Automatic License Plate Recognition Using Mobile Device

Hung Ngoc Do, Minh-Thanh Vo, Bao Quoc Vuong, Huy Thanh Pham, An Hoang Nguyen, and Huy Quoc Luong
School of Electrical Engineering
International University, Vietnam National University
Email:{dnhung, vmthanh, vqbao}@hcmiu.edu.vn, {thanhhuysf, nguyenhoangan2503, luongquochuy1995}@gmail.com

***Abstract*— The rapid development of intelligent transport systems (ITS) brings us a safer and more convenient life. In this field, automatic license plate recognition (ALPR) plays an important role in many applications which have been deployed in reality such as stolen car detection, parking system management, and automatic transport charging system. The traditional ALPR methods usually need a high resolution camera to capture good quality images and a powerful computer to process the complex algorithms. However, with many great scientific breakthroughs in mobile device technology, we can easily buy smartphone/tablet which equipped strong central processing unit (CPU) and excellent camera to meet the requirements of ALPR. In this paper, we develop an Android program which processes the image captured by built-in camera of mobile device to have license plate number and save it into database for further applications. Open-source OpenCV libraries are imported into project for some image processing steps with purpose of programming time saving. We apply Tesseract engine and neural network which are two optical character recognition (OCR) methods to convert license plate image to machine-encoded text.**

***Index Terms*—Android, smartphone, ALPR, ITS, Tesseract, OCR, neural network.**

I. INTRODUCTION

In recent years, the great progress of technology in many areas such as sensors, embedded systems, control systems, communications, and signal processing has prepared a solid ground for deployment of ITS in reality. Although there are many types of ITS, they can be grouped into two classes including intelligent infrastructure systems and intelligent vehicle systems [1]. In intelligent infrastructure systems, the license plate numbers must be in machine-encoded text for easier store and processing, so OCR plays the important role in extracting data from license plate image. Because the complexity of OCR algorithms, the traditional methods usually use a combination of high resolution camera and powerful computer to perform recognition.

Since the mobile devices have been equipped strong hardware and installed mobile operating system, many complex applications for daily activity, entertainment, etc. have been transferred successfully from computers to them. The authors in [2] built a tagging application runs on Android devices. They used OpenCV libraries and image processing techniques like morphological and contour finding in object detection. An application runs on a smartphone sends the captured image to a computer via fourth generation (4G) mobile network for implementation of vehicle retrieval and recognition and receives the results was introduced in [3]. Although this application exploits the mobility of mobile phone, its disadvantage is the need of 4G network and computer. The authors in [4] developed a software runs on mobile device to recognize speech and synthesize voice for helping visually impaired people in Azerbaijani reading. A. Mutholid used two different methods template matching [5] and neural network [6] in OCR step for automatic number plate recognition (ANPR) systems based on smartphone.

In this paper, we develop an application runs on Android mobile device which can extract license plate number in machine-encoded text type from image captured by available camera of that device. Some image processing techniques are used to crop the license plate from image including noise reduction, adaptive binarization, and skew correction. Based on the properties of letters/numbers in the plate, we can separate each of them for recognition. We considered two OCR methods: Tesseract engine [7] and neural network. In order to evaluate the performance of the program, we created a database and used three measuring parameters including correct localization rate, OCR rate, and processing duration.

The rest of this paper is organized as follows. The system description which details the methodology to design the application is presented in section II. Section III shows the review of the method to build an Android application and the structure of our program. The results of our application are provided in section IV. Finally, section V concludes the paper.

II. SYSTEM DESCRIPTION

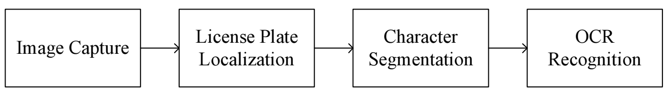

Fig. 1. System Overview

The overall system can be described in figure 1. The image captured by built-in camera of mobile device will be processed in three main stages including license plate localization, character segmentation, and OCR recognition.

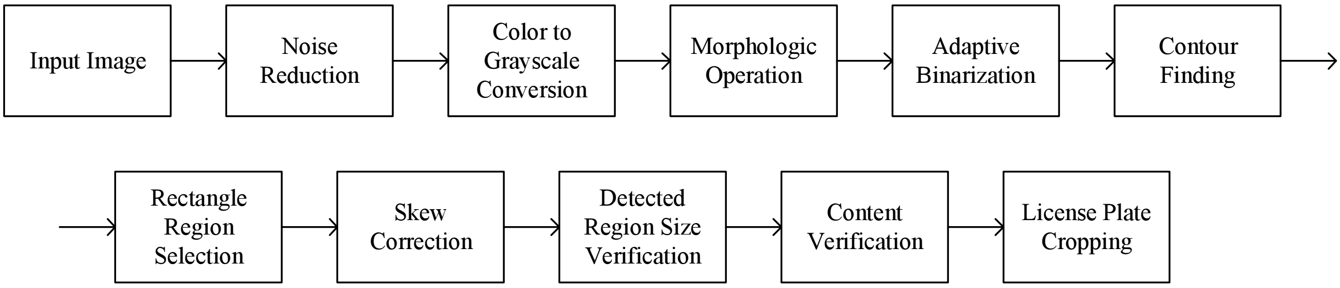

Fig. 2. License Plate Localization

A. License Plate Localization

The main purpose of the localization is to detect a region containing a license plate in the captured image. The procedure of license plate localization can be described in figure 2. In this program, we use adaptive bilateral filter to enhance the quality of the captured image. This filter proposed by Tomasi [8] is a nonlinear filter which removes noise efficiently while preserves the edge structures of images. In this method, the Gaussian distribution weights are used for neighboring pixels to calculate the new value of the considered pixel. The filtered image I^* can be determined as in equation (1),

$$I^* = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\parallel p - q \parallel) G_{\sigma_r}(\parallel I_p - I_q \parallel) I_q, \quad (1)$$

where $W_p = \sum_{q \in S} G_{\sigma_s}(\parallel p - q \parallel) G_{\sigma_r}(\parallel I_p - I_q \parallel)$ is normalization factor, σ_s and σ_r determine the filtering amount, S is considered window, G_{σ_s} and G_{σ_r} are Gaussian function kernels, and I is the intensity value. B. Zhang and J. P. Allebach introduced an improved version of this filter in 2008 [9] which is called adaptive bilateral filter. In this method, a new offset parameter ζ is introduced and the equation (1) is rewritten as follows:

$$I^* = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\parallel p - q \parallel) G_{\sigma_r}(\parallel I_p - I_q - \zeta \parallel) I_q. \quad (2)$$

The width of range filter σ_r and offset parameter ζ are used in locally adaptive way for having better result in smoothing and sharpening.

The background colours of license plates usually indicate the vehicle is property of individual, organization or government. However, colour information is not used in localization stage, so we convert the RGB into grayscale image. This step can be done by using equation (3),

$$Y = 0.299R + 0.587G + 0.114B, \quad (3)$$

where Y is a new grayscale value of the pixel and R, G, B are the values in three channels of each input pixel.

The captured image from camera of mobile device can contain small details and suffer from noises. Dealing with this problem, we apply morphologic operation which is a popular method for noise reduction, isolation of individual elements, and join of disparate elements [10]. Two basic operations of morphology are erosion and dilation. Dilation can fix the broken characters by expanding the object using structuring element to control. Equation (4) expresses this operation.

$$\begin{aligned} S = A \oplus B[i,j] = \max\{&A[i-r, j-c] + B[r,c], \\ &(i-r, j-c) \subset A, [r,c] \subset B\}. \end{aligned} \quad (4)$$

In order to remove the elements which have smaller detail than structuring element, we apply erosion operation which shrinks objects in an image. This operation can be described as in equation (5).

$$\begin{aligned} S = A \ominus B[i,j] = \min\{&A[i-r, j-c] - B[r,c], \\ &(i-r, j-c) \subset A, [r,c] \subset B\}. \end{aligned} \quad (5)$$

Figure 3(b) and figure 3(c) show the results of applying erosion and dilation operations.

Binarization is a process of converting grayscale image to binary image which can be classified into global method and local method. While the former scheme uses one threshold for all pixels, the latter scheme divides image into $m \times n$ blocks and calculates threshold for each block. In this paper, we apply the improved Bernsen algorithm introduced in [11]. Consider $(2w+1) \times (2w+1)$ size block with center at (x, y), the threshold for $Y(x, y)$ is determined by equation (6).

$$\tau(x,y) = \beta((1-\alpha)\tau_1 + \alpha\tau_2); \beta \in (0,1),\ \alpha \in [0,1], \quad (6)$$

where τ_1 and τ_2 are calculated as in equations (7) and (8), respectively.

$$\begin{aligned} \tau_1(x,y) = \frac{1}{2}(&\max_{-w \le k,l \le w} Y(x+l, y+k) \\ &+ \min_{-w \le k,l \le w} Y(x+l, y+k)), \end{aligned} \quad (7)$$

$$\begin{aligned} \tau_2(x,y) = \frac{1}{2}(&\max_{-w \le k,l \le w} \hat{Y}(x+l, y+k) \\ &+ \min_{-w \le k,l \le w} \hat{Y}(x+l, y+k)), \end{aligned} \quad (8)$$

and $\hat{Y}(x,y)$ is result of applying Gaussian filter to $Y(x,y)$.

In order to improve the performance of this process, we determine the text characteristic of each block before thresholding. The variance of ith block can be calculated by using equation (9).

$$\sigma_i^2 = \mathbb{E}\{(Y(u,v) - \mu_i)^2\}, \quad (9)$$

(a) Original Image

(b) Erosion Operation

(c) Dilation Operation

(d) Binarization

Fig. 3. Morphologic and Binarization Operations.

where $\mathbb{E}$ is the expected value operator, μ_i is the mean of the intensity values of ith block. This block is considered as non-text if $\sigma_i^2 < \sigma_{\text{th}}^2$, where σ_{th}^2 is the predefined threshold. Only the text blocks are applied improved Bernsen algorithm, the others will be set as background.

The next step is to find contour which is a list of point presenting a curve in an image. The algorithm proposed by S.Suzuki and K.Abe [12] is applied to find out the contours and the results are stored in a sequence. Among these detected contours, we only consider rotated rectangular shapes as candidates of license plate.

In ideal condition, the license plate images has rectangle shape. However, they can be shown in skew direction due to the way of using camera to capture image. In this paper, we use Radon transform in [13] to perform skew correction. In Radon transform, the Cartesian coordinates will be mapped into a distance and angle (s, θ) coordinates. The Radon function projects an image matrix into particular directions, so the result of projection of a two-dimensional function $f(m, n)$ is a set of line integrals. Each line in original image will be projected into one point in new plane. From position of these points, we can determine the inclination angle.

In Vietnam, license plate size (Height$\times$Width) of vehicles is standardized as follows: $110mm \times 470mm$ for car and $200mm \times 280mm$ for motorbike. These size characteristics are used to choose the detected rectangle regions with similar Height$\times$Width ratio. In some cases, several rectangle regions are detected, so we use another license plate property which is the number of characters in one license is 7, 8, or 9.

B. Character Segmentation

In character segmentation stage, we also process the cropped image in a similar way in previous section: binarization, contour finding and rectangle region detection. Figure 4a and figure 4b show the results of these steps. Based on the size property of characters in a license plate, only the regions have width in range of 35 pixels to 180 pixels and height in range of 110 pixels to 200 pixels are kept. This step will remove the regions which don't meet size standard criteria as in figure 4c. The result of next step, morphologic operation, is shown in figure 4c.

a) b) c) d)

Fig. 4. Character Segmentation

C. Character Recognition

1) Tesseract OCR: Tesseract OCR engine, released by HP, is a popular OCR engine has very high correct rate [7]. We can install this engine in almost popular operating systems like Windows, iOS, Andoid, etc. Furthermore, when combining with Leptonica libraries, it supports to convert from several image formats into machine-encoded text in over 60 languages. However, to get better results, we used a collection of real license plate character images to have a good trained data file to import to Tesseract engine.

2) Neural Network OCR: Kohonen networks are one of the well-known neural networks which have self-organizing property. This helps Kohonen networks can show the most natural learning ways where input pattern are undefined [14]. In this project, Kohonen networks are used for OCR stage in which the input data is a matrix where values determined by mapping the image to binary matrix.

III. Android Application Development

Android is a popular open source mobile operating system based on the Linux kernel which was released by Google under the Apache license [15]. In this project, we used Java language to write program in Android studio which is a popular integrated development environment (IDE) with built-in Android development tools (ADT) [16]. In order to compile and embed the functions written in C/C++ language, Native Development Kit (NDK) toolset is installed and configured in project.

A. OpenCV Library

OpenCV is developed by the researchers of Intel and the first version was announced at the IEEE conference on computer vision and pattern recognition in 2000 [17]. It is an open source library including over 1000 functions written in C/C++ language. These functions implement many tasks from basic to advanced in computer vision field.

B. SQLite

Structured Query Language (SQL) is a popular programming language for data management in relational databases. We can do many tasks like insert, update, delete, get queries, etc. SQLite is a scaled-down version of SQL which is used for applications run on systems with restricted hardware like smartphone and tablet [18]. We designed database to save number plate and also can export the database into excel file and upload to cloud storage.

IV. RESULTS

In this section, the results of our project are shown and discussed. In order to evaluate the performance of this project, we created a license plate image database of car and motorbike which includes 200 images for each type. The smartphone was used for image capturing and testing has 13 megapixel camera, 2GB RAM, and Quad-core 1.9 GHz CPU.

TABLE I
LOCALIZATION RESULTS

Vehicle Type	Standard Size (mm)	Localization Rate (%)
Motorbike	200×280	98
Car	110×470	96

Table I shows the results of license plate localization when we test all image database. It is clear that the correct rates of both vehicle types are similar and reach over 96%. The results will fluctuate around these values when we test application with images captured in different light conditions and quality of license plate (dirty, deformed, etc.).

TABLE II
OCR RECOGNITION RATE AND PROCESSING DURATION

Vehicle Type	Tesseract OCR		Neural Network OCR	
	Correct Rate (%)	Average Processing Duration (mS)	Correct Rate (%)	Average Processing Duration (mS)
Car	97	619	94	449
Motorbike	98	624	95	455

Table II shows the correct recognition rate and average processing duration of two OCR methods: using Tesseract engine and using neural network. We can see that the correct recognition rate of Tesseract scheme is higher than neural network scheme and reaches 98% for Motorbike type. The processing duration depends on the hardware and working status of the testing device. In this test, we closed all running applications before opening the testing program. Although neural network shows shorter processing duration, both methods have that measured data is less than 1 second, so we can receive the results immediately.

V. CONCLUDING REMARKS

We proposed and built an Android application for recognizing the license plate number. The main advantage of this application is its mobility characteristics while reaching high correct recognition rate and short processing duration. Some applications can be developed by using this results such as verifying the license plate number, checking insurance submission status, parking management system, etc.

REFERENCES

[1] C.-N. E. Anagnostopoulos, "License plate recognition: A brief tutorial," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 1, pp. 59–67, Jan. 2014.

[2] G. Savitha, P. S. Venugopal, Sarojadevi, and N. Chiplunkar, "An approach for object detection in Android device," in *International Conference on Signa and Image Processing*, Jeju Island, Korea, Jan. 2014, pp. 9–142.

[3] W. Pei, X. Jia, X. Huang, M. Lu, and Y. Fu, "A vehicle retrieval and recognition system based on smart phones," in *International Congress on Image and Signal Processing*, Shenyang, China, Oct. 2015, pp. 527–532.

[4] K. Aida-zade, E. Mustafaev, and J. Hasanov, "Intelligent reading system based on mobile platform," in *International Conference on Problems of Cybernetics and Informatics*, Baku, Azerbaijan, 2012, pp. 1–4.

[5] A. Mutholib, T. S. Gunawan, J. Chebil, and M. Kartiwi, "Optimization of ANPR algorithm on Android mobile phone," in *the IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, Kuala Lumpur, Malaysia, Nov., pp. 1–5.

[6] A. Mutholib, T. S. Gunawan, and M. Kartiwi, "Design and implementation of automatic number plate recognition on Android platform," in *International Conference on Computer and Communication Engineering*, Kuala Lumpur, Malaysia, 2012, pp. 540–543.

[7] R. Smith, "An overview of Tesseract OCR engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR)*, Parana, Argentina, 2007, pp. 629–633.

[8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision*, Bombay, India, Jan. 1998, pp. 839–846.

[9] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE transactions on Image Processing*, vol. 17, no. 5, pp. 664–678, May 2008.

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2007.

[11] Y. Wen, Y. Lu, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Transactions on Intelligent transportation systems*, vol. 12, no. 3, pp. 830–845, 2011.

[12] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 32–46, 1985.

[13] S. R. Deans, *The Radon Transform and Some of Its Applications*. Dover Publications, 2007.

[14] T. Kohonen, *Self-Organizing Maps*. Springer, 2001.

[15] W. Jackson, *Learn Android App Development*. Apress, 2013.

[16] M. Drongelen, *Android Studio Cookbook*. PACKT publishing, 2015.

[17] S. Brahmbhatt, *Practical OpenCV*. Apres, 2013.

[18] J. Wei, *Android Database Programming*. Packt Publishing Ltd., 2012.