

Automatic Number Plate Recognition (ANPR) through smart Phones using Image Processing Techniques

Priyamvadha Krishnakumar
SSN College of Engineering

Abstract: Correct recognition of vehicles violating traffic rules and regulations is a major challenge in the present complex traffic environment. An automatic Number Plate Recognition (ANPR) system plays an important role in the detection of such events. In this paper, automatic number plate recognition (ANPR) was designed and implemented on Android mobile phone platform. First, the graphical user interface (GUI) for capturing image using built-in camera was developed to acquire car plate number. Second, the preprocessing of raw image was done using contrast enhancement, filtering, and straightening. Next, an optical character recognition (OCR) using neural network was utilized to extract texts and numbers. The proposed ANPR algorithm was implemented and simulated using Android SDK on a smart phone or PC. Future research includes optimizing the system for mobile phone implementation with limited CPU and memory resources, and geo-tagging of the image using GPS coordinates and online database for various mobile applications.

Keywords: ANPR, number plate, smart phone, android, image processing.

I. Introduction

Violation of traffic rules and regulations especially in third world countries is an area of concern. There has been an increase in the violation by the vehicles. The detection of the vehicles violating traffic signals is a major problem in this context. ANPR acts a solution for the above problem. Automatic Number Plate Recognition (ANPR) is simply the combination of hardware and software which have the ability to read the character and number on the vehicle's license plate. Commonly, the ANPR systems are used in various access control and traffic law enforcement, namely toll gate access, parking area access, speed trap and traffic light tress passing. There are numerous algorithms that can be used to improve the accuracy and processing time of ANPR, such as RGB or YCbCr, image binarization, image filter, fuzzy algorithm, template matching and neural network. Generally, the ANPR system works in five steps, image acquisition, image pre-processing, number plate localization, character segmentation and the last is optical character recognition (OCR) to recognize the individual character with the help of database stored for each and every alphanumeric character [3]. In this paper we eliminate the number plate localization step as we can capture only the number plate using Android mobile phone's camera. ANPR as an intelligent device eradicates the need for human involvement and has a high accuracy. There are numerous variables that need to be paid attention for gaining a high accuracy of the number plate recognition, for example, time of day, weather and angles between the cameras and the license plates [1]. Android as the popular mobile operating system has capability to complement those variables. Android offers open development environment which is built on open-source Linux kernel. Accessing the hardware resources is available to all applications through the API libraries. The third-party and native Android applications are constructed with the same APIs and executed on the same run time. Developers can eliminate and replace any native application with a third-party developer alternative [4]. The specification of the Android mobile phone used in this research can be seen in Table 1. Minimizing the computational cost, maximizing the recognition result and recognition time are the main objectives

Table I. Android Mobile Phone Specification Details

Component	Specification
CPU	800 MHz 1 Core ARM v6
GPU	Adreno 200
Chipset	Qualcomm MSM7227
Memory	278.34 MB
ROM	178.39 MB
Network Data	2G / 3G / HSDPA 7.2
OS	Android 2.3.7
Screen size	3.5 inch
LCD technology	TFT 16M colors multi touch screen
Resolution	HVGA (480 x 320) ~165 ppi pixel density
Camera Resolution	5 MP (2592x1944 pixels)
Digital/Optical Zoom	Up to 2x digital zoom
Shot Mode	Single / Smile / Continuous / Panorama

Photo Effects	Normal / Negative / B&W / Sepia
White Balance	Auto / Daylight / Cloudy / Incandescent / Fluorescent
GPS	AGPS
Wi-Fi	802.11b/g/n

II. Anpr On Android

Android is a Linux-kernelled open source mobile phone operating system, middleware with key applications. Android applications are primarily written in Java and compiled into Dalvik executable (DEX) format, a custom byte code. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. Developers have full access to all the frameworks and APIs that the core applications use and to Google-developed software libraries. In 2005, Google merged the mobile phone OS developer Android, and Google continued to develop it after the merger. Until November 5th 2007, open Handset Alliance which is composed of Google and other 33 mobile device manufacturers announces this operating system [5] [6]. Android’s software architecture is designed to simplify component reuse. Any application can publish its capabilities, and any other application can then use those capabilities, subject to security constraints enforced by the framework. The detail of the android architecture is shown in figure 1.

The Android software development kit (SDK) supports authoring applications with rich functionality. Like the iPhone, it can handle touch screens, accelerometers, 3D graphics, and GPS as well as collaboration among applications like email, messaging, calendars, social networking, and location-based services.

Many researchers have proposed and published ANPR in several ways including simulation using Matlab or developing using java and C language. With the android architecture and support of java library, it is possible to implement the ANPR on android devices.

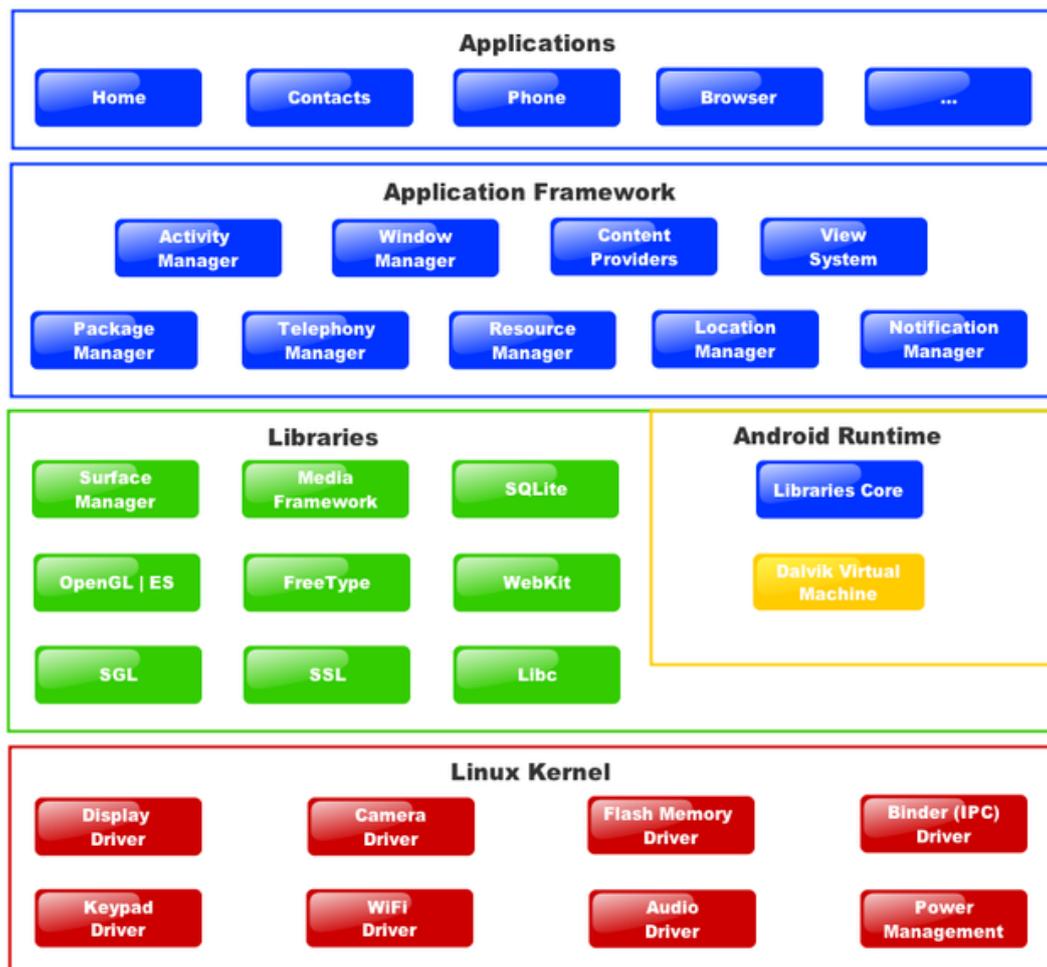


Figure 1. Android Architecture

III. System Design

The overall ANPR system can be subdivided into the software design and hardware design. In this section will discuss the both designs in detail.

A. Software Design

The most important part of this system is the software design. The software design uses series of image processing techniques which are implemented in Android mobile platform which is supported minimum API 5 or android 2.0 (cupcake).

The ANPR algorithm designed in this paper is roughly divided into four parts:

- Capture vehicle number plate image
- Image preprocessing and filtering
- Segmentation of the number plate image
- Recognize the numbers plate image using OCR algorithm.

The first step is the capturing of an image using the camera in the mobile phone. The images are captured in RGB format so it can be further processed easily for the number plate segmentation. The following step is preprocessing and filtering of the number plate image. It involves the following process.

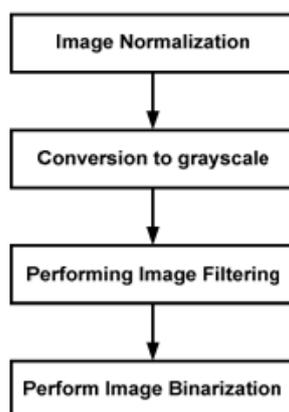


Figure 2: Image Preprocessing and filtering process

The processing time is proportional to the size of the input images. So the foremost task involves reducing the image size by performing image normalization. The images have been normalized to 420x312 resolutions. Image normalization not only reduces the size of the image but it also defines a fixed image size standard for all the successor processing. Usually the images occupy three channels i.e. Red, Blue and Green. For further processing, these three channels need to be converted to a single channel. This is done by the conversion of images to gray scale. The gray scale images only contain the intensity information with white and black lying on the two extremes. cvCvtColor() library function is used with CV_RGB2GRAY for converting images to gray scale. The gray scale converted images are presented in Figure 3. The next step involves filtering out the noise component from these images. Various blurring techniques are present including Gaussian, Median and Kalman that removes the noise and reduces sharp edge details present in the image. These techniques create a water color image of the real image, removing the sharp edge and discontinuities that are present within the image.



Figure 3

Among those techniques, Gaussian filter has a better performance in high noise image processing as compared to the Median filter. The iterative behavior of Gaussian filter makes it time complex. This complexity can be reduced by reducing the 2D Gaussian function parameter. In the current case, 7x7 Gaussian function is used. cvSmooth() function along with CV_GAUSSIAN type is used for the application of Gaussian blur.

The next process is the segmentation process. The algorithm of segmentation basically finds the maximum peak in the graph of vertical projection iteratively. The peak is treated as a space between characters, if it meets some additional conditions, such as height of peak, the algorithm then zeroes the peak and this process will repeat until no next space is found.

The final step is recognizing the character using Optical Character Recognition (OCR) algorithm by compared the image character that we have in second step against the alphanumerical database that we trained using artificial neural network (ANN) algorithm approach. Target for the ANN are 26 elements for ANN's letter and 10 elements for ANN's number. For ANN's number, there will be 10 neurons for hidden layer and output layer. For ANN's letter, there will be 20 and 26 neurons for hidden layer and output layer respectively. The value of element of each targets are all zero except one element on specific position which represent the number or letter. The output from the network will be two dimensional matrixes with size 26 x n and 10 x n for ANN's letter and ANN's number, respectively. The value of output will be in range 0 and 1. To be recognize, output should be processed first by converted the highest value to be 1 and other will be 0. Then, the location of element which has value 1 will be founded and the result will represent number or letter. For example, output from ANN's letter which has value 1 at first element will represent the letter A. Correlation method is used by the OCR to counterpart individual character and as a final point the number is identified and stored a variable as a string format. The string will be compared against the stored database for the vehicle authorization.

B. Hardware Design

The hardware design is all consisting in mobile phone device, including camera to capture the image, central processing unit which is inside the mobile phone to process the ANPR algorithm. In this research, we are using android mobile phone with the specification:

Processor: ARM v6 800MHz
RAM: 285 Mega Bytes
Screen Size: 3.5 Inches
Camera: 5 Mega Pixels
OS Version: 2.3.7 Ginger Bread

As the camera capture the number plate, the ANPR algorithm on the mobile phone receives the image and performs the processing, which yields the vehicle number. This number is then showing in the mobile phone's screen as string format.

IV. Construction And Implementation

After the system design has been completed, the system is ready for the construction and implementation. First of all the software design is constructed under android platform using android API 7 for android 2.1 Éclair. The system environment itself we used eclipse as android editor and JDK 1.6 for the java environment.

The construction step consists of graphical user interface (GUI) and constructs the algorithm into the android framework based on java programming language using java and android library. Furthermore, several Google's libraries have been imported in this construction step. Figure 4 shows the GUI for ANPR on android mobile phone. Finally as the construction finish, we implement the ANPR into real android mobile phone. We connected the android mobile phone to the computer using USB cable and compile the source code to install directly into android mobile phone.



Figure 4

V. Conclusion

This paper has generally discussed the design and implementation of automatic number plate recognition (ANPR) on android mobile phone platform. The proposed system can be redesigned for multinational car license plate in future. This ANPR algorithm has been tested over a wide range of images yielding a high accuracy rate.

References

- [1]. Muhammad Tahir Qadri and Muhammad Asif, "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition", 2009 International Conference on Education Technology and Computer, pp. 335–338. 2009.
- [2]. Neng-Sheng Pai, Sheng-Fu Huang, Ying-Piao Kuo¹, Chao-Lin Kuo, "License Plate Recognition Based on Extension Theory," 2010 International Symposium on Computer, Communication, Control and Automation, pp. 164–167, IEEE 2010.
- [3]. Thanongsak Sirithinaphong and Kosin Chamnongthai, "The Recognition of Car License Plate for Automatic Parking System", Fifth International Symposium on Signal Processing And its Applications, ISSPA '99, Brisbane, Australia, 22-25 August, 1999.
- [4]. S. Thanongsak, and C. Kosin, 1998, "Extracting of Car License Plate Using Motor Vehicle Regulation and Character Pattern Recognition," Proceedings of the 1998 IEEE APCCAS '98 Asia-Pacific Conference on Circuit and Systems, pp. 559-562, November 24-27, 1998.
- [5]. Sheng-Wen Chen, Chung-Huang Yang and Chien-Tsung Liu, "Design and Implementation Of Live SD Acquisition Tool in Android Smart Phone", 2011 Fifth International Conference on Genetic and Evolutionary Computing, pp. 157–162, IEEE Computer Society, 2011.
- [6]. "What is Android?", android developer, <http://developer.android.com/guide/basics/what-is-android.html>, retrieved 2012-02-11.
- [7]. Ondrej Martinsky, "Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems", Brno University of Technology, pp. 20–23, 2007