DESIGN OF SMARTPHONE APPLICATION FOR CHARACTER LOCALIZATION AND RECOGNITION

Snehal Charjan*

Prof. R. V. Mante**

Dr. P. N. Chatur***

Abstract

The complete design of Character localization and recognition application system for Smartphone with four modules are explained. Images taken from camera of mobile device will be browsed from gallery and pre-processed. The text of these images will be accurately localized within the device in a fraction of a second using special localization method called "Low Complexity Localization method". Localized text sub-image will be fed for text extraction to the best OCR engine called "Tesseract". The text output is then fed to web to search for web search results. The result will be displayed on browser of mobile .The system can be deployed on android operating system based smartphones. More exciting applications can be developed over the text extraction method with a high performance while also being computationally inexpensive. In this paper we have described all modules and explained localization module in detail. We have explained android architecture to be useful for deployment also explained Tesseract engine to be used. Finally we concluded with a good design for Smartphone OCR application.

Keywords: Pre-process, Text localization, text extraction, OCR, android, Smartphone

* M.Tech 2nd year,

** Asst. Professor,.

*** Head of Department, Department of Computer Science and Engineering,Government College of Engineering,Amravati (Maharashtra), India

1. Introduction

Finding areas of text in a natural image is a difficult problem. One reason for this is that, from a texture and geometric standpoint, many objects (e.g., tree branches or electrical wire on a blue sky) resemble text. In addition, the text size, font, color, orientation and skew are generally unpredictable. In the following, we introduce an efficient method for finding text areas in natural images. The only assumption we make regarding the text is that it is written on a more or less uniform background using a contrasting color.[1] OCR stands for Optical Character Recognition, which is the process of taking an image, and being able to interpret the image and obtain textual data from it. OCR enables the program to literally scan the image and find text, which your program can then use elsewhere in its workings. The automatic localization of text within a natural image is an important problem in many applications. Once identified, the text can be analyzed, recognized, and interpreted. However, many objects in natural images, such as tree branches or electrical wires, are easily confused for text by existing optical character recognition (OCR) algorithms. For this reason, applying OCR on an unprocessed natural image is computationally expensive and may produce erroneous results. Hence, robust and efficient methods are needed to identify the text-containing regions within natural images before performing OCR [5]. Smart Phones have Internet access anywhere. The automatic text localization and recognition of text within a natural image is very useful for many problems. Once identified, the text can be used for many purposes. User can get current information about the product, place or boards. More exciting applications can be developed over the text extraction method with a high performance while also being computationally inexpensive. Design of such localization application is given in this paper.Tesseract is used in this method wich is best The handheld device equipped with a 3-megapixel camera, CPU 330MHz and RAM 128 MB capable of acquiring images of text to be readable by a human viewer. Images taken from camera of mobile device will be browsed from gallery and pre-processed. The text of these images will be accurately localized within the device in a fraction of a second. Localized text sub-image will be fed for text extraction. The text output is then fed to web to search for web search results. The result will be displayed on browser of mobile. So Objectives are

- browse image from gallery of mobile
- localize text with low complexity detection method

- Apply rapid character recognition
- Apply post processing algorithm
- feed extracted text to search engine to display search results

So, these objectives can be accomplished in four modules as shown in fig 1.1.In this paper design of modules and special localization method for faster localization is explained.

Fig. 1.1

2. Android architecture

Figure 2.1 shows the diagram of Android Architecture. The Android OS can be referred to as a software stack of different layers, where each layer is a group of several program components. Together it includes operating system, middleware and important applications. Each layer in the architecture provides different services to the layer just above it.

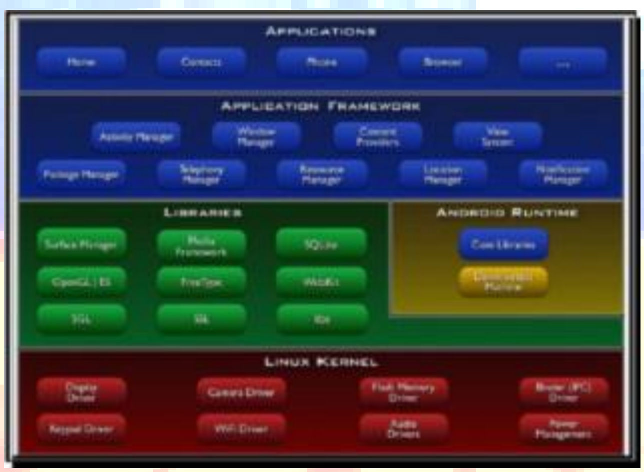

Fig.2.1

a. Linux Kernel

The whole Android OS is built on top of the Linux 2.6 Kernel with some further architectural changes. It is this Linux that interacts with the hardware and contains all the essential hardware drivers. Drivers are programs that control and communicate with the hardware. For example, consider the Bluetooth function. All devices has a Bluetooth hardware in it. Therefore the kernel must include a Bluetooth driver to communicate with the Bluetooth hardware. Android uses the Linux for all its core functionality such as Memory management, process management, networking, security settings etc. As the Android is built on a most popular and proven foundation, it made the porting of Android to variety of hardware, a relatively painless task.

b. Libraries

The next layer is the Android's native libraries. It is this layer that enables the device to handle different types of data. These libraries are written in c or c++ language and are specific for a particular hardware. Some of the important native libraries include the following**:** Surface Manager**:** It is used for compositing window manager with off-screen buffering. Off-screen buffering means you cant directly draw into the screen, but your drawings go to the off-screen buffer. There it is combined with other drawings and form the final screen the user will see. This off screen buffer is the reason behind the transparency of windows.

Media framework**:** Media framework provides different media codecs allowing the recording and playback of different media formats

SQLite**:** SQLite is the database engine used in android for data storage purposes

WebKit**:** It is the browser engine used to display HTML content

OpenGL**:** Used to render 2D or 3D graphics content to the screen

Android Runtime**:** Android Runtime consists of Dalvik Virtual machine and Core Java libraries.

Dalvik Virtual Machine : It is a type of JVM used in android devices to run apps and is optimized for low processing power and low memory environments. Unlike the JVM, the Dalvik Virtual Machine doesn't run .class files, instead it runs .dex files. .dex files are built from .class file at the time of compilation and

provides hifger efficiency in low resource environments. The Dalvik VM allows multiple instance of Virtual machine to be created simultaneously providing security, isolation, memory management and threading support.

Core Java Libraries : These are different from Java SE and Java ME libraries. However these libraries provides most of the functionalities defined in the Java SE libraries.[2]

c. Application Framework

These are the blocks that our applications directly interact with. These programs manage the basic functions of phone like resource management, voice call management etc. As a developer, you just consider these are some basic tools with which we are building our applications. Important blocks of Application framework are:

March 2013

IJMIE Volume 3, Issue 3 ISSN: 2249-0558

Activity Manager: Manages the activity life cycle of applications

Content Providers**:** Manage the data sharing between applications

Telephony Manager**:** Manages all voice calls. We use telephony manager if we want to access voice calls in our application.

Location Manager**:** Location management, using GPS or cell tower

Resource Manager**:** Manage the various types of resources we use in our Application

d. Applications

Applications are the top layer in the Android architecture and this is where our applications are going to fit. Several standard applications come pre-installed with every device, such as:

- SMS client app
- Dialer
- Web browser
- Contact manager

As a developer we are able to write an app which replace any existing system app. That is, you are not limited in accessing any particular feature. You are practically limitless and can whatever you want to do with the android. Thus Android is opening endless opportunities to the developer.[3]

3. System Design

The whole system design is as in fig 3.1

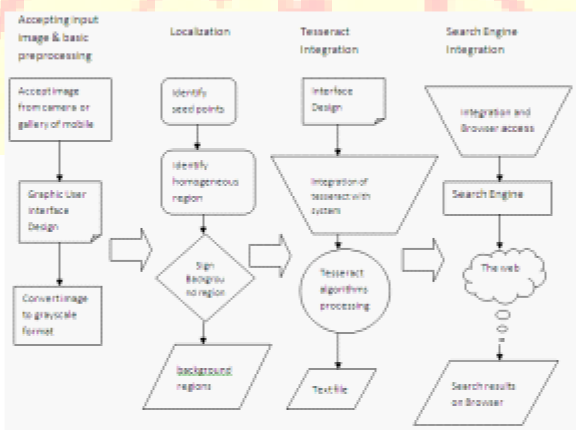

Fig.3.1

The four main stages of the project are as follows:

1. Accepting input image & basic preprocessing
2. Localization
3. Tesseract Integration
4. Search Engine Integration

3.1 Accepting input image & basic preprocessing

First module have three main tasks as accepting image from camera or gallery of mobile, Graphic User interface design, convert image to grayscale format. Modern day handheld devices are usually capable of capturing color images. A color image consists of color pixels represented by a combination of three basic color components viz. red (r), green (g) and blue (b). The range of values for all these color components is 0-255. So, the corresponding gray scale value f(x,y) for each pixel, which also lies between 0-255, may be obtained by using

Eq. 1.

f(x,y)=0.299 x r(x,y)+0.587 x g(x,y) +0.114 x b(x,y)

Eq.(1)

Applying this transformation for all pixels, the gray scale image is obtained and is represented as a matrix of gray level intensities, $I_{PxQ} = [\hat{f}(x,y)]_{PxQ}$ where P and Q denote the number of rows i.e. the height of the image and the number of the columns i.e. the width of the image respectively , $f(x,y) \in G_L = \{0,1,\dots,L-1\}$ the set of all gray levels, where L is the total number of gray levels in the image. Such a gray level image is fed as input to the proposed character recognition system.[4]

3.2 Localization

Identify seed points, identify homogeneous region, sign background region, background region. The text is assumed to be located on a homogeneous background using a contrasting color. A schematic representation of our proposed method of sign detection and text localization in natural images is shown in Fig. 3.2. In order to identify text containing regions, we first isolate areas within the image where luminance is homogeneous. We do this by calculating the homogeneity of equal sized blocks within the image and then identifying the homogenous blocks. The pixels in these homogenous blocks are the seed points used by a region growing algorithm to identify homogenous regions.

Once regions containing homogenous blocks have been identified, we search the hull of each homogenous region for holes that potentially contain text. In order for these holes to be labeled as text, they must be a minimum size and have an average intensity that contrasts against the corresponding detected back-ground region.

If no text region is found, the image is divided again into smaller blocks and the above steps are repeated. If no text region is found after the smallest block size is used, a relaxed search is performed. In the relaxed search, the threshold for the minimum size of a character is reduced and the hulls of all previous homogenous regions are once again searched for text. This final search is designed to find text areas that contain small characters.[5]

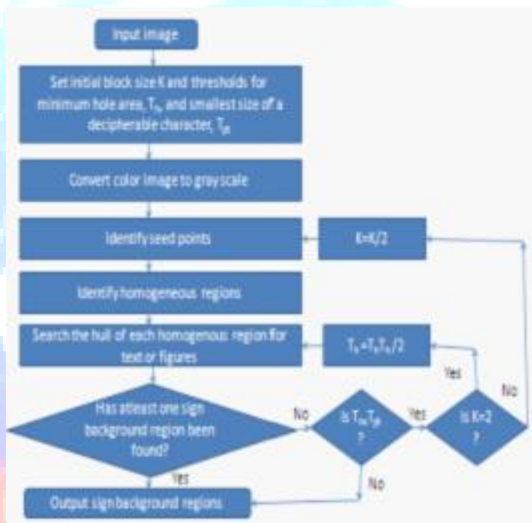

Fig 3.2

3.3 TESSERACT INTEGRATION

A lot of OCR software have been developed to accomplish this mission. Tesseract, originally developed as proprietary software at Hwelett-Packard between 1985 and 1995, now sponsored by Google, is considered to be one of the most accurate open source OCR engine currently available. It is capable of recognizing text in variety of languages in a binary image format. [6]Tesseract is a raw OCR engine. It has no document layout analysis, no output formatting, and no graphical user interface. It only processes a TIFF image of a single column and creates text from it. TIFF compression is not supported unless libtiff is installed. It can detect fixed pitch vs proportional text. The engine was in the top 3 in terms of character accuracy in 1995. It compiles and runs on Linux, Windows and Mac OS X, however, due to limited resources only Windows and Ubuntu Linux are rigorously tested by developers. Tesseract can process English, French, Italian, German, Spanish, Brazilian

Portuguese and Dutch. It can be trained to work in other languages as well. It can only handle left-to-right languages such as our Nepali script.

Tesseract is suitable for use as a backend, and can be used for more complicated OCR tasks including layout analysis by using a frontend such as OCRopus. Further integration with programs such as OCRopus, to better support complicated layouts, is planned.

Likewise, frontends such as FreeOCR can add a GUI to make the software easier to use for manual tasks.

It runs from the command line, and may be called with the command.tesseract image.tif output

Tesseract handles image files in TIFF format (with filename extension .tif); other file formats need to be converted to TIFF before being submitted to Tesseract. After lying dormant for more than 10 years, Tesseract is now behind the leading commercial engines in terms of its accuracy. Its key strength is probably its unusual choice of features. Its key weakness is probably its use of a polygonal approximation as input to the classifier instead of the raw outlines.[7]

3.4 SEARCH ENGINE INTEGRATION

Text file extracted from image by tesseract will be feed to the search engine on mobile and results will be shown on browser of mobile, so integration of overall system with search engine is last step.[2]

4. Conclusions

This paper presented the Design of Character localization and recognition application for Smartphone using best method from various available methods proposed for Text Localization and Optical Character Recognition by Gao and J. Yang, X. Chen, J. Yang, J. Zhang, and A. Waibel, P. Shivakumara, T. Q. Phan, and C. L. Tan using laplace approach, Kim, K. Jung, and J. H. Kim using support vector machines [8]to[11]. “Low Complexity Localization method“

explained is also fast. Compared to all OCR engine all methods tesseract has highest accuracy. Tesseract, sponsored by Google, is one of the most accurate open source OCR engine currently available. So fast application can be developed using design from this paper. Future work includes implementing Smartphone application for localizing text in image using "Low Complexity Localization method" and recognizing localized text using Tesseract which can be used for various functions.Future work include implementing this method along with text correction method on extracted text from tesseract.

References

[1] S. A. R. Jafri, M. Boutin, and E. J. Delp, "Automatic text area segmen-tation in natural images," inProc. ICIP, pp. 3196–3199, 2008.

[2] http://developer.android.com/about/index.html.

[3] Ms. Vaishali Zilpe, Dr. P. N.Chatur,"iNavigate-An Android Based Navigation Application", International Journal of Advanced Research in Computer Science and Electronics Engineering, Volume 1, Issue 4, June 2012

[4] Ayatullah Faruk Mollah, Nabamita Majumder, Subhadip Basuand Mita Nasipuri "Design of an Optical Character Recognition System for Camera-based Handheld Devices" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No1, July 2011

[5] Katherine L. Bouman, Golnaz Abdollahian,"A Low Complexity Sign Detection and Text Localization Method for Mobile Applications", IEEE Transactions on multimedia, VOL.13, NO. 5, OCTOBER 2011.

[6] Derek Ma, Qiuhau Lin, Tong Zhang "Mobile Camera Based Text Detection and Translation" Stanford University, Nov 2000.

[7] Ray Smith, "An Overview of the Tesseract OCR Engine," Google Inc. IEEE 0-7695-2822-8/07 ,2007.

[8] Gao and J. Yang, "An adaptive algorithm for text detection from natural scenes," inProc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, vol. 2, 2001.

[9] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection and recognition of signs from natural scenes,"IEEE Trans. Image Process., vol. 13, no. 1, pp. 87–99, Jan. 2004.

[10] P. Shivakumara, T. Q. Phan, and C. L. Tan, “A Laplacian approach to multi-oriented text detection in video,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 2, pp. 412–419, Feb. 2011.

[11] K. Kim, K. Jung, and J. H. Kim, “Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm,”IEEE Trans. Pattern Anal. Mach. In-tell., vol. 25, no. 12, pp. 1631–1638, Dec. 2003.

.

Author's Profile:

Snehal Charjan received her B.E. degree in Computer science and Engineering, from G.H.Raisoni College of Engineering, Nagpur, Maharashtra, India, in 2011, pursuing M.tech Degree in Computer Science and Engineering from Government college of Engineering, Amravati, Maharashtra, India. Her research interests include Pattern recognition, Optical Character recognition. At present, she is engaged in Character localization and recognition application for Smartphone.

Ravi V. Mante received his B.E. degree in Computer science and Engineering, from Government college of Engineering, Amravati, Maharashtra, India in 2006, the M.tech Degree in Computer science and Engineering, from Government college of Engineering, Amravati, Maharashtra, India, in 2011. He is Assistant professor in Government college of Engineering, Amravati, Maharashtra, India since 2007. His research interests include ECG signal analysis, soft computing technique, cloud computing. At present, He is working with Artificial neural network.

P. N. Chatur has received his M.E. degree in Electronics Engineering from Govt. College of Engineering, Amravati, India and PhD degree from Amravati University. He has published twenty papers in national and ten papers in international journal. His area of research includes Neural Network, data mining. Currently he is head of Computer Science and Engineering department at Govt. College of Engineering, Amravati.