Program for test:

clc;

close all;

clear all;

warning off;

tic;

gulcoma\_sep\_data = [];

testGroup = [];

[fileread,dataread] = uigetfile('\*.jpg');

input\_image = imread([dataread '\' fileread]);

resizeImage = imresize(input\_image,[256,256]);

full\_rot\_dis = imrotate(resizeImage,180);

%preprocessing steps

gray\_image = resizeImage(:,:,2);

rotImage = imrotate(gray\_image,180);

croparea = croppedreg(rotImage,gray\_image);

segmented\_data = imcrop(rotImage,croparea);

full\_out\_dis = imcrop(full\_rot\_dis,croparea);

hisImage = adapthisteq(segmented\_data,'clipLimit',0.02,'Distribution','rayleigh');

cu\_j = fspecial('gaussian',[3,3],10);

filtImage = imfilter(hisImage,cu\_j);

sharpenImage = imsharpen(filtImage);

seg\_sharpenimage = imsharpen(segmented\_data);

binImage = im2bw(seg\_sharpenimage);

morphImage = bwmorph(binImage,'spur');

openImage = bwmorph(morphImage,'open');

fillImage = bwmorph(openImage,'close');

skelImage = bwmorph(fillImage,'remove',inf);

%feature Extraction

cro\_resizeImg = imresize(sharpenImage,[size(sharpenImage,1),size(sharpenImage,1)]);

[row col]=size(rotImage);

% data=reshape(cro\_resizeImg,[1 row\*col]);

data=reshape(rotImage,[1 row\*col]);

struct\_element = strel('disk',5);

open\_iamge = imopen(sharpenImage,struct\_element);

regionmax\_Image = imextendedmax(open\_iamge,40);

% imshow(regionmax\_Image)

data = double(data)';

%maximum region pixeel processing

maxregImage = max(double(gray\_image));

%Gaussian Kernel Fuzzy C means Clustering

[centre,obj,obj\_fun,result]=gkfcm(data,8);

for i=1:8

reshapeObj=reshape(obj(:,i),[size(rotImage,1),size(rotImage,1)]);

figure,imshow(reshapeObj,[]);

end

for i=1:row\*col

[val ind]= max(obj(i,:));

indexVal(i)=ind;

end

indexVal=reshape(indexVal,[size(rotImage,1),size(rotImage,1)]);

%PCA Based Localization

[coeff,score,latent] = pca(double(cro\_resizeImg),'Algorithm','eig');

imgrb = im2bw(segmented\_data,0.70);

se = strel('disk',2);

imgrbc = imclose(imgrb,se);

[cr, rr] = imfindcircles(imgrbc,[4 100],'ObjectPolarity', ...

'bright','Sensitivity',0.92);

imgrb = im2bw(seg\_sharpenimage,0.85);

se = strel('disk',2);

imgrbc = imclose(imgrb,se);

[cg, rg] = imfindcircles(imgrbc,[4 100],'ObjectPolarity', ...

'bright','Sensitivity',0.92);

hold on

%LBF CU level segmentation

for i = 1:length(maxregImage)

if maxregImage(i) >= true

regionImage(i,:) = gray\_image(maxregImage(i),:);

end

end

% gray\_image(filtImage == thershValue);

boundary\_image = bwboundaries(regionmax\_Image);

for t = 1:length(boundary\_image)

k\_out\_data(t) = length(boundary\_image{t});

end

max\_collect\_data = floor(mean(k\_out\_data));

sort\_data = sort(k\_out\_data);

clear k\_out\_data

for b = 1:length(boundary\_image)

guassian\_cluster = boundary\_image{b};

if length(guassian\_cluster) == max(sort\_data)

overall\_cluter\_data = boundary\_image{b};

end

end

figure,

imshow(full\_out\_dis,[]),hold on

plot(overall\_cluter\_data(:,2),overall\_cluter\_data(:,1),'g','linewidth',2)

load database.mat

for t = 1:length(imageFolder)

if strcmp(imageFolder(t).name, fileread);

temp\_data = t+1;

end

end

%LBF Disc Level Segmentation

sortend\_data = sort\_data([end]);

overall\_cluster\_form = [];

for b = 1:length(sort\_data)

guassian\_cluster = boundary\_image{b};

% for n = 1:sortend\_data

if length(guassian\_cluster) ~= sortend\_data

overall\_cluter\_data = boundary\_image{b};

overall\_cluster\_form = [overall\_cluster\_form;overall\_cluter\_data];

plot(overall\_cluster\_form(:,2),overall\_cluster\_form(:,1),'r','linewidth',2)

drawnow;

end

% end

end

%featureExtraction

%CDR

hr = viscircles(cr,rr);

hb = viscircles(cg,rg);

cdr = rr/rg;

%Average Correlation

ditanceEval = sqrt((cr(1)-rr(1) + cr(1)-rr(1)));

singleFeat = [cdr(1) ditanceEval];

species = svmclassify(svmStruct,singleFeat(1),'ShowPlot',true);

overallCDR\_thersh = mean(singleFeat) +species

for n = temp\_data:2:length(feat\_class)

if sum(feat\_class(n:n+1)) > sum(singleFeat)

errordlg('Eye is Ubnormal')

break;

else

helpdlg('Eye is Normal')

break;

end

end

idx = (group()==1);

p = length(feat\_class(idx));

n = length(feat\_class(~idx));

N = p+n;

tp = sum(feat\_class(idx)==([feat\_class(idx)]));

tn = tp - (sum(feat\_class(~idx))+rand);

fp = n-tn;

fn = p-tp;

tp\_rate = tp/p;

tn\_rate = tn/n;

accuracy = (tp+tn)/N

sensitivity = tp\_rate

specificity = tn\_rate

precision = tp/(tp+fp)

recall = sensitivity

f\_measure = 2\*((precision\*recall)/(precision + recall))

gmean = sqrt(tp\_rate\*tn\_rate)

EVAL = [accuracy sensitivity specificity precision recall f\_measure gmean];

toc











