

Identification of Alphanumeric pattern using Android

Anamika Sharma (M. Tech Student), Suman Kumar Jha, Lingayas University, Faridabad.

Abstract: The “Identification of Alphanumeric pattern using Android” is a smart phone apps using Android platform and combines the functionality of Optical Character Recognition and identification of alphanumeric pattern and after processing, data is stored in server. This paper present, to design an apps using the Android SDK that will enable the Identification of Alphanumeric pattern using optical character reader technique for the Android based smart phone application. Camera, captures the document image and then the OCR is convert that image in to text (Binarization of captured data) according to the Alphanumeric (alphabetic and numeric characters) database and data stored in server.

Keywords—*Android, Tesseract OCR engine, API, , Tesseract Android Tools, Android SDK, Android NDK, Tess-two, Eclipse, ADT plug in, cloud computing.*

I. INTRODUCTION

Android is a Linux-based operating system whose code was released by Google under the Apache license, is used primarily in touch screen devices.

Android structure consist the Linux kernel layer which provides memory management, security settings, power management, hardware abstraction, etc. This Android application mainly used OCR systems to convert the documents, either printed alphabet or handwritten into machine encoded text ,its used in converting books and documents into electronic files and to publish the text on a website. Early OCR techniques were based mostly on template matching, simple line and geometric features, stroke detection, and the extraction of their derivatives. Such techniques were not sophisticated enough for practical recognition of data handwritten on forms or documents. Nowadays, in OCR, once a printed or handwritten text has been captured optically by a scanner or some other optical means, the digital image goes through the following stages of a computer recognition system:

1. The preprocessing stage that enhances the quality of the input image and locates the data of interest.
2. The feature extraction stage that captures the distinctive characteristics of the digitized characters for recognition.
3. The classification stage that processes the feature vectors to identify the characters and words.

To extract symbolic information from millions of pixels in document images, each component in the

character recognition system is designed to reduce the amount of data. As the first important step, image and data preprocessing serve the purpose of extracting regions of interest, enhancing and cleaning up the images, so that they can be directly and efficiently processed by the feature extraction component. Finally, it describes how to correct defects such as skewed text lines and slanted characters, how to normalize character images, and how to extract contours and skeletons of characters that can be used for more efficient feature extraction. One of the most important feature of Android application is cloud computing to deliver exciting applications to enterprises.

II. Objectives

- (a) Design and implement an Android application that is able to take a picture on machine-printed characters.
- (b) Design a server that is able to take the image, process the image and return the text to the client.

III. Design and Implementation

OCR system has the following particular processing steps:

1. Scanning.
2. Preprocessing.
3. Feature extraction or pattern recognition.
4. Recognition using one or more classifier.
5. Contextual verification or post processing

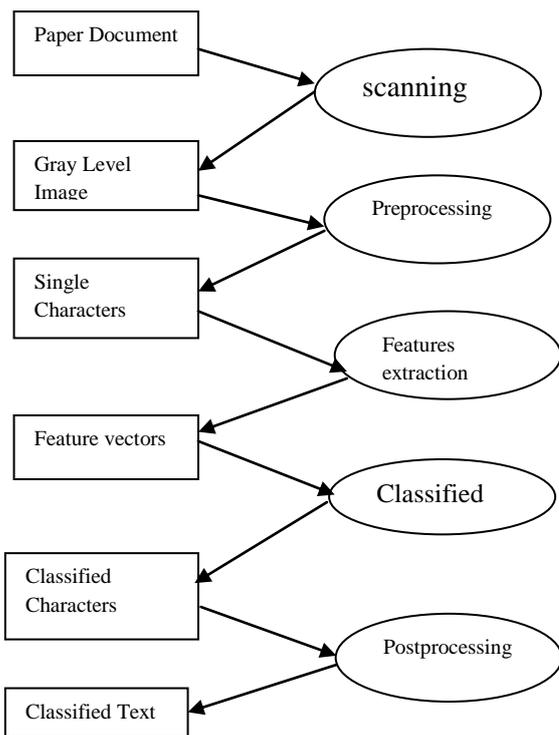


Figure 1: Steps of an OCR

1. Scanning

Scanning is a method for extract characters from scanned images and converts the image into proper digital image. This process is called text digitization.

2. Preprocessing:

Preprocessing consists of number of processing steps to formulate the raw data, usable for the recognizer. The typical preprocessing steps included the following process:

- (a) Binarization
- (b) Noise Detection & Reduction
- (c) Skew detection & correction
- (d) Page layout analysis
- (e) Segmentation

(A)Binarization methods

Binarization is a technique by which the gray scale images are converted to binary images. Some binarization methods are given below

1. Global Fixed Threshold: The algorithm chooses a fixed intensity threshold value I . If the intensity value of any pixel of an input is more than I , the pixel is set to white otherwise it is black. If the source is a color

image, it first has to be converted to grey level using the standard conversion.

2. Otsu Global Algorithm: This method is both simple and effective. The algorithm assumes that the image to be threshold contains two classes of pixels and calculates the optimum threshold separating those two classes so that their combined spread (intra-class variation) is minimal.

3. Nib lack’s Algorithm: Niblack’s algorithm calculates a pixel wise threshold by sliding a rectangular window over the grey level image. The threshold is computed by using the mean and standard deviation, of all the pixels in the Window.

4. Adaptive Nib lack’s Algorithm: In archive document processing, it is difficult to identify suitable sliding window size SW and constant k values for all images, as the character size of both frame and stroke may vary image by image. Improper choice of SW and k values results in poor binarization. Modified Niblack’s algorithm.

5. Sauvola’s Algorithm: Sauvola’s algorithm is a Modification of Niblack’s which is claimed to give Improved performance on documents in which the Background contains light texture, big variations and Uneven illumination. In this algorithm, a threshold is Computed with the dynamic range of the standard deviation.

(B) Noise detection & correction methods

Noise can be produced during the scanning section Automatically. Some noise detection and correction approaches are (a) to design low pass filter to reduce noise , it remove a the noise a while retaining the entire Signal (b) averaging method and the median method are used to removed the noise from character images.

(c) Skew detection & correction methods

Sometimes digitized image may be skewed and for this situation skew correction is necessary to make text lines horizontal. Skew correction can be achieved in two steps. First, estimate the skew

Angle θ_t and second, rotate the image by θ_t , in the opposite direction. The idea is based on the detection of DSL segments from the upper envelope. they applied Radon transform to the upper envelope to get the skew angle. Radon transform and the Hough transform are related but not the same. Then applied generic rotation algorithm for skew correction and then applied-cubic interpolation.

(D) Page Layout analysis

The overall goal of layout analysis is to take the raw input image and divide it into non-text regions and "text lines"—sub images of the original page image that each contains a linear arrangement of symbols in the target language. Some page layout algorithms are: RLSA, RAST etc. Existing page layout analyzers are described in. An algorithmic approach is described in with its limitation. Still this section is developing and thus become a major research field. Using page layout analysis in OCR helps to recognize text from newspaper, magazine, documents etc.

(E) Feature extraction

Feature extraction is an OCR method for classifying characters. With feature extraction all characters will be divided into geometric elements like lines, arcs and circles and the combination of these elements will be compared with stored combinations of known characters. This method provides much more flexibility than the formerly used pattern recognition and also copes well with variations in font style and size.

(F) Classification

Several approaches have been used to identify a character based on the features extracted using algorithms described in previous. In choosing classification algorithms, use of Artificial Neural Network (ANN) is a popular practice because it works better when input data is affected with noise. Methods are Decision tree, MLP, Kohonen Neural Network

Tesseract OCR Engine

Tesseract Engine is a well-known open source OCR engine that released under the [Apache License 2.0](#). using this to construct the OCR library for Android, and to implement a Android OCR application with it. OCR engines converts images of machine-printed form.

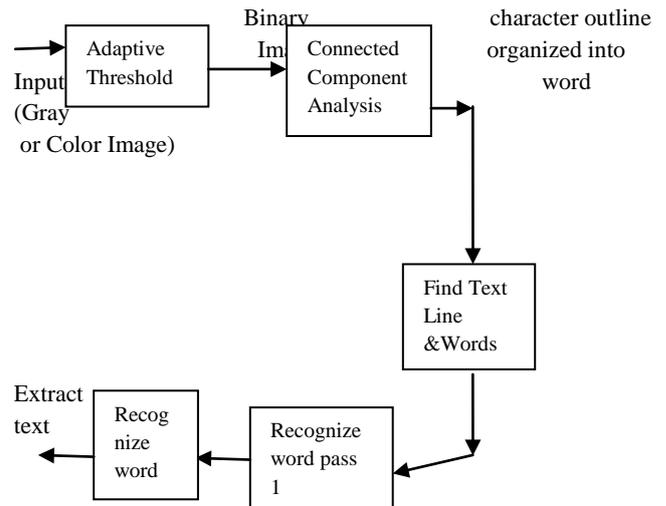


Figure 2: Structure of Tesseract

a) **Tesseract Android Tools:** Tesseract Tools for Android provides a set of Android APIs and build files for the Tesseract OCR and Leetonia image processing libraries. Android SDK: The Androids provides you the API libraries and developer tools.

b) **ADT:** Android Development Tools (ADT) is a plugin for the Eclipse IDE that is designed to give you a powerful, integrated environment.

c) **Eclipse:** In computer programming, Eclipse is a multi-language software development environment comprising a base workspace and an extensible plug in system for customizing the environment. It is written mostly in Java. It can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, FORTRAN, Haskell, Perl, PHP.

3.2 Result

The built-in OCR converts the text embedded in the captured image into Unicode text format. Also Android platform has been increasingly being common in accordance with its features like low cost, customizable, lightweight operating system. When we capture the image from any picture or any document with the help of camera then this application extract the numeric value just like phone number and save to backend server site and this application also include cloud computing feature and digital signature feature.

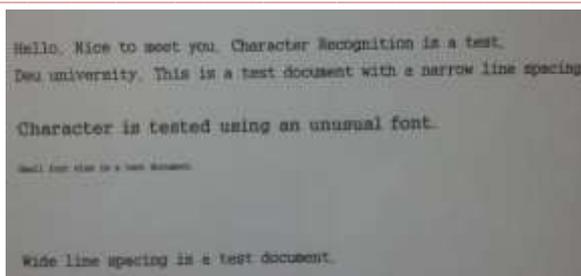


Figure:-3 original document for recognition

The recognition result is shown as follows. Fig :-shows original document with English language. After being captured by smart phone camera, the data is processed by binarization for object segmentation. Then we can see the recognition results on the screen of smart phone as shown in

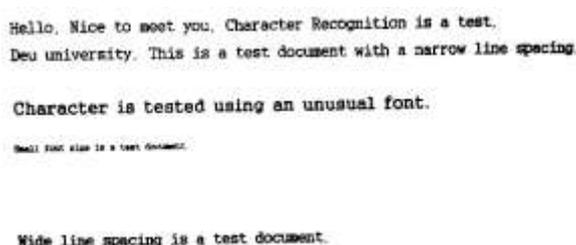


Figure.4. Binarization of captured data

Conclusion

In this paper, character recognition system was implemented by using the Android smart phones. The implementation process of the system was described to recognize the characters in the document using the camera screen. Photo data taken by a smart phone can be compared with the database of the system, then the characters can be recognized, the recognized character can be created to a text file to take advantage of the as applications of the Internet and pre-retrieval and various strategies Smart phones character recognition system does not need hardware such as a computer or a scanner. Therefore, there are the advantages that the recognition cannot be spatially restricted and simple character recognition is possible.

REFERENCES

[1] Argha Roy, Diptam Dutta, Kaustav Choudhury," Training Artificial Neural Network using Particle Swarm Optimization Algorithm" IJARCS SE Volume 3, Issue 3, March 2013 pp 43—434

[2] Amir Bahador Bayat Recognition of Handwritten Digits Using Optimized Adaptive Neuro-Fuzzy Inference Systems and Effective Features Journal of Pattern Recognition and Intelligent Systems Aug. 2013, Vol. 1

[3] Chirag I Patel, Ripal Patel, Palak Patel Handwritten Character Recognition using Neural Network International Journal of Scientific & Engineering Research Volume 2, Issue 5, May-2011

[4] Sushree Sangita Patnaik and Anup Kumar Panda Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter Applied Computational Intelligence and Soft Computing Volume 2012, Article ID 897127.

[5] Dileep Kumar Patel, Tanmoy Soml, Sushil Kumar Yadav, Manoj Kumar Singh," Handwritten Character Recognition Using Multiresolution Technique and Euclidean Distance Metric" JSIP 2012, 208-214

[6] .Offline Handwritten Character Recognition Techniques using Neural Network: A Review International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064 Volume 2 Issue 1, January 2013

[7] Gurpreet Singh Chandan Jyoti Kumar Rajneesh Rani Dr. Renu Dhir," Feature Extraction of Gurmukhi Script and Numerals: A Review of Offline Techniques" IJARCSSE Volume 3, Issue 1, January 2013 pp 257-263

[8] Majida Ali Abed, Hamid Ali Abed Alasadi," Simplifying Handwritten Characters Recognition Using a Particle Swarm Optimization Approach" European Academic Research, Vol. I, Issue 5/ August 2013 pp-532-552

[9] Argha Roy, Diptam Dutta, Kaustav Choudhury," Training Artificial Neural Network using Particle Swarm Optimization Algorithm" IJARCS SE Volume 3, Issue 3, March 2013 pp 43—434

[10] Amir Bahador Bayat Recognition of Handwritten Digits Using Optimized Adaptive Neuro-Fuzzy Inference Systems and Effective Features Journal of Pattern Recognition and Intelligent Systems Aug. 2013, Vol. 1

[11] Swagatam Das, Arijit Biswas, Sambarta Dasgupta, and Ajith Abraham Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications Hindawi Publishing Corporation Applied Computational Intelligence and Soft Computing Volume 2012, Article ID 897127

[12] J.L. Blue, G.T. Candela, P.J. Grother, R. Chellappa, C.L. Wilson, Evaluation of pattern classifiers for fingerprint and OCR applications, *Pattern Recognition*, 27(4), 1994, 485–501.

- [13] R. Smith, An overview of the Tesseract OCR engine, Proc. Int. Conf. Document Analysis and Recognition (ICDAR 2007), 2007, 629-633.
- [14] M. Seeger, Binarising camera images for OCR, Proc. Int. Conf. Document Analysis and Recognition), 2001, 54-58.
- [15] K. Wang, J. A. kangas, Character location in scene images from digital camera, *Pattern Recognition.*, 36(10), 2003, 2287-2299.
- [16] C. C. Chang, S. M. Hwang, D. J. Buehrer, A shape recognition scheme based on relative distances of feature points from the centroid. *Pattern Recognition*, 24(11), 1991. 1053-1063.
- [17] T. Bernier, J.-A. Landry, A new method for representing and matching shapes of natural objects. *Pattern Recognition*, 36(8), 2003. 1711-1723