**Handwriting digit recognition**

Jiaxuan Zhang

Handwriting recognition (or HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. Alternatively, the movements of the pen tip may be sensed "on line", for example by a pen-based computer screen surface.



Handwriting number recognition is a challenging problem researchers had been research into this area for so long especially in the recent years. There are many fields concern with numbers, for example, checks in banks or recognizing numbers in car plates and the subject of digit recognition appears.

Handwriting recognition belongs to the scope of character recognition and pattern recognition. For types of the recognition process, the character recognition is divided into off-line identification and on line identification. For the identification object, it can be divided into a printed handwriting recognition, and identify two categories, we often say that the handwriting recognition refers to the on-line handwriting recognition.

Task: the input consists of black or white pixels, the digits are usually well-separated from the background, and there are only ten output categories. Yet the problem deals with objects in a real two-dimensional space and the mapping from image space to category space has both considerable regularity and considerable complexity. The problem has added attraction because it is of great practical value.

**Popular method:**

There are several classifiers:

Baseline linear classifier: possibly the simplest classifier that one might consider is a linear classifier. Each input pixel value contributes to a weighted sum for each output unit. The output unit with the highest sum (including the contribution of a bias constant) indicates the class of the input character, for this experiment, we used deslanted 20X20 images. The network has 4010 free parameters, the deficiencies of the linear classifier are well documented (Duda & Hart 73) and it is included here simply to form a basis of comparison for more sophisticate classifiers. The test error rate is 8.4%.

Baseline nearest neighbor classifier: Another simple classifier is a K-nearest neighbor classifier with a Euclidean distance measure between input and images. This classifier has the advantage that no training time, and no brain on the part of the designer, are required. However, the memory requirement and recognition time are large.

Pairwise linear classifier: A simple improvement of the basic linear classifier was tested (Guyon et al. 89). The idea is to train each unit of a single-layer network to classify one class from one other class. But the result of this is only slightly better than a linear classifier.

Principal component analysis and polynomial classifier: Following (Schurmann 78), a preprocessing stage was constructed which computes the projection of the input pattern on the 40 principal components of the set of training vectors. To compute the principal components, the mean of each input component was first computed and subtracted from the training vectors. This classifier can be seen as a linear classifier with 821 inputs, preceded by a module that computes all products of pairs of input variables. Error on the test set was 3.3%.

According to Salch, “Ernst Kussul and Tatiana Baidyk have efficiently developed a neural classifier Limited Receptive Area (LIRA). The classifier LIRA is contained of three neuron layers: output, sensor and associative layers. The classifier was tested over two image databases. The first database is the MNIS database which contained 60,000 handwritten digit images for the classifier training and 10,000 handwritten digit images for the classifier testing. The second database has 441 images of the assembly micro device. For dividing the database a random procedure was used to test and train subsets. The classifier LIRA provides error rate of 0.61% as a mean value of three trials.





This is a general diagram for Arabic digit recognition system.

While what I suppose to do is another way called “Direction” method, which determines by the direction and trace the black pixels going through. With the difference of change in angles of lines and trace, we can figure out what the number is. The advantage of this method is we can achieve digit recognition without the interference of size of number and make it more accurate.

Details: We can divide the digit image into pixels. And according to the difference of the color (black or white), we can figure out where is black (represents -1) and where is white (represents +1).

|  |  |  |
| --- | --- | --- |
| B | B | W |
| W | B | W |
| W | B | W |

B means “Black” and W means “White”, we can figure out the “path” it come from. Then we get the “direction” (the direction is not the writing direction but the test direction).

Then we go down first, until we get the point we can only go back without other pixel to go. This position can be called “start point”. Then we start at start point, follow the exactly path, we can find out what the number is.

|  |  |  |
| --- | --- | --- |
| W | B | W |
| W | B | W |
| W | W | W |

Examples:

The path to “1”:



Direction address example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 16 | 1 | 2 | 3 | 6 |
| 15 | 8 | X | 4 | 7 |
| 14 | 7 | 6 | 5 | 8 |
| 13 | 12 | 11 | 10 | 9 |

We can give each black pixel a “direction address” which stored the direction it comes from. The value would be (X,Y), X∈[1,8],Y ∈[1,16]. When the value of direction address change more than 2 exclude 7 for X and more than 2 for Y exclude 15, the point would be call a “change point”.

For example: the point next to start point of “1” is goes bottom,

So the direction address of the start point would be “(6,11)”, “(6,12)” or “(7, 12)”.

When we sum up all of the direction address, it would be something like“(6,11), (6,12), (6,12), (7, 12)…” and so on.



## Similar method for number “7”:

Clean the noise points first;

## The direction address of each point before angle would be “(4, 7), (4, 7), (4, 7), (4, 7), (4, 7), (4, 7)…” while the direction point’s direction address would be “(6,11)”, “(6,12)” or “(7, 12)”, and it would go through that way.

## So whole direction addresses for “7” would be “(4, 7), (4, 7), (4, 7)…(6,11), (6,12), (7, 12)…” according to the change in direction address, we can figure out what the number it is.

According to the method I mentioned above, this recognition system can simply deal with the number without circles, such as1, 2, 3, 5, 7, but it’s hard to handle with numbers with circles such as4, 6, 8, 9 and especially 0 which even don’t has a single point!

The way to handle these problems is to add a circle detect system. Because the trace of the black pixels has already been recorded, we can find out whether there is a circle by the change of direction angle (360°) and repeat points which means we should have a new variable to cache the change in angle. The shortcoming for circle detect system is that we need more time to calculate the angle changed, but by theory aspect it would work.

But the handwriting digit can be with a great variety, there are also some problems I cannot deal with, for example:





Those numbers in above pictures are too crowd to read.

**Conclusion:** although there are many problems in “Direction Address“ method, such as low running speed a high hardware required and cannot work with too crowd numbers. But it still has some advantages, for example, numbers are figured out by the trace so we don’t need to worry about the size of number anymore, which means we will no longer need a huge database to store every possible results.

“

**Work Cited**

Salch Ali K. Al-Omari, Putra Sumari, Sadik A. Al-Taweel and Anas J.A. Husain, “Digital Recognition using Neural Network”, School of Computer Sciences, University Sains Malaysia.

Y. LeCun, L.Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, and V. Vapnik, “Comparison of Learning Algorithms for Handwritten Digit Recognition”

E. Boser, I. Guyon, and V. N. Vapnik, A Training Algorithm for Optimal Margin Classifiers, in Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5 144-152, Pittsbutgh (1992).

Schurmann, A Multi=Font Word Recognition System for Postal Address Reading, IEEE Trans., G27, 3(1978).

O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, Chapter 4, John Wiley and Sons (1973).