Handwritten Character Recognition

Status Report 2

Patrick Rippl, CM 3353

Brandon Rudolph, CM 3359

Chelsey Yin, CM 2040

Kevin Wang, CM 1996

CSSE – 463 – 01

Project Report #2

Due: Sunday, 02/02/2018

2/2/18

Contents

1 Achievements

We were able to implement two functions: one for training a CNN and one for segmentation of single line handwritten words.

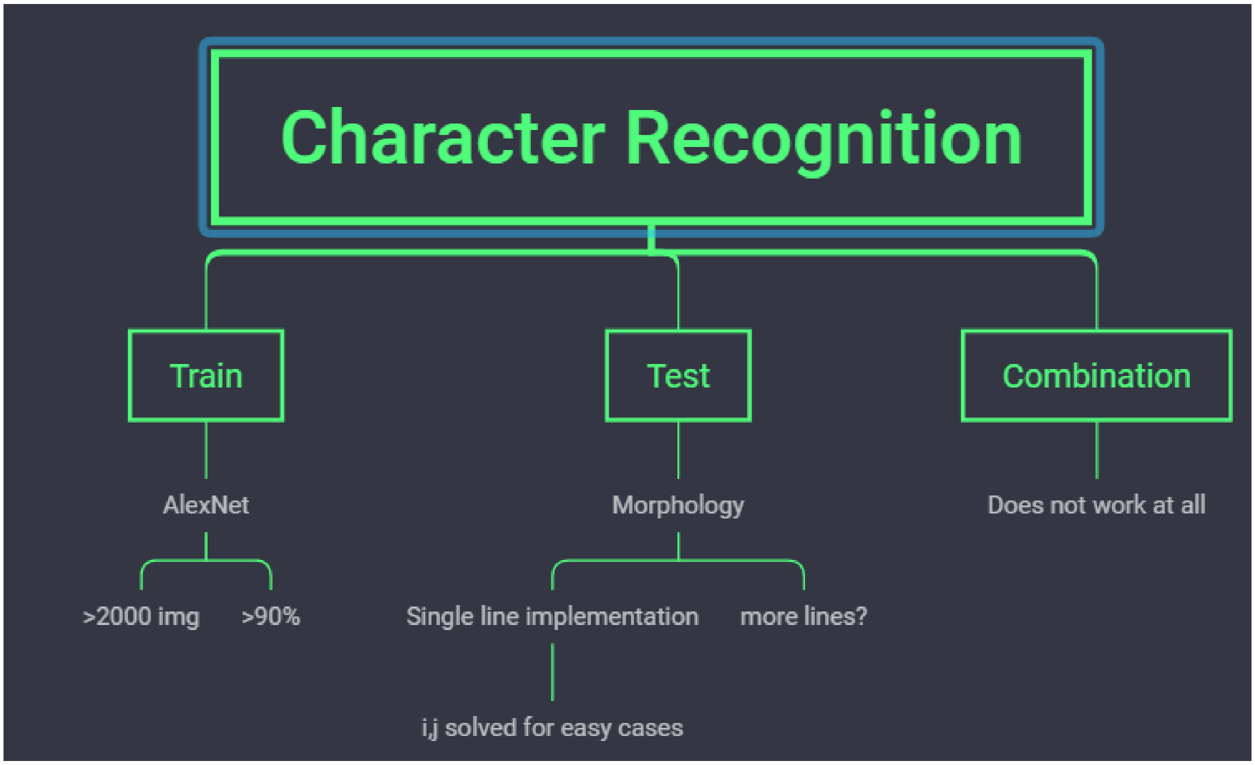

Figure 1 Results so far

1.1 Train set for character recognition

To approach our handwritten recognition, we train a CNN (AlexNet) to recognize handwriting with a dataset of ~2500 images of single characters, such as depicted in Table 1. We were able to reach an accuracy of 93% on this data set.

	Labels		Labels		Labels
0	Sample01	**A**	Sample11	**a**	Sample37
1	Sample02	**B**	Sample12	**b**	Sample38
2	Sample03	**C**	Sample13	**c**	Sample39
3	Sample04	**D**	Sample14	**d**	Sample40
4	Sample05	**E**	Sample15	**e**	Sample41
5	Sample06	**F**	Sample16	**f**	Sample42
6	Sample07	**G**	Sample17	**g**	Sample43
7	Sample08	**H**	Sample18	**h**	Sample44
8	Sample09	**I**	Sample19	**i**	Sample45
9	Sample10	**J**	Sample20	**j**	Sample46
		K	Sample21	**k**	Sample47
		L	Sample22	**l**	Sample48
		M	Sample23	**m**	Sample49
		N	Sample24	**n**	Sample50
		O	Sample25	**o**	Sample51
		P	Sample26	**p**	Sample52
		Q	Sample27	**q**	Sample53
		R	Sample28	**r**	Sample54
		S	Sample29	**s**	Sample55
		T	Sample30	**t**	Sample56
		U	Sample31	**u**	Sample57
		V	Sample32	**v**	Sample58
		W	Sample33	**w**	Sample59
		X	Sample34	**x**	Sample60
		Y	Sample35	**y**	Sample61
		Z	Sample36	**z**	Sample62

Table 1 Sample Train Character Data

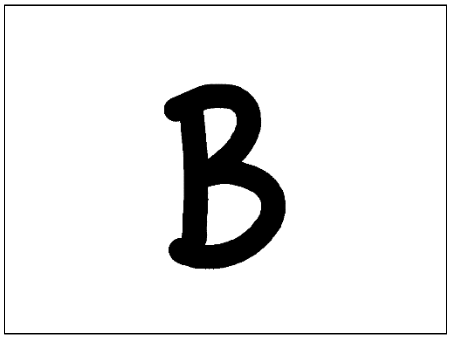

Figure 2 Example of Train Data

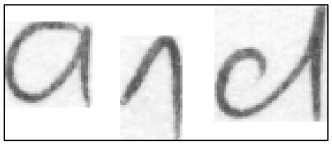

Figure 3 Example of handwritten test data

1.2 Morphology method function for test data set

The test data is handwritten words, which are not clearly separable. We first converted the image into a grayscale image and then binarized the image. We threw out the parts that has less than a hard-programmed threshold of 30 pixels. We used $bwlabel$ to different each object and use $regionprops$ to find their bounding box. Then we can crop each letter out according to the box.

For letter i and j, we go over each column to see if there are any parts of different letters in the same line. Since we can only recognize words where every letter is clearly separated, we have to take care of situations where the dot of is i and j is considered. Therefore, we combined single pixels above recognized pixels to one character.

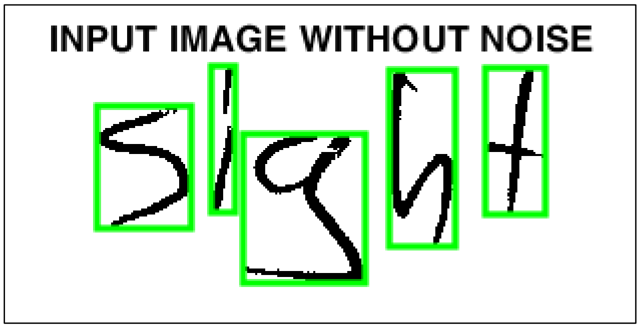

Figure 4 Output image of advanced segmentation method

Out of this segmentation, we changed the single letters from the bounding boxed into single images and saved them into separate files, so that we can then feed them into our trained CNN net.

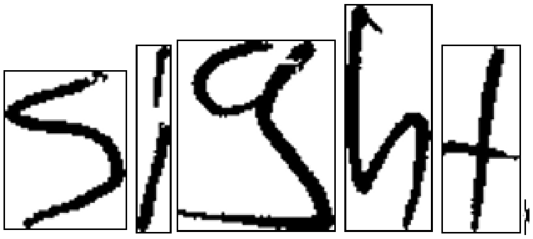

Figure 5 Single characters saved into separate .png files

2 Difficulties

The trained CNN provided an accuracy of approximately 90%. The single characters from above were fed into this recognition set, which did not perform well on the recognition at all.

3 Questions

We tried to classify a very simple word to see if the accuracy of our net works on a different test set as well. As mentioned, the accuracy is very low. One of the reasons might be that the size of the trained and tested data set images may affect the performance, even if the trained AlexNet should be scale invariant.

One approach for improvement might be to train our net on a bigger data set such that other wrong recognized characters will be fed into it as well. Furthermore, we thought about using a different net structure than AlexNet to increase the overall performance.