

---

## Table of Contents

Q 7 .....	1
Training stage .....	1
Classifying stage .....	1
Used functions .....	2

## Q 7

```
% Code:
clear all, close all, clc;
addpath([genpath('./materials') genpath('./functions')])
```

## Training stage

```
cell112_circ = objects_features('cell112.bmp');
cell112_circ = cell112_circ(:);
[IDX, C] = kmeans(cell112_circ, 2);

C1_indices = find(IDX==1);
C2_indices = find(IDX==2);

fprintf('Cluster 1:\n')
fprintf('Center Circularity: %g | Size: %g\n', C(1),
    length(C1_indices))
fprintf('Cluster 2:\n')
fprintf('Center Circularity: %g | Size: %g\n', C(2),
    length(C2_indices))

Cluster 1:
Center Circularity: 0.673643 | Size: 13
Cluster 2:
Center Circularity: 0.348888 | Size: 15
```

## Classifying stage

```
test_circ = objects_features('test.bmp');
test_circ = test_circ(:);
test_dist = [(test_circ-C(1)).^2 (test_circ-C(2)).^2];
[~, TAG] = min(test_dist, [], 2);

fprintf('\nClassify by min distance:\nclusters size is:\nC1 = %g | C2
    = %g\n', sum(TAG==1), sum(TAG==2))

mu1 = C(1);
sigma1 = std(cell112_circ(C1_indices));

mu2 = C(2);
sigma2 = std(cell112_circ(C2_indices));
```

---

```
test_prob = [normpdf(test_circ, mu1, sigma1) normpdf(test_circ, mu2,
    sigma2)];
[~, TAG] = max(test_prob, [], 2);
```

```
fprintf('\nClassify by max probability:\nclusters size is:\nC1 = %g |
    C2 = %g\n', sum(TAG==1), sum(TAG==2))
```

```
Classify by min distance:
clusters size is:
C1 = 7 | C2 = 8
```

```
Classify by max probabiltiy:
clusters size is:
C1 = 7 | C2 = 8
```

## Used functions

```
function [ C ] = objects_features( file_name )
%OBJECTS_FEATURES Summary of this function goes here
% Detailed explanation goes here
img = imread(file_name);

    [seg_im, vals] = segmentation(img);
    C = [];
    for val=vals(1 : end-1)
        c = circularity(seg_im, val);
        C = [C c];
    end

end

function [ seg_pic, vals ] = segmentation( image )

if size(image,3) == 3
    gray_pic = rgb2gray(image);
else
    gray_pic = double(image);
end
gray_pic(gray_pic==255) = 1;
seg_pic = gray_pic;
[nrows, ncols] = size(gray_pic);
vals(1) = 1;
ind = 1;
for x = 2:ncols-1
    for y = 2:nrows-1
        if seg_pic(y, x) == 0
            seg_pic = region_grow(seg_pic, vals(ind), x, y);
            ind = ind + 1;
            vals(ind) = vals(ind-1) + 10;
        end
    end
end
end
```

---

```
function [gray_pic] = region_grow(gray_pic, val, x, y)
    gray_pic(y, x) = val;
    [nrows, ncols] = size(gray_pic);
    for i = -1:1
        for j= -1:1
            if y+i < 1 || x+j < 1
                continue
            end
            if y+i > nrows || x+j > ncols
                break
            end
            if gray_pic(y+i, x+j)==0
                gray_pic = region_grow(gray_pic, val, x+j, y+i);
            end
        end
    end
end

function [ c ] = circularity( image, val )
% circularity Calculates circularity for segmented shape.
% c = circularity calculation.
% image = segmented image.
% val = segmented shape value.

    area = length(find(image == val));
    perim = myPerim(image, val);
    c = (4*pi*area)/perim^2;
end
```

*Published with MATLAB® R2017a*