4 OCR system construction and system testing

Summary

Note* - Error rate = 1 - hitrate

The hitrate for version 1:

Short1: 62%

Short2: 72%

Home1: 5.6%

Home2: 5.9%

Home3: 6.1%

The hitrate for version 2:

Short1: 72%

Short2: 74%

Home1: 52%

Home2: 56.6%

Home3: 57.9%

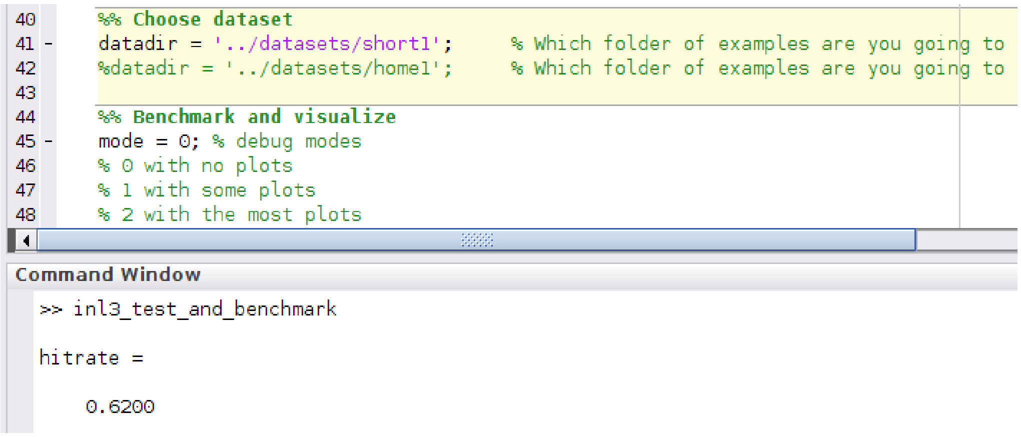
Based on the result for both version, version 2 performed 9 times better on correctly classifying characters. The reason version 2 did better because the feature extraction is not relying on the input images to have the same size when the characters segmented. Version 1 heavily relied on same size image and character size, thickness inorder to have a consistent and accurate result.

Changed the feature extraction for version 2. After getting the extracted image, it zooms in on the character, resizes all image to be around 50pixel tall. That way, there is some consistency with the dimension of the image. Then I feed it into the feature extraction from version 1 and added a few more features such as the orientation, EulerNumber and such from region props.

I have a hard time push improving the hitrate the homes' data set. I tried to improve on the segmentation, but then it did not perform well for the shorts' dataset. I google scholar papers and techniques research had used to improve OCR, but a lot of them uses a neural network. I tried but did not succeed in the time being. I will have to work it out after this course.

Version 1 of OCR Results for 5 datasets

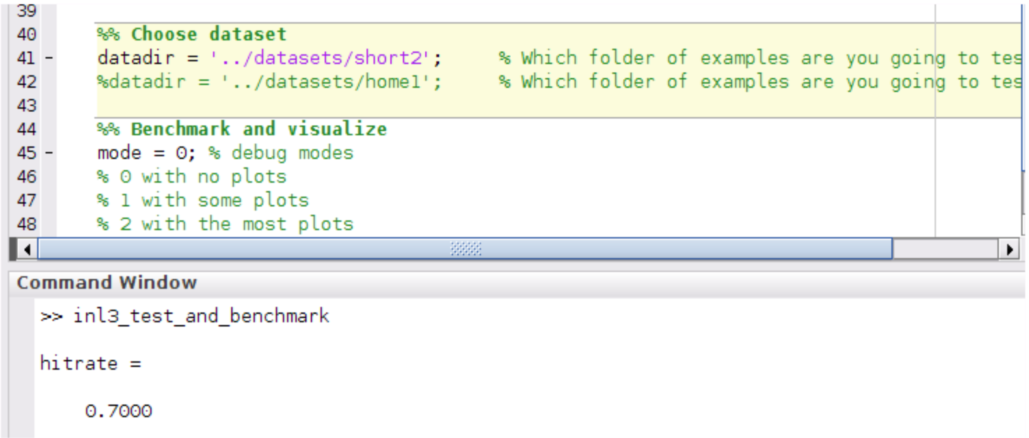
```
[im, s] = imread('q4t1s1','png');
imshow(im);
```



```
%% Choose dataset
datadir = '../datasets/short1';     % Which folder of examples are you going to
%datadir = '../datasets/home1';     % Which folder of examples are you going to

%% Benchmark and visualize
mode = 0; % debug modes
% 0 with no plots
% 1 with some plots
% 2 with the most plots
```

```
>> inl3_test_and_benchmark

hitrate =

    0.6200
```

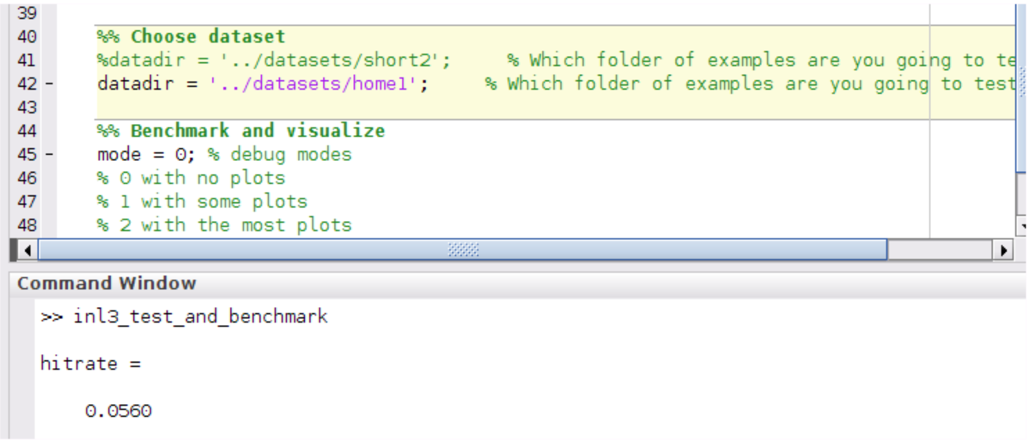
```
[im, s] = imread('q4t1s2','png');
imshow(im);
```



```
%% Choose dataset
datadir = '../datasets/short2';     % Which folder of examples are you going to tes
%datadir = '../datasets/home1';     % Which folder of examples are you going to tes

%% Benchmark and visualize
mode = 0; % debug modes
% 0 with no plots
% 1 with some plots
% 2 with the most plots
```

```
>> inl3_test_and_benchmark

hitrate =

    0.7000
```

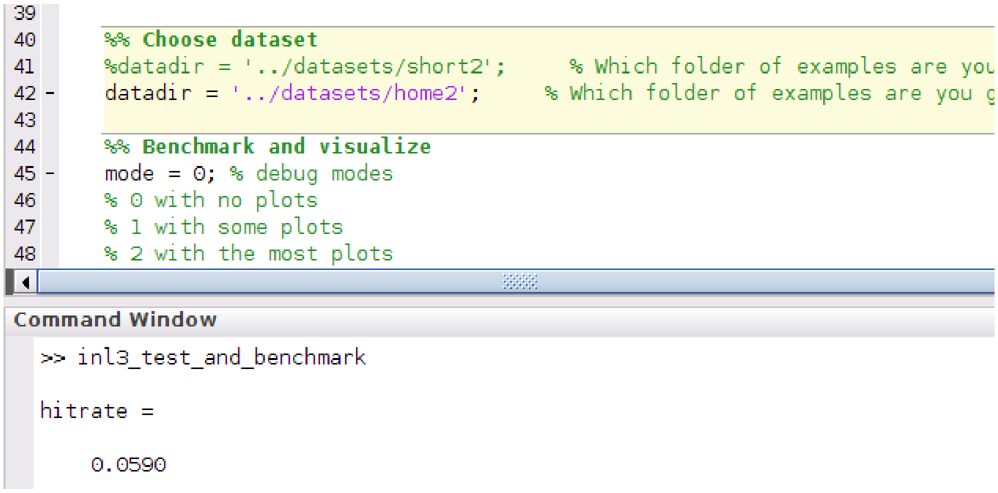
```
[im, s] = imread('q4t1h1','png');
imshow(im);
```



```
%% Choose dataset
%datadir = '../datasets/short2';     % Which folder of examples are you going to te
datadir = '../datasets/home1';     % Which folder of examples are you going to test

%% Benchmark and visualize
mode = 0; % debug modes
% 0 with no plots
% 1 with some plots
% 2 with the most plots
```

```
>> inl3_test_and_benchmark

hitrate =

    0.0560
```

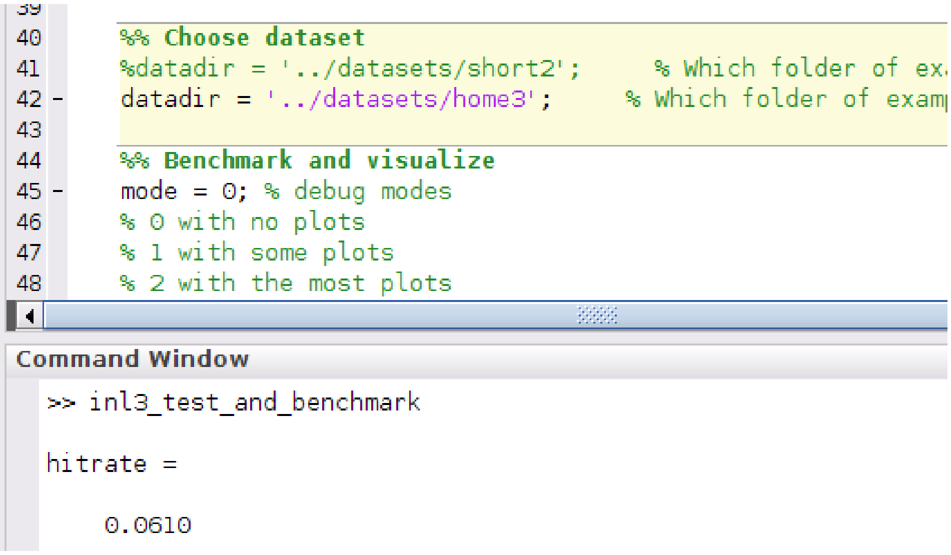
```
[im, s] = imread('q4t1h2','png');
imshow(im);
```



```
%% Choose dataset
%datadir = '../datasets/short2';     % Which folder of examples are you
datadir = '../datasets/home2';     % Which folder of examples are you g

%% Benchmark and visualize
mode = 0; % debug modes
% 0 with no plots
% 1 with some plots
% 2 with the most plots
```

```
>> inl3_test_and_benchmark

hitrate =

    0.0590
```

```
[im, s] = imread('q4t1h3','png');
imshow(im);
```

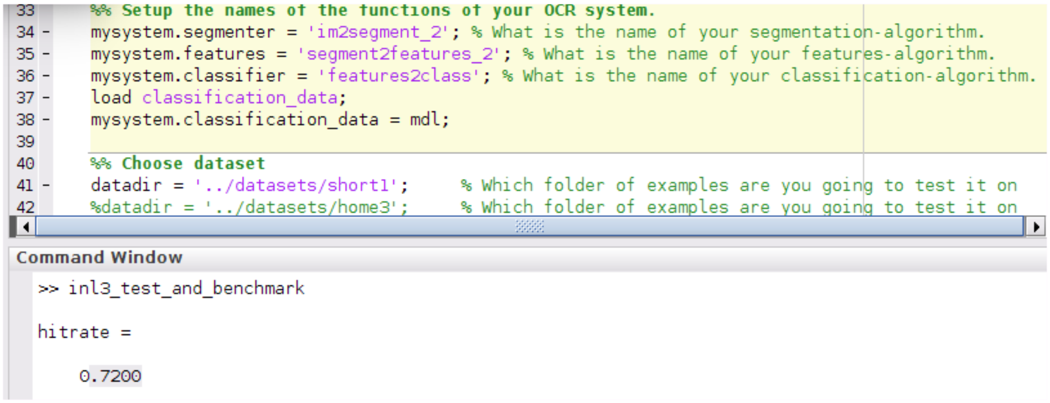


```
%% Choose dataset
%datadir = '../datasets/short2';      % Which folder of ex
datadir = '../datasets/home3';      % Which folder of exam

%% Benchmark and visualize
mode = 0; % debug modes
% 0 with no plots
% 1 with some plots
% 2 with the most plots
```

```
>> inl3_test_and_benchmark

hitrate =

    0.0610
```

Version 2 of OCR Results for 5 datasets

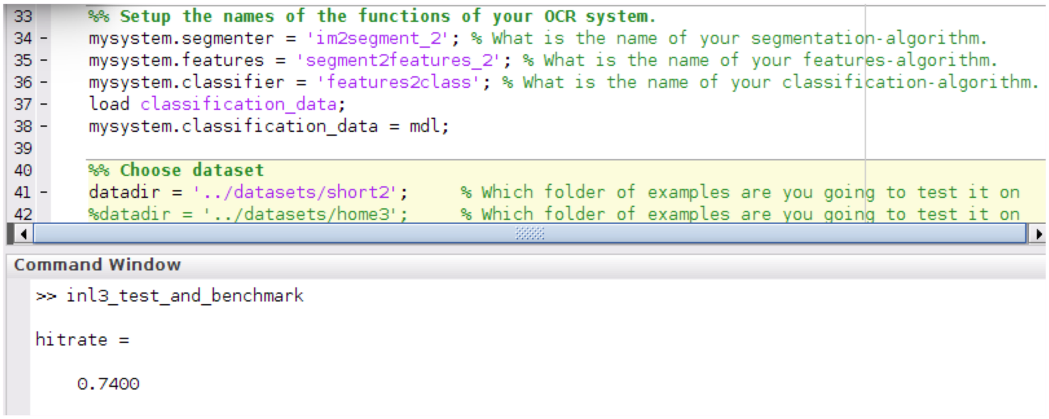
```
[im, s] = imread('q4t2s1','png');
imshow(im);
```



```
%% Setup the names of the functions of your OCR system.
mysystem.segmenter = 'im2segment_2'; % What is the name of your segmentation-algorithm.
mysystem.features = 'segment2features_2'; % What is the name of your features-algorithm.
mysystem.classifier = 'features2class'; % What is the name of your classification-algorithm.
load classification_data;
mysystem.classification_data = mdl;

%% Choose dataset
datadir = '../datasets/short1';      % Which folder of examples are you going to test it on
%datadir = '../datasets/home3';      % Which folder of examples are you going to test it on
```

```
>> inl3_test_and_benchmark

hitrate =

    0.7200
```

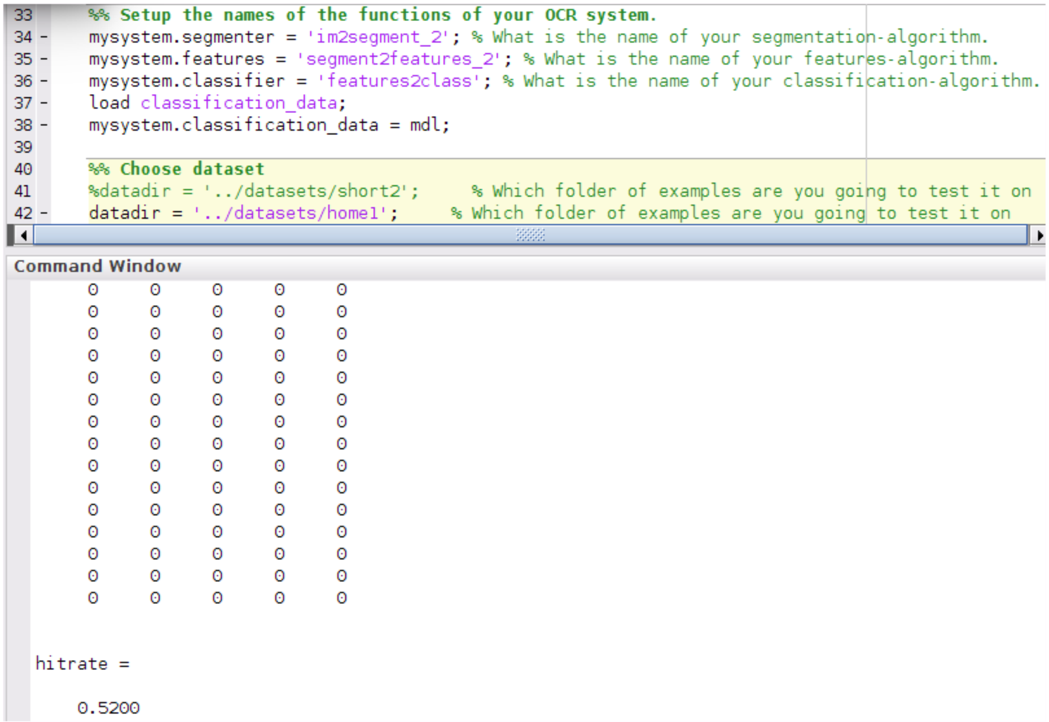
```
[im, s] = imread('q4t2s2','png');
imshow(im);
```



```
%% Setup the names of the functions of your OCR system.
mysystem.segmenter = 'im2segment_2'; % What is the name of your segmentation-algorithm.
mysystem.features = 'segment2features_2'; % What is the name of your features-algorithm.
mysystem.classifier = 'features2class'; % What is the name of your classification-algorithm.
load classification_data;
mysystem.classification_data = mdl;

%% Choose dataset
datadir = '../datasets/short2';     % Which folder of examples are you going to test it on
%datadir = '../datasets/home3';     % Which folder of examples are you going to test it on
```

```
>> inl3_test_and_benchmark

hitrate =

    0.7400
```

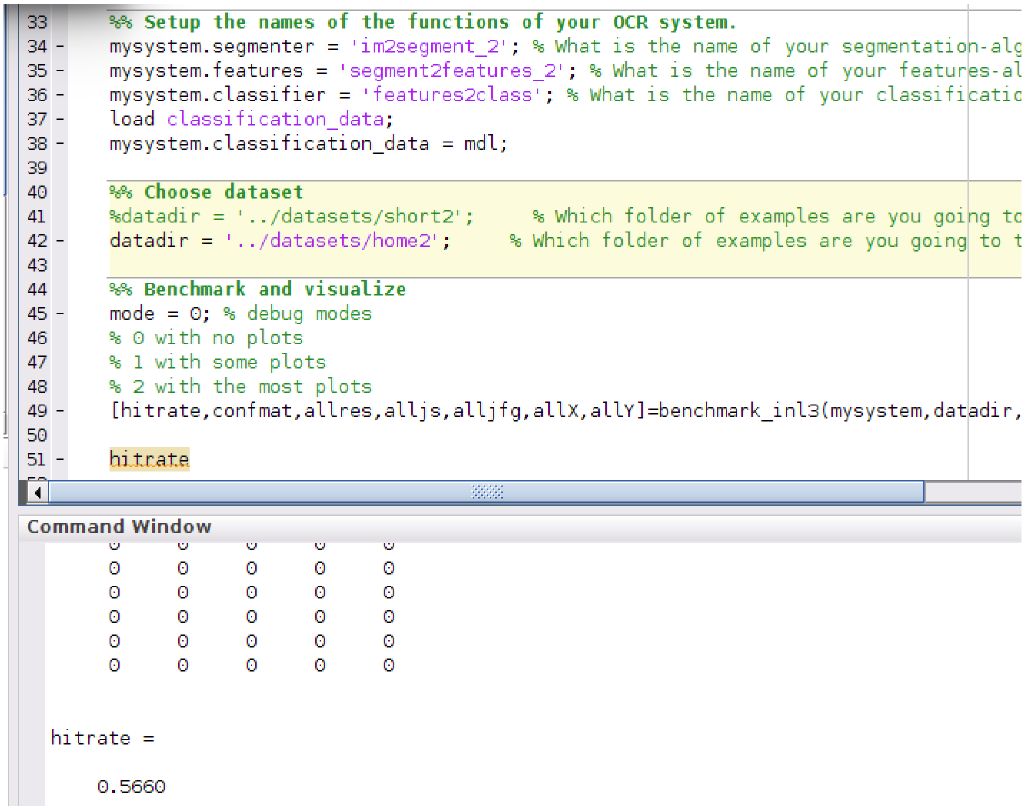
```
[im, s] = imread('q4t2h1','png');
imshow(im);
```



```
%% Setup the names of the functions of your OCR system.
mysystem.segmenter = 'im2segment_2'; % What is the name of your segmentation-algorithm.
mysystem.features = 'segment2features_2'; % What is the name of your features-algorithm.
mysystem.classifier = 'features2class'; % What is the name of your classification-algorithm.
load classification_data;
mysystem.classification_data = mdl;

%% Choose dataset
%datadir = '../datasets/short2';     % Which folder of examples are you going to test it on
datadir = '../datasets/home1';     % Which folder of examples are you going to test it on
```

```
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0


hitrate =

    0.5200
```

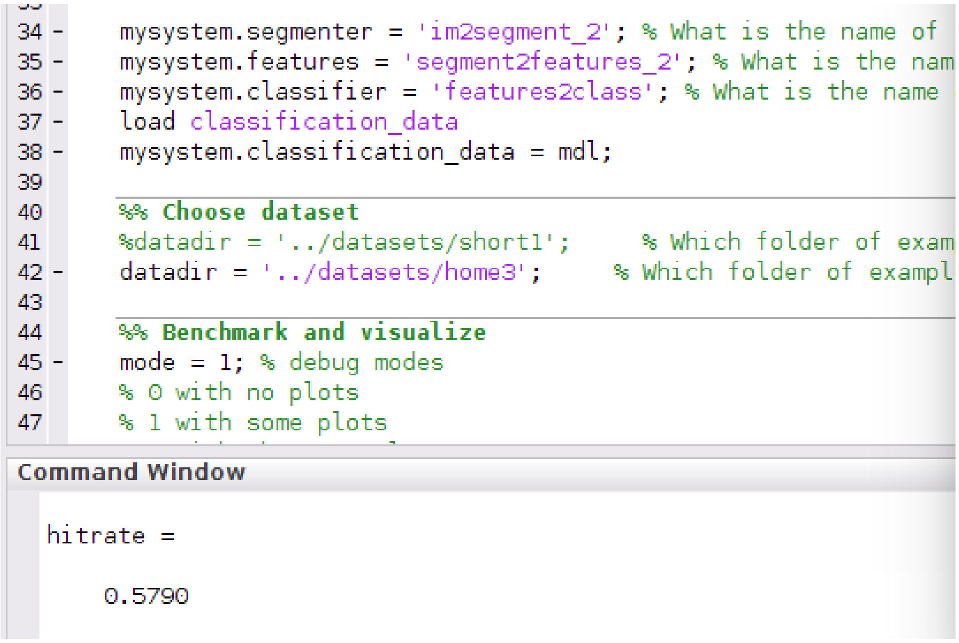
```
[im, s] = imread('q4t2h2','png');
imshow(im);
```



```
%% Setup the names of the functions of your OCR system.
mysystem.segmenter = 'im2segment_2'; % What is the name of your segmentation-alg
mysystem.features = 'segment2features_2'; % What is the name of your features-al
mysystem.classifier = 'features2class'; % What is the name of your classificatio
load classification_data;
mysystem.classification_data = mdl;

%% Choose dataset
%datadir = '../datasets/short2';     % Which folder of examples are you going to
datadir = '../datasets/home2';     % Which folder of examples are you going to t

%% Benchmark and visualize
mode = 0; % debug modes
% 0 with no plots
% 1 with some plots
% 2 with the most plots
[hitrate,confmat,allres,alljs,alljfg,allX,allY]=benchmark_inl3(mysystem,datadir,

hitrate
```

```
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0


hitrate =

    0.5660
```

```
[im, s] = imread('q4t2h3','png');
imshow(im);
```

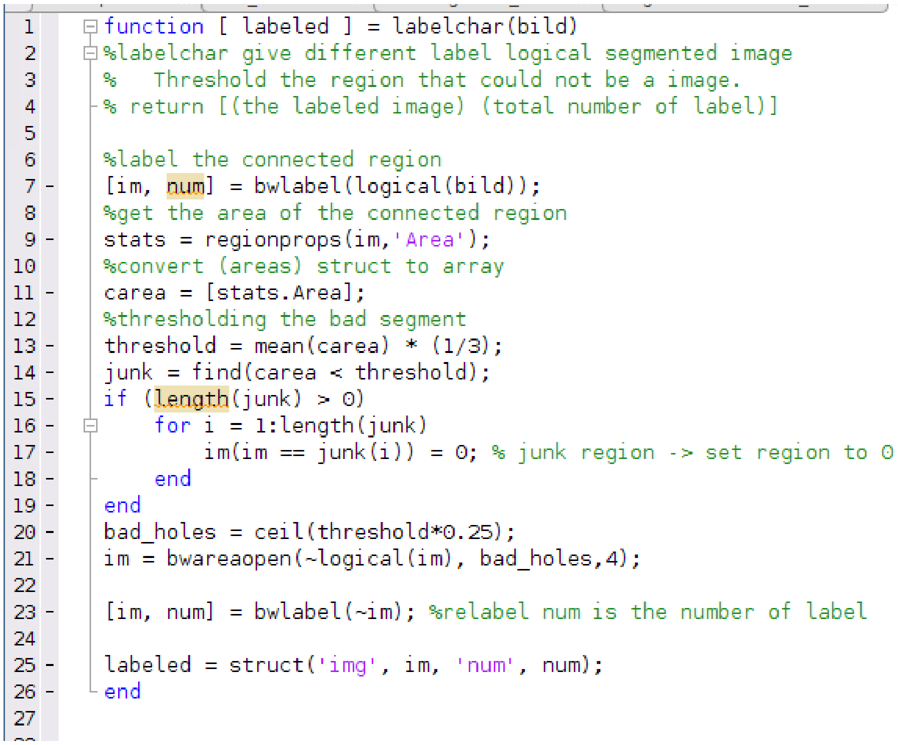


```
mysystem.segmenter = 'im2segment_2'; % What is the name of
mysystem.features = 'segment2features_2'; % What is the nam
mysystem.classifier = 'features2class'; % What is the name
load classification_data
mysystem.classification_data = mdl;

%% Choose dataset
%datadir = '../datasets/short1';     % Which folder of exam
datadir = '../datasets/home3';     % Which folder of exampl

%% Benchmark and visualize
mode = 1; % debug modes
% 0 with no plots
% 1 with some plots
```

```
hitrate =

    0.5790
```

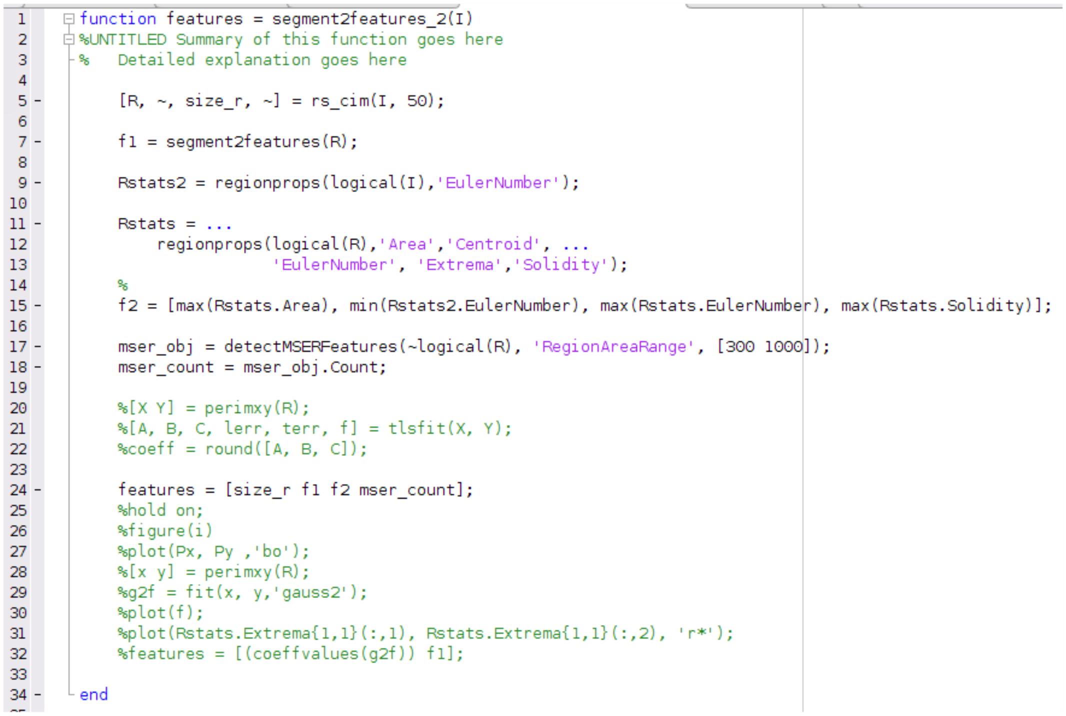
Code

```
[im, s] = imread('q4c1','png');
imshow(im);
```



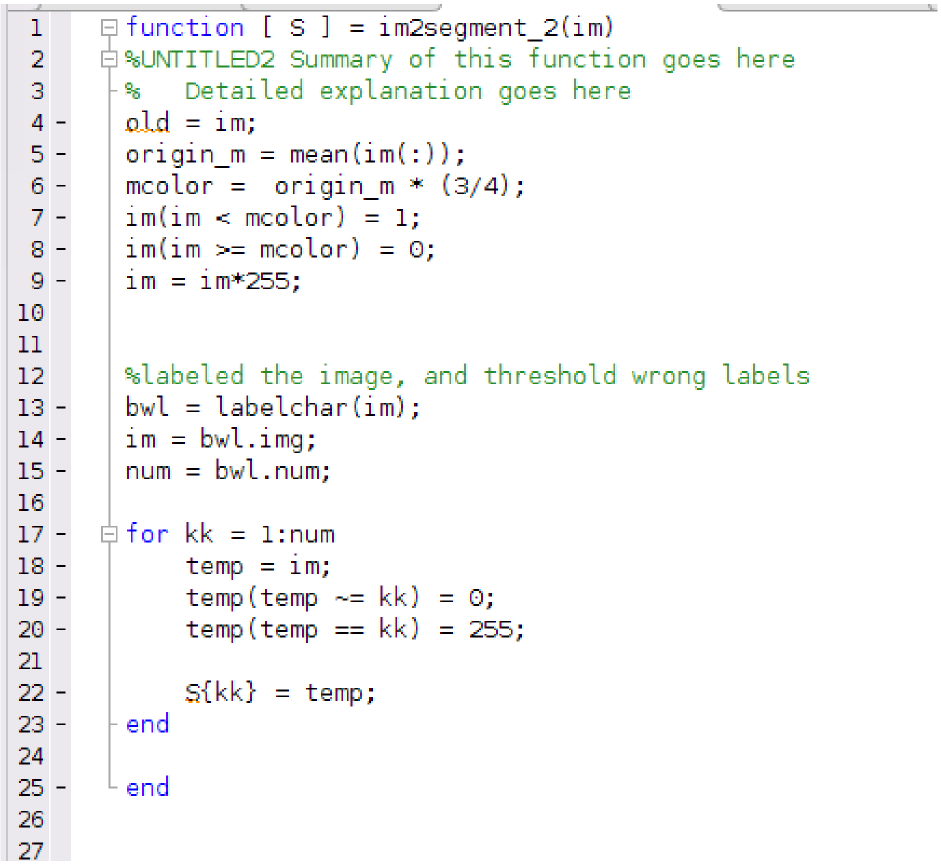
```
function [ labeled ] = labelchar(bild)
%labelchar give different label logical segmented image
%   Threshold the region that could not be a image.
% return [(the labeled image) (total number of label)]

%label the connected region
[im, num] = bwlabel(logical(bild));
%get the area of the connected region
stats = regionprops(im,'Area');
%convert (areas) struct to array
carea = [stats.Area];
%thresholding the bad segment
threshold = mean(carea) * (1/3);
junk = find(carea < threshold);
if (length(junk) > 0)
    for i = 1:length(junk)
        im(im == junk(i)) = 0; % junk region -> set region to 0
    end
end
bad_holes = ceil(threshold*0.25);
im = bwareaopen(~logical(im), bad_holes,4);

[im, num] = bwlabel(~im); %relabel num is the number of label

labeled = struct('img', im, 'num', num);
end
```

```
[im, s] = imread('q4c2','png');
imshow(im);
```



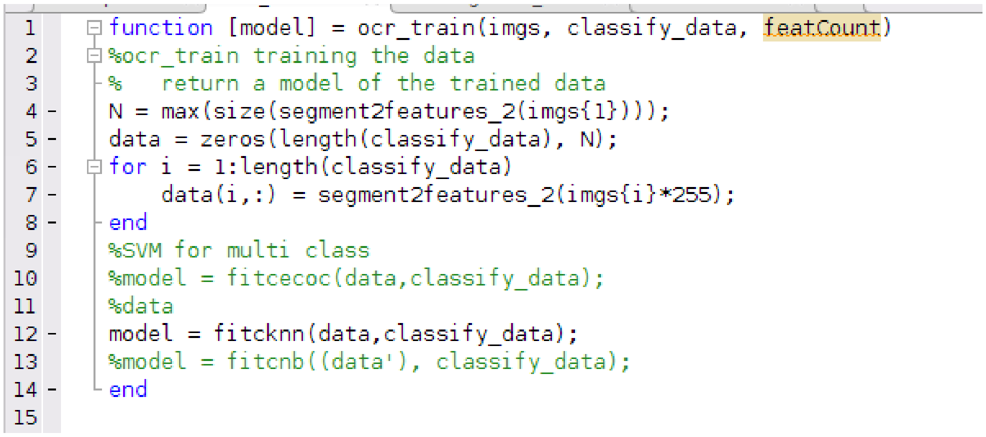
```
function features = segment2features_2(I)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here

    [R, ~, size_r, ~] = rs_cim(I, 50);

    f1 = segment2features(R);

    Rstats2 = regionprops(logical(I),'EulerNumber');

    Rstats = ...
        regionprops(logical(R),'Area','Centroid', ...
                    'EulerNumber', 'Extrema','Solidity');
    %
    f2 = [max(Rstats.Area), min(Rstats2.EulerNumber), max(Rstats.EulerNumber), max(Rstats.Solidity)];

    mser_obj = detectMSERFeatures(~logical(R), 'RegionAreaRange', [300 1000]);
    mser_count = mser_obj.Count;

    %[X Y] = perimxy(R);
    %[A, B, C, lerr, terr, f] = tlsfit(X, Y);
    %coeff = round([A, B, C]);

    features = [size_r f1 f2 mser_count];
    %hold on;
    %figure(i)
    %plot(Px, Py ,'bo');
    %[x y] = perimxy(R);
    %g2f = fit(x, y,'gauss2');
    %plot(f);
    %plot(Rstats.Extrema{1,1}(:,1), Rstats.Extrema{1,1}(:,2), 'r*');
    %features = [(coeffvalues(g2f)) f1];

end
```

```
[im, s] = imread('q4c3','png');
imshow(im);
```



```
function [ S ] = im2segment_2(im)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
old = im;
origin_m = mean(im(:));
mcolor =  origin_m * (3/4);
im(im < mcolor) = 1;
im(im >= mcolor) = 0;
im = im*255;


%labeled the image, and threshold wrong labels
bwl = labelchar(im);
im = bwl.img;
num = bwl.num;

for kk = 1:num
    temp = im;
    temp(temp ~= kk) = 0;
    temp(temp == kk) = 255;

    S{kk} = temp;
end

end
```

```
[im, s] = imread('q4c4','png');
imshow(im);
```



```
function [model] = ocr_train(imgs, classify_data, featCount)
%ocr_train training the data
%   return a model of the trained data
N = max(size(segment2features_2(imgs{1})));
data = zeros(length(classify_data), N);
for i = 1:length(classify_data)
    data(i,:) = segment2features_2(imgs{i}*255);
end
%SVM for multi class
%model = fitcecoc(data,classify_data);
%data
model = fitcknn(data,classify_data);
%model = fitcnb((data'), classify_data);
end
```