ISSN:0975-9646

Online Hand Writing Recognition System on Android Based Mobiles

Phani Sridhar A

Assistant Professor, Computer Science and Engineering,
Sanketika Vidya Parishad Engineering College
P.M.Palem, Visakhapatnam, Andhra Pradesh, India.

***Abstract* – Now a days smart mobiles with androido*s ar*e very relentlessly scattered in the technological world. There may be many features and apps which make the effective use of these mobiles as well as serve the needs of a user. This paper presents a system for online hand written character recognition that involves alphabets of lower case and upper case, numerical and some basic expressions together with a user-interface. The aim of this study is to utilize the most convenient man-machine-interface like pen for input of mathematical expressions. In pen-enabled devices handwriting sequences are collected by the digitization of pen movements which outputs an array of coordinates called strokes .A neural network is trained for recognizing each stroke and a recursive algorithm parses the expression by combining neural network output and structure of the expression. After the recognition, all output is combined into file and hence enables easy-input of characters in all pen enabled devices such as tablet external tablet pads, electronic pen-boards etc.**

***Keywords*— Smart mobiles, androido*s*, recognition, strokes, neural network, recursive algorithm, digitization.**

I. INTRODUCTION

From the conception of the first alphabet, handwriting has been a medium of communication. As the literacy rate in most societies improved, handwriting has played a major role in technological advancement, keeping historical records and also as a persistent means of communication. With the advancement of technology more and more technical barriers have been broken. The advent of computers was a great enhancement to mankind's everyday life which also revolutionized writing systems. The pervasive nature of small handheld computing devices is spear heading a new movement in information technology. Small devices like handheld computers, smart phones and PDAs are a few of the gadgets that are making this phenomenon become a reality. In spite of this, handwriting has still prevailed in this day and age of modern technology.

Interaction between human beings and most computing devices employed keyboards and pointing devices like the mouse. However, these input methods are inappropriate when it comes to the application of small devices; mostly because of their size. This necessitates the need for innovative input methods. Handheld computing devices required easier methods of interaction for use. Researchers have come up with yet another means of interaction, handwriting recognition. Smart Phones, Palmtop computers and PDAs utilize a stylus as one of their main input devices. The stylus is used as both a pointing device and also for text entry. Handwriting Recognition systems (HWR) with PDAs, comprises of the software component that facilitates data entry, recognition and interpretation. Handwriting recognition can be broadly classified into two groups: online recognition and offline recognition. Online handwriting recognition makes use of pressure put upon an electrostatic-sensitive writing surface upon which the user forms handwriting with the stylus. Online recognition system considers samples of the movement of the pen-tip, the coordinates of the sampled points, and information on pen-up and pen-down states. On the other hand offline handwriting recognition utilizes the handwriting image after completion of the handwriting process. This type of handwriting recognition utilizes a scanner as an input to get the handwriting image. As a result it lacks the temporal input sequence information provided directly by the user. On-line data, in general, is more compact compared to off-line data because of the different dimensionalities in representation. The difference in the data size results in substantial difference in the processing time.

Recognition of characters has been a widely studied problem in last twenty years Moreover; we believe it has the still growing importance. Its origins can be found in conventional OCR (optical character recognition) methods which deal with the recognition of scanned texts. There are two main fields of study in the area of character recognition.

1) The first more traditional one deals with an automatic conversion of Scanned characters to an electronic form. It is called offline recognition.
2) On the other hand, online recognition deals with interpretation of character entered to the computer by drawing them by hand, using some kind of an input device[2].

From these two, offline recognition is considered to be a more difficult task, as scanned images of formulas contain less information than formulas entered by an input device. Moreover, the offline methods have to deal with the artifacts originating from the process of scanning. Offline recognition can be further categorized to the recognition of printed text and the recognition of handwritten characters. Apart from the usual process of recognizing the characters it is needed to implement in the android mobiles.

Handwriting Recognition is the task of transcribing a language message represented in a spatial form of graphical marks, into a computer text. Studies in this field of pattern recognition have been ongoing for more than four decades. Nevertheless, various applications exist that necessitate this

ever continuing research in search of better, more robust and reliable recognition systems. One such application, handwriting interpretation, deals with the task of determining the most likely meaning of a sample of handwriting and the generic process of hand writing recognition is shown in fig.

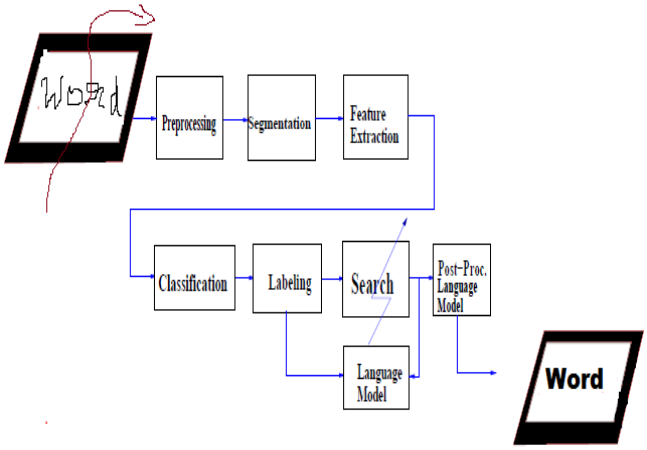

Fig. 1.1 Traditional Hand writing Recognition Process

II. LITERATURE SURVEY

On - line recognition refers to the recognition mode in which the machine recognizes the handwriting while the user writes on the surface of a digitizing tablet with an electronic pen. The digitizing tablet captures the dynamic information about handwriting such as number of strokes, stroke order, writing speed, etc., all in real time. Off-line recognition, by contrast, is performed after the handwriting has been completed and its image has been scanned in. Thus, dynamic information is no longer available. Another advantage of on - line recognition over off-line recognition is that there is close interaction between the user and the machine. [3] The user can thus correct any recognition error immediately as it occurs. The problem of on-line handwriting recognition can be defined in various ways. Variables in the problem definition include character set, writing style and desired recognition rate. In general, each problem definition lends itself to different algorithmic approaches, which in turn make use of different features for classification. With respect to alphanumeric character recognition, it seems that the most discriminating features come from their shape. This has led to study in the field of handwritten alphanumeric character recognition to use geometrical and topological features, statistical features, and other algorithms to perform recognition based on character shape. In recent years, some effort has been made in modeling pen - tip movement. In the handwriting generation model, for example, the movement of the pen tip is described as a velocity vector controlled by two synergies: one is the curve-linear velocity and the other is the angular velocity. Such a model gives us a better understanding of handwriting generation. Later on various approaches have been proposed for variety of problems that are raised in various languages in terms of curve difficulty, stroke identity, space realization, expression identity, memory management of processors, express reliability, data type flexibility and its scope etc.

III. ALPHANUMERIC CHARACTER RECOGNITION

There is a great need of recognizing the alphanumeric characters even though it has been given a flexible and reliable solution for identifying the single and multiple alphabets in various languages. As there are many fields where the handwriting recognition is not only sufficient but also calculation and solutions to small arithmetic numerical are mandatory. It is also identified that many programmed devices are sold out in the market which are responsible for making a clear view of characters that are written on these devices. In this an approach to online handwritten alphanumeric character recognition based on sequential handwriting signals has been proposed. The recognition process consists of the following stages:

- Data preprocessing
- Feature extraction
- Character classification
- Reference set evolution

Feature extraction is a very crucial step, as the success of a recognition system is often attributed to a good feature extraction method. The main idea here is to come up with an efficient and reliable method for extracting features from the sequential handwriting signals. In particular, points in strokes corresponding to local extreme of curvature are detected. These points correspond to the maxima of angular velocity or the minima of curvilinear velocity of the pen-tip movement in fluent writing. n addition, the mid-point between two consecutive points that correspond to curvature extreme or pen-down/pen-up locations is also used to represent a local minimum of angular velocity (or a local maximum of curvilinear velocity).

An expected advantage of this approach over other methods comes from the fact that our character matching processes are elastic and hence can tolerate local variation and deformation. Besides, dominant points are quite easy to extract using the technique. Moreover, our approach can handle large alphabets due to its fast pre-classification step. However, since our approach is based on sequential handwriting signals, it is intrinsically stroke-order dependent unless multiple models corresponding to different stroke orders are used.

A. *Choosing a Correct Symbol :*

Choosing the right symbol is in fact the hardest stage in the process of recognition of any kind. After the learning process is complete we have 10 'digit chain code's representing the 10 digits matching the user hand writing. Once the user is writing an input symbol, the system has to decide what symbol in our alphabet it resembles the most. In order to compare the input character with our alphabet, the following methods were tested:

1. *Normalizing without using interpolation*: For each pair of input and one of the alphabets 'digit chain code's, the two chains are normalized to the same length (the maximum length of both chains). The missing entries are filled with duplicating present entries in equally spaced location along the vector, according to the "stretching factor". Once equal in size, the sum of differences in each entry is measured (basically it means measuring the differences using 1-NORM), and normalized.
2. *Normalizing using interpolation*: The only difference from the first method is in the process of normalizing both vectors to the same length: the additional entries are calculated using linear interpolation in between two adjacent entries. This way the matching process is invariant to scaling.
3. *Using Relative chain codes*: Both previous approaches were implemented on a relative 'digit chain codes. We follow the same process on a sequence of differences between following angles measured. This way the matching process is invariant to rotation and scaling.
4. *Divide and conquer*: Dividing each digit chain code to a number of smooth segments. The number of smooth segments is first compared, narrowing down the number of candidates. Then, for each segment, a similar procedure is taken as in 1, 2.

IV. CONNECTIVITY

As it is well known fact that android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown in the architecture diagram. This is an intelligent system which binds any sort of applications into it with little difficulty that prevail the usability for the customer as well as the developer. Here preciously the connectivity and internal requirements are needed to be satisfied.[3]

But necessarily the issues that are specified in the architecture are shown in fig 2.

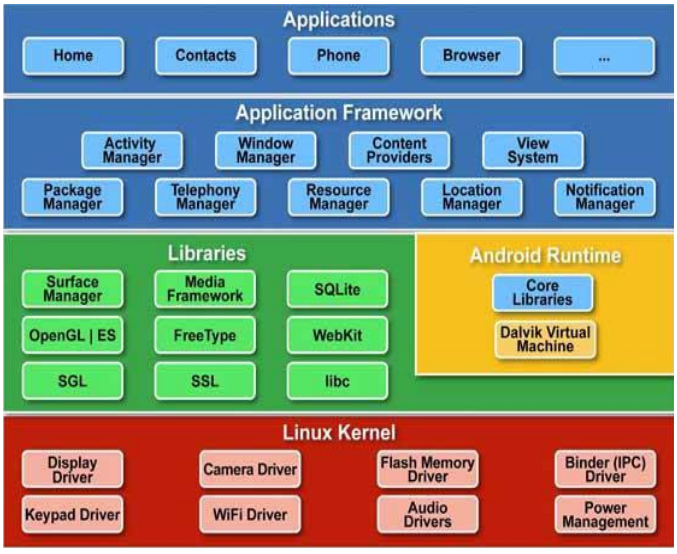

Fig. 4.1 Android Architecture

Even after recognizing the alphabets it is needed to match the words so that the perfect match could be displayed as output. For this purpose some clustering mechanisms are used.

A. *Agglomerative hierarchical clustering*

Clustering is a data mining (machine learning) technique used to place data elements into related groups without advance knowledge on the group definitions. In this paper the authors provides an in depth explanation of implementation of agglomerative and divisive clustering algorithms for various types of attributes. Database - the details of the victims of Tsunami in Thailand during the year 2004, was taken as the test data. The algorithms are implemented using Visual programming and the formation of the clusters and running time needed of the algorithms using different linkages (agglomerative) to different types of data are taken for analysis. Clustering is the most popular method that makes an attempt to separate data into disjoint groups such that same-group data points are similar in its characteristics with respect to a referral point whereas data points of different-groups differs in its characteristics. Such described groups are called as clusters. Thus clusters are comprised of several similar data or objects with respect to a referral point. Cluster is one of the most important methods in the disciplines of engineering and science, including data compression, statistical data analysis, pattern recognition, data mining, artificial intelligence, and so on. Some real time applications such as handwritten character recognition, fingerprint recognition, speech/speaker recognition, and document classification, require the use of clustering techniques in order to reduce the training data amount or to find representative data points. Clustering methods are broadly understood as hierarchical and partitioning clustering. A hierarchical clustering is a nested sequence of partitions. This method works on both bottom-up and top-down approaches. Based on the approach hierarchical clustering is further subdivided into agglomerative and divisive. The agglomerative hierarchical technique follows bottom up approach whereas divisive follows top-down approaches. Hierarchical clustering use different metrics which measures the distance between 2 tuples and the linkage criteria, which specifies the dissimilarity in the sets as a function of the pair-wise distances of observations in that sets. The linkage criteria could be of 3 types' single linkage, average linkage and complete linkage An important step in any clustering is to select a distance measure, which will determine how the similarity of two elements is calculated. This will influence the shape of the clusters, as some elements may be close to one another according to one distance measure. Agglomerative algorithms in which we introduce intermediate clusters obtained by partitional clustering algorithms to constrain the space over which agglomeration decisions are made. We referred to them as constrained agglomerative algorithms. These algorithms generate the clustering solution by using an agglomerative algorithm to build a hierarchical sub tree for each partitional cluster and then agglomerate these clusters to build the final hierarchical

tree. Our experimental evaluation shows that these methods consistently lead to better solutions than agglomerative methods alone and for many cases they outperform partitional methods, as well. To understand these improvements, we studied the impact that constraining has on the quality of the neighborhood of each document and found that constraining leads to purer neighborhoods as it can identify the right subspaces for the various classes.

B. *Agglomerative Algorithm*

For n samples, agglomerative algorithms begin with n clusters and each cluster contains a single sample or a point. Then two clusters will merge so that the similarity between them is the closest until the number of clusters becomes 1 or as specified by the user.

- Start with n clusters, and a single sample indicates one cluster.
- Find the most similar clusters Ci and Cj then merge them into one cluster.
- Repeat step 2 until the number of cluster becomes one or as specified by the user.

The distances between each pair of clusters are computed to choose two clusters that have more opportunity to merge. There are several ways to calculate the distances between the clusters Ci and Cj.

V. Data Capture And Implementation

After match up of a alphabet with the data set values these are to be displayed accordingly with the font style and size appropriately. But here is the problem with acceptability of the given test alphabet and the generated output. Here comes with the access of various input values that can capture easily and improvise the pre dedicated values. This could be done by the capture sensible practices. Some of the examples for these UI have been shown below. The data capture dialog box will be something like the slots are dedicated to individual alphabets which will fall into that slot.

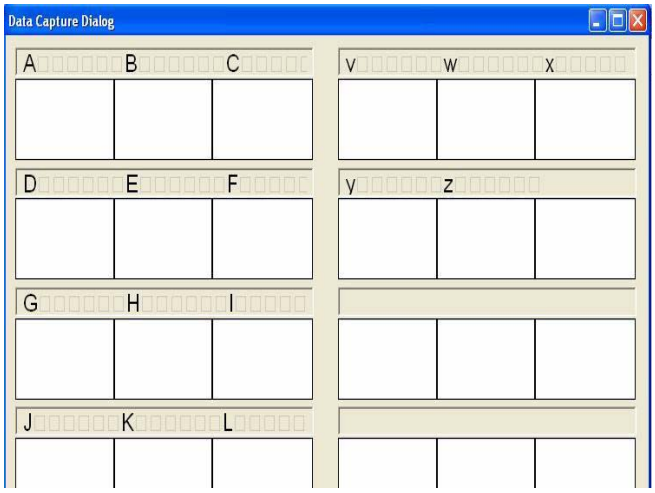

Fig. 5.1 Data Capture Dialog Box

The data is resembled using the mark or identity on the sensible pop up on these displays. This could be identified by the dialog box shown below and the available display screens are relatively good to specify the abrupt styles of the user. The writing style illustrates the capability on which the exact data set represents. Also this includes the stoke and style of the tip or pen which is used. Some writing stokes can be given as Basic strokes and cursive strokes [1].

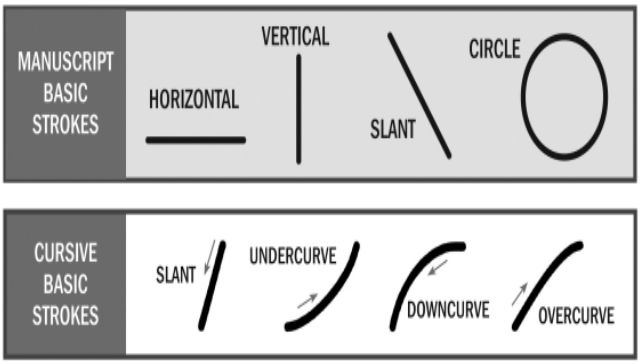

Fig. 5.2 Some Hand Writing Strokes

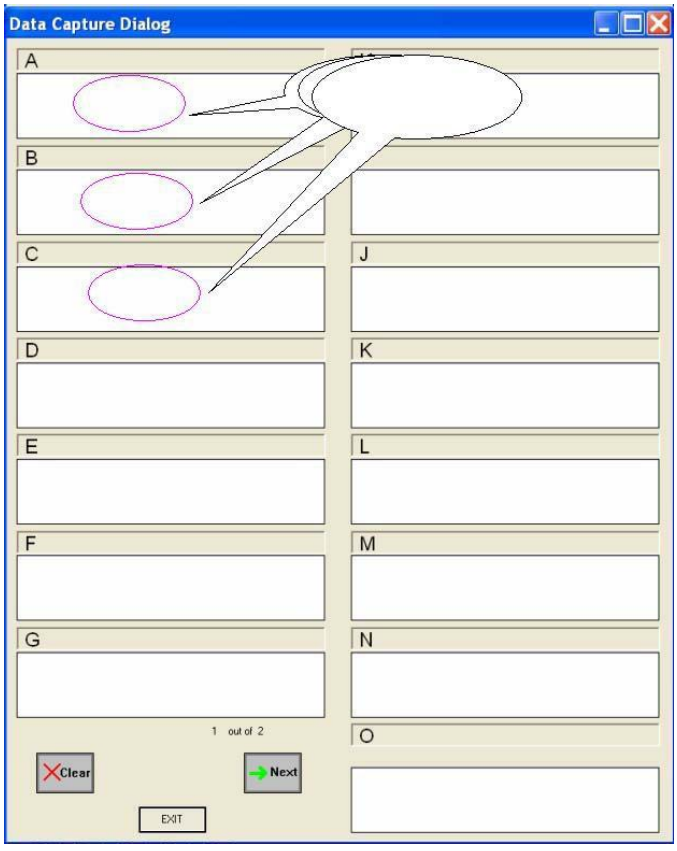

Fig. 5.3 Dialog Box identifies an Alphabet

VI. Results

After design and implementation this is installed as an apk in a basic smart phone. Here the work area is enables and it has been given with a data dialog box where the entry of data is identified. Later on a alphabet is written on the display screen that will allow the style of writing and the stroke style. Based on these criteria the alphabet has been compared with the data set and the output is displayed accordingly.

Fig. 6.1 Work Area of data input

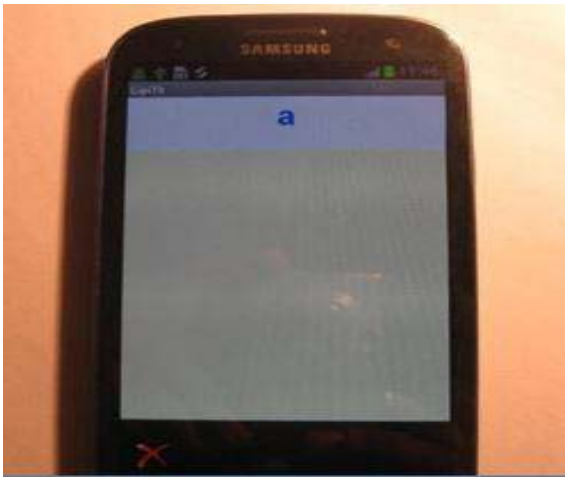

Fig. 6.2 Work Area of data output

VII. CONCLUSION AND FUTURE WORK

This work has been given an efficient application to be used in android based smart phones that could recognize the alphabets and words written with the help of a finger tip or pen on the screen which is displaying the same as output on the display for pre specified style and size.

Further this work can be extended for the expression calculation by the use of arithmetic operators. It is also suggested to enhance with the help of major image processing techniques for the visually impaired (VH) persons. Also the same could be extended for other disciplines and as an embedded application too.

REFERENCES

1. Basic strokes and its importance in handwriting **"**Blog.upub.net/writing.../Basic-Strokes-The-Foundation-of-Handwriting**"**.
2. **"**A Novel Approach to On-Line Handwriting Recognition Based on Bidirectional Long Short-Term Memory Networks**"** by Marcus Liwicki, Alex Graves, Horst Bunk, Jurgen Schmidhuber.
3. **"**The Writer Independent Online Handwriting Recognition System frog on hand and ClusterGenerative Statistical Dynamic Time Warping**"**
by Claus Bahlmann and Hans Burkhardt, Member, IEEE.
4. http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture.
5. http://www.square-9.com/blog/advancements-of-smartphones-in-data-capture.