Computer Vision CS534

Homework 1:OCR

Hao Luo

RUID: 162006265

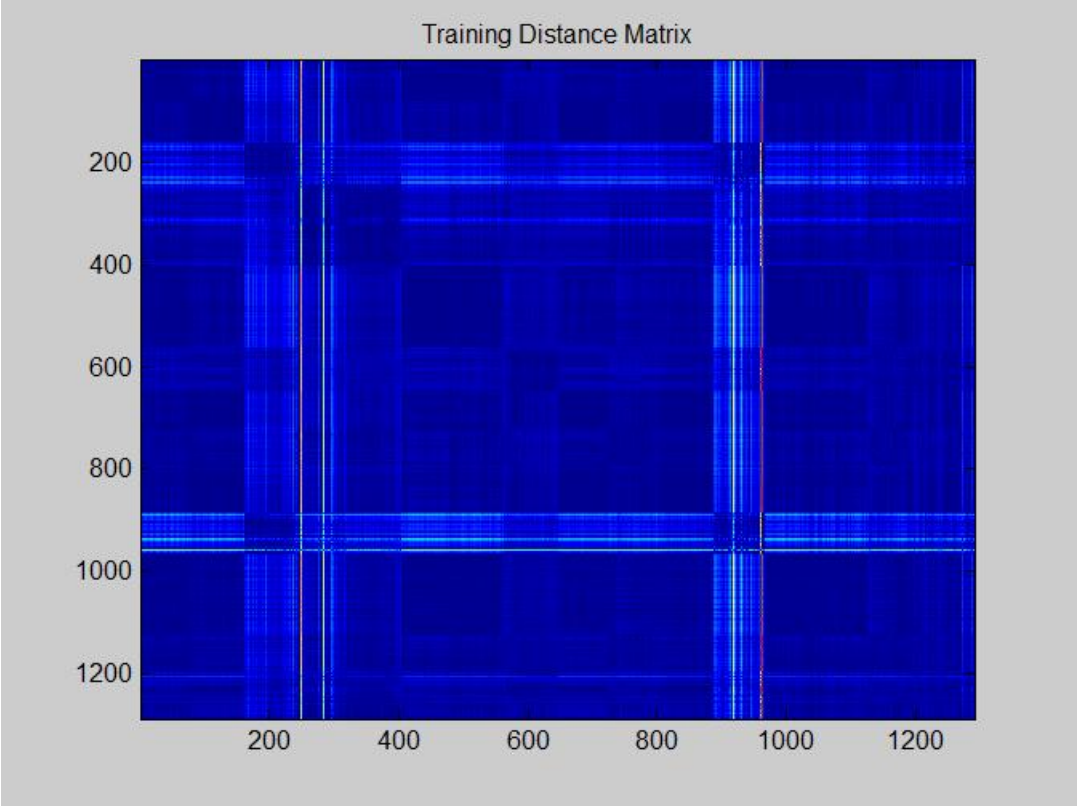
1, TrainingData and Testingdata recognition result

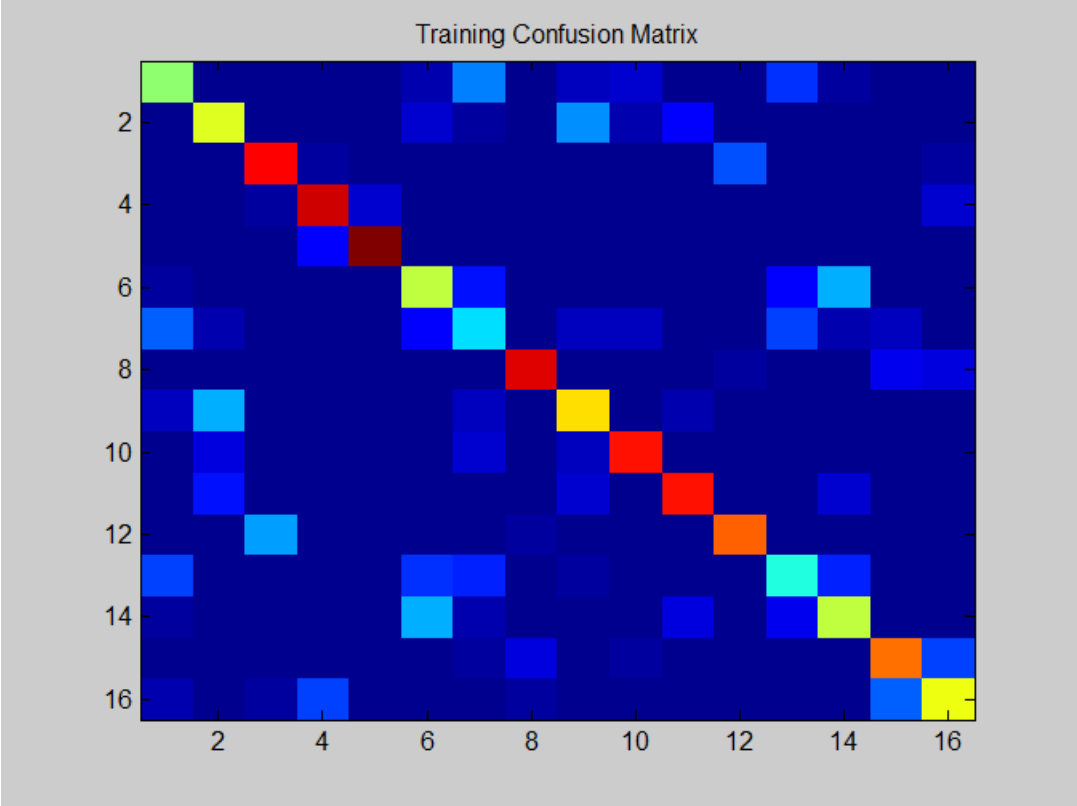
By Choosing the Threshold = 200.

1. The basic Training Data recognition rate:

A: 0.4500	D: 0.5250	F: 0.7875	H: 0.8500
K: 0.9000	M: 0.5000	N: 0.3000	O: 0.8250
P: 0.5875	Q: 0.7500	R: 0.7625	S: 0.7000
U: 0.3625	W: 0.5000	X: 0.6875	Z: 0.5422

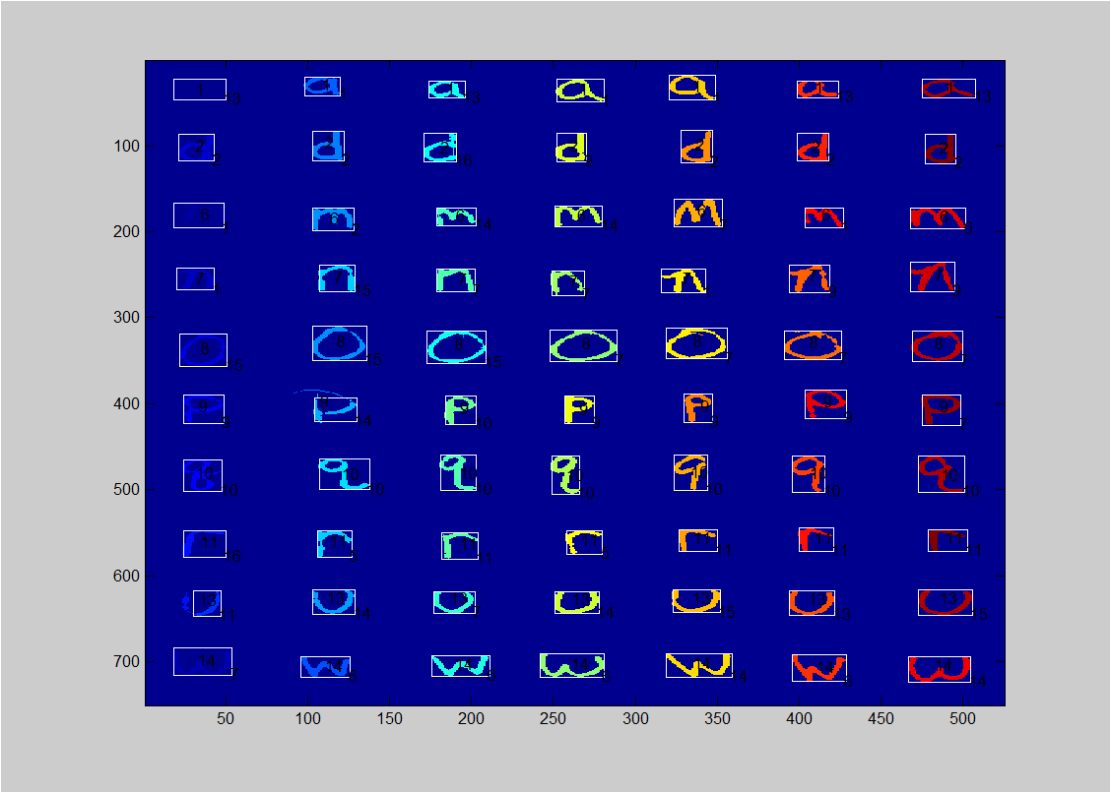
The overall recognition rate for training image is 62.69%



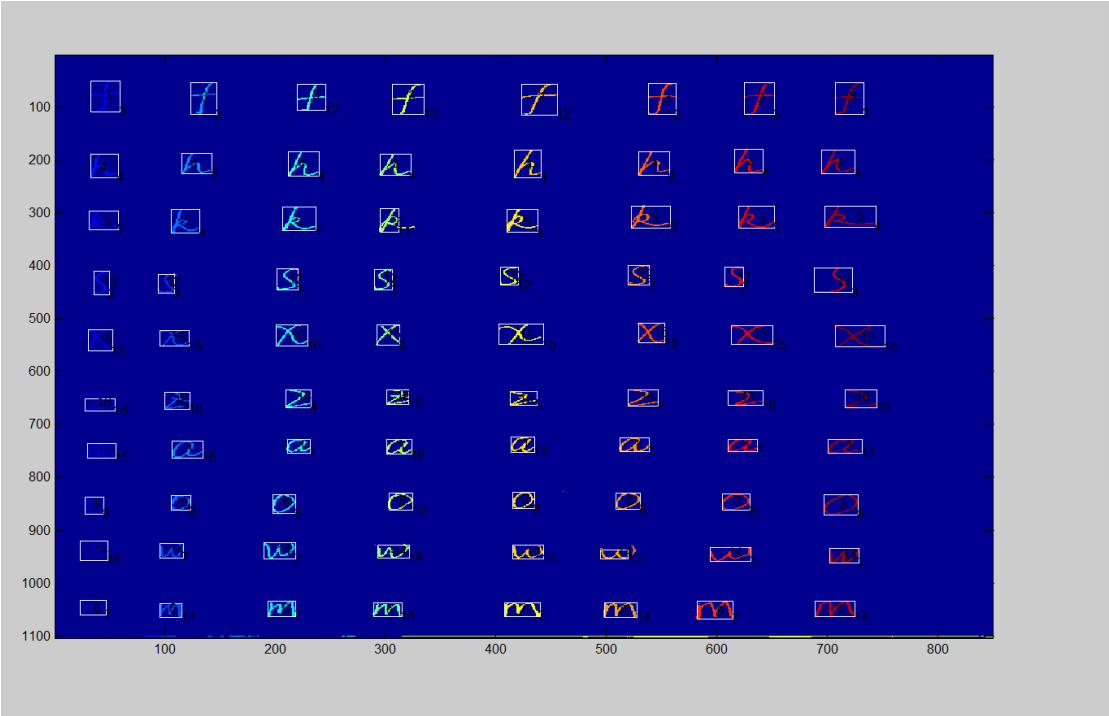


2, For Test data, the rate is

The rate of Test1 is 0.4143



The rate of Test2 is 43.75%



2,Enhancement Techniques used :

THRESHOLDING:

Correctly recognizedcharacters	test1.bmp (70 characters)	test2.bmp (80 characters)
threshold = 200 (original)	29	35
threshold = 176	20	40
threshold = 210	35	34
threshold = 215	36	39
Using Otsumethod	34	29

At first, I tried to find a best threshold. But I quickly realized the entries of the confusion matrix were a ratio. So while I was able to decrease the overall confusion by changing the threshold, I did not increase overall recognition accuracy.

After several attempt , my optimal threshold was 215.

I also tried using Otsu's method. Matlab has the in-built function (graythresh) which computes the threshold when any image is passed to it. But it doesn't work well.

Bounding Box

My condition of choosing bounding box is :

a = maxr-minr; b = maxc-minc; AreaThreshold = a*b;

(1)

AreaThreshold >120 for train data ,and AreaThreshold >150 for test data

To eliminate small rectangles

(2)

(a>10)&&(b>10)&&(b<120)&&(a<120)

Define the boundary of box

(3)

(a/b<5)&&(b/a<5)

Eliminate the long line.

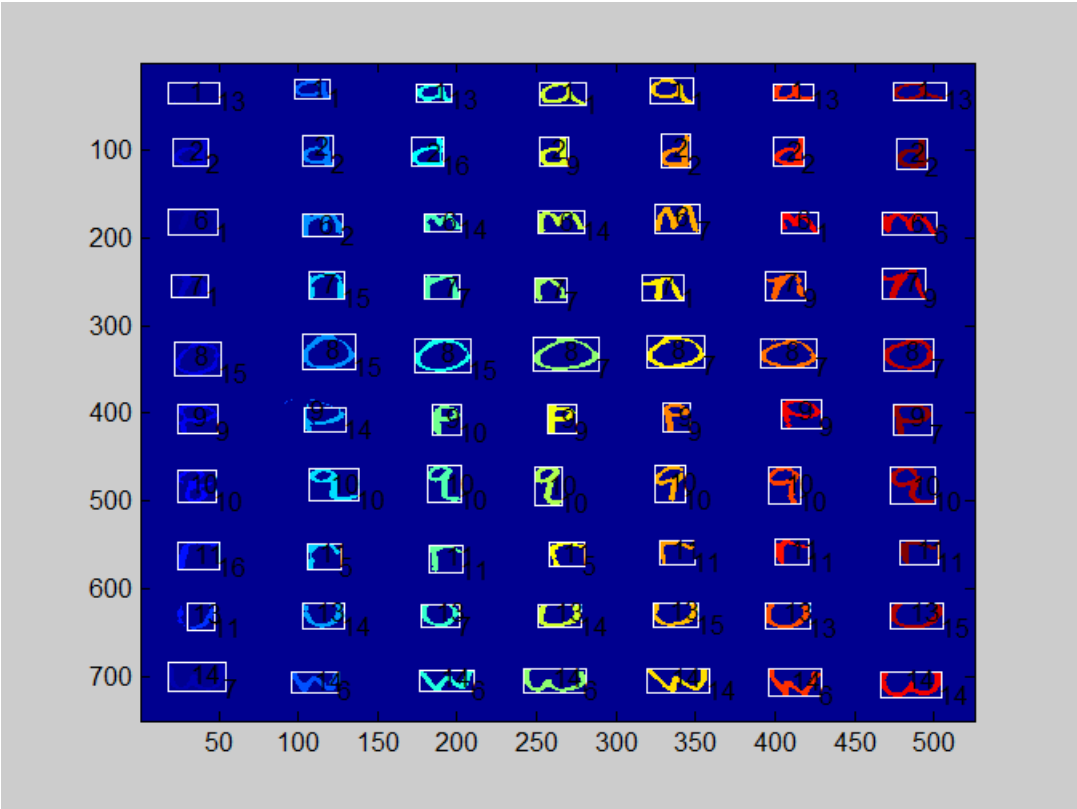
Nearest Neighbor Search:

During matching process, we can take ‘n’ nearest neighbors and compute the most frequently occurred value in those neighbors .

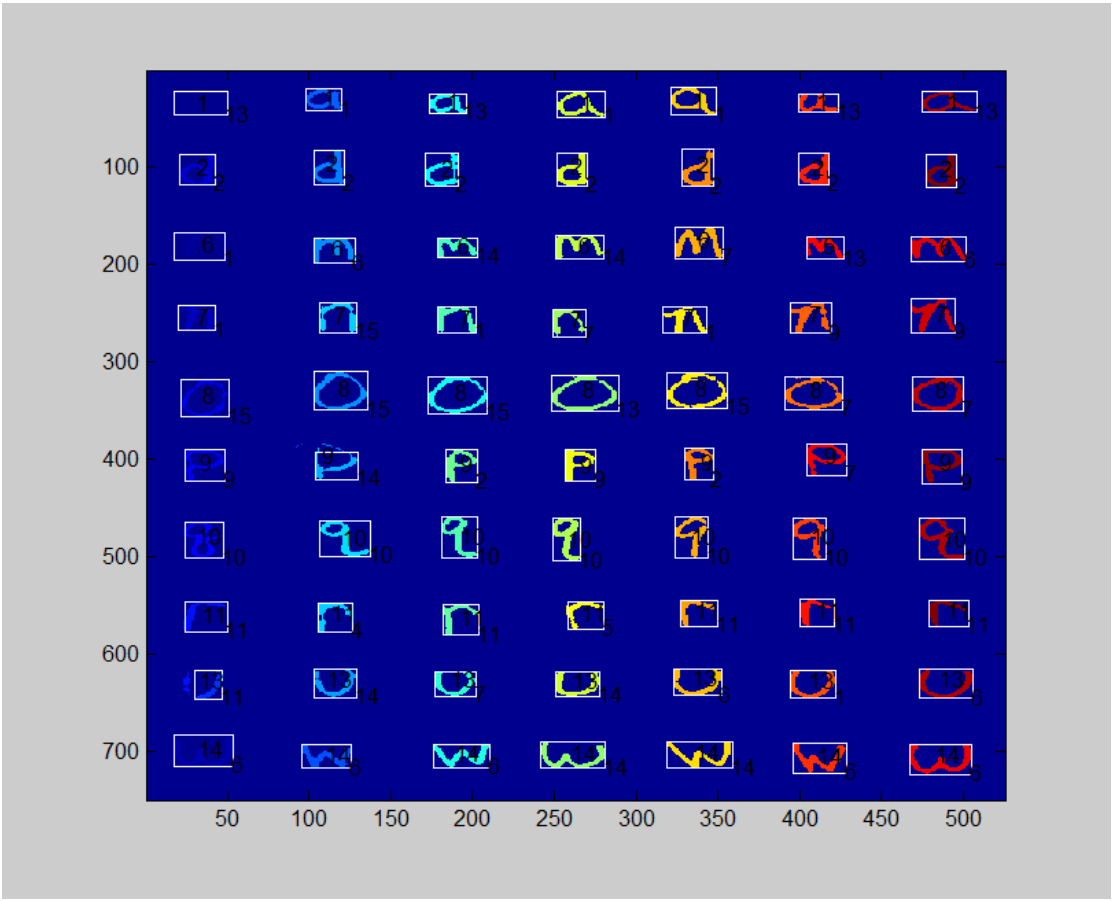
In the recognition rate for training images is around 62.69 percent while implementing this n-nearest neighbor with n=4,the recognition rate is around 78 percent.

Test1.bmp

Original: 0.4143.Before using this technique :

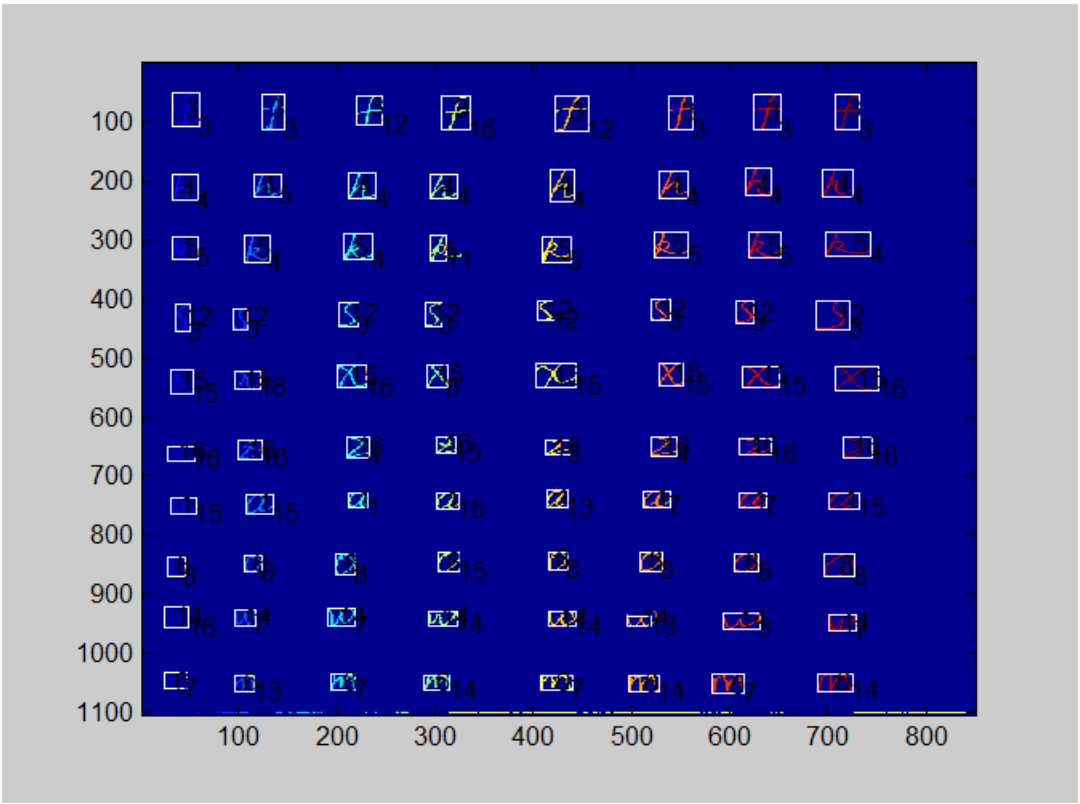


Test1 0.4286 After using this technique:

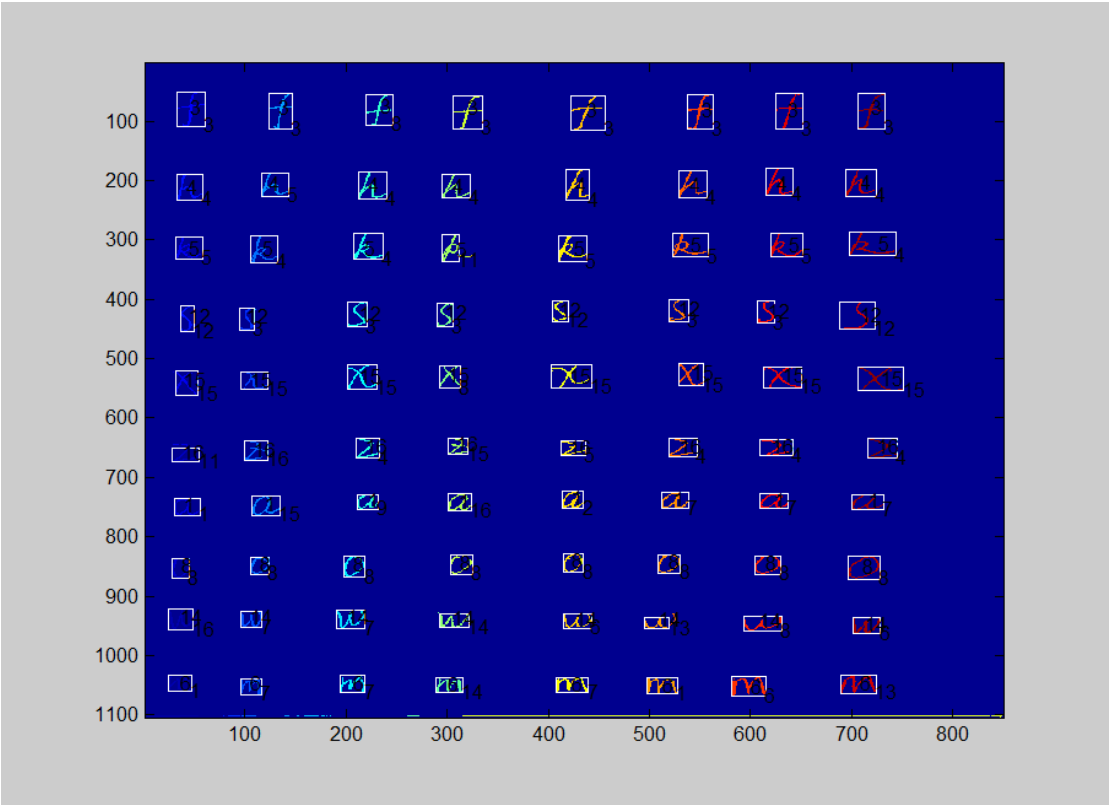


Test2.bmp

Original: 0.4375 Before using this technique :



Test2 0.500 After using this technique:

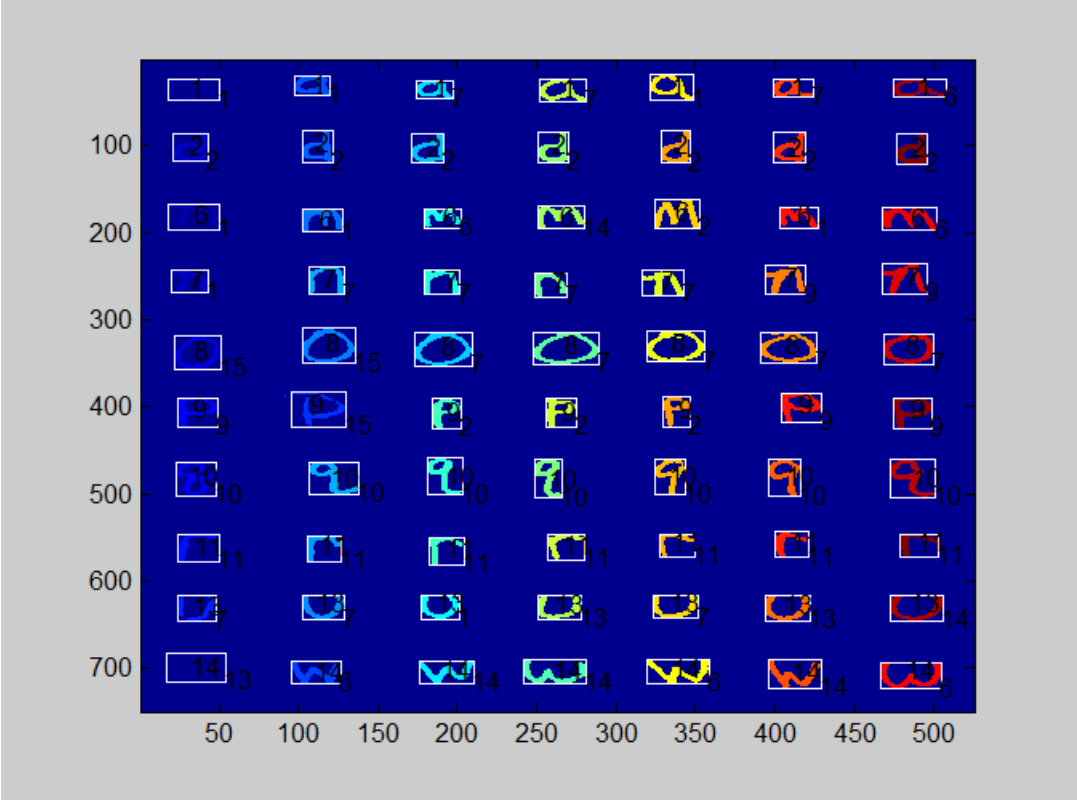


Combine all the enhancement method:

Use threshold = 215, boundingbox and Nearest Neighbor Search.

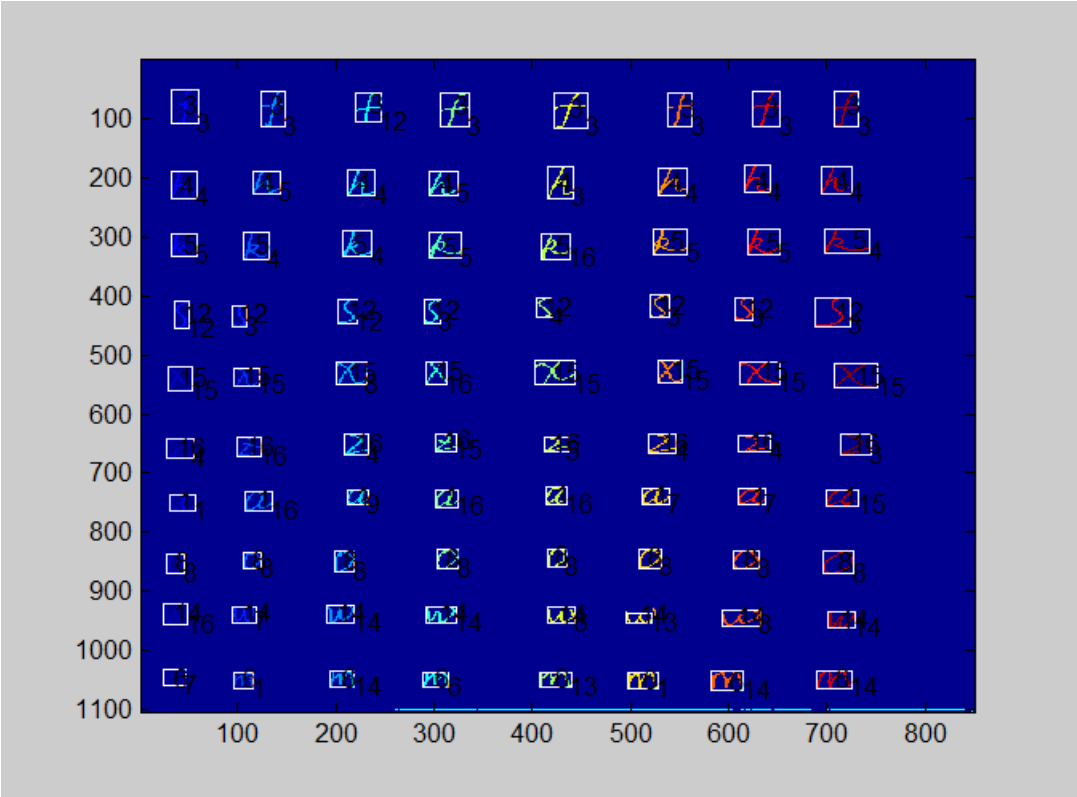
For test1.bmp:

Rate = 0.5429



For test2.bmp:

Rate = 0.4750



3, Running Code

The Location and class of test1.bmp and test2.bmp is stored in the file '**test1_class.mat**, **test1_ location.mat**, **test2_class.mat**, **test2_ location.mat**. load these data.
The main code is in the function file '**RunMyOCRRecognition.m**'. And the pictures need to put in the current folder.
Then run [trainRateArray,totaltrainRate,testRate] =RunMyOCRRecogiton(filepath, locations, classes)
Like in matlab
>>[trainRateArray,totaltrainRate,testRate] =RunMyOCRRecogiton('test2.bmp', Test1_Loc,Test1_Class)