International Journal of Engineering & Technology, 7 (3.24) (2018) 566-569

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Design and Implementation of Lightweight Vehicle License Plate Recognition Module utilizing OpenCV and Tesseract OCR Library

[1]Yong Gyu Jung, [*2]Hee-Wan Kim

[1]Dept. Of Medical IT, Eulji University, 553Sangeong-daero, Sujeonggu, Seongnamsi, Gyeonggido, 13135 Korea
*[*2]Division of Computer-Mechatronics, Shamyook University, 815 Hwarangro Nowongu, Seoul, 01795 Korea*
**Corresponding author Email: hwkim@syu.ac.kr*

Abstract

Background/Objectives: In order to recognize the license plates automatically, we design and implement a vehicle license plate recognition module that extracts characters of license plate area using open source OpenCV and Terreract OCR library.

Methods/Statistical analysis: The static image was binarized using OpenCV 's binarization function. After binarizing the image by adjusting the pixel values between adjacent pixels, the candidate region judged to be a license plate was derived. The final candidate was derived according to the proposed algorithm in the candidate region. The extracted plate area was analyzed by using the Tesseract OCR library, and characters were extracted as a character string.

Findings: The vehicle license plate recognition module relates to character recognition in the field of computer vision. In this paper, we designed and implemented a module that recognizes a license plate by using open source, applying a proposed algorithm to a moving object as a static image. The proposed module is a relatively lightweight software module and can be used in other applications. It is possible to install the camera at the entrance of the apartment and can read the license plate to identify whether it is a resident or not. When speeding and traffic violations occur on the highway, the vehicle numbers can be automatically stored and managed in the database. In addition, there is an advantage that it can be applied to various character recognition applications through modification of a slight algorithm in the module.

Improvements/Applications: In addition to character recognition, the OpenCV library can be applied to various fields such as pattern recognition, object tracking, and motion recognition. Therefore, we will be able to create technologies corresponding to various services that are becoming automated and unmanned.

Keywords: *OpenCV, Terreract OCR Library, Static Image, Plate Recognition, Binarization*

1. Introduction

Compared to the past, modern people enjoy the benefits of civilization developed in daily life. Now people are accustomed to unmanned automation, and people feel it is annoying when they do it themselves. The automation of these services is expected to develop in various forms in the future. The development of the artificial intelligence field is the original technology that made the modern service form. Computer vision, one of the artificial intelligence fields, has had a great influence on the development of this unmanned system. Computer vision is the field of machine vision, which means to interpret data from static images by performing digital image processing. In this paper, we propose a module that obtains a static image from a captured image of a vehicle, extracts a car license plate area, and extracts a character string within the license plate area. Since the license plates must be extracted from the entire image, there are many calculations to be processed, which greatly affects the performance of the entire system. In the method of extracting license plate area, Hough transformation[1] is applied to a binary image to search vertically and vertically edges, and extraction is performed using information of width and height ratio of license plate area [2],an adaptive background subtraction technique to separate vehicles from the background[3], headlight detection method includes high-intensity region detection and classification for cars and bikes[4], methods of identification of the scene area containing traffic signs[5],and methods of using color image information [6,7]. In this study, we used various libraries provided by OpenCV[8] to perform image processing tasks required at each stage. We used OpenCV'scvCaptureFromFile() function to obtain a static image and binarize it. Since binarization identifies the license plate area, the color is meaningless, so it is a process that shows the boundaries of the area as black and white. The next step is to label the binarized image by adjusting pixel values between adjacent pixels. The labeling derives candidate regions that are considered to be license plates from the separated regions. After labeling, the candidate region of the license plate is displayed using OpenCV 'scvDrawRect () function, and the final candidate is derived according to the algorithm applied in the module of the candidate region. Using the Tesseract OCR library as a derived plate area, characters in the image are analyzed and characters are extracted as string values. We saved the derived area as an image file using OpenCV's cvSaveImage () function. In the last step, we could output the recognized string value from TessBaseAPI object of Tesseract OCR. Using the proposed module, it will be easy to install and operate the system according to the scale.

2. OpenCV and Tesseract OCR

2.1. Development Environment for Implementing Opencv

OpenCV(Open Source Computer Vision) is an image processing library created by Intel. Many algorithms from basic image processing to advanced image processing are implemented as functions. OpenCV is most commonly used in the C ++ and C # environments for handling image processing, but it has been developed for use in the Java environment. In other words, it is a library that can be used in an OS of Java environment such as Android. In order to utilize OpenCV in Android OS, OpenCV library which is made in C ++ is compiled to be recognized by Java language through NDK (Native Develop Kit) for implementing JNI interface in advance[9]. OpenCV for providing effective solutions for complex image processing and vision algorithm for real time application for UG and PG students projects. Computer Vision (CV) applications require extensive knowledge of digital signal processing, mathematics, statistics and perception [10, 11].The compiler includes Eclipse with the SDK plug-in for writing the built libraries into Android application programs. When Open CV is used on a PC, Open CV is basically made in C ++ or C # language, and libraries can be included in the program project. Since Android's SDK language is Java, it is necessary to re-distribute programs that are written in C ++ so that they can be used in Java Native Interface (JNI) through NDK (Native Development Kit).Figure 1 shows the structure of the JNI that acts as a gateway between source code and JAVA.

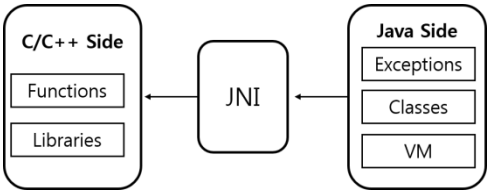

Figure 1: Location of JNI

2.2. Ocr

Optical character recognition converts characters included in the image into characters. Typically, the text of the scanned image includes printed letters, handwritten letters, and the like. The character is scanned to analyze the image and then the character is obtained. Digitally processing printed and handwritten characters, then storing them on a computer, searching them, and launching them is an area of everyday computer vision and pattern recognition research. In the early days of the optical character recognition system, we used a method of recognizing one letter of each word in turn, and only one type of font could be used in the character recognition process. Most OCR systems now handle almost all fonts very accurately. With the development of pattern recognition technology, the leading OCR system recognizes non-character elements in well-organized texts. Non-letter elements include paragraphs, multi-level, and visuals.

2.3. Tesseract OCR

Tesseract is a well-known OCR engine developed by Ray Smith at HP Labs from 1985 to 1995 [12]. The TETRECT OCR engine supports over 60 languages and different image formats. It was one of the top three OCR engines in the 1995 UNLV Accuracy Test. Since then, Google has made significant improvements. It is now the most accurate open source optical character recognition engine. Tessalite is available for Windows, Mac OS X, Linux, Android, and iOS. However, in the case of engines, only command line tools are provided. EmguCV provides Tessnet2's Net wrapper, tessnet2, which is an OCR engine to enable the project to use teletext through the EmguCV library [12, 13]. However, this OCR wrapper can only be run on Windows, and is incompatible with Linux and other operating systems.

3. Design and Implementation of Module

A process of extracting a character string from a captured image of a vehicle through a digital image processing process will be described. We used an open source library called OpenCV to build this vehicle license plate recognition module. We also used the Tesseract OCR library for string extraction.

When the captured image of the vehicle is prepared, the first step is to obtain a static image. OpenCV'scvCaptureFromFile () function was used to capture a single frame from the image.

The key idea of license plate recognition can be divided into two stages. First, car license plate detection is performed. Second, optical character recognition is used in the area to obtain license plate letters and numbers. More specifically, the following five basic procedures [14] must be followed in order to recognize license plates:

-License plate localization: Detects and separates license plates in a given video.

-Normalization: Adjust the size, dimension, contrast, and brightness of the plate.

-Character segmentation: Gets each individual character in the region.

-Optical character recognition: recognizes characters.

-Parsing: Compare license plate rules.

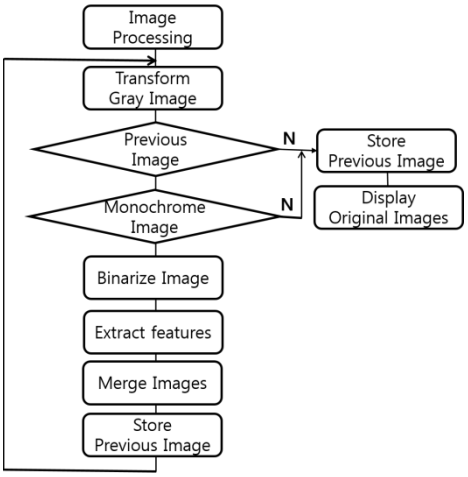

Figure 2:.Algorithm of motion detection

Figure 1 shows a flowchart for recognizing a moving object. Removes the contents of MIME from the header parse among the data transmitted from the surveillance camera, passes the length of data excluding the start and end parts of the image to Parse Content Length, and acquires image data. If there is an image, pass the factor to the image processing processor to process the image, and acquire the image again if there is no image. In order to track moving objects, we used a method of removing the background to recognize moving objects. In order to remove the background, the pixel values of the image obtained in the previous frame are stored in and the pixel values of the image obtained in the current frame are stored in. The difference between two images is defined in equation (1), and the difference image is obtained by taking absolute values to normalize from 0 to 255.

$$h(x,y) = |g(x,y) - f(x,y)| \quad (1)$$

Here, denotes a difference image, denotes a previous image frame, and denotes a current image frame. When the background is removed, the difference image remains. Then, the background image is binarized. Find feature points in binarized images. If an image with a feature point is matched with a current image, an

image in which a moving object is recognized can be obtained.

A static image can be obtained as shown in Figure. 3 by an algorithm that recognizes a moving object.

Figure 3: Static image from original image

The next step is to obtain a static image and binarize the image. Since the acquired image is a color image, it is converted into a monochrome image for easy image processing. A difference image of an image frame is obtained through a monochrome image. This method has a feature to detect boundaries of moving objects. The binarized image by the threshold value T is obtained by the equation (2).

$$f\big(h(x,y)\big) = \begin{cases} 255, & if\ h(x,y) > T \\ 0, & if\ h(x,y)\ \leq T \end{cases} \quad (2)$$

Figure 3 shows the binarized image. After acquiring the binarized image, we use the algorithm that extracts the feature points of OpenCV algorithm to find the features existing in the binarized image. Feature points are detected using a corner algorithm. The definition of the most widely used corner has been proposed by Harris [8,15].

The Harris corner detection method uses a method of moving a window defined in an image up and down and left and right, and determining a corner by analyzing changes in pixel values in the window. The image was reconstructed by matching the original image with the image of the feature point. Both images were matched through the equation(3).

$$I(x,y) = I_n(x,y) + I_f(x,y) \quad (3)$$

Where is the matched image, is the original image, and is the feature pot detected image.

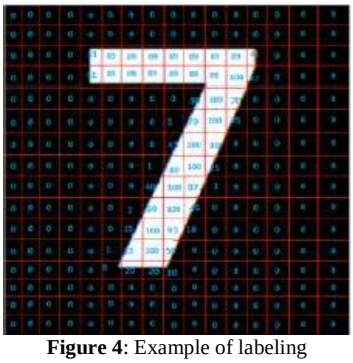

Figure 4: Example of labeling

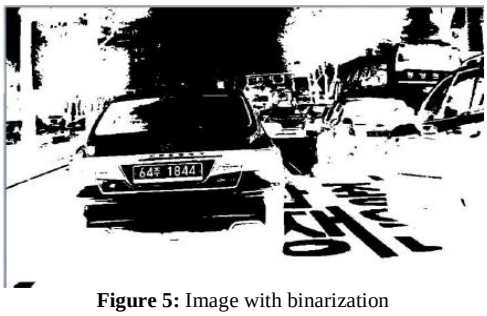

Figure 5: Image with binarization

Binarization determines the threshold value and represents pixel values only in black and white depending on whether the pixel value is greater or less than the threshold value. Our goal is to extract the license plate area, so the various colors in the image are meaningless. Only the boundaries of the region are needed, so they are represented by black and white as shown in Figure 4. It is also related to the labeling to be performed in the next step.

The next step is labeling. Labeling refers to assigning the same value to a pixel if adjacent pixels have the same pixel value. The binarization in the previous step was to facilitate labeling more easily. If the colors are varied, there will be many kinds of values to be given, and unnecessary areas will be derived, which will reduce the efficiency of analysis. By performing binarization and labeling as shown in Figure 5, black and white regions can be separated. And derives suspect candidate regions from the isolated regions.

After performing the labeling, you can display the candidate region of the license plate using OpenCV'scvDrawRect () function. Now, it is only necessary to perform the task of leaving only the license plate area among the candidate areas as shown in Figure 6.

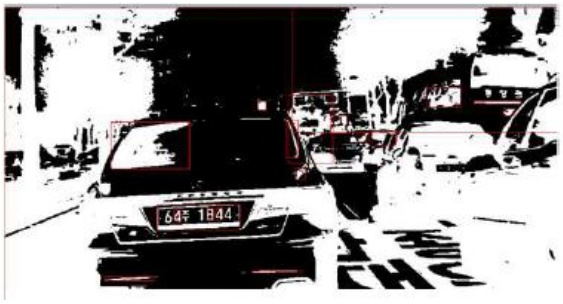

Figure 6: The derived candidate region

Figure 7: Final derived license plate area

Areas that cannot be regarded as license plate areas should be removed in the first candidate area. In this case, various logic can be applied. But we applied as the following in this module.

(1) The horizontal length of the license plate is longer than the vertical length

(2) The horizontal length of the license plate is about four times longer than the vertical length.

(3) There are at least seven character regions.

We excluded unsatisfactory areas by the above conditions (1) and (2). And the license plate area, which is characteristic of (3), applies that there are at least seven other labels in the area. Therefore, we first reduce the number of candidate regions by applying the feature of length, and then perform labeling once again, so that only the region with 7 or more internal labels was selected as the final candidate as shown in Figure7.

Now that you have extracted the license plate area, you need to perform character recognition. We used the Tesseract OCR library for character recognition. Although there is a method to perform character recognition through the template matching method, it has been concluded that the OCR library is superior in terms of

recognition rate because Hangul has many kinds and is not well matched according to the angle. OCR is optical character recognition, which analyzes characters in an image and derives the result as a string value. In Figure 8, Tesseract OCR's character recognition function requires an image to be recognized as an argument, so the derived plate area must be saved as an image file. We used OpenCV'scvSaveImage () function.

Figure 8: Image by Tesseract OCR function

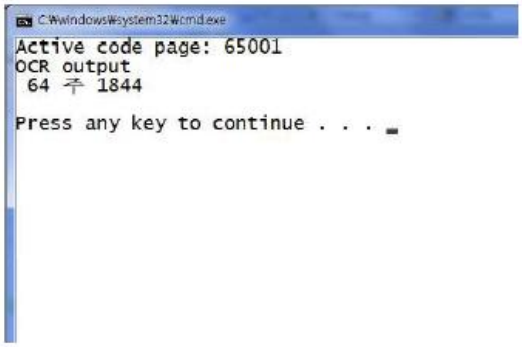

Figure 9: Recognized character string

The final step is to generate the string value through character recognition as shown in Figure 9. Now that we have extracted the license plate area, we can get the recognized string value from TessBaseAPI object of Tesseract OCR.

4. Conclusion

This paper was designed and implemented a vehicle license plate recognition module that extracts license plate area using OpenCV and Terreract OCR library. The static image is binarized using OpenCV 'sbinarization function. The binarized images were labeled by adjusting pixel values between adjacent pixels. The labeling derived a candidate region that is considered to be a plate from the separated region. After labeling, the license plate candidate area is displayed using OpenCV function, and the final candidate is derived according to the proposed algorithm in the candidate area. Using the Tesseract OCR library, the characters were analyzed to derive the characters as string values. The derived character area was saved as an image file using OpenCV function and the recognized string value was output using TessBaseAPI object of Tesseract OCR.

The vehicle license plate recognition module relates to character recognition in the field of computer vision. As this module has already been commercialized, it is possible to install a camera at the entrance of an apartment to read the license plate and identify whether it is a resident or not. When the camera is installed on the expressway, it is possible to automatically store the vehicle number in the database when speeding and other traffic violations occur.

In addition, the OpenCV library can be applied to various fields such as pattern recognition, object tracking, and motion recognition in addition to character recognition. Therefore, you will be able to create technologies corresponding to various services that are becoming automated and unmanned.

Acknowledgment

This paper was supported by the Fund of the Sahmyook University in 2018.

References

[1] J.Illingworth, & J.Kittler (1988). A survey of the hough transform. Computer Vision, Graphics, and Image Processing, 44(1), 87-116.
[2] Y. Wang, J. F Doherty, and R. E. Vandyck (2000). Moving object tracking in video.29th Applied Imagery Pattern Recognition Workshop, 95-101, doi: 10.1109/AIPRW.2000.953609
[3] K. P. Karmann and A. von Brandt(1990). Moving object recognition using an adaptive background memory. Time-Varying Image Processing and Moving Object Recognition, 2, 289-296.
[4] T. H. Chen, J. L. Chen, C. H. Chen, and C. M. Chang(2007). Vehicle detection and counting by using headlight information in the dark environment. IEEE 2007 International Conference on Intelligent Information Hiding and Multimedia Signal Processing IIHMSP07, 519-522.
[5] Abderrahim Salhi, Brahim Minaoui, and Mohamed Fakir(2014). Robust Automatic Traffic Signs Detection using fast polygonal approximation of digital curves. International Conference onMultimedia Computing and Systems(ICMCS), 255-260, doi: 10.1109/ICMCS.2014.6911185
[6] M. Lalonde, & Y. Li(1995). Detection of Road Signs Using Color Indexing, Technical Report CRIM-IT-95/12-49, Centre de Recherche Informatique de Montreal. Available from: <http://www.crim.ca/sbc/english/cime/>
[7] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, & T. Koehler (2005). A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information. Proceedings of the 2005 IEEE Intelligent Vehicles Symposium, doi: 10.1109/IVS.2005.1505111
[8] OpenCV (2009), Open source Computer Vision library. Retrieved from http://opencv.willowgarage.com/wiki/.
[9] A. Anuar, K. M. Saipullah, N. A. Ismail, & Y. Soo (2012). OpenCV based real-time video processing using android smartphone. International Journal of Computer Technology and Electronics Engineering, 1(3), 58-62.
[10] N. J Uke, & R. C Thool (2012). Cultivating research in computer vision within graduates and post-graduates using open source. International Journal of Applied Information Systems, 1(4), 1-6.
[11] Nilesh J. Uke.,&Ravindra C. Thool (2013). Moving Vehicle Detection for Measuring Traffic Count Using OpenCV. Journal of Automation and Control Engineering, 1(4), 349-352.
[12] GitHub, Inc (2018). Tesseract is an Open Source OCR engine. Retrieved from http://code.google.com/p/tesseract-ocr
[13] [13]Pixel technology (2009). Open Source OCR assembly using Tesseract engine. Retrieved from http://www.pixel-technology.com/freeware/tessnet2/
[14] B. Enyedi, L. Konyha, & K. Fazekas (2006). Real Time Number Plate Localization Algorithms. Journal of Electrical Engineering, 57(2), 69-77.
[15] G. Bradski, & A. Kaehler (2008).Learning OpenCV. O'Reilly Media, Inc., CA, USA.