clc

clear all

close all

%% Read & Plot Target Image

targetImage = imread ('C:\Users\Bhavana\Documents\MATLAB\FinalCode\Blood

SnearImages\img09.jpg');

figure

imshow(targetImage);

%% Convert to Greyscale

grayscale = rgb2gray(targetImage);

figure

imshow(grayscale);

%% Imhmin and WS algorithm

newGrayscale = imhmin(grayscale,25);

wsImg = watershed(newGrayscale);

cellMask = segmentImageFcn(targetImage);

cellMask = bwareaopen(cellMask,30);

wsEdges = wsImg == 0;

wsEdges = bwareaopen(wsEdges,200,8);

cellMask(wsEdges) = 0;

figure

imshow(cellMask);

%% Detect circles

detectCircles = @(x) imfindcircles(x,[20 35], ...

'Sensitivity',0.89, ...

'EdgeThreshold',0.04, ...

'Method','TwoStage', ...

'ObjectPolarity','Dark');

[centers, radii, metric] = detectCircles(grayscale);

figure

imshow(targetImage)

viscircles(centers,radii,'edgecolor','r');

%% differentiate types of infections from blood-smear images

imgSet = imageSet(fullfile(pwd,'.\BloodSnearDatabase'),...

'recursive') ;

disp(['Your imageSet contains ', num2str(sum([imgSet.Count])),...

' images from ' num2str(numel(imgSet)) ' classes.']);

subset = select(imgSet,1:3);

subsetNames = [subset.ImageLocation];

subsetLabels = {};

for ii = 1:numel(subset)

subsetLabels{ii} = repelem({subset(ii).Description},subset(ii).Count,1);%#ok

end

subsetLabels = vertcat(subsetLabels{:});

togglefig('Sample Images',1)

[hpos,hdim] = distributeObjects(numel(subset),0.05,0.95,0.01);

[vpos,vdim] = distributeObjects(3,0.95,0.05,0.025);

ax = gobjects(numel(subset),1);

[hind,vind] = meshgrid(1:numel(imgSet),1:subset(1).Count);

for ii = 1:numel(subsetNames)

ax(ii) = axes('Units','Normalized',...

'Position',...

[hpos(hind(ii)) vpos(vind(ii)) hdim vdim]);

imshow(subsetNames{ii});

title(subsetLabels{ii},'fontsize',8)

end

expandAxes(ax);

%% PARTITION the imageSet into training and test sets & create a visual BAG OF FEATURES

rng default

[trainingSets, testSets] = partition(imgSet,0.6,'randomized');

extractorFcn = @customParasitologyFcn;

bag = bagOfFeatures(imgSet,...

'CustomExtractor',extractorFcn,...

'StrongestFeatures',1,...

'VocabularySize',500);

togglefig('Encoding',true)

for ii = 1:numel(imgSet)

img = read(imgSet(ii), randi(imgSet(ii).Count));

featureVector = encode(bag, img);

subplot(numel(imgSet),2,ii\*2-1);

imshow(img);

title(imgSet(ii).Description)

subplot(numel(imgSet),2,ii\*2);

bar(featureVector);

set(gca,'xlim',[0 bag.VocabularySize])

title('Visual Word Occurrences');

if ii == numel(imgSet)

xlabel('Visual Word Index');

end

if ii == floor(numel(imgSet)/2)

ylabel('Frequency of occurrence');

end

end

%% TRAIN category classifier on the training set

Classifier = trainImageCategoryClassifier(trainingSets,bag);

%% EVALUATE the classifier on the test-set images:

[confMat,knownLabelIdx,predictedLabelIdx,~] = ...

evaluate(classifier,testSets);

avgAccuracy = mean(diag(confMat));

togglefig('Prediction')

imagesc(confMat)

colorbar

%% Now we can use the classifier to PREDICT class membership!

togglefig('Prediction')

ii = randi(size(imgSet,2));

img = read(imgSet(ii),randi(imgSet(ii).Count));

[labelIdx, predictionScore] = predict(classifier,img);

bestGuess = classifier.Labels(labelIdx);

actual = imgSet(ii).Description;

imshow(img)

t = title(['Best Guess: ',bestGuess{1},'; Actual: ',actual]);

if strcmp(bestGuess{1},actual)

set(t,'color',[0 0.7 0])

else

set(t,'color','r')

end

%%

sendEMail('reddybhavana09',bestGuess{1},'MESSAGE');