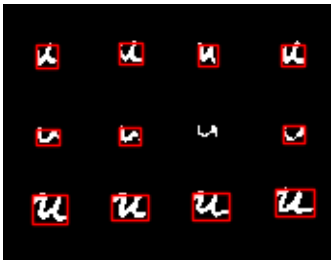
Assignment 1
Optical Character Recognition

Kshitij Shah – ks1223

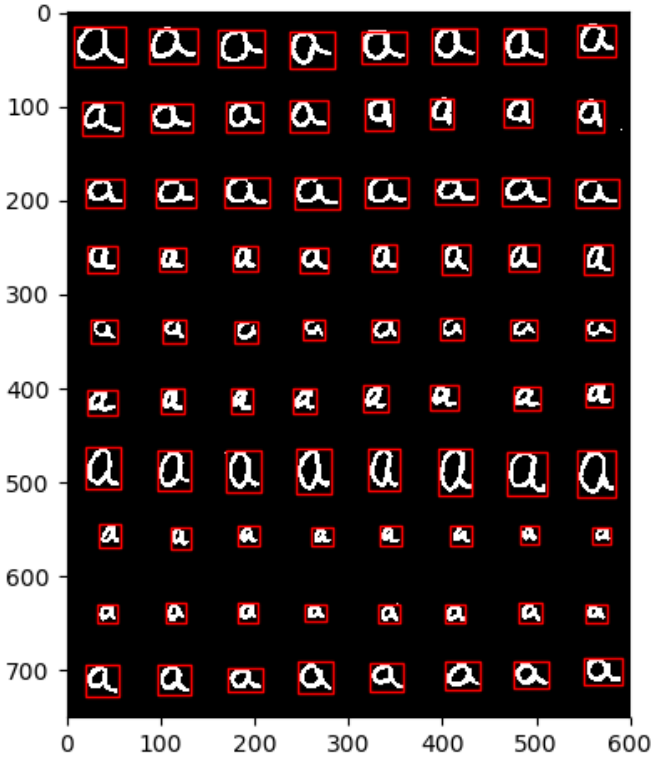
Training

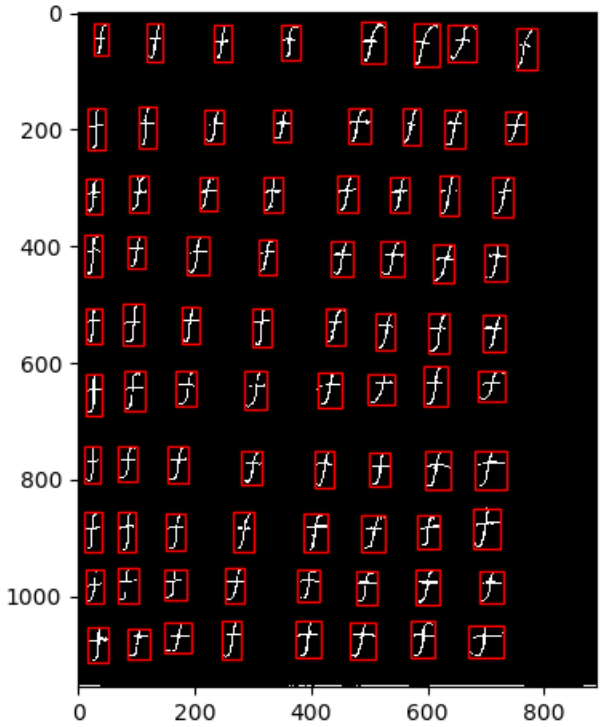
Connected Components

In basic settings with sizing threshold of 10, some characters were not recognized as shown in below image.

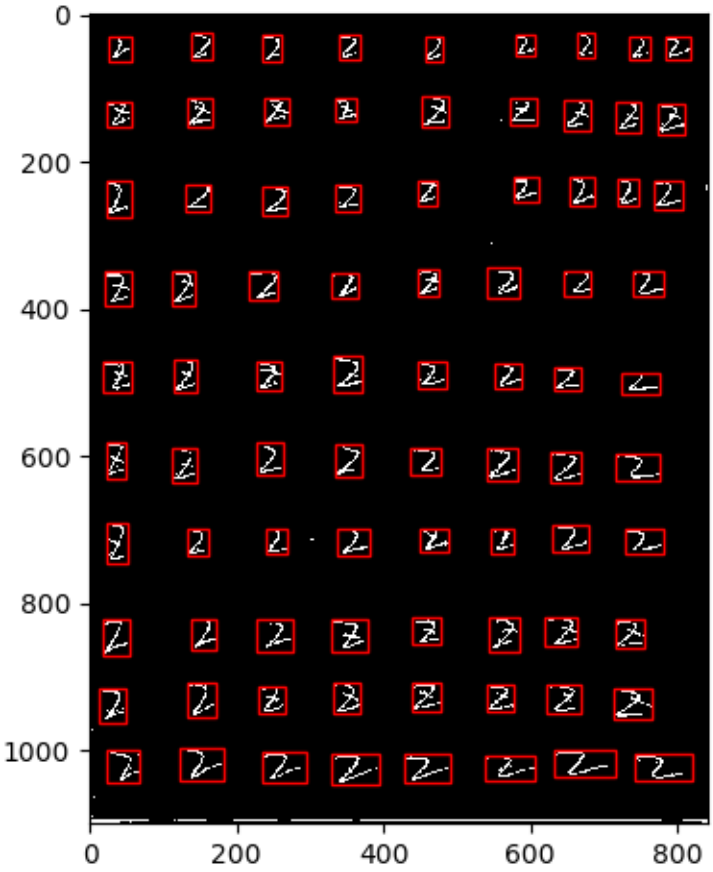


This problem was fixed with dilation and dynamic thresholding using Yen's algorithm. Here are the sample images with bounding box using enhancements. All the 16 images can be found in bounding_box.zip attached with report.


0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

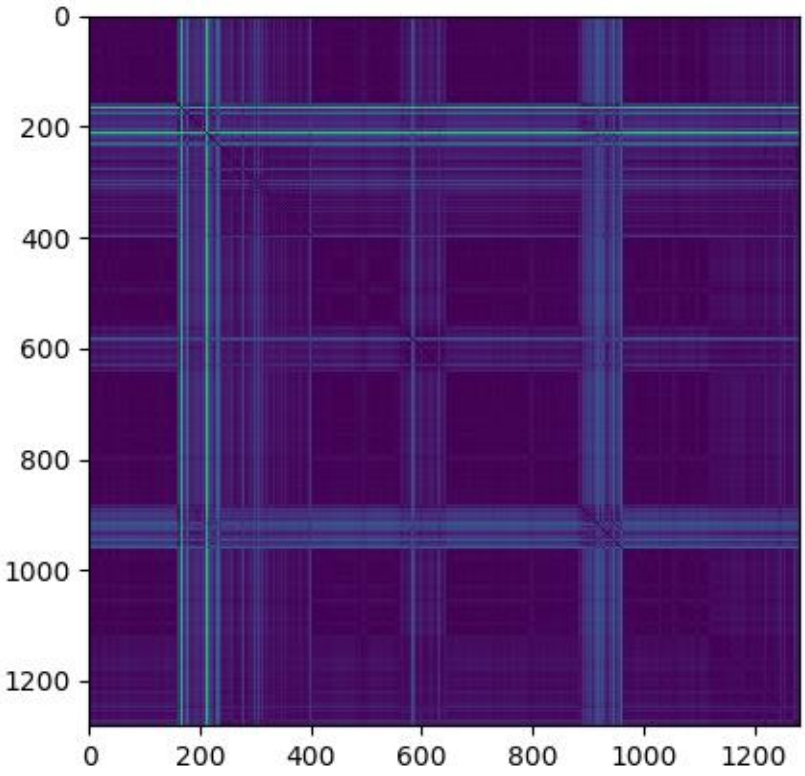

0
200
400
600
800
1000
0
200
400
600
800


0
100
200
300
400
500
600
700
0
100
200
300
400
500
600

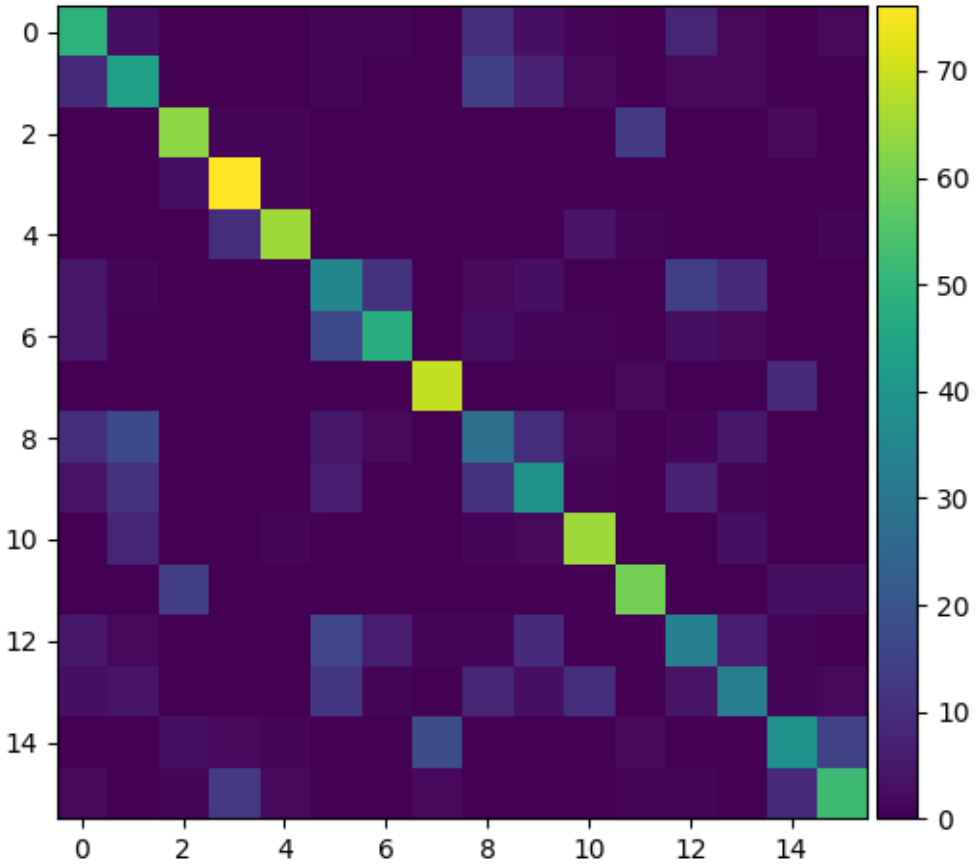

0
200
400
600
800
1000
0
200
400
600
800

Recognition Phase

Distance Matrix



Confusion Matrix



Training data analysis

Class	Correct Instances	Total Instances	Percentage
a	49	80	61.25
d	43	80	53.75
f	63	80	78.75
h	76	80	95
k	65	80	81.25
m	35	80	43.75
n	48	80	60
o	69	80	86.25
p	28	80	35
q	39	80	48.75
r	65	80	81.25
s	60	80	75
u	33	80	41.25
w	32	80	40
x	39	80	48.75
z	52	84	61.9047619
Total	796	1284	61.9937695

Recognition Phase

Basic

Basic method without threshold for bounding box was detecting 1448 components in training data instead of 1284. This results are not taken in to consideration. After applying the minimum and maximum threshold for the size of box number of components detected on training data became 1281. On both the test images number of components were correct.

Instead of too many different enhancement I tried to keep it simple. The profiles are there. Basic, Enhanced-which combines multiple enhancements optimized for better result and Enhanced with filter descriptor which has really good improvement.

	Morphology	**Threshold**	**Descriptor**	**Selector**
Basic	None	200	Hu's moments	Nearest match
Enhanced	Partial Dilation	Yen's method	Hu's moments	Weighted k neighborhood match
Enhanced with Filter	Dilation	Yen's method	Response Filter	Convolution

	Training Recognition Rate	**Training Number of Component**	**Test 1 RR**	**Test 1 NoC**	**Test 2 RR**	**Test 2 NoC**
Basic	61.43%	1281	47.14%	70	47.5%	70
Enhanced	61.99 %	1284	51.42%	70	51.25%	70
Enhanced with Filter	N.A.	1284	82.86%	70	93.75%	70

Enhancements

With enhancement, I noticed that, it was not like after each enhancement the result improved. Some enhancement improved results individually, but combined together worsened it.

Here are the four areas in which enhancements were tried.

1. Thresholding
2. Morphology
3. Classifier
4. Descriptor

1. Thresholding

After trying various fixed values I found 200 as the best threshold. However, performance was improved further by finding threshold by Yen's algorithm.

2. Morphology

After trying dilation, erosion, opening and closing as expected only dilation and closing had positive effect. The improvement with closing was negligible and was abandoned in favor of dilation. Dilation is done using a disk of radius 3.

3. Classifier

Nearest match was already giving good results. With most matches k-nearest neighbors reduced performance. Ultimately, weighted k-nearest neighbors with decreasing weights improved performance slightly.

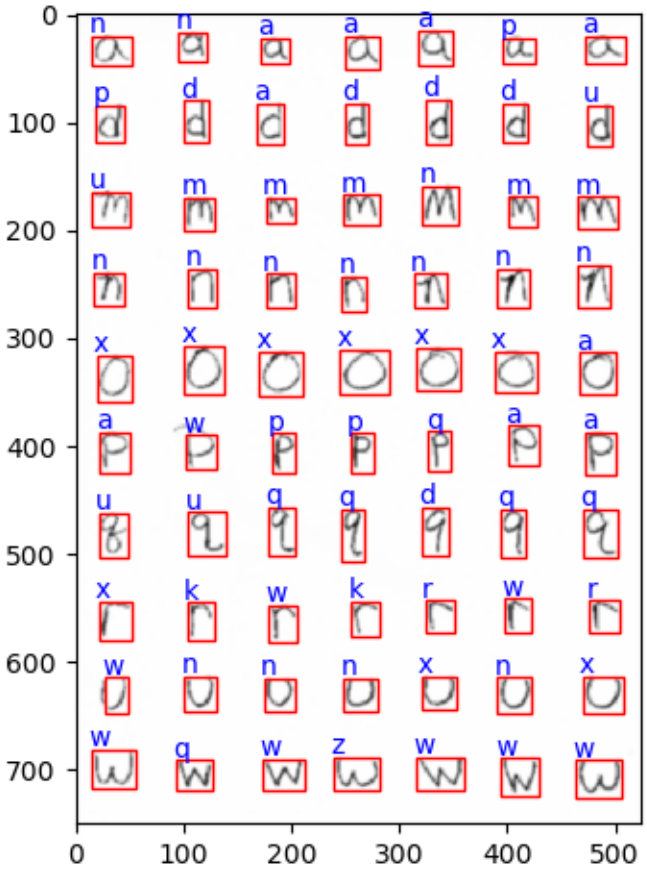
4. Descriptor

I couldn't find many rotation invariant descriptors. After that rotation variant descriptor were also considered. Normalized moments up to order 3 didn't improve performance. The most successful enhancement was the response filters (like convolution filters) which is trained from training data. The filters use only and only training data and are trained at runtime only. This shot of up the recognition rate dramatically on test images. No pretrained models are used. The description of training is in following section.

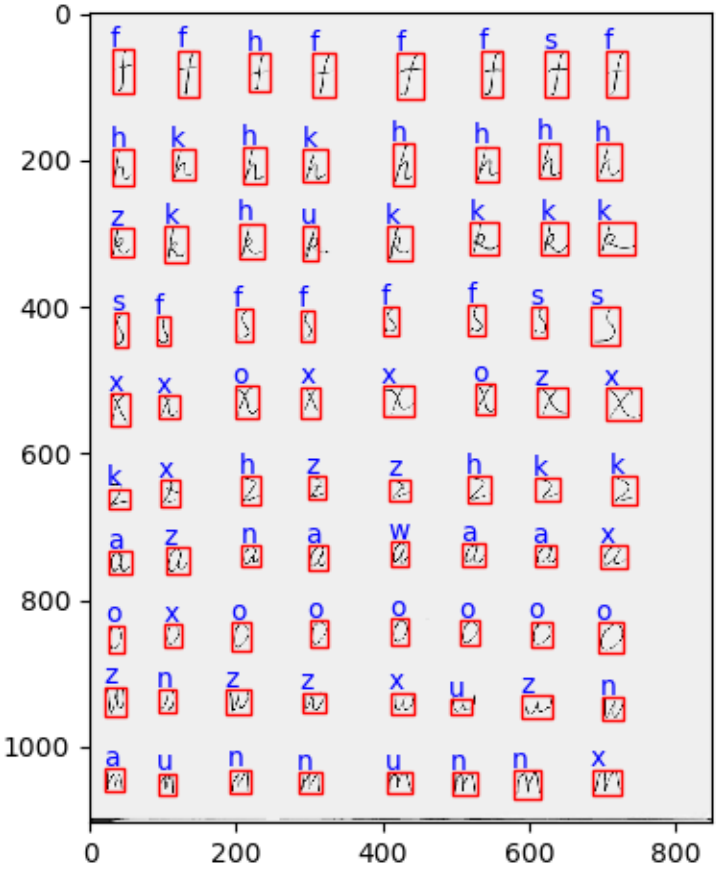
Results and Distance Matrix

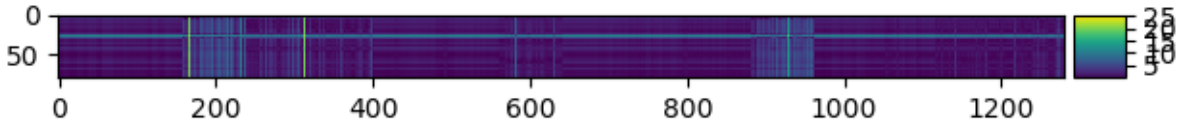
Basic

Test 1



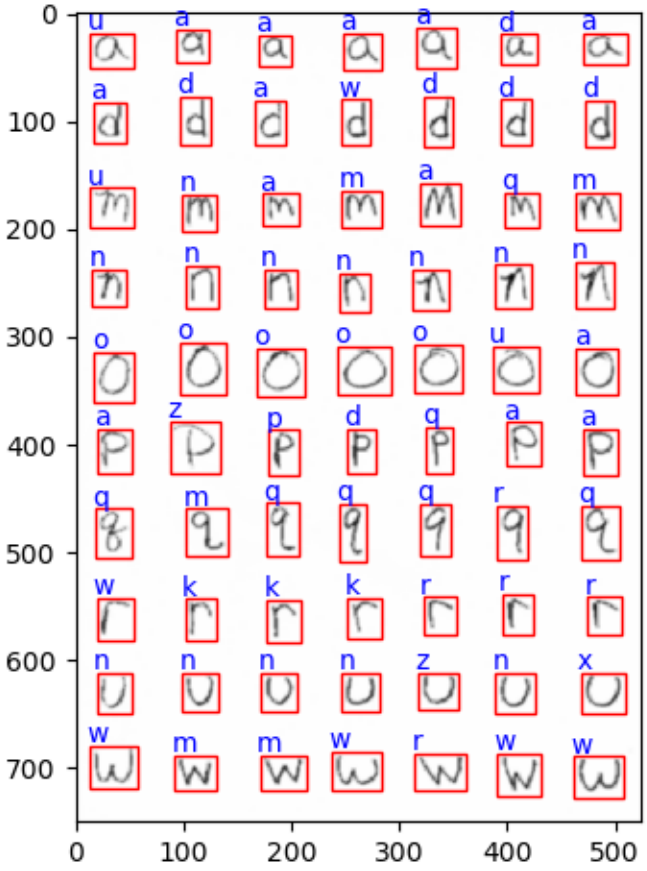
Test 2


0
200
400
600
800
1000
0
200
400
600
800

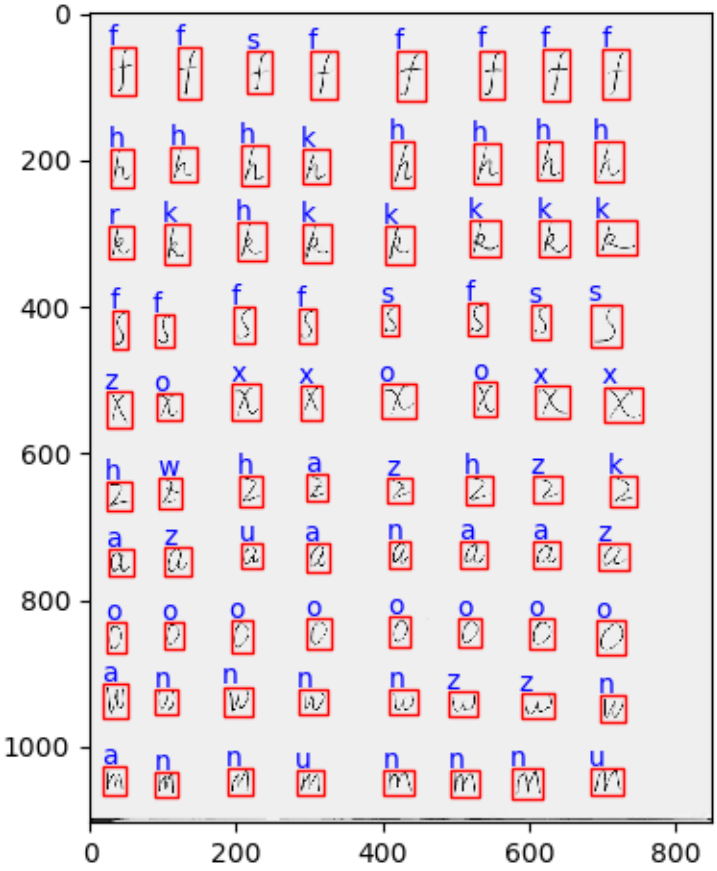

0
50
0
200
400
600
800
1000
1200
25
20
15
10
5

Enhanced

Test 1



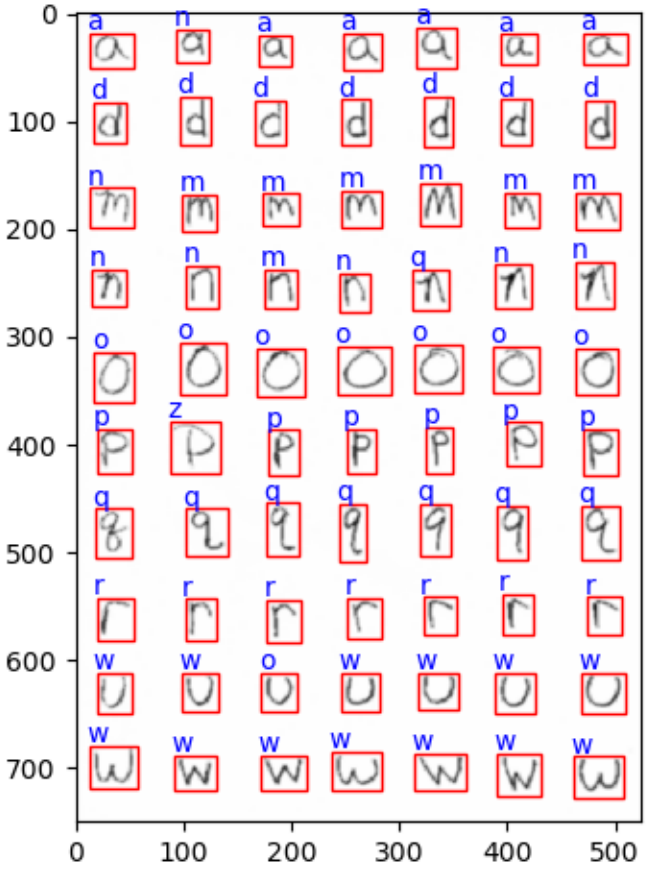
Test 2


0
200
400
600
800
1000
0
200
400
600
800

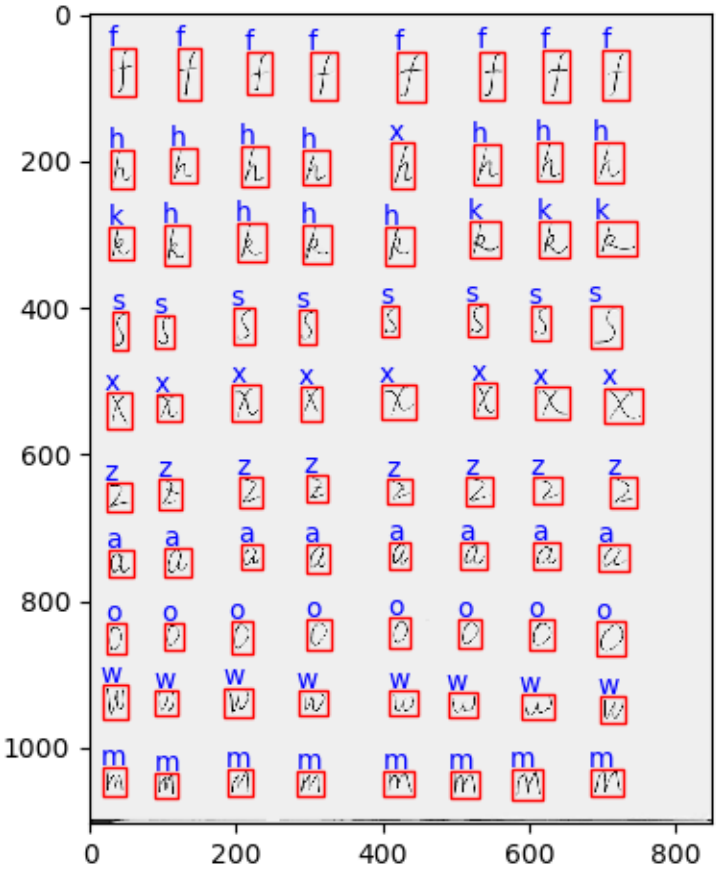

0
50
0
200
400
600
800
1000
1200
40
30
20
10

Enhanced with Filter

Test 1



Test 2



Response Filter

Response filters were trained with using very primitive method that it cannot be even called proper filters. Here are the steps for training. There would be one filter for every class.

For very class

Initialize $n \times n$ zero matrix.

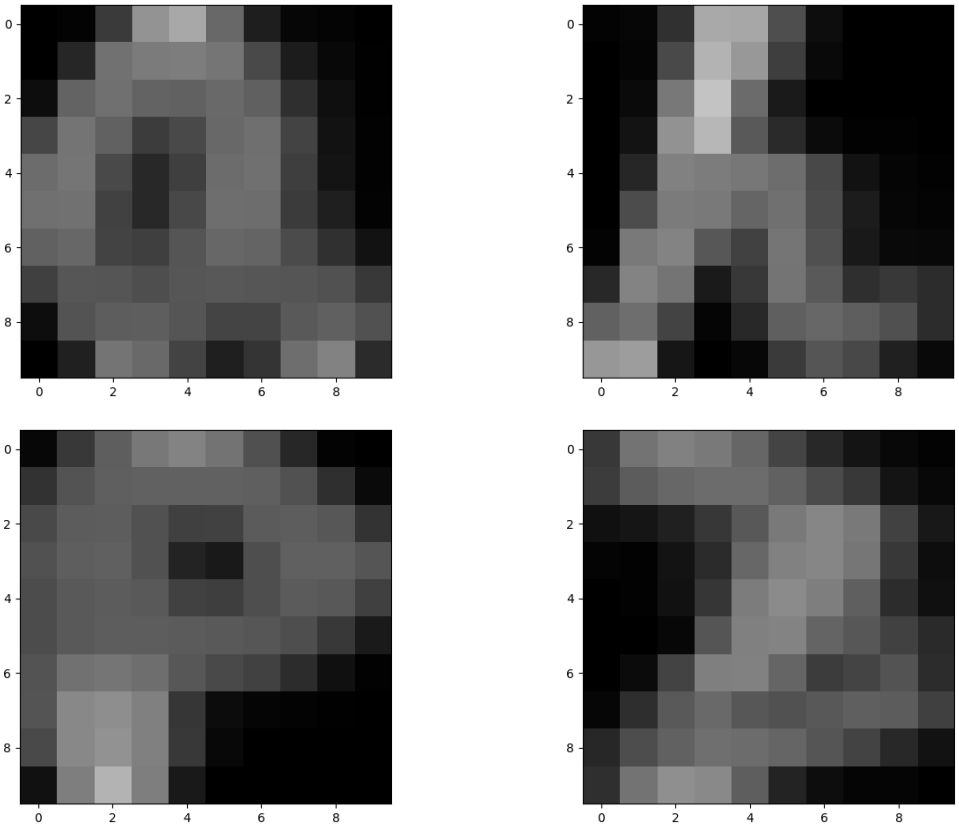
For every component

Resize component to $n \times n$.

Add the component with matrix.

Normalize the matrix

Sample Filters



Above are the filters for a, h, z, p (clockwise from above)

Size of the filter is set to 10 as increasing it further did not improve recognition rate on test images. However, if the result is deteriorating it can be increased from setting.py -> filter_size

Issues faced

Rotation Invariance

It was very hard to find good rotation invariant descriptors. The problem with rotation invariance is characters like 'M' and 'W'. Most of the instances of 'w' are recognized incorrect with Hu's moments in my results. Even human can't distinguish between 'M' and rotated 'W' without context.

Lack of uppercase 'U' on training data

There was no uppercase 'U' on training data, while they were there on tests. Resulting in almost 0 percent recognition with every method.

Limitations of the program

1. Program will perform poorly on rotated characters with filters at least.
2. It is assumed that characters will not exceed 100 pixels in size, no character in the training or testing image exceeds 40 pixels. If that is the case, a parameter can be set in the program to increase that limit.
3. Not tested on color text.

Misc.

Environment Details

Complier: Anaconda Python 2.7

Platform: Windows 10

IDE: PyCharm 2016

Tested on CentOS at prototype.cs.rutgers.edu

Annexure A

Performance of enhanced filter method on training data

The filters are themselves trained using all the characters in the training data. Hence, I am reluctant to make claims based on training data recognition rate. Therefor filter results for training data is not mentioned in the report.

However, the recognition rate of filter method on training data is reported 91.2%.

Following is the breakdown table for each class.

Class	Correct	Total	Percentage
a	76	80	95
d	73	80	91.25
f	76	80	95
h	69	80	86.25
k	70	81	87.5
m	78	80	97.5
n	67	80	83.75
o	78	80	97.5
p	74	80	92.5
q	65	80	81.25
r	70	80	87.5
s	73	80	91.25
u	71	80	88.75
w	79	80	98.75
x	74	80	92.5
z	78	83	92.8571429
Total	1171	1284	91.1993769