ISSN 2395-1621

Handwriting Recognition Keyboard Application for Android

#1Prof. Swati Sagar, #2Nikita Bobade, #3Suchitra Pujari, #4Rhea Shetty

1swati.a.sagar309@gmail.com
2nikitabobade96@gmail.com
3suchitrapujari148@gmail.com
4rheashetty95@gmail.com

#1234Department of Information Technology

Bharati Vidyapeeth's College of Engineering for Women, Pune, Maharashtra, India.

ABSTRACT

We present a handwriting recognition keyboard application for Android OS which aims to replace the onscreen keyboard with handwriting recognition technology and can be used for text input in any Android application. It is an application where user writes a single character on the touchscreen, the application recognizes it and gives a possible list of suggestions. We use the techniques of cropping and down sampling to transform the image into a second image and then recognize character. The application uses machine learning/AI technique called Kohonen's neural network based on Self Organizing Map (SOM) algorithm for Character Recognition. This approach provides much higher performance than the traditional neural network.

Keywords: Android, Self-organizing map, Kohonen Learning Algorithm

ARTICLE INFO

Article History

Received: 31st March 2017

Received in revised form : 31st March 2017

Accepted: 3rd April 2017

Published online : 9th April 2017

I. INTRODUCTION

The keyboard application plays a major role in controlling functions of a smartphone. We have developed a handwriting recognition keyboard application which recognizes the character written by user on touch screen of mobile phone and gives possible suggestions for it. This application will replace the default keyboard which will allow user to draw character over the entire screen of phone without having space restriction. It is a simple handwriting recognition keyboard. It detects a single letter provided to it and also recognizes it. The fat finger problem that occurs with the traditional keypad is no more a problem here.

Computational models of handwriting recognition have been an active area of research since late 2000s, for they can contribute not only to theoretical insights but also to practical applications, such as optical scanning (optical character recognition), intelligent character recognition, handwriting movement analysis and artificial intelligence. However, developing a computational model of handwriting recognition is quite difficult, because handwriting are complex, multinational.

This android application is different in looks from other text entry applications. User can easily use the entire smartphone screen to write. It is a translucent screen so you can also view the background application or screen. The translucent screen has two rows on top of the screen. First row shows the output and second row gives list of suggestions. In GyroPen, character forms the input when a user draws on a touch screen of high resolution. Unfortunately, this resolution is too high to be directly presented to the neural network. To resolve this problem, we use the techniques of cropping and down sampling to transform the image into a second image that has a much lower resolution. Pre-processing techniques are applied to image for making it fit for recognition engine. The paper is believed to develop a method of handwriting recognition for native languages that is fast, robust, reasonably simple and accurate with a relatively simple and easy to understand algorithms and techniques. The method we present here has a promising performance.

II. LITERATURE SURVEY

There has been significant research and study in the field of handwriting recognition and till date it has proved to be challenging at every stage.

In **An Answer to "Who Needs a Stylus?" on Handwriting Recognition on Mobile Devices [1]**, a detailed study of handwriting recognition system that pre-

exist is given. It also mentions all the pros and cons of using handwriting recognition in mobile devices.

In **Bayesian network modelling of strokes and their relationships for on-line handwriting recognition [2]**, the use of a very promising Bayesian network is elaborated. This paper discusses the technicalities and algorithm to be followed to develop a stroke-model for character recognition.

In **Multi-Language Online Handwriting Recognition [3]**, a theoretical study of the approach and problems regarding multi lingual handwriting recognition is given.

In **Hand-Written Character Recognition Using Kohonen Network [4]**, character recognition is done over character which is drawn by dragging mouse over computer screen using simple neural network learning algorithm.

III. System Architecture

The following represents the typical blocks for our android application.

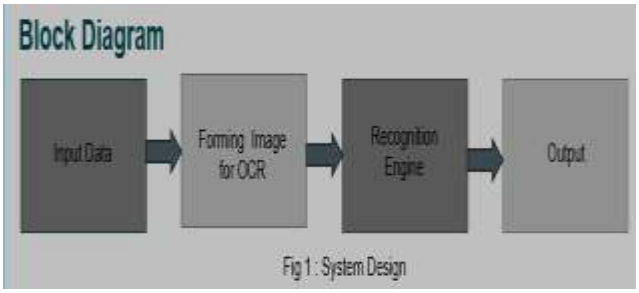

Figure 1: Block Diagram

I. Input:

Input is acquired from user. User writes on the keyboard layout a character using finger over touchscreen of smartphone.

II. Forming Image for OCR :

The character drawn is now converted into image format and then sent to the handwriting recognition engine. The preprocessing techniques are applied.

Image is down-sampled and cropped to remove all whitespaces by capturing the bit map of the image before giving it as input to the recognition engine. Basically we reduce the resolution of the image to a 5X7 resolution to make it work efficiently.

III. Recognition Engine:

This engine uses Kohonen Learning Algorithm. There are three basic steps involved in the application of the algorithm after initialization: sampling, similarity matching, and updating. These three steps are repeated until formation of the feature map has been completed.

The network that we implement is the Kohonen's network having two layers. There is an input layer and an out layer. The size of the input layer is decided by the user and has to match each row's size in the input data file. The Kohonen's network has neurons that compete with each other. In Kohonen's layer, input is fed to neurons and output is calculated using a weighted sum formula. Different patterns for input lead to many different neurons. Only the winning neuron and its neighbours qualify for learning of given input pattern.

IV. Output:

The character drawn on the keyboard layout will be recognized and given as output in the form of digitized character along with other matching and possible suggestions.

IV. Kohonen's Self-Organising Maps

A. Initialize Network:

For each of the nodes A, set the initial weight vector Wa(0) randomly to any value. Set the initial value for neighbourhood Na(0) to a large value.

B. Present Input:

For input we feed it in the form of binary pixels of 1 for white pixels and zero for black pixels. This is achieved by passing the input array to the computation of Kohonen's "winner". The result is which of the neurons won, this is stored in the "best" integer. Calculating winning node c based on the maximum activation among all p neurons participating. Hence the neuron with the largest activation is results into a winner. The winning or firing neuron has 1 as the final. Every other neuron apart from winning neuron has 0 as the final output.

C. Update Weights:

The original method that was proposed by Kohonen for calculating the changes in weights is known as the additive method.

The variable used in the additive method is the training vector that was presented to the network. The variable is nothing but the weight of the winning neuron, and the variable itself is the updated weight.

D. Training and Learning to Recognize Letters:

Kohonen's SOM is an unsupervised algorithm. It continuously learns. The ability of neurons increases since different neurons win for different kinds of patterns. A constant called learning rate is used by the learning algorithm. The learning rate has be a positive integer less than 1. Typically the learning rate is one such as .2 or .3. When we set the learning rate to a larger value it causes the training to progress much faster. If we set it too large it could cause the network to never converge.

In each epoch, the weights are adjusted. An epoch occurs when input of training data is given to the neural network and the weights are updated based on the results. As the adjustments of the weights occur, the next time the same training data is given to it, it must produce a network that will give much better matching results. As data is given and weights are updated and adjusted, epochs continue. Eventually the return on the weight updations and

adjustments will disappear to the point that it will longer be values to continue.

V. RESULT

One major complexity issue is an almost infinite variability in the user's hand writing due to many different kinds of writes and their nature of hand writing. A single character may be written in infinite ways and also can be represented in many different ways.

The user draws a character on screen and with the help of said methodology the character is recognised and different matching suggestions are presented. User can make an entire word character by character. This application is compatible with any other application where a traditional keyboard would be used.

Figure 2: Result

VI. CONCLUSION

This android application aims to make text-entry and communication easy. As smartphones are essential part of our lives today, people like senior citizens or people stuck with language barrier face difficulty in communication through smartphones. This application when developed in native languages in future can prove useful and make it easier for such people to handle communication through applications.

This application helps to solve various text entry problems in phones like "fat finger problem" while using touch screen with soft keyboards and small keypad. The handwriting recognition engine efficiently recognizes the input text written by user on touchscreen and converts it into digital text. Our application's layout occupies the entire screen of mobile thus giving user the freedom to input text of varying size and without any space restriction.

REFERENCES

[1] Holzinger, A.; Searle, G.; Peischl, B.; Debevc, M. An Answer to "Who Needs a Stylus?" On Handwriting Recognition on Mobile Devices. In E-Business and Telecommunications; Springer Berlin Heidelberg: Seville, Spain, 2012; Volume 314, pp. 156–167.

[2] S. Cho and J. Kim. Bayesian network modeling of strokes and their relationships for on-line handwriting recognition. Proc. of the sixth ICDAR, Seattle, WA, pages 86–90, Sep. 2001.

[3] Daniel Keysers, Thomas Deselaers, Henry A. Rowley, Li-Lun Wang, Victor Carbune . Multi-Language Online Handwriting Recognition .In IEEE Transactions on Pattern Analysis and Machine Intelligence (2016)

[4] Dr. Pankaj Agarwal "Hand-Written Character Recognition Using Kohonen Network" IJCST Vol. 2, Issue 3, September 2011 .

[5] Thomas Deselaers, Daniel Keysers, Jan Hosang, and Henry A. Rowley, "GyroPen: Gyroscopes for Pen-Input With Mobile Phones", IEEE transactions on human-machine systems, vol. 45, no. 2, April 2015, 263

[6] Vuokko Vuori, Erkki Oja, "Analysis of different writings styles with the self organizing map". In Proceedings of the 7th International Conference on neural information processing. Vol. 2., 2000, pp 1243-1247. November 2000

[7] Purnendu Mishra "Recognition of any Character by using SOM Technique" Int. J. Adv. Res. Sci. Technol. Volume 1, Issue1, Oct-2012, pp 48-50.

[8] Bailing Zhang, Minyue Fu, Hong Yan, and Marwan A. Jabri "Handwritten Digit Recognition by Adaptive-Subspace Self-Organizing Map (ASSOM)" IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 10, NO. 4, JULY 1999 .

