

Assignment A7: Image Features

CS 4640
Fall 2019

Assigned: 5 November 2019

Due: 21 November 2019

1. Use image called *mask* in file *chars45.mat* to study shape analysis using Hu's moments for the characters:

A, B, C, D, E, F, G, H, I, K, L, M, N, O, R, S, T, U, V, Y
a, c, d, e, f, g, h, i, k, l, m, n, o, p, r, s, t, u, v, y
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Use a character from the image as a model, $(A_1, A_2, A_3, A_4, A_5, A_6, A_7)$, for each character above. Develop the functions listed below and report performance in terms of overall and per character success.

```
function Mpq = CS4640_central_moment(pts,p,q)
% CS4640_central_moment - compute a central moment
%   Mpq = sum sum (x^p*y^q)
%           x   y
% On input:
%   pts (nx2 array): row and cols of points
%   p (int): exponent for x
%   q (int): exponent for y
% On output:
%   Mpq (float): Mpq moment
% Call:
%   M00 = CS4640_central_moments([1 1; 2 2; 3 3],0,0);
% Author:
%   <your name>
```

```

%      UU
%      Fall 2019
%
function Epq = CS4640_normal_moment(pts,p,q)
% CS4640_normal_moment - compute a central normal moment
%   Epq = <pq/M00^b   where b = 1+(p+q)/2
% On input:
%   pts (nx2 array): row and cols of points
%   p (int): exponent for x
%   q (int): exponent for y
% On output:
%   Epq (float): Epq moment
% Call:
%   E00 = CS4640_normal_moment([1 1; 2 2; 3 3],0,0);
% Author:
%   <your name>
%   UU
%   Fall 2019
%
function H = CS4640_Hu_moments(pts)
% CS4640_Hu_moments - compute Hu's 7 moments
% On input:
%   pts (nx2 array): row and cols of points
% On output:
%   H (7x1 vector): Hu moments
% Call:
%   H = CS4640_Hu_moments([1 1; 2 2; 3 3]);
% Author:
%   <your name>
%   UU
%   Fall 2019
%
function H_models = CS4640_Hu_build(templates)
% CS4640_Hu_models - produce Hu models for image templates
% On input:
%   templates (n-element vector struct): template images
%   (k).im (MxN binary image): image template
% On output:
%   H_models (nx7 array): Hu models
% Call:
%   Hm = CS4640_Hu_build(templates);

```

```

% Author:
%     <your name>
%     UU
%     Fall 2019
%
function classes = CS4640_Hu_classify(im,H_models)
% CS4640_Hu_classify - classify characters using Hu models
% On input:
%     im (MxN binary image): input image
%     H_models (nx7 array): Hu models for n characters
% On output:
%     classes (kx2 array): class and distance for each CC
% Call:
%     Hm = CS4640_Hu_classify(im,Hm);
% Author:
%     <your name>
%     UU
%     Fall 2019
%

```

2. Implement an eigenchars classification approach similar to the eigenfaces method discussed in the text. Develop a template database of 100 images (2 examples of each character). Build the models and then report performance on overall and per character success on *mask* from *chars45.mat*. Develop the following functions.

```

function [V,MM,PCA_models] = CS4640_PCA_model(templates)
% CS4640_PCA_model - build PCA model from templates
% On input:
%     templates (vector struct): n template images
%         (k).im (MxN binary array): template image for character k
% On output:
%     V (M*nxM*n array): eigenvectors
%     MM (M*nx1 vector): mean vector
%     PCA_models (nxk array): weight values for first k eigenvectors
% Call:
%     [V,MM,PCA_models] = CS4640_PCA_model(templates);
% Author:
%     <your name>
%     UU
%

```

```

%      Fall 2019
%
function c = CS4640_PCA_classify(im,V,MM,PCA_models)
% CS4640_PCA_classify - classify image using PCA models
% On input:
%      im (MxN binary array): input image
%      V (M*NxM*N array): eigenvectors
%      MM (M*Nx1 vector): mean vector
%      PCA_models (nxk array): weight values for first k eigenvectors
% On output:
%      c (int): class
% Call:
%      [V,MM,PCAm] = CS4640_PCA_model(templates);
% Author:
%      <your name>
%      UU
%      Fall 2019
%

```