Bulletin of the *Transilvania* University of Braşov
Series I: Engineering Sciences • Vol. 5 (54) No. 1 - 2012

AN IMAGE RECOGNITION SOFTWARE TOOL FOR CAR IDENTIFICATION BASED ON LICENSE PLATE

R. HANDARIC[1] D. FLOROIAN[1]

Abstract: *The aim of this paper is to present an image recognition software tool based on classical algorithm but rearranged in manner to obtain a good answer and a little possibility of fault. Also for validating the software tool a scenario was proposed: car identification based on license plate number (CILP). The software program was designed in C++ using OpenCV library. The software tool uses a custom approach of classical algorithms in order to achieve better results.*

Key words: *image recognition, software tool, computer vision.*

1. Introduction

Nowadays computers become more and more powerful and the need for use large calculations programs becomes feasible. In this purpose our paper presents a software tool which use sophisticated algorithms with large amounts of calculations. Because the calculations are in the same manner and only the repetitions of them are big, a powerful computer will complete the task in a finite time.

Furthermore, the algorithms become more and more efficient so it is possible in present time to obtain real time results in computer vision.

In this paper we want to present a software tool created for image recognition. In order to demonstrate the functionality a study case was choose. In this study case a car is identified by its license plate. The main idea is simple but the complexities of practical implementation impose a systematic approach. This is because a simple thing can be found in many forms. Also the state of the license plate (due to weather conditions or damages or depreciation compose a new scene every time. From this scene a simple area must be used. The software tool will recognize this particular area and will process only the relevant information. In the second step of the process the software tool will try to recognize the characters or digits that compose the image. In the next step the information is transformed in digital object in order to recognize the license plate in the database.

For implementing this process there are a few important steps. The most important is image recognition and the paper will present this in the first. The second important step is to digitize the object. In the near future is in development a multiagent system which will approach the problem in other way using intelligent agents for recognition and searching. We hope that in this manner the procedure for digitize the license plate will be faster and

[1] Centre "Control Process Systems", *Transilvania* University of Braşov.

more reliable. Obviously the final step is to manage the database which is a trivial operation, but in the case of large objects and large amounts of objects became a time eating operation. The next version of the software tool will implement also an intelligent mechanism for database also based on a multiagent system.

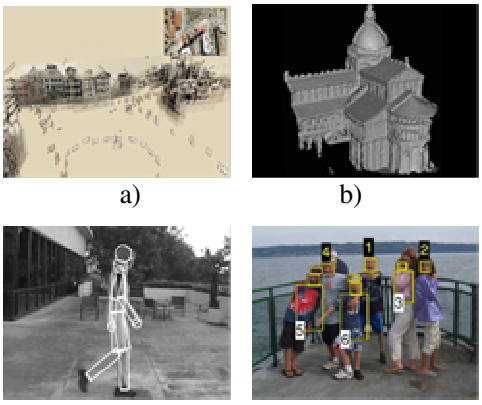
The paper is organized as follow: in second section we present the supporting concepts as "computer vision" and "image processing"; in section 3 we present the algorithm used for CILP and in section 4 the study case for car identification is presented. Finally in conclusion section we present the results and the future possibilities.

2. Supporting Concepts

As humans, we perceive the three-dimensional structure of the world around us with apparent ease. Perceptual psychologists have spent long times trying to understand how the visual system works and researchers in computer vision have been developing. In parallel, mathematical techniques for recovering the three-dimensional shape and appearance of objects in imagery are also developed.

During the recent years many scientist contribute to the improvement in this field. In present we have reliable techniques for accurately computing a partial 3D model of an environment from thousands of partially overlapping photographs (see Figure 1a) [7].

Given a large enough set of views of a particular object or facade, we can create accurate dense 3D surface models using stereo matching (see Figure 1b). We also, can track a person moving against a complex background (see Figure 1c). Also is possible, with moderate success, attempt to find and name all of the people in a photograph using a combination of face, clothing, and hair detection and recognition (see Figure 1d).



a) b) c) d)

Fig. 1. *Computer Vision elements*

However, despite all of these advances, the dream of having a computer interprets an image at the same level as a two-year old (for example, counting all of the animals in a picture) remains elusive.

The main problem of why is vision so difficult, is given in part, because vision is an inverse problem, in which we seek to recover some unknowns given insufficient information to fully specify the solution. For resolve this situation we must therefore resort to physics-based and probabilistic models to disambiguate between potential solutions.

However, modelling the visual world in all of its rich complexity is far more difficult than, modelling the vocal tract that produces spoken sounds because of quantity of information which must be computed [2], [7], [8].

The only hope is that computer vision is being used today in a wide variety of real-world applications, which include:

• Optical character recognition (OCR): reading handwritten postal codes on letters and automatic number plate recognition (ANPR);

• Machine inspection: rapid parts inspection for quality assurance using stereo vision with specialized illumination to measure tolerances on aircraft wings or auto body

parts or looking for defects in steel castings using X-ray vision;

• Retail: object recognition for automated checkout lanes 3D model building (photogrammetry): fully automated construction of 3D models from aerial photographs used in systems such as Bing Maps;

• Medical imaging: registering pre-operative and intra-operative imagery or performing long-term studies of people's brain morphology as they age;

• Automotive safety: detecting unexpected obstacles such as pedestrians on the street, under conditions where active vision techniques such as RADAR or LIDAR do not work well;

• Match move: merging computer-generated imagery (CGI) with live action footage by tracking feature points in the source video to estimate the 3D camera motion and shape of the environment. Such techniques are widely used in movies production and they also require the use of precise matting to insert new elements between foreground and background elements;

• Motion capture (MOCAP): using retro-reflective markers viewed from multiple cameras or other vision-based techniques to capture actors for computer animation;

• Surveillance: monitoring for intruders, analysing highway traffic, and monitoring pools for drowning victims;

• Fingerprint recognition and biometrics: for automatic access authentication as well as forensic applications [6], [7], [9].

In electrical engineering and computer science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or, a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

Image processing usually refers to digital image processing, but optical and analog image processing also are possible. An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows as seen in Figure 2 [1].

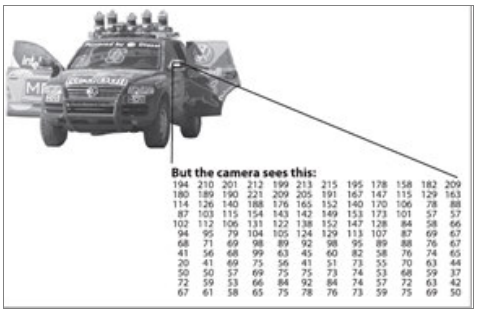


Fig. 2. *To a computer the side's mirror is just a grid of numbers*

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real time computer vision, developed by Intel Corporation and now supported by Willow Garage [1]. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, Matlab, and other languages.

OpenCV was designed for computational efficiency and with a strong focus on real-time applications. One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. The OpenCV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics. Because computer vision and machine learning often go hand-in hand, OpenCV also contains a full, general-purpose Machine Learning Library (MLL). This sublibrary is focused on statistical pattern recognition and clustering. The MLL

is highly useful for the vision tasks that are at the core of OpenCV's mission, but it is general enough to be used for any machine learning problem.

Although Intel started OpenCV, the library is and always was intended to promote commercial and research use. It is therefore open and free, and the code itself may be used or embedded (in whole or in part) in other applications, whether commercial or research.

All these aspects recommend our software tool to be based on OpenCV.

In this paper we use this OpenCV in order to obtain a rapid software tool.

3. Algorithms

Gray scale conversion: From the 24-bit color value of each pixel (*i,j*) the R, G and B components are separated and the 8-bit gray value is calculated using the formula:

$$\text{Gray}(i,j) = 0.59 \cdot R(i,j) + 0.30 \cdot G(i,j) + 0.11 \cdot B(i,j) . \quad (1)$$

Median filter is a non-linear filter, which replaces the gray value of a pixel by the median of the gray values of its neighbors. We have used 3 × 3 masks to get eight neighbors of a pixel and their corresponding gray values. This operation removes salt-and-peeper noise from the image.

Contrast enhancement: Contrast of each image is enhanced through histogram equalization technique [4], [6]. Total 256 numbers of gray levels (from 0 to 255) are used for stretching the contrast. Let total number of pixels in the image be N and the number of pixels having gray level k be n_k. Then the probability of occurrence of gray level k is, $P_k = n_k/N$. The stretched gray level S_k is calculated using the cumulative frequency of occurrence of the gray level k in the original image using the Eq. (2), where, 255 indicates the maximum gray level in the enhanced image:

$$S_k = \sum_{j=0}^{k} \frac{nj}{N} \cdot 255 . \quad (2)$$

We have extracted the edges created by the characters within the license plate. Sobel edge operator [4], [10] is used for detection of edge gradients. It is seen that when the characters of the license number are written horizontally the vertical edges of each character appear at regular interval and they have a specific height. The pattern and concentration of the vertical edges also remains in conformity with the pattern of the license number. This appearance of vertical edge pattern is statistically seen to occur within the license plate, no where else within the natural scene of the image. In the present work, we have explored this phenomenon to fiind the license plate region within the image. The vertical edge at point (*x,y*) is found using the following formula:

$$gradV(y,x) = \sqrt{\left(\sum_{m=-1}^{+1} \sum_{n=-1}^{+1} V\_mask(n,m) \cdot \frac{img\_contrast(y+n, x+m)}{4} \right)^2} , \quad (3)$$

where, *imag_contrast* is the enhanced image over which the edge detection algorithm is operated upon, *V_mask* is the Sobel's mask for vertical edge detection as given below and *gradV* is the vertical edge gradient:

$$V_{mask} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 2 \end{bmatrix} . \quad (4)$$

Depending on the value of the *gradV* we have binarized the edge image using the

threshold μ_{gradV} mean edge gradient value) and formed the edge image *img_edge*. Deciding the threshold value for binarization of edges is a key factor.

In the present paper, we use a novel approach based on prominent vertical edges computed from vehicle images [5] for localization of significant license plate regions. It may be observed that the pattern of the vertical edges at the license plate region is very dense and prominent. Also, the vertical run-lengths of edge pixels within the license plate regions are almost equal to the height of the characters there in. Using the aforesaid attributes, the overall localization algorithm may be subdivided into the following intermediate stages: identification of potential band of rows, primary localization of license plate regions based on statistical distribution of vertical edge pixels, refinement of license plate regions based on prominent vertical edges and finally, localization of license plate bounding box by removing the noise segments. The steps are discussed hereunder in an algorithmic approach.

First Stage:

1. For row=1 to height
   For col= 1 to width
edgepixel[row]=edgepixel[row]+edge (row,col)
2. For row=1 to height
   If edgepixel[row]>T_{min}
mean[row]=mean(); //of edge pixel positions
variance[row]=variance( )
//of edge pixel positions
3. For row=1 to height
   Find the set of continuous rows satisfying
variance[row]> maximum variance (V_{xmax}).
   //This set actually gives the probable n
   //bands having license plates
4. For each band,
set top= starting row
set bottom= ending row.

The value of T_{min} and V_{max} are calculated from the generated dataset.

Up to this point what we get is the statistically obtained potential license plate region whose dimension indicates the maximum area in which the license plate can appear.

Second Stage:

1. For each band, calculate minimum and maximum values of μ_x (μ_{xmin} and μ_{xmax})
calculate maximum value of v_x (v_{xmax}).
2. For each band,
set left= (μ_{xmin} – v_{xmax})
set right= (μ_{xmax} + v_{xmax}).
3. For each band,Draw box having diagonal corners (left, top) and (right, bottom)
   //This box will localize the position of license plate

The value of H_{min} is actually dependent on the height of the characters in the character set and is obtained by averaging the heights of the vertical edges within the bounding box obtained in second stage.

Third Stage:

1. For each bounding box in second stage,
   From left to right
find first prominent vertical edge having
height> predefined minimum height (H_{min})
if found, set new_left=left
  From right to left
find first prominent vertical edge having
height> predefined minimum height (H_{min})
if found, set new_right=right
2. Draw box with left-top and right-bottom corners'coordinates as (new_left, top) and (new_right, bottom)
3. Among these new bounding boxes, the overlapped or very close bounding boxes are merged to get common bounding box. This case actually occurs in case of multiline character set license plate.

Fourth Stage:

1. For each bounding box in third stage,
aspect_ratio=box_width/box_height
area=box_width×box_height.

2. Among all bounding boxes, noise boxes are removed by allowing boxes having specific range of aspect ratio and area.

The outcome of this stage is the true license plates and along with them there may be additionally generated boxes indicating false license plates. These noisy boxes are removed in the fourth stage depending on the aspect ratio and the area of the generated boxes.

The average value of aspect ratio and the area are calculated from the generated dataset for single-line and multiline character set license plates separately.

4. Study Case

As discussed, a number of 20 images of resolution 704 x 576 pixels were selected for preparation of data set.

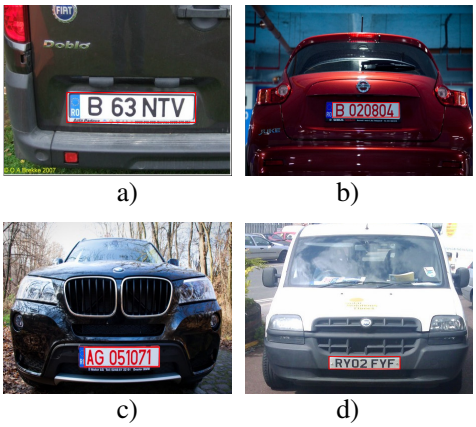
The experimental threshold values are statistically computed from the image data sets and are obtained as mentioned below:

- Minimum number of edge pixels in a row, $T_{min} = 6$.
- Maximum variance of the position of the edge pixels in a row containing license plate, $V_{max} = 35$.
- Minimum height of the selected bands of rows containing license plate $H_{min} = 8$.
- Total area of the license plate is considered to be 3000 to 6000 pixels.

We have considered the case for the standard Romanian license plate that consists of a blue vertical strip (the European strip) on the left side of the plate displaying the 12 stars of the European Union and the country code of Romania (RO), always followed on a white surface using black font by the county code and a combination of two or three digits and three capital letters, or six digits for red plates and plates for cars that fall under a leasing agreement. All numbers issued before January 1, 2007, used the flag of Romania instead of the 12 European stars [3], [9-10].

The processing is made with an extra step that consists in a dilation used for the binary image, in this operation the value of the output pixel is the maximum value of all the pixels in the input pixel's neighbourhood.

Using the above considerations, experiments are conducted with 20 images from personal database. Figure 3 (a-d) shows the some of the cases where perfect localization of license plate is done, the license plate region is being marked by the thick red box. Figure 3 (a-d) shows the intermediate results of the license plate localization part of experiment when done on Figure 4a.



a) b) c) d)

Fig. 3. *Sample images with successfully localized license plates regions*

Figure 4a shows the case where license plate is wrongly localized and figure 4b shows the case where no license plate is detected by our program.

Fig. 4. *Sample images where the current technique fails to localize true license plate regions with the current program*

The result can be analysed by considering two different cases of finding the proper license plate. We consider the result to be false negative if true license plate is not found or false locations are detected as license plate. True positive cases ate those where only true license plate is detected as a license plate.

It is observed from the experimentation on the collected dataset of vehicle images that the false negative rate is 15% and the true positive accuracy is 75%.

The confused characters mainly are B and 8, E and F, D and O, S and 5, Z and 2. To increase the recognition rate, some criteria tests were used in the system for the confused characters defining the special features of the characters. With these features of characters and applied tests during recognition algorithm, recognition rate is increased with the minimum error.

After finding the proper license plate from the image, the program will continue with processing this area in order to recognize the characters, and compare them with those in the database. This software will be used for granting access in a parking lot by recognizing the plate number for the registered users.

For example an institution has a parking lot where entry and exit access if often controlled, so we can grant access to the employees by entering their license number in the database, and this way when they want to enter or exit the parking lot, a camera will take a picture of the car, the software will process the image and recognize the license plate number, compare it with those in the database, and automatically raise the barrier. Also this software can be used in public parking lots, to keep a evidence of the cars that enter, and help track stolen cars, or cars that are searched by police. In the Figure 5 the GUI interface is depicted.

5. Conclusions

In our present paper, we have developed an effective method for localization of license plate regions from true color images.

The technique is extensively tested with more than 20 different types of image samples and gives satisfactory results.

We think that limitation of the hardware was the main factor for software limitation. In the very immediately period a new hardware will be available with a self-build controller fitted on this application.

In the same time, as the technique is edge based, the main limitation of our algorithm is that it performs well for less noisy images and having well printed characters over the license plates. This is why in the future the technique will be enhanced by applying some soft computing techniques for training the patterns of edge gradients.

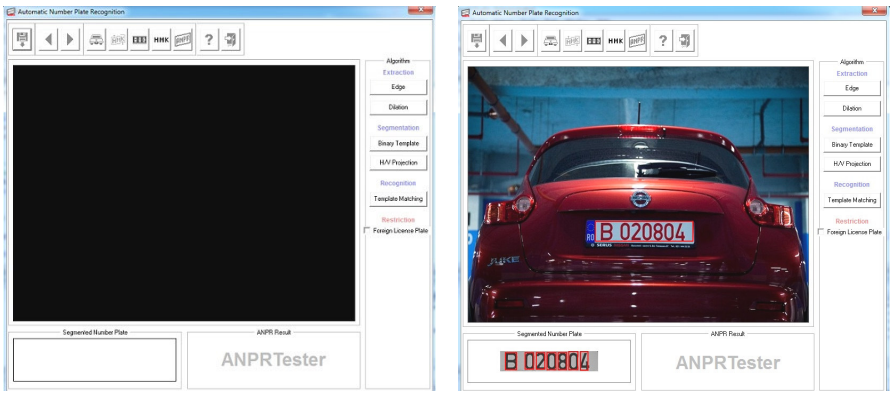


Fig. 5. *The GUI Interface*

We are currently working at improving and optimizing the detection algorithm implementation, our goal is to adapt the algorithm to better fit to the conditions of processing image frames acquired with standard video camera without any additional equipment and also with specific hardware attached to a powerful camera. In the same time, we will improve the mechanism for localization of the license plate so the localized license plate regions are to be subsequently processed by a segmentation and OCR module for extraction of vehicle registration numbers.

In the segmentation of plate characters, license plate will be segmented into its constituent parts obtaining the characters individually. Before recognition algorithm, the characters will be normalized to fit equal size, step necessary in template matching: the character image is compared with ones in the database and best similarity is measured. Some confused characters (mainly like B and 8, E and F, Z and 2) were differentiated by defining the special features of the characters. With these features of characters and applied tests during recognition algorithm, recognition rate is increased with the minimum error.

We also need a very good computing time, because this is a real time application and we hope that the next version of the software will have an improved algorithm, so that they satisfy our needs and the process time. We prove that the algorithm that we proposed before is competitive and the software tool is able to satisfy the desired requests.

References

1. Atanassov, A.: *Advanced Software Architecture of an Automatic Vehicle Number Plate System*. In: Journal of the University of Chemical Technology and Metallurgy **47** (2012), p. 77-82.
2. Horowitz, M.: *Efficient Use of a Picture Correlator*. In: Journal of the Optical Society of America **47** (2005), p. 327-331.
3. Mahini, H., Kasaei, S., Dorri, F.: *An Efficient Features - Based License Plate Localization Method*. In: 18th International Conference on Pattern Recognition, Hong Kong, China, 20-24 August 2006, Vol. 2, p. 841-844.
4. Nguyen, C.-D., et al.: *Real-Time License Plate Localization Based On a New Scale and Rotation Invariant Texture Descriptor*. In: 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, China, 12-15 October, 2008, p. 956-961.
5. Saha, S., Basu, S., et al.: *License Plate Localization from Vehicle Images - An Edge Based Multi-Stage Approach*. In: International Journal of Recent Trends in Engineering **1** (2009), p. 284-288.
6. Sozbay, S., Ercelebi, E.: *Automatic Vehicle Identification by Plate Recognition*. In: World Academy of Science, Engineering and Technology Journal **9** (2005), p. 222-225.
7. Szeliski, R.: *Computer Vision: Algorithms and Applications*. New York. Springer, 2010.
8. Tatale, S., Khare, A.: *Character Recognition and Transmission of Characters using Network Security*. In: International Journal Advances in Engineering & Technology **1** (2011), p. 351-360.
9. Wu, P., Chen, H., et al.: *License Plate Extraction in Low Resolution Video*. In: 18th International Conference on Pattern Recognition, Hong Kong, China, 20-24 August 2006, Vol. 1, p.824-827.
10. Zhang, Y., Zhang, C.: *A New Algorithm for Character Segmentation of License Plate*. In: IEEE Intelligent Vehicles Symposium, Beijing, China, 9-11 June 2003, p. 106-109.