International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958, Volume-2, Issue-2, December 2012

# Design of Multimedia Application for Fast and Efficient Text Input from Touch Screen Input Devices using Character Recognition

**K S Jagadeesh, Chandramouli.H, Naveen Ghorpade**

***Abstract:*** ***We are wasting a lot of our time texting and typing messages through mobile's and keyboards, so we have come up with software which can recognize the set of character u scribble on the screen and make it visible in the normal times new roman format. This would save lot of our time as we write or scribble faster than typing through other input devices and more efficient user interface is also achieved. Character recognition is a task of determining handwritten characters /digits. This is done by having some of the sample sets of characters written by numerous people. The task entails matching the handwritten characters with characters in the sample set and determining the character in the sample set which best matches the Test Character. The aim of the second step of the recognition structure is to extract discriminant information from an image of a character, as well as to reduce its dimensions of representation. This reduction is required in order to make easier the conception of the classification system, when discriminant feature extraction allows to present competently a character to the classifier. This paper envisages using a number of benchmark datasets to carry out the task. The first step is a feature extraction. Features such as shape, orientation, outline, character frontiers etc, have to be extracted from the character image. The features are then used for the pattern classification task. The output gives the class to which the character belongs. The results obtained using neural networks was compared with other methods of classification for character recognition and classification provides highest accuracy of 96%.***

***Keywords:*** ***Feature extraction, transducer, Character Recognition, Pattern Recognition.***

## I. INTRODUCTION

The world is fastening up so we require fastening up too, hence we are trying to implement a very good user interface which helps us in generating our text faster than we type using input devices.

In this project we take the text written on the touch screen mobiles or touch screen monitors in the image format and get the text out of it to normal text format recognizing the pattern of the text. The text obtained can be then sent to the other person. This is more efficient and time saving than our normal process of sending mail's or messages.

**Manuscript received on December, 2012.**

**K S Jagadeesh,** Research Scholar-JJT University, Rajasthan
**Chandramouli.** Research Scholar-JJT University, Rajasthan
**H, Naveen Ghorpade,** Assistant Professor, Department of CSE, Sri Krishna Institute of Technology, Bengaluru.

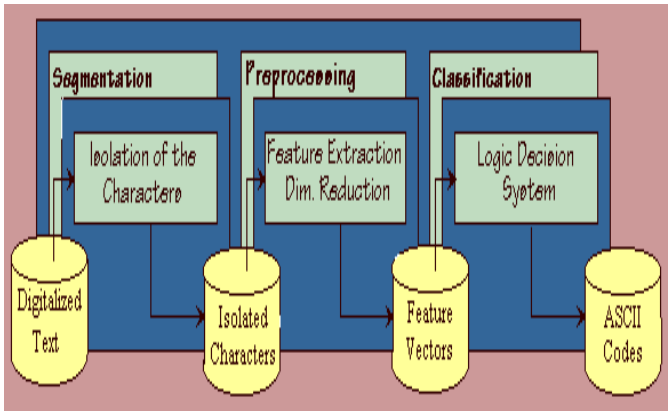
We have implemented this project of character recognition using algorithms for determining individual symbols and symbol image matrix mapping. We have coded the software using C++ View cut as front end, TC Editor as user agent and C language as back end.

### *A. Need of Character recognition*

The need for character recognition in our project deals with providing a very good user interface so that the work of an user becomes less and easy. It saves the time of the user as he will just be scribbling the information and it would automatically get converted to normal text. It's efficient as it saves lot of time of the user and work goes on in a easy way.

### *B. Properties*

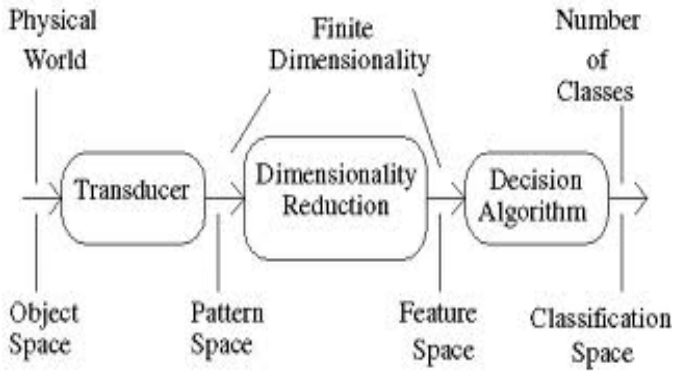
***a) Character Recognition***: The main structure of recognition system is illustrated shown in the figure below.




The first step is the segmentation one, which consist in analyzing the digitalized image provided, so as to localize the limits of each character, and to isolate them one from each other. The aim of the second step of the recognition structure is to extract discriminant information from an image of a character, as well as to reduce its dimensions of representation. This reduction is required in order to make easier the conception of classification system, when discriminant feature extraction allows to present competently a character to the classifier. Once discriminant features have been extracted, they are submitted to a logic decision system whose task is to identify the character that they represent and to assign them to corresponding ASCII code, this is the classification step at last.

***b) Pattern Recognition:***

The pattern recognition in our project has 2 stages. First stage is feature extraction and second stage is classification. It can be shown as below.




The first stage feature extraction where the input is given from the user through screen of a mobile or monitor i.e. as shown from physical world is taken as an input from the transducer in the screen then converted to image and sent into the dimensionality reduction phase it reduces the image to required dimension. Feature extraction is measurement on a population of entities that will be classified. This assists the classification stage by looking for features that allows fairly easy to distinguish between different classes. Several different features have to be used for classification. The decision algorithm helps in stage of classification.

The methods used for recognizing the patterns in this project are

1) Bayesian decision theory
2) Linear Classification or Discrimination

***C. Applications:***

The major application of implementing this software would be the user friendliness. The user will be very helpful as they would be saving lot of their work without typing. They may easily scribble as they write it for reference and the text would be automatically converted to standard form. The user interface would be very good and efficient as it saves lot of time.

***D. Problem Formulation:***

This project aims at designing and implementing efficient software which would deal with character recognition. The text written on the screen will be converted to image and then converted to the standard form using many algorithms, and would be helpful in time utilization.

## II. PROBLEM DEFINITION

The high level task in off-line handwriting recognition is to classify an ordered sequence of images of handwritten characters. Each character is represented by a scaled, demised, segmented 16 by 8 array of 1-bit pixels.

This means that the problem of character segmentation, in which the image of the text is split into images of individual characters, has been solved for this dataset. To simplify the problems posed by capital letters, the dataset does not include the first letters of the original words. The data was collected from 159 different subjects with wide variance in handwriting style, and it appears that some of the characters have been segmented from cursive handwriting. Therefore, our data set might be considered as a mixed modal environment, which is a much harder domain compared to a small group of collaborative writers that mimic a specific style of writing as in. Each character entry includes, along with the sampled bitmap image, a label for the correct character classification, word information (to indicate boundaries between words), and ordering information. The ordering information and the word ID have been crucial to our approach to the problem.

## III. IMPLEMENTATION

***A. Determining Character Lines:***

Enumeration of character lines in a character lines in a image is essential in delimiting the bounds within which the detection can proceed. Thus detecting the next character in an image does not necessarily involve scanning the whole image all over again.

***Algorithm:***

1. Start the first x and y pixel of the image pixel(0,0), set number of lines to 0.
2. Scan up to the width of the image on the same y-component of the image
   a. If a black pixel is detected register y as top of first line
   b. If not continue with next pixel.
   c. If no black pixel is found up to the width increment y, reset x to scan the next horizontal line.
3. Start at the top of the line found and first x-component pixel(0,line_top).
4. Scan up to the width of the image on y-component of the image
   a. If no black pixel is detected register y-1 as bottom of the first line. Increment number of lines.
   b. If a black pixel is detected increment y and reset x to scan the next horizontal line.
5. Start below the bottom of the last line found and repeat steps 1-4 to detect subsequent lines.
6. If bottom of image is reached stop.

***B. Detecting Individual Symbols:***

Detection of individual symbols involves scanning character lines for orthogonally separable images composed of black pixels.

***Algorithm:***

1. Start at the first character line top and first x-component.
2. Scan up to image width on the same y-component
   a. If black pixel is detected register y as top of the first line
   b. If not continue to next pixel
3. Start at the top of the character found and first x-component, pixel(0,character_top)
4. Scan up to the line bottom on the same x-component

a. If black pixel found register x as the left of the symbol
b. If not continue to the next pixel
c. If no black pixels are found increment x and y to scan the next vertical line

5. Start at the left of the symbol found and top of the current line, pixel(character_left, line_top)
6. Scan up to the width of the image on the same x-component
   a. If black characters are found register x-1 as right of the symbol
   b. If a black pixel is found increment x and reset y to scan the next vertical line
7. Start at the bottom of the current line and left of the symbol, pixel(character_left, line_bottom)
8. Scan up to the right of the character on the same y-component
   a. If a black pixel is found register y as the bottom of the character
   b. If no black pixel are found decrement y and reset x to scan the next vertical line.

### *C. Symbol Image Matrix Mapping:*

The next step is to map the symbol image into a corresponding 2 dimensional binary matrix. An important issue to consider here will be deciding the size of the matrix. If all pixels of the symbol are mapped into the matrix, one would definitely be able to acquire all the distinguishing pixel features of the symbol and minimize overlap with other symbols. However this strategy would imply maintaining and processing a very large matrix(up to 1500 elements for a 100*150 pixel image). Hence a reasonable tradeoff is needed in order to minimize processing time which will not significantly affect the separate ability in patterns.

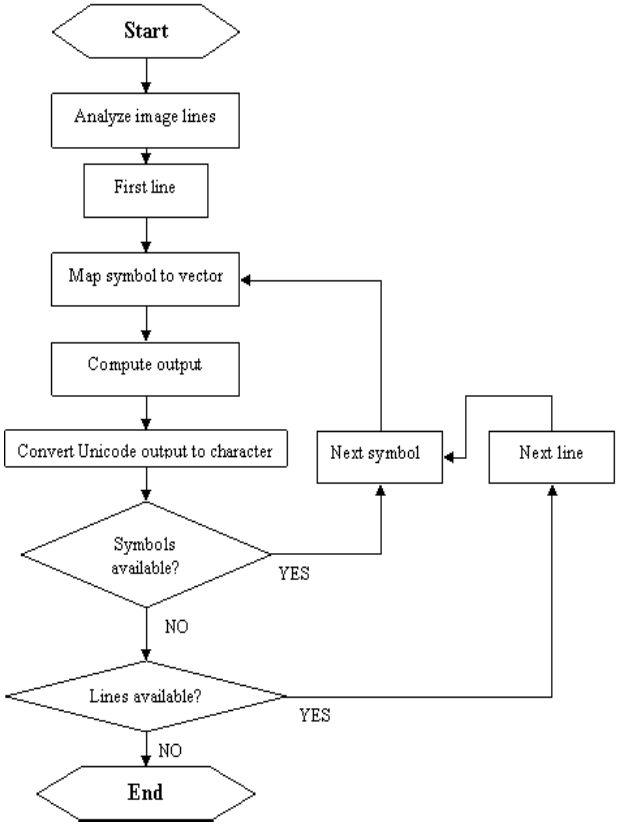
### *Algorithm:*

1. For the width (initially 20 elements wide)
   a. Map the first (0,y) and last (width,y), pixel components directly to the first(0,y) and last (20,y) elements of the matrix
   b. Map the middle pixel component (width,2/y) to the $10^{th}$ matrix element
   c. Subdivide further divisions and map accordingly to the matrix
2. To the height (initially 30 elements high)
   a. Map the first (x,0) and last (x,height) pixel components directly to the first(x,0) and last (x,30) elements of the matrix
   b. Map the middle pixel component (x,height/2) to the $15^{th}$ matrix element
   c. Subdivide further divisions and map accordingly to the matrix
3. Further reduce the matrix to 10*15 by sampling a factor of 2 on both the width and the height.

## IV. DESIGN

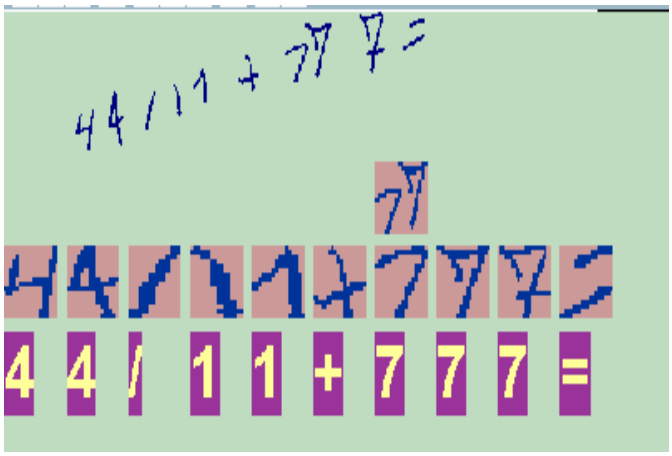
The software is developed and designed considering the aspects such as extensibility, robustness, reliability, fault-tolerance, security, maintainability, compatibility, modularity and reusability.

Design flowchart:




## V. RESULTS

The results are found to be successful for the input image given, we have obtained the result as expected and the conversion is accurate. The scribbled digits was converted to standard format as shown below and its very accurate.



## VI. CONCLUSION

The software has its own advantages and disadvantages, for future enhancement as below

- We can implement it in future for the traffic systems to recognize the number plate of vehicle's which would be not clearly visible in d picture or would be in different fonts etc.
- It can be implemented in creating software for blind so that they can text friends through writing characters in their own learnt language and get it converted through this software in a smart phone or system as explained.

## VII. REFERENCES

[1] E.Cohen, J.J.Hull, and S.N.Srihari. "Control Structure for interpreting handwritten Address" IEEE Transactions.

[2] E.Cohen, J.J.Hull, and S.N.Srihari. "Understanding Handwritten Text in a Structured Environment" Proceedings of IEEE.

[3] J.C.Simon O.Barat and N.Gorski."A System for Recognition of Handwritten" International Conference on Document Analysis System.

[4] G.Kim and V.Govindaraju. "Bank check Recognition Using Cross Validation between Legal and Courtesy Amount" Automatic Bank check Processing, World Scientific pages.

[5] S.N.Srihari, Y.C.Shin, V.Ramprasad and D.S.Lee. "Name and Address Block Reader" Proceedings of IEEE.