

ATMA: Android Travel Mate Application

Nitin Mishra

Department of Physics & Computer Science
Dayalbagh Educational Institute
Dayalbagh, Agra, India

C Patvardhan

Department of Electrical Engineering
Dayalbagh Educational Institute
Dayalbagh, Agra, India

ABSTRACT

Every day a Smartphone user may look for a new application dedicated for his need. Android makes it easier for consumers to get and use new content and applications on their Smart phones. This paper presents an extremely on-demand, fast and user friendly Android Application ATMA. ATMA stands for Android Travel Mate Application. This application is useful for native Tourists and Travelers who possess Android Smart phones. It enables Travelers and Tourists to easily capture the native country language Books pages, signboards, banners and hotel menus etc. The built-in OCR converts the text embedded in the captured image into Unicode text format. It also provides translation facility so that Tourists can translate the Native Language Unicode text into their own country language. This Application has an advanced search feature so that recognized as well as translated text can be used to copy, paste, share and search for travel related queries like museums, places, restaurants, books, culture, hotels, etc. There is no remote computing overhead because the application has built in OCR suite as well as Image Processing suite both installed in the Android device. It provides fast, robust and extremely high Quality performance because of having improved Auto focus behavior, continuous dynamic preview and improved noise tolerance feature.

General Terms

Android Application development, Image Processing

Keywords

Android, Tesseract OCR engine, Bing Translator API, Shirorekha Chopping, Tesseract Android Tools, Android SDK, Android NDK, Tess-two, Eclipse, ADT plug in

1. INTRODUCTION

The next generation open operating systems are not on desktops or mainframes but on the small mobile devices people carry every day. The openness of these new environments leads to new applications and markets and enables greater integration. As the demand grows for mobile phone applications, research in optical character recognition, a technology well developed for scanned documents, is shifting focus to the recognition of text embedded in digital photographs. Optical character recognition (OCR) [1] is a powerful tool for bringing information from our analog lives into the increasingly digital world. This technology has long seen use in building digital libraries, recognizing text from natural scenes, understanding hand-written office forms etc.

The Personal Computer and the Internet have found revolutionary ways to connect people, to entertain them and let them exchange information. But none of these is able to reach each person anywhere and anytime like the cell phone does. Current global mobile phone usage is 4 billion, which is equivalent to around half of the world's population. The cell phone has become very important in today's society. Google

has come out with the new open and comprehensive platform for mobile devices called Android. It includes an operating system, middleware, user-interface and applications. It is manufacturer spanning and able to run on every cell phone. By 2014 mobile internet usage should take over Desktop internet usage. Google's approach is to develop an operating system which can run on every mobile device and not for their specific mobile devices itself, enables them to reach as many people as possible.

In Indian market there was always a huge demand of such an Android Application, which would enable a Tourist sitting in a restaurant to capture, OCR and translate the Hindi language menu in his own country language to order his favorite dish without any need to know the language of menu. ATMA enables people to understand any native country language and also seamlessly fire a travel specific search to get the info regarding the country like museums, restaurants, hotels, culture, temples, books, movies, songs, etc. Image processing on mobile phones is a new and exciting field with many challenges due to limited hardware, limited CPU utilization, image Quality and connectivity. Phones with high quality cameras, powerful CPUs, and memory storage devices are becoming increasingly common. Most of the existing OCR technology for smart phones are likely to be remotely executed by the servers which do not face limited memory and limited processing power challenge moreover they suffer from networking delay. The proposed application would have image processing suite as well as OCR engine both installed on the smart phone implying no server communication overhead and quick and far better processing thus would not just only tackle the limited processing power and limited memory challenge but also provide improved performance. The Challenge is to develop an OCR for Mobile Phones with an improved OCR Algorithm that must be capable enough to handle the trade off between Processing time and Quality of results with proper segmentation issues and noise handling [2]. Moreover, Devanagari Script Recognition is an extremely tedious task in comparison of Roman Script because of having features like Shirorekha, modifiers and the conjunct characters [3, 4]. The Proposed Android Travel Mate Application enables Tourists and Travelers so that he/she can:

- OCR the native country language Books pages, Signboards, Banners and hotel menus etc.
- Translate the Recognized text into one of 12 languages.
- Copy, paste and share the Translated Text as well as Recognized Text.
- Search for text related museums, songs, images, videos, hotels and restaurants.

Currently ATMA supports English and Hindi OCR only. The improvements and the support for other languages like Urdu in ATMA are being carried out continuously.

2. ANDROID

The term “Android” has its origin in the Greek word *andr-*, meaning “man or male” and the suffix *-eides*, used to mean “alike or of the species”. This together means as much as “being human”. Android is a software stack for mobile devices which means a reference to a set of system programs or a set of application programs that form a complete system. This software platform provides a foundation for applications just like a real working platform. Android is intended to revolutionize the mobile market by bringing the internet to the cell phone and allowing its use in the same way as on the PC. Android is an operating system based on Linux with a Java programming interface. It provides tools, e.g. a compiler, debugger and a device emulator as well as its own Java Virtual machine (Dalvik Virtual Machine - DVM). Android is created by the Open Handset Alliance which is lead by Google. Android uses a special virtual machine, e.g. the Dalvik Virtual Machine. Dalvik uses special bytecode. Therefore one cannot run standard Java bytecode on Android. Android provides a tool "dx" which allows converting Java Class files into "dex" (Dalvik Executable) files. Android applications are packed into an .apk (Android Package) file by the program "aapt" (Android Asset Packaging Tool). To simplify development Google provides the Android Development Tools (ADT) for Eclipse. The ADT performs automatically the conversion from class to dex files and creates the apk during deployment. Android supports 2-D and 3-D graphics using the OpenGL libraries and supports data storage in a SQLite database. Every Android application runs in its own process and under its own userid which is generated automatically by the Android system during deployment. Therefore the application is isolated from other running applications and a misbehaving application cannot easily harm other Android applications.

2.1 ANDROID ARCHITECTURE

The Android Architecture layers are as follows:

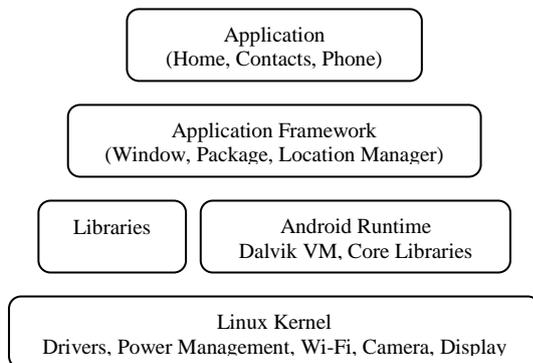


Fig 1: Android Architecture

2.1.1 Applications

Android comes with a set of built-in core applications including an email client, SMS program, calendar, maps, browser, contacts, and others which are written using Java.

2.1.2 Application Framework

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user. Underlying all applications is a set of services and systems, including:

- A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even a web browser.
- Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data.
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.
- A Notification Manager that enables all applications to display custom alerts in the status bar.
- An Activity Manager that manages the lifecycle of applications and provides a common navigation back stack.

2.1.3 Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

- **System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices.
- **Media Libraries** - based on Packet Video's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- **Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.
- **LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view
- **SGL** - the underlying 2D graphics engine.
- **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer.
- **FreeType** - bitmap and vector font rendering.
- **SQLite** - a powerful and lightweight relational database engine available to all applications.

2.1.4 Android Runtime

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

2.1.5 Dalvik Virtual Machine

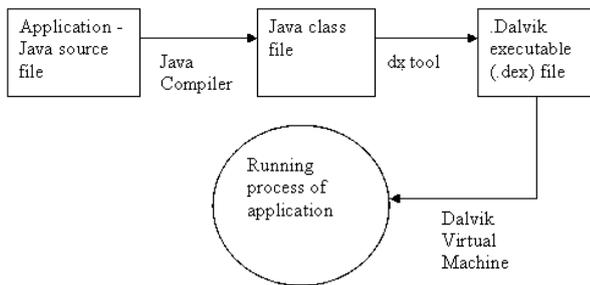


Fig 2: Dalvik Virtual Machine

The Dalvik virtual machine is simple Java interpreter machine, completely optimized for Android platform and which is developed to run on low-end memory mobile devices. One of the prominent aspects in Dalvik is its capability to run along an application compilation enhancing the runtime performance of the applications. DVM delivers a good performance over various stages of an application runtime environment, conserving more battery-power during long run of an application. A small drawback with Dalvik is its non compatibility with Java SE libraries, Java Applet, Java ME class libraries and Swing Java libraries. It uses its own Apache Harmony Java implementation libraries.

2.2 FEATURES OF ANDROID

The Android continues to lead the Smart Phone market because of having a great number of features. Android allows one to access core mobile device functionality through standard API available open -source. One can combine information from the web with data on the phone such as contacts or geographic location to create new user experiences. Android does not differentiate between the phones basic and third party applications even the dialer or home screen can be replaced. The SDK contains what one need to build and run Android applications, including a true device emulator and advanced debugging tools. Thus provides fast and easy development. It enables reuse and replacement of components. Dalvik virtual machine is optimized for mobile devices. Integrated browser based on the open source WebKit engine. Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional). SQLite is used for structured data storage. Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF). Android supports GSM Telephony (hardware dependent), Bluetooth, EDGE, 3G, and Wi-Fi (hardware dependent), Camera, GPS, compass, and accelerometer (hardware dependent) etc. Android possesses a rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plug in for the Eclipse IDE.

3. Smart Phone usage Statistics

The Recent Researches on Mobile phones have boost the Image processing applications development over it. The interesting facts about Smart phones usage statistics are as follows:

- The Global Mobile usage is 4 billion, out of which 1.08 billions are Smart Phones.
- As the Mobile Internet graph is moving up continuously, it is being estimated that by 2014 mobile internet usage should take over desktop internet usage.
- One half of all local searches are performed on the mobile devices because of the increasing use of mobiles. Their versatility is also growing.
- About 29% of mobile users are open to scanning a mobile tag to get coupons, as mobile tags can serve up coupons that can be redeemed instantaneously in store.
- About 91% of mobile internet access is to socialize compared to 79% on Desktops.
- Women aged 35 to 54 are the most active group in mobile socialization.
- Over 1/3 of total facebook users (about 600 million+ users) use facebook mobile.
- Over 1/2 of total twitter users (about 165 million users) use twitter mobile.
- About 200 million+ YouTube views occur on mobile devices per day.

As the Smart Phone usage has grown up to tremendous level. The Travel Mate Application for Android Smart Phones will surely foster the need of tourists and travelers. It will also boost the advancement in the future technologies and researches on the Smart Phones.

4. Tesseract OCR Engine

Today, Tesseract is considered one of the most accurate open source OCR engines available. Tesseract OCR Engine was one of the best 3 engines in 1995 UNLV Accuracy Test. Between 1995 and 2006 however; there was little activity in Tesseract, until it was open sourced by HP and UNLV in 2005. It was again re-released to the open source community in August of 2006 by Google [5]. Tesseract has ability to train for newer language and scripts as well [6]. A complete overview of Tesseract OCR engine can be found in [7]. While Tesseract was originally developed for English, it has since been extended to recognize French, Italian, Catalan, Czech, Danish, Polish, Bulgarian, Russian, Greek, Korean, Spanish, Japanese, Dutch, Chinese, Indonesian, Swedish, German, Thai, Arabic, and Hindi etc [8]. Training the Tesseract OCR Engine for Hindi language requires in-depth knowledge of Devanagari script in order to collect the character set. Moreover, Tesseract OCR Engine does not just require training of the collected dataset but also to tackle the character segmentation and clubbing issues based on the script specific features i.e. Shirrekha, maatra etc.

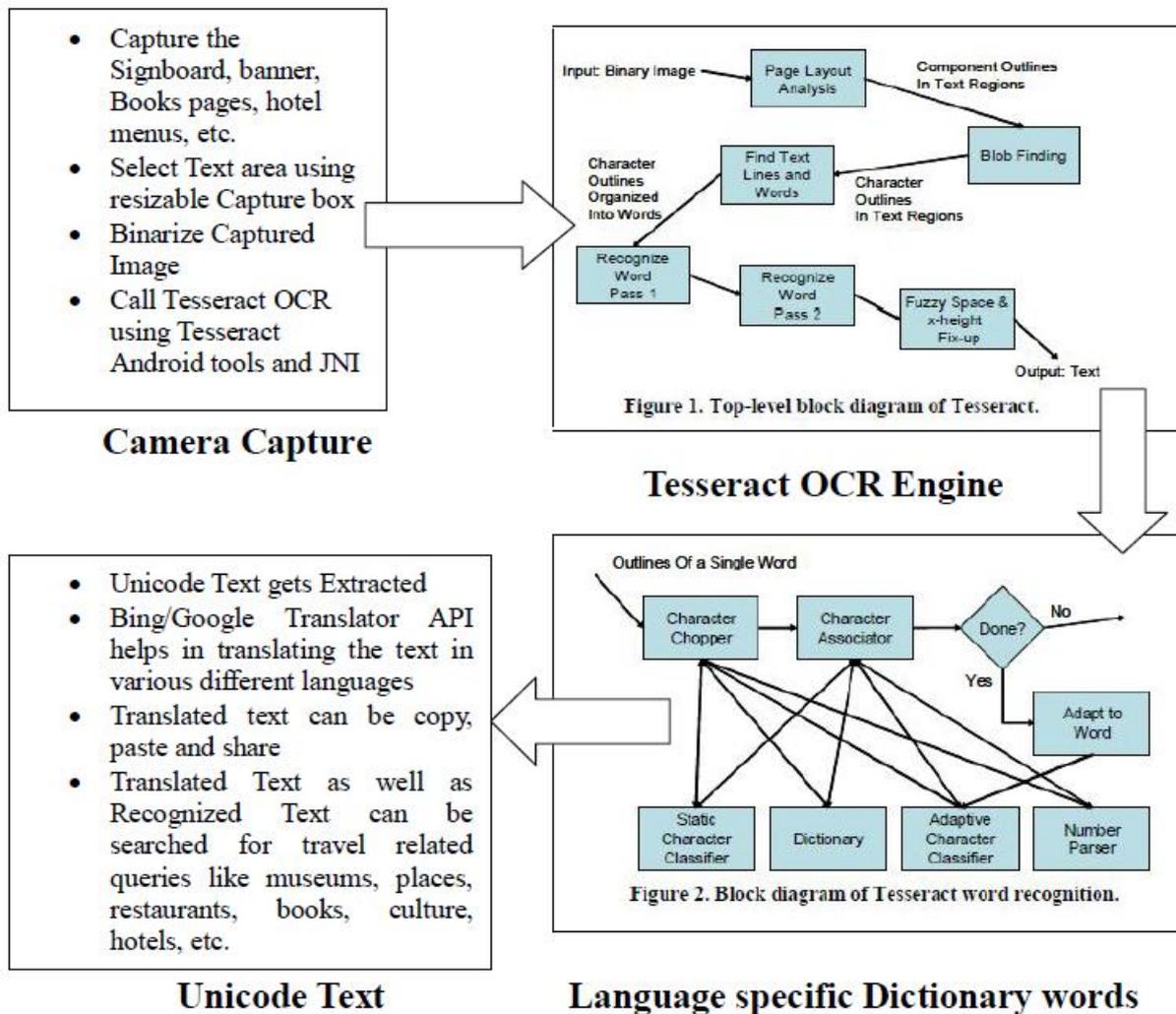


Fig 3: Methodology used

5. METHODOLOGY

As Fig 3 shows the application consists of four major components described below:

5.1 Camera Capture Module

In this module the user is allowed to resize the camera capture box by touching the box corners on the screen so as to capture the only concerned text image from signboard, banner and book pages [9]. The camera keeps continue auto focusing the image automatically throughout the session. Once the capture button is pressed the beep sound plays and the captured image is sent to Tesseract OCR engine module.

5.2 Tesseract OCR Engine Module

In this module, The Binarization of Captured Image takes place, after that the text layout is analyzed, Blobs are detected and finally words and lines are detected. The words are sent to a number of passes. In these passes each word is chopped into characters and characters are checked for the need of joining the broken characters or the breaking of associated characters. Finally chopped characters are recognized with the help of inbuilt fuzzy features matched to language specific training

data of Unicode characters. After each pass the words are matched back and forth with the Language specific Dictionary words.

5.3 Dictionary words Matching Module

In this module each group of sequential characters is searched for a dictionary based word match, which helps in identifying the word more accurately rather than just giving an meaningless word as result. Finally the recognized text is transferred to Unicode text Post processing Module.

5.4 Unicode Text Post processing Module

In this module, the recognized characters are displayed as Unicode characters and the user is allowed to translate the recognized text into his desired language available in the drop down list from settings. From there user may choose one of the two available translators from the drop down list including Bing Translator and Google translator [10]. Moreover the user can use the advanced search feature to search the travel specific related queries like museums, books, videos, songs, culture, images, places and hotels etc. related to recognized or translated text.

6. TOOLS USED

The Development of ATMA: Android Travel Mate Application required the following open source tools to be used:

6.1 Ubuntu 10.04 Operating System

Ubuntu is an ancient African word meaning 'humanity to others'. It also means 'I am what I am because of who we all are'. The Ubuntu operating system brings the spirit of Ubuntu to the world of computers. Ubuntu is a Linux Distribution that is extremely preferred for the Development of Applications.

6.2 Tesseract version 3.01

The introduction about Tesseract has been clearly explained in section 4. This section explains the steps of Training Data Generation. The basic guideline to prepare training data has very clearly explained in [11], which is followed to prepare the customized training data. It has following phases described below:

6.2.1 Training image generation

It involves the sufficiently spaced out single font specific text image creation. For each new font Tesseract OCR Engine suggests preparation of a new image file.

6.2.2 Box file generation

The information about the Bounding Boxes for all the characters present in the training image is generated for specifying language specific components in the box file. The default generated Bounding boxes can easily be edited using box file editors i.e. cowboxer tool etc.

6.2.3 Train file generation

Box file editors also allow editing the corresponding Unicode characters against appropriate Bounding boxes.

6.2.4 Character set file generation

Character set file is required to specify the information like uppercase, lowercase, digits, punctuation marks etc. about the Unicode characters. Since Devanagari does not distinguish upper and lower case characters, only digits and punctuation marks have to be specified.

6.2.5 Font properties selection

Font properties like italic, bold, fixed, serif etc. are required to be specified before training the data. In this work only normal fonts have been considered.

6.2.6 Feature extraction

This phase extracts the features of the shape of characters from the Training Data Image.

6.2.7 Clustering

This phase clusters the character shape features into prototypes.

6.2.8 Dictionary data preparation

Tesseract may use up to 5 types of Dictionary files which are converted into Directed Acyclic Word Graph (DAWG) files.

6.2.9 Post processing ambiguity removal

Editing the unicharambigs file allows removing the intrinsic ambiguity between two similar looking characters or their combinations by using a substitution rule.

6.3 Android SDK r16

The Android SDK [12] archive initially contains only the basic SDK tools. It does not contain an Android platform or any third-party libraries. In fact, it doesn't even have all the tools one need to develop an application. In order to start developing applications, one must install the Platform-tools and at least one version of the Android platform, using the

SDK Manager [13]. Platform-tools contains build tools that are periodically updated to support new features in the Android platform (which is why they are separate from basic SDK tools), including adb, dexdump, and others. To install Platform-tools, Android platforms and other add-ons, one must have an Internet connection, so if one plans to use the SDK while offline, please make sure to download the necessary components while online. To start the SDK Manager, please execute the program "SDK Manager.exe". From the command-line one can also directly trigger an update by Executing:

```
tools\Android.bat update sdk
```

An AVD Emulator may have default or one of the following skins:

- QVGA (240x320, low density, small screen)
- WQVGA (240x400, low density, normal screen)
- FWQVGA (240x432, low density, normal screen)
- HVGA (320x480, medium density, normal screen)
- WVGA800 (480x800, high density, normal screen)
- WVGA854 (480x854 high density, normal screen)

The development has been done using **SDK API 8** for Android 2.2 and onwards version as platform-tools for Target Devices, because **Android 2.2** supports following features:

- App installation on external storage media
- Data Backup
- OpenGL ES 2.0 Graphics
- YUV to JPEG compression
- Audio Manager
- Media Recorder
- Speech Recognition Engine
- Camera and Camcorder
- Flexible UI Framework

6.4 Android NDK r7

The Android NDK is a toolset that lets someone to embed components that make use of native code in Android applications. Android applications run in the Dalvik virtual machine. The NDK allows one to implement parts of applications using native-code languages such as C and C++. This can provide benefits to certain classes of applications, in the form of reuse of existing code and in some cases increased speed.

6.5 Eclipse Helios SR2

The Prerequisites Packages must be installed on the system as follows:

```
sudo apt-get install ia32-libs  
sudo apt-get install sun-java6-jdk
```

Create home directory for Android-specific instance of Eclipse [14]. It is recommended to have Eclipse configured separately for each project rather than to have a single instance loaded with every imaginable plug-in. e.g create a directory named ~/.eclipse-android for this instance. Unpack the Eclipse distribution in its home directory.

```
gunzip -c eclipse-java-helios-SR2-linux-gtk-x86_64.tar.gz |  
tar xopf -
```

This resulted in ~/.eclipse-android/eclipse/
Place the downloaded Android SDK file in the android development home directory. In a command line window, change to the android development home directory. Unpack the distribution.

```
gunzip -c android-sdk_r16-linux_x86.tgz | tar xopf -
```

The developer will end up with `~/android/android-sdk_r16-linux_x86/`. Where, the directory name contains the release number.

6.6 ADT plug in 16.0.0

Now, Install and configure the Eclipse ADT Plugin as follows:

```
ln -s ~/android/android-sdk-linux_x86 ~/android-sdk
```

6.7 Tesseract Android Tools 1.00

Tesseract Tools for Android [15] provides a set of Android APIs and build files for the Tesseract OCR and Leptonica image processing libraries. Tess-two is a fork of Tesseract Android Tools which can also be used for native building of Tesseract API for Android environment.

7. SNAPSHOTS

आगरा पर्यटन

Fig 4: Text to be recognized

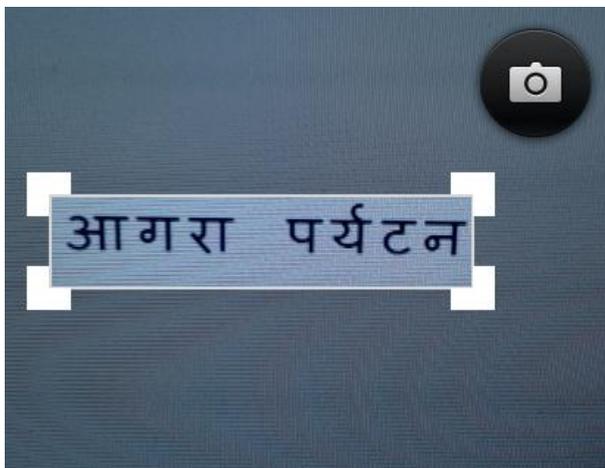


Fig 5: Capture Area Selection Box

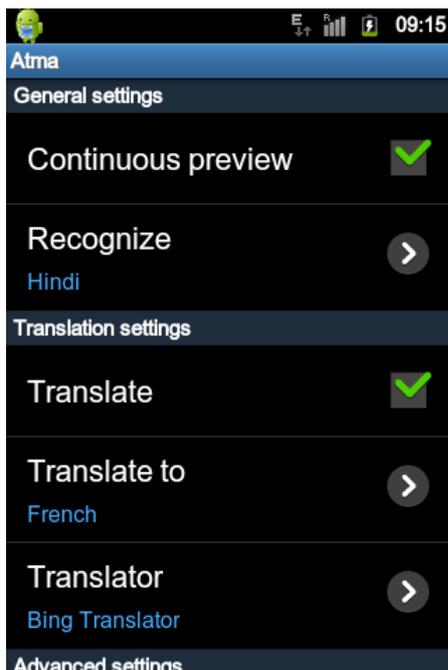


Fig 6: General Settings

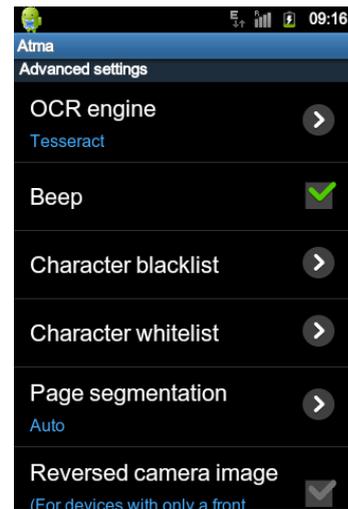


Fig 7: Advanced Settings



Fig 8: OCR and English Translation



Fig 9: OCR and German Translation



Fig 10: OCR and French Translation

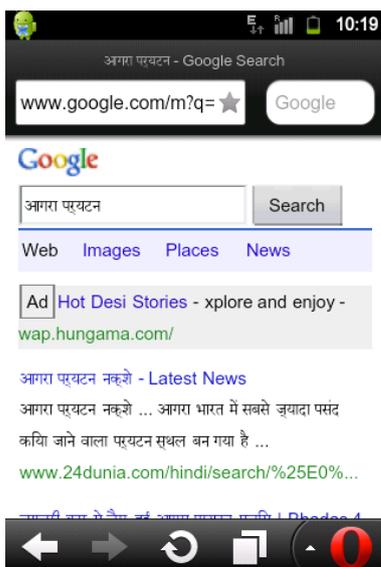


Fig 11: Copy, Paste, share and search for Recognized Text related web content

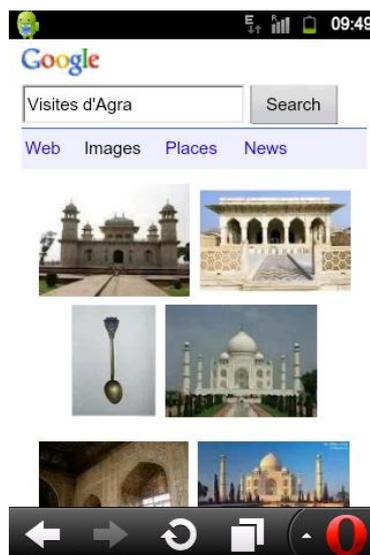


Fig 14: Copy, Paste, share and search for Translated Text related Images

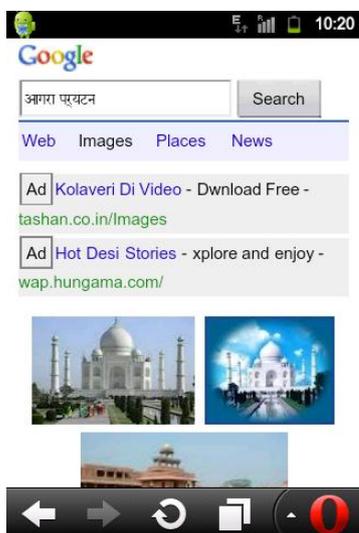


Fig 12: Copy, Paste, share and search for Recognized Text related Images

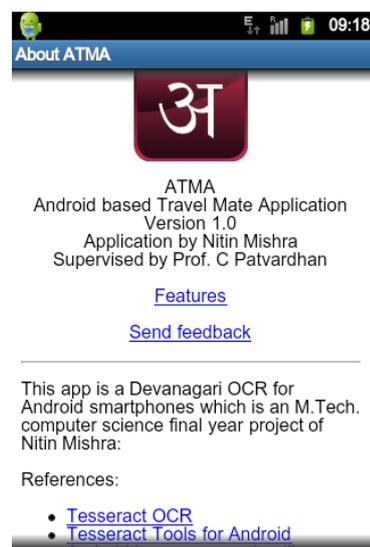


Fig 15: About ATMA and its Features



Fig 13: Copy, Paste, share and search for Translated Text related web content

8. EXPERIMENTAL RESULTS

The Tests were conducted on 100 samples each of book's pages, banners, signboards and posters of Hindi and English language captured under Light variations. The results are as follows:

Table 1: Average Recognition results

Language	English	Hindi
Avg. Mean Confidence per word	69	24
Avg. Time taken per word	153 ms	681 ms
Light and Noise susceptibility	LOW	HIGH
Average Accuracy	97.9%	79.2%

9. CONCLUSIONS

This Application provides fast, robust and extremely high Quality performance because of having improved Auto focus behavior, continuous dynamic preview, improved noise tolerance feature and no remote computing overhead.

10. REFERENCES

- [1] Bansal, V. and Sinha, R.M.K. "A Complete OCR for Printed Hindi Text in Devanagari Script", Sixth International Conference on Document Analysis and Recognition, IEEE Publication, Seattle USA, 2001, Page(s):800-804.
- [2] Saba, T., Sulong, G. and Rehman, A. "A Survey on Methods and Strategies on Touched Characters Segmentation", International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 1, No. 2, June 2010.
- [3] Pal, U., Chaudhuri, B. B. "Indian Script Character recognition: A survey", *Pattern Recognition*, vol. 37, pp. 1887-1899, 2004.
- [4] Jindal, M.K., Sharma, R.K., lehal, G.S. "A Study of Different Kinds of Degradation in Printed Gurmukhi Script", Proceedings of the International Conference on Computing: Theory and Applications (ICCTA'07), 2007.
- [5] Google code : <http://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html> (last accessed 8 January, 2012)
- [6] Smith, R. "An Overview of the Tesseract OCR" in proc. ICDAR 2007, Curitiba, Paraná, Brazil.
- [7] Open source Tesseract OCR Engine, available at <http://code.google.com/p/tesseract-ocr/>
- [8] Tesseract OCR training data downloads. Available at: <http://code.google.com/p/tesseract-ocr/downloads/list>
- [9] Simple Android Photo Capture. Available at: <http://labs.makemachine.net/2010/03/simple-android-photo-capture/>
- [10] Microsoft Translator Java API. Available at: <http://code.google.com/p/microsoft-translator-java-api>
- [11] Tesseract-OCR An OCR Engine that was developed at HP Labs between 1985 and 1995... and now at Google. Available at: <http://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract3>
- [12] Download the Android SDK. Available at: <http://developer.android.com/sdk/>
- [13] Android Developers Homepage. Available at: <http://developer.android.com/index.html>
- [14] Eclipse IDE for JAVA Developers. Available at: <http://eclipse.org/downloads/packages/eclipse-ide-java-developers/galileosr2>
- [15] A fork of Tesseract Tools for Android. Available at: <https://github.com/rmtheis/tess-two>