.extractfeature(arrytemp4)
array1 = obj1.features()
array2 = obj2.features()
array3 = obj3.features()
array4 = obj4.features()

csvinputdata = [array1, array2, array3, array4]
with open('person.csv','w') as csvfile:
    writer=cs.writer(csvfile)
    writer.writerows(csvinputdata)
csvfile.close()

csvoutputdata = []
with open('person.csv', 'r') as csvFile:
    reader = cs.reader(csvFile)
    for row in reader:
        csvoutputdata.append(row)
csvFile.close()
print(csvoutputdata)

res = b.Blob()
resdata = res.findobjects(assignment,csvoutputdata)
print(resdata)
```

Output:

Objects with these labels:

[1, 1, 4, 4, 2, 1, 1, 4, 2, 1, 1, 1, 1, 1]

data in csv:

```
0.47079467963881205,1.13347481286401,0.031246549735029123,1
0.20702064053931896,1.0221431206196976,0.030172936659572287,2
0.4193313431149398,1.1015238657266109,0.01639014580191051,3
0.8323897034944049,1.8044328772264322,0.02435419635922391,4
```