Optical Character Recognition as an approach to Retrieve Information on Books

W.V.D. Fernando
Department of Computer Science and Engineering
University of Moratuwa, Sri Lanka
vimuth.10@cse.mrt.ac.lk

Abstract - ***This paper discusses an Android application that allows users to search for information on books more intuitively using several methods the main one being the use of optical character recognition (OCR) to read the title of a book from its cover. The application uses Tesseract OCR engine to read the text and OpenCV image-processing library to process the image to make it more suitable for OCR. This paper describes the design decisions in the process of development, the final result and the issues identified during the development of the application.***
Index: Android, OCR, Books

I. INTRODUCTION

Internet has become a very useful tool for shoppers today to find information and make better choices with regards to their money. But the vast amount of information in the Internet is distributed over a large number of different websites and sources. So a tool that can collect the necessary information to a single location and makes searching for information easier is very important. This paper describes an application developed for the Android mobile platform that makes searching for information on book easier, which in turn allows users to make better shopping decisions.

In today's world where everyone has a mobile device, we can use features previously unavailable such as cameras and mobile Internet connectivity to provide users with a better user experience when searching the web. While users were generally limited to searching the Internet using keywords, this influx of mobile devices provides more intuitive ways of looking for information.

The idea behind the application is to automate the process of searching for information on books. Users are provided with a variety of intuitive ways to look for information that replace going to a search engine and manually searching. This application provides the users with the following functionalities

I. Users can scan the cover of a book to search
J. Scan the barcode available on books to search
K. Search by manually entering the details about the book
L. Add books to "favorites" list and save details to view later

After the user scans or enters the necessary data, information about the book is gathered from multiple sources such as Google Books, Amazon Online Store and Goodreads social cataloging website and displayed for the user. Information gathered includes.

A Summary
B Reviews
C Prices on online stores
D Similar books

This application is developed for the Android platform. Android is the most popular mobile platform in use today. According to the International Data Corporation (IDC) Android devices make up about 80% of the worldwide mobile market [1]. Android is also open source and available under Apache License version 2.0. This restriction free development environment makes it very popular among developers all over the world. So Android was chosen as the platform for the app.

In the next section some literature found about OCR algorithms that were used to implement the OCR progress in the application are discussed. Following sections then discuss the design decisions associated with the project and its implementation details. Finally, the results of the implementation are described.

II. RELATED WORK

The biggest challenge in the development of this application is the process of extracting text from a book cover. Book covers are generally designed with highly stylized text and complex images. Filtering these out and identifying the text alone is a very complex task.

One proposed method is to use a technique named Stroke Width Transform [2]. Text normally consists of a stroke with a uniform width. So if we can identify the most likely stroke for every pixel in the image and identify continuous strokes we can recreate the letters in the image as in figure 1

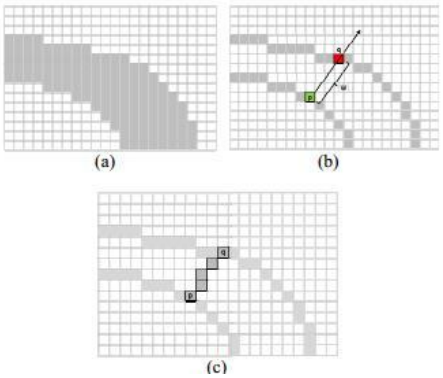

Figure 1 Stroke Width Transform

Another method that has been proposed is to identify contours in the image and find out possible candidates for letters from those contours [3]. First all the contours in the image are identified, then all the identified contours are examined and relevant contours that match a certain set of conditions are selected as possible letters. These parameters include the proportions of the contours, their size in contrast to the size of the full image, etc. An example of the process is given in figure 2. The first part shows all the identified contours and the second part of the image shows the contours that were selected as possible candidates for letters.

Figure 2: Contour identification. Top: all identified contours, Bottom: only possible letters

The second method was selected for this application, as it is easier to implement. The second method is also more suitable because it uses a low amount of processing power compared to the other method.

The main issue with this method as well as other methods for character identification is false positives. Contours that are not letters are also going to be mistaken for letters. This makes the final result very unreliable for complex images such as book covers.

III. DESIGN AND IMPLEMENTATION

A. Design

To implement the search by scanning the cover feature, Optical character recognition is used. Tesseract-ocr library is an open source OCR library currently maintained by Google. Tesseract Android Tools is a set of APIs that provide the functionality of Tesseract in the Android platform. This library is used in the application to scan the image for text due to its easy to use API.

Images captured using the camera have to be optimized first to make it more suitable for character recognition. To do the image processing OpenCV (Open Source Computer Vision Library) image processing library was used. OpenCV is an open source library designed for real time computer vision. It was designed originally by Intel and is available under the open source BSD license.

The Barcode reader functionality was implemented using ZXing ("zebra crossing") library.ZXing is an open-source, multi-format 1D/2D barcode image processing library in Java.

The final architecture of the application comprised of 6 main packages.

Image Processor
OCR
Barcode Scanner
Connection management
Database Management
Main controller

1) Image Processor module

This module captures the photo using the camera available. Then it fixes the image orientation. After that, the image is put through the procedure identified in the literature review to identify possible letters. Then a new image is created that only contain possible contours that can be letters. The new image is passed in to the OCR engine.

2) OCR Module

This package runs the Tessaract-ocr engine on the image provided by the image processer module. Then possible false positives are removed. Characters that are not numbers or

letters are removed. Single letters that don't belong to a word other than 'a' is also removed, as they are mostly false positives.

3) Barcode Scanner

Barcode scanner module scans the barcode using the camera and finds out the ISBN of the book, which is encoded in the barcode. Then the identified ISBN number can be used with online APIs to get the information about books.

4) Connection Management

Google Books API is used to search for the book and retrieve the information about the book based on the information gathered through the Barcode or the OCR module. Then the ISBN of the book is used to query other sources such as Amazon Web store and Goodreads for more information. This module handles all the communication that happens between the application and the online APIs.

5) Database Management.

To implement the "favorites" list, information about the books have to be saved in the phone memory. Saving and retrieving data is handled through the Database Management module. Data storage in the Android platform is handled through an SQLite database, so this module works with the available Android helper modules to provide data storage.

6) Main Controller

Main controller coordinates all the packages and manages the application. It handles user input and manages other packages.

Figure 3 shows the basic overview of the architecture of the application. It's broken down as such so that changes made to any part doesn't affect the other part of the application

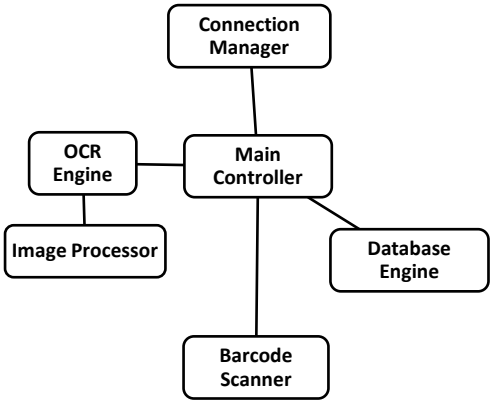

Figure 3: Basic architecture of the application.

B. Implementation

Due to the fragmentation of the Android platform many devices are still using outdated versions of the Android platform. Because of that the application was implemented so that it covers Android versions 3.0 to 4.3.

Modular design of the application allows for the creation of a loosely coupled application where independent modules can be altered without breaking others. As this application uses some third party APIs this modular design is important to handle changes to these APIs without breaking the application.

IV. RESULTS

The final application produced mixed results while the application worked well on newer phones with good hardware, OCR did not perform well on older hardware with limited resources. The main issue found here was the lack of proper memory handling. The image processing steps in the OCR algorithm require the image to be processed in many different ways with multiple copies of the image. This resulted in older phones running out of memory and crashing when processing complex photos. The low quality of the cameras available in older hardware also affected the final outcome of the application. As these cameras result in photos with low dots per inch (dpi), accurately identifying contours is very hard.

Another issue found with the character recognition was the inability of the OCR engine to identify obscure fonts and the difficulty in separating out the text from the background. This led to many issues of falsely identifying the background as a part of the text.

Even with newer devices the identified text was never 100% accurate. Most of the time only part of the text on the page was identified and the addition of false positives made the results less accurate.

Another hurdle found on the implementation was the lack of APIs to get the price of books for mobile platforms. Amazon Web Store, which is the most popular online bookstore in the world doesn't allow mobile apps to use their API to get the price of a book on their store.

But other than the OCR feature, all other features in the application worked flawlessly in all tested phones.

V. FUTURE WORKS

The future of this application depends on proper memory optimization and resource utilization. So the next step in the development needs to be focused on the efficient management of memory, which would allow the application to run on older devices.

Another aspect that can be improved is the

book recommendations. Features such as recommendations based on previous buying patterns and interested genre can be incorporated in to the application for a better user experience.

VI. CONCLUSIONS

Identifying text in complex backgrounds is a very complex task that needs more attention. This is a very important field that has a wide array of real world applications through multiple industries and could change the way we interact with computers in the future.

But a major issue in doing OCR in a mobile platform is the limited amount of computing hardware. This factor is a major limitation on the amount of computation that can be archived in a user-friendly time frame. This led to final results with incomplete optimizations that could have been improved further on a regular computing platform.

REFERENCES

[1] "Worldwide Quarterly Mobile Phone Tracker", International Data Corporation (IDC), Q3, 2013.

[2] Epshtein, Boris, EyalOfek, and Yonatan Wexler. "Detecting text in natural scenes with stroke width transform." *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.

[3] Yangxing, L. I. U., and Takeshi IKENAGA. "A contour-based robust algorithm for text detection in color images." *IEICE transactions on information and systems* 89.3 (2006): 1221-1230.