Propose A Simple and Practical Vehicle Logo Detection and Extraction Framework

Article *in* International Journal of Emerging Trends & Technology in Computer Science · May 2015

2 authors:

Mustafa Salam Kadhm
Mustansiriyah University
26 PUBLICATIONS **224** CITATIONS
SEE PROFILE

Shu Yun Lim
Unitar International University
18 PUBLICATIONS **339** CITATIONS
SEE PROFILE

International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 3, May-June 2015 **ISSN 2278-6856**

Propose A Simple and Practical Vehicle Logo Detection and Extraction Framework

[1]Mustafa S. Kadhm, Lim Shu Yun [2]

[1]University of Technology, Department of Computer Science,
Baghdad, Sinaa st. 52, Iraq

[2]Department of Computer Science and Information Technology, University of
Tun Abdul Razak ,KL, Malaysia

Abstract

There are several algorithms and techniques that might help in recognizing vehicle logo, but the need for simple framework especially in terms of processing power (computationally) and memory consumption as well as the flexibility to such framework to be attached to an existed system is crucial. In this paper a framework for Vehicle Brand Logo Detection and Extraction based on OpenCV library has been proposed. The proposed framework can be used as complete tool for Vehicle Brand Logo Detection, or as part for image noise removal since noise removal is an important initial step for most computer vision applications. After the framework has been implemented and tested, it gives a good result for image matching and feature extraction with the focus on Vehicle Logo Detection and Extraction process.

Keywords: OpenCV, Framework, Pattern Recognition, Logo Detection.

1. INTRODUCTION

In this paper a framework for the process of extracting a car-brand logo from a captured image or video frame (as still image) in order to recognize this particular logo (if existed within the image) brand has been presented. The framework intended to be run in real-time bases and for real world scenarios. Besides, it could extract a particular logo from an image without any initial steps or knowledge such as the car number plate position, etc. Furthermore, it's not limited to front view of the car, though it can work also in image that captured in various angles. After implementing the proposed framework a capable of extracting the logos from average to high quality images was achieved, with high accuracy and in small processing time.

The car brand's logo is a small object that should be unique to represent the car brand and model. It could be found in several parts of the car especially in the front part of the car (in middle area that lays among the car lights and number plate) comparing to other car parts. Mostly, car logo shape is unique in terms of color, texture, and geometry which make the logo easy to be noted or seen by human.

Thus, each car definitely has one unique logo, but this unique logo could be vary in size, color, texture, etc. For several models of the same car, one logo possible to have different attributes, which make the detection and comparing/matching logos is a non-trivial task.

However, using SIFT or SURF for feature Extraction and Matching is a common approach, but both of the algorithms are computationally expensive in terms of memory and processing time consumption which make integrating/ utilizing them in a big software systems is difficult and sometimes impractical. Therefore, there is a need to figure out a simple but effective process/ framework to be used.

The main goal of the proposed framework is to search any image for a given Logo (as a template) by identifying and extract it then it will be so easy to tell the model/ brand of the car based on the template and the extracted logo image. For instance, a PROTON car logo image used as a template. The framework will check the original care image to find out is there any logo match the logo in the template image, if there is a logo in the image it will be easily notice the brand/ model of the car.

2. RELATED WORKS

Many researchers have searched pattern/object recognition and feature extraction where several techniques and algorithms were discussed that might be suitable for logo detection such as in [1] where the researchers have suggested a real-time method for animated television logo detection developed based on a unified logo boundary, which is extracted from the multi-frame logo sequence, in [2] the researchers have proposed an extension of the color object detection method devised by Luo and Crandall, intended for use in detecting logos and trademarks in unconstrained color images, while in [3] a solution was given for logo detection and recognition in degraded document images using Bayesian Model, whereas in [4] the researchers proposed a framework for detection and recognition on document images. Regarding a logo as a group of features with restricted geometrical relationships, in [5] the researchers proposed a framework that utilizing SIFT to solve various kinds of image transformations, as in [6] the researchers have presented using boundary extension of feature rectangles performs initial detection of logo candidates. Then each logo candidate is further prescreened by a simple decision tree classifier so as to reduce the false positive, and in [7] the researchers proposed string

matching technique shows that logo recognition is possible if the logo foregrounds clearly separated and does not contain any false segments.

On the other hand there are researchers have focused on 'Vehicle' Logo Detection and Matching such as [8] where the researchers have proposed method is based on the prior information and texture information of the vehicle, also in [9] the researchers proposed to use a framework based on edge detection and morphological filter, in [10] the researchers proposed Laplacian-Gaussian differential operator and Gaussian filtering and Sobel operator for logo edge detection, and for last not least in[11] the researchers have described an improved Adaboost algorithm, and wavelet moment by which the told that they can handle transformation issues (e.g. scale and rotation).

3. LOGO-DETECTION AND EXTRACTION PROCESS

The main process of the proposed framework has several steps. Assume first that there is already an image (a frame of captured video stream) as a still image, with image dimensions smaller than 800 X 600 to make the detection process fast and practical for the image to be processed and searched for the logo in the best way. All the proposed method steps start with read a RGB image then convert it into gray scale .Secondly, convert the gray scale image to binary image. After that, resize the binary image into half size of the original size. Furthermore, resize the image to its original size. Besides, applying edge detection technique, then the matching process. The all framework steps can be illustrates in figure (1).

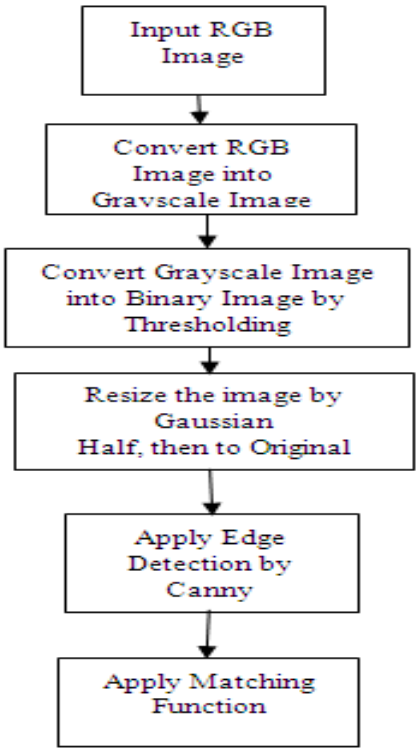

Figure 1 Flow chart of the proposed framework

Step1: Grayscale conversion

The first step of logo detection and extraction is convert the colored image (as in Figure. 2: Original Colored Image) to Grayscale (could be accomplished by using OpenCV function "cvCvtColor()" (as in Figure. 3-Scale Image (Pixels values 0-255) the range of color pixels limited to 0-255 in order to simplify the calculations and processing time and reducing required memory for storing the images.

Figure 2 Original Image

Figure 3 Gray-Scale Image (Pixels values 0-255)

Step 2: Binary threshold

The second step is converting the gray scale image into binary image by applying binary threshold method according to the formula in Eq.1 (Open CV 2.1 documentation)[12]:

$$dst(x,y) = \begin{cases} maxValue & if\ src(x,y) > T(x,y) \\ 0 & otherwise \end{cases} \quad (1)$$

Where 'dst' represents the resulted image, 'src' is the original image, 'maxValue' is the threshold value, x and y are the pixels representations of an image. In this step the output image will limit the pixel data values of the grayscale image to be in 1 and 0 to remove the unwanted pixels data based on the threshold value, the result binary image illustrated below:

Figure 4 Image after Binary Threshold

Step3: Half size image

Resize the image to half of its original size and then apply Gaussian Blur with size of 5X5.

After this step, the resulted image will be less noise. This step helps for improving the blurring of the image by blinding the sharp-geometrical shapes such as the cells of the car-grill and in the same time still maintaining the logo in a good shape.

Step 4: Resize the Image to its original size

In this step the resized image after Gaussian blurring process will be resize to its original size again which make us got less noise and good edges for the logo as shown in (Figure. 5):

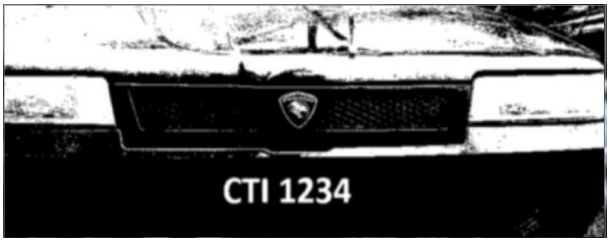

Figure 5 Same image (resulted from the previous processing steps) after Removing the Noise

Therefore, the main goal of the previous processing steps is to prepare the image for edge detection process by removing any irrelevant pixel data, which will make edge detection algorithm run faster and to provide better results.

Step 5: Edge Detection

Run edge detection using either **Sobel** Operator or **Canny** Algorithm, in the proposed framework Canny is much has been used, Figure. 6 illustrate the result after applying the canny algorithm.

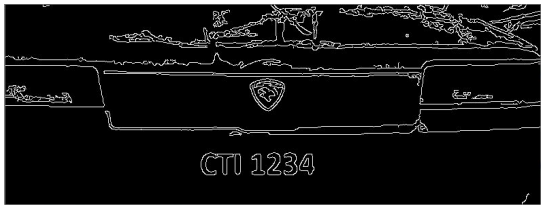

Figure 6 Result after applying canny algorithm

Although most of the image noise has been removed (as shown above) , the image still not ready to be searched for the template logo yet .There are three factors make the detection so hard and affect the accuracy rate of the framework .The angle of the logo will be not significant, the size of the logo within the image against the logo template that we will search for within the images will be handled, and simple rotation should be also handled too in order to extract the logo from the image by using similar template that might have differences and variations in these terms.

Step 6: Matching

The final step in the proposed framework is the matching process. It is done by applying template matching function "cvMatchTemplate()" in openCV library to find the desired logo. Figure(7) shows the result of the proposed framework.

Figure 7 Logo Detection

4. CONSIDERATIONS AND EXPERIMENTAL RESULTS

There are several considerations that should take place for recognizing and extracting a car brand logo or any type of logos that have similar scenarios and conditions, mainly they are:

- The scale of the car logo in the image against (comparing to) the template logo image.
- The rotation of the car logo in the image against (comparing to) the template logo.
- The angle of both camera and the logo.

Assuming that the quality of the image, the clarity of the logo within the image and other stuff like luminance should be all in good/reasonable situation, then the focus on the scale, rotation and angle.

In order to handle and accommodate such differences in these terms, a smoothing for the resulted image has been used again on the final image that is removed the noise and applied the canny algorithm.

Therefore, with this technique it is possible to match the logo template and search for the logo within the image and if the logo exists then the framework can extract it. Fig 8, 9, 10, 11 shows an examples of applying the proposed frame work logo detection.

Figure 8 Logo Detection – 01

Figure 9 Logo Detection - 03

Figure 10 Logo Detection -04

5. CONCLUSION

There are several algorithms that could help to detect and extract certain patterns, but most of them are complex and computationally expensive, since this process is commonly wanted to be attached as a subsystem that could fits nicely with the existed systems.

A simple but effective framework for logo detection and extraction has been proposed. The accuracy of proposed framework is high, and still handle some rotation and scaling scenarios, especially when Templates (Logos) that have close similarities to the ones that may exist in the used images. Exploratory results with artificial and real life images shows an efficient accuracy of the proposed framework for the recognition process.

References

[1] Esen, E. Soysal, M. Ates, T.K. Saracoglu, A. Aydin Alatan, A., A Fast Method For Animated TV Logo Detection, Content-Based Multimedia Indexing, June 2008, p 236.

[2] Phan, R. Chia, J. Androutsos, D., Colour Logo And Trademark Detection In Unconstrained Images Using Colour Edge Gradient Co-Occurrence Histograms, Electrical and Computer Engineering, 2008. CCECE 2008, May 2008, p. 000531.

[3] Wang H., Document Logo Detection and Recognition Using Bayesian Model, Pattern Recognition (ICPR), 2010 20th International Conference on Aug/2010, pp. 196991-1964.

[4] Zhe Li, Schulte Austum, M. Neschen, M., Fast Logo Detection and Recognition in Document Images, Pattern Recognition (ICPR), 2010 20th International Conference on Aug/2010, p. 2716.

[5] Arafat, S.Y. Husain, S.A. Niaz, I.A. Saleem, M., Logo detection and recognition in video stream Digital Information Management (ICDIM), 2010 Fifth International Conference on July 2010, p. 163.

[6] Hongye Wang Youbin Chen ,Logo Detection in Document Images Based on Boundary Extension of Feature Rectangles, Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on Oct/2009, p. 1335.

[7] den Hollander, R.J.M. Hanjalic, A., LOGO RECOGNITION IN VIDEO STILLS BY STRING MATCHING, Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on Sept/ 2003, Vol.3, p. III.

[8] Wang Yunqiong Liu Zhifang Xiao Fei, Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on, Dec. 2007, p. 691.

[9] Li W., Li L., A Novel Approach for Vehicle-logo Location Based on Edge Detection and Morphological Filter, Electronic Commerce and Security, 2009, vol. 1, p.343.

[10] Yuanli Wang Na Li, Ying Wu, Intelligent Systems and Applications (ISA), 2nd International Workshop, May 2010, p. 1.

[11] Xiang Pan, Xiang Nian Huang, Shao-Hui Zhang, Systems, Wavelet Moment and Improved Adaboost Application to Vehicle-logo Location Man and Cybernetics, SMC 2009. IEEE International Conference on Dec. 2009, p. 5174.

[12] Itseez , Open Computer Vision Library , http://opencv.org/ (2015)

AUTHOR

Mustafa Salam received the B.S. degrees in Software Engineering from Al-Mansour University College, Baghdad, Iraq 2009 and M.S. in Information Technology from University of Tun Abdulrazak, Malaysia 2012. During 2012-2013, he worked as Instructor in Ishik University, Turkey .Currently, PhD candidate at University of Technology, Baghdad, Iraq.