CS534 : Computer Vision
Optical Character Recognition

Chaitanya Mitash (cm1074)

I. Basic Framework

The basic framework for OCR was implemented based on the assignment description, which uses the following parameters.

- **thresholding** : manually set threshold (th = 150) was used for background removal.
- **denoising** : removal of small components (components with < 10 pixel width/height are removed)
- **features** : hu moments (each of the 7 components are normalized).

The feature dataset was created with 16 training images consisting of 80 character instances each. Following were the observations with the training set.

- **imprecise thresholding** leads to several isolated components. Thus, the number of components computed in the training image is more than the 80 character instances present. In worst case, it was found to be 144 components for image with character 'o' when using the default threshold of 150.
- **denoising**, needs another threshold parameter as in the minimum allowed width and height of the component. We use 10 pixels as the default value, which reduces the number of components to the range (66-86). The trade off here is between removing noise and some of the actual small components.
- **training set is evaluated** using the computed feature database. For each feature vector, it's nearest neighbor (other than itself) is found and compared to it's original label. This was correct **48.6%** times corresponding to 1265 labeled components. The distance matrix and confusion matrix are plotted in figure.
- **evaluation of the test images** is performed using the nearest neighbor classifier. The results are tabulated in figure.

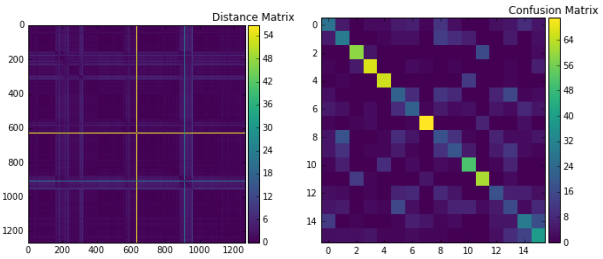


Fig. 1: the distance matrix and confusion matrix for training images. The diagonal in distance matrix represents the distance of feature vector with itself while in confusion matrix the diagonal represents the instances when labels are identified correctly.

image	thres	success	numComp
test1	150	18.57	72
test2	150	46.25	81
test1	200	24.28	70
test2	200	51.25	80

Fig. 2: Recognition rate on test images

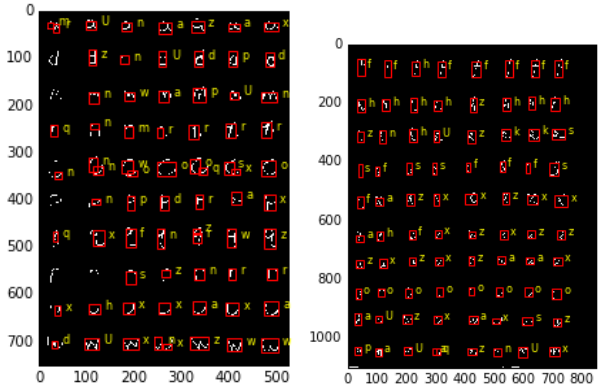


Fig. 3: evaluation on test images.

the results indicate several weakness in thresholding, feature learning and classification. By manually tuning the threshold, we observe an improvement in performance.

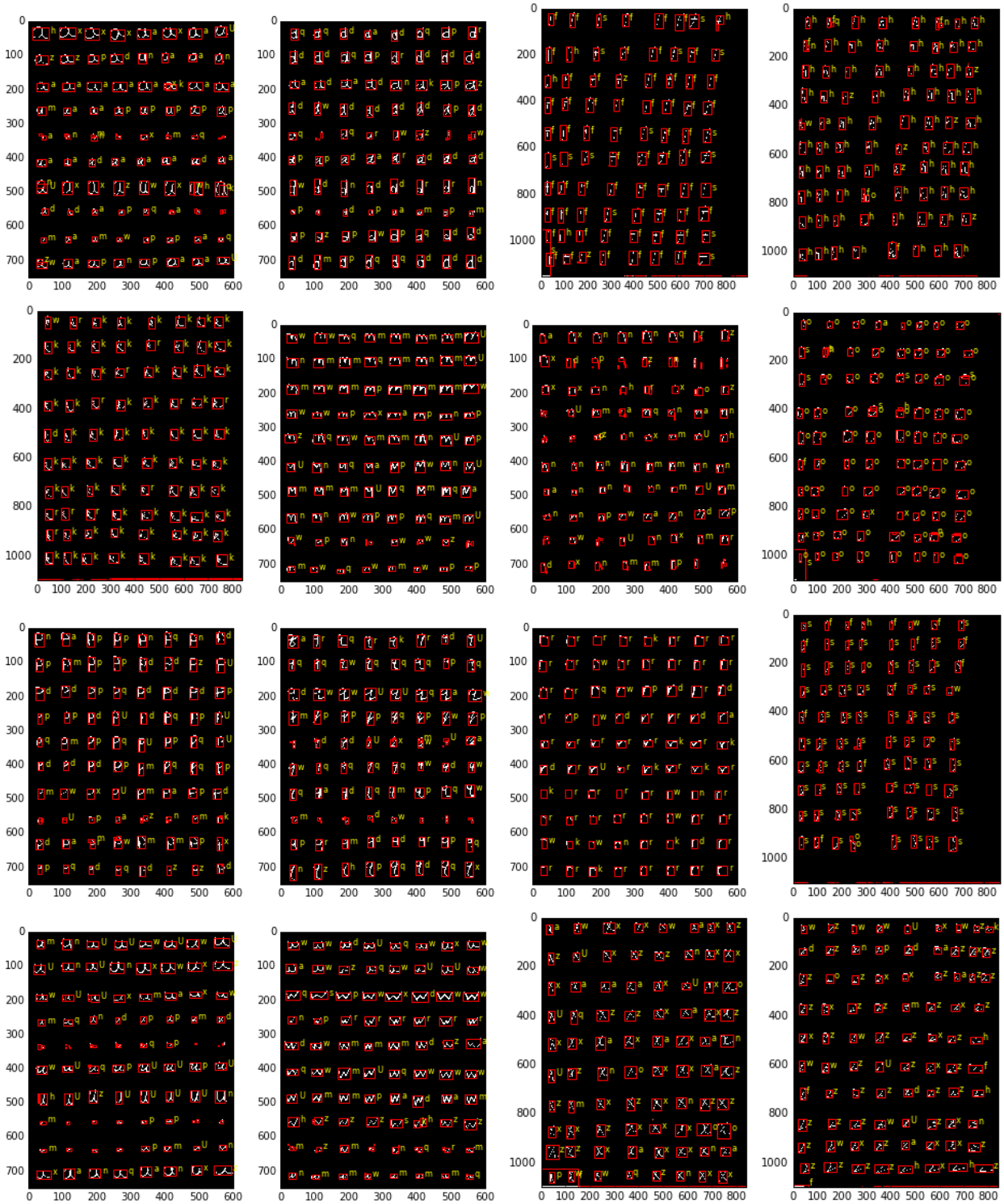


Fig. 4: Evaluating the training dataset

II. ENHANCEMENTS

Several enhancements are attempted, and the observations are recorded.

- **Thresholding** : Several global and local thresholding methods were tested, namely **Otsu, Li, Yen and adaptive thresholding**. From the experiments, it was observed that Otsu's thresholding performed the best on average when extended with different features and better classifier. The other interesting choice was local or adaptive thresholding which, combined with morphological operations like closing gives good results 5

image	method	success	numComp
trainset	Otsu	55.13	1295
test1	Otsu	32.85	70
test2	Otsu	41.25	81
trainset	li	51.42	1295
test1	li	28.57	70
test2	li	42.50	81
trainset	yen	47.05	1288
test1	yen	30.00	70
test2	yen	51.25	80
trainset	adaptive	45.50	1325
test1	adaptive	31.42	70
test2	adaptive	38.75	80

Fig. 5: Recognition rates with thresholding methods

- **Morphological operations** : The thresholding can lead to several small broken components, thus **gaussian blur** was used to average out the neighborhoods and morphological operation of **CLOSING** was used, which is a combination of dilation and erosion, used to fill the gaps between broken components. These operations enhance the performance as presented in 6
- **Classification** The k-nearest neighbor approach was used to find the label with maximum counts amongst the top k predictions. Different values of k was used and the best performance was achieved, with k=5. The results in 7 use this classifier.
- **Features** : The basic implementation used the hu moments computed from the region of interest for each segmented character. Adding to this, several geometric features like **eccentricity, convex density, orientation, euler number and bounding box ratio** improved the performance as in 7. Most significant contribution was when using the ratio of convex hull of the character to the area of the bounding rectangle. This feature in a way represents the shape of the characters and is less effected by broken segments.

image	method	success	numComp
trainset	gaussian	54.73	1288
test1	blur	35.71	70
test2	(otsu)	51.25	80
trainset	gaussian	50.77	1294
test1	blur	34.20	70
test2	(adap)	45.00	81
trainset	gaussian	55.52	1286
test1	blur,closing	32.85	70
test2	(otsu)	53.75	80
trainset	gaussian	51.20	1293
test1	blur,closing	44.28	70
test2	(adap)	45.00	80

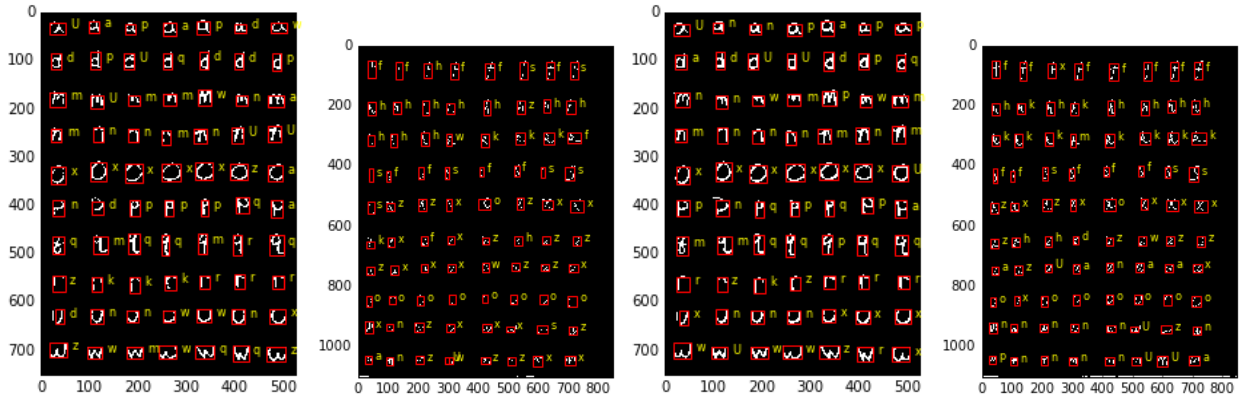
Fig. 6: Recognition rate with morphological operations

image	method	success
trainset	Bbox ratio	72.93
test1		35.71
test2		55.00
trainset	eccentricity	75.81
test1		38.57
test2		53.75
trainset	convex density	79.39
test1		41.42
test2		70.00
trainset	orientation	82.81
test1		58.57
test2		63.75
trainset	euler number	84.44
test1		62.85
test2		70.00
trainset	all features (adap)	84.43
test1		62.85
test2		78.75

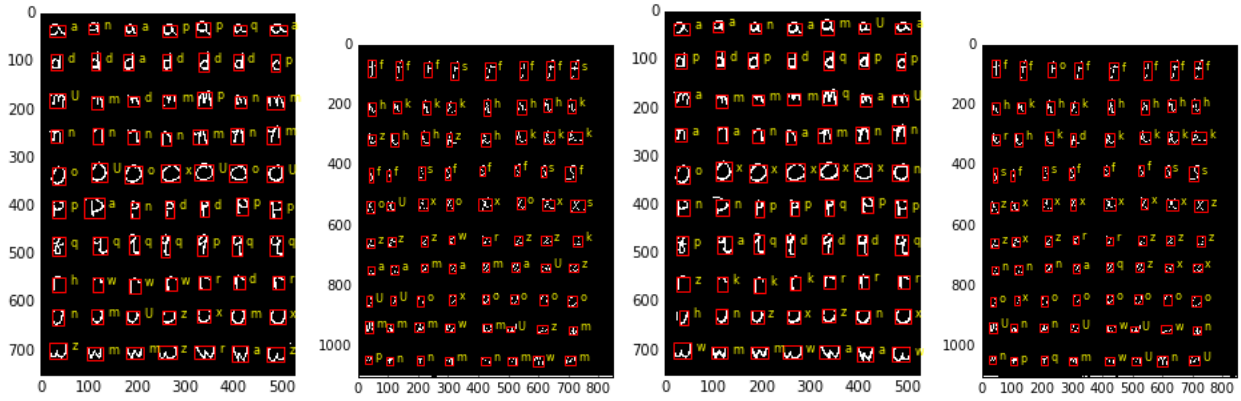
Fig. 7: Recognition rate with adding features

III. BEST PERFORMANCE

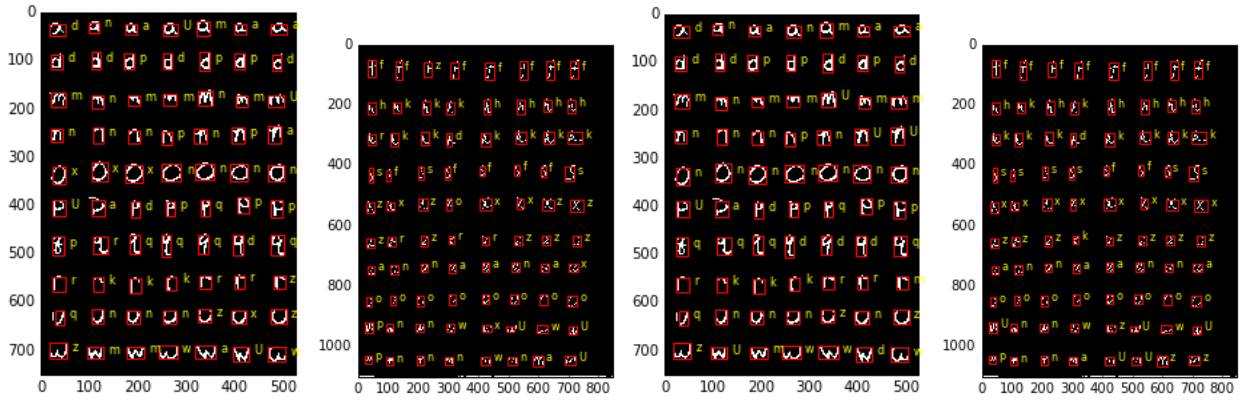
The best performance was achieved by using gaussian blur followed by adaptive thresholding, and then using the closing operation with (neighborhood = 3). All the features mentioned above were used and KNN classifier



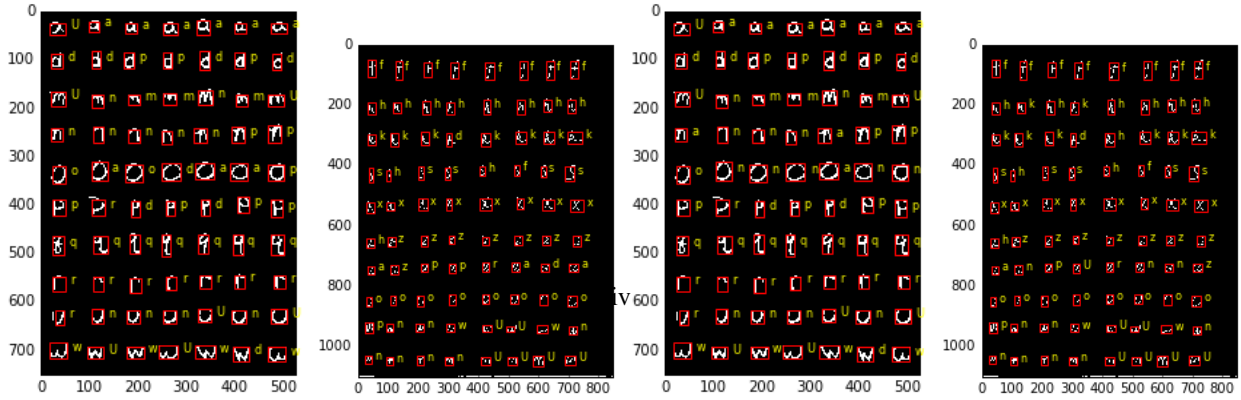
(a) test1: Otsu thresholding **(b)** test2: Otsu thresholding **(c)** test1: gaussian blur, closing with Otsu **(d)** test2: gaussian blur, closing with Otsu



(e) test1: gaussian blur, closing with adaptive **(f)** test2: gaussian blur, closing with adaptive **(g)** test1: bbox ratio feature **(h)** test2: bbox ratio feature



(i) test1: eccentricity feature **(j)** test2: eccentricity feature **(k)** test1: convex density feature **(l)** test2: convex density feature



(m) test1: euler number feature **(n)** test2: euler number feature **(o)** test1: orientation feature **(p)** test2: orientation feature

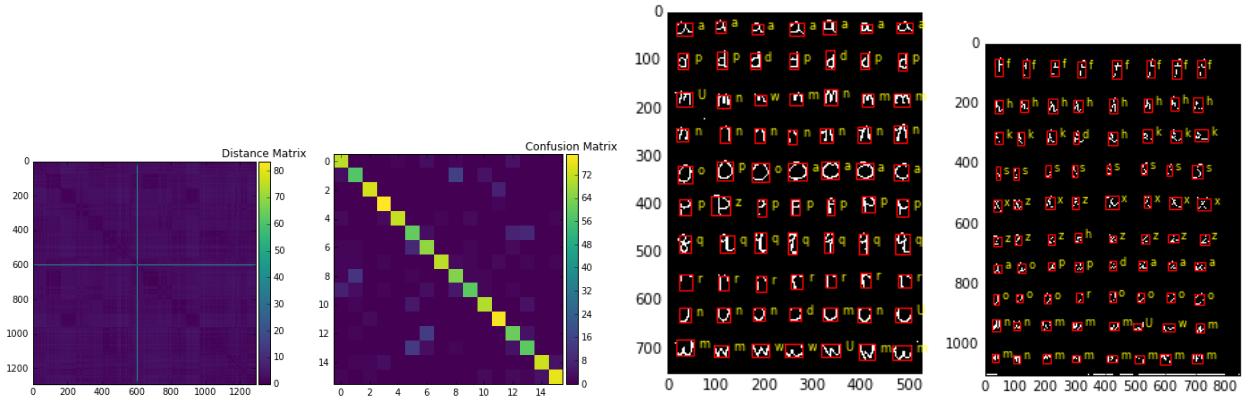


Fig. 9: Final results on training and test data

was used with k=5. The training and test plots corresponding to this result is 9

IV. Notes

- The binary images were used to display results to better show the effect of thresholding and morphological operations instead of using the connected component images.
- opencv and skimage libraries were used for evaluating different enhancements.
- To run the code with different test image, you should add the test image and ground truth file names in the file **RunMyOCRRecognition.py**
- Loading the pickle file might cause an error due to python version mismatch. I used the following API.

  $mydict = pickle.load(pkl\_file, encoding =' latin1')$

  In case this does not work, remove the encoding specification and it should run.