RESEARCHERID
THOMSON REUTERS

ISSN: 2277-9655
[Lippaiová* *et al.*, 7(8): August, 2018] Impact Factor: 5.164
IC™ Value: 3.00 CODEN: IJESS7

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AUTOMATIC LICENSE PLATE RECOGNITION SYSTEM USING OPENCV LIBRARY

Denisa Lippaiová, Miroslav Michalko, Ondrej Kainz, František Jakab
Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical university in Košice, Slovakia

DOI: 10.5281/zenodo.1336680

ABSTRACT
The purpose of this paper is to analyze the problematics of object detection in images and to introduce a proper solution for car license plate detection used for access restriction and monitoring cars accessing the parking area through the entrance ramp. The solution is implemented using OpenCV library and designed as a desktop application able to detect and read car license plates from video file. Achieved results are tested and evaluated in the conclusion of this paper.

KEYWORDS: image and video processing, object detection, video compression, optical character recognition, OpenCV, Tesseract

I. INTRODUCTION
Computer vision [1] is a science discipline focused on technologies enabling machine extraction of important features from images for specified task solutions. In other words, computer vision is human vision imitated by computer. The main purpose is description of objects we see by one or more images and reconstruction of its features as shape, illumination or color.

Computer vision is used in many different areas. One of them is image and video processing for purposes of object detection in these images or video files. Nowadays, very common application of object detection in images is detection of car license plates. There already exists few license plate detection systems, which are mostly being used in parking areas with paid parking or areas with restricted entry for vehicles. These systems are unified under common name ALPR systems, what stands for Automatic License Plate Recognition systems.

In this paper we would like to introduce a solution for car license plate detection used for access restriction and monitoring cars accessing the parking area through the entrance ramp. The cars are being monitored by camera on the entrance ramp. After video processing the access can be granted. The purpose of the solution is detection of license plate area in this video file and recognition of license plate number.

The solution proposed in this paper was implemented using OpenCV library. OpenCV (Open Source Computer Vision Library) [2] is an open source library used for creating applications in field of computer vision and machine learning. The library supports many programming languages, including Java which was used for implementation of the solution. The library is compatible with operating systems Windows, Mac OS, iOS, Linux or Android. During implementation of designed solution, we used OpenCV library for detection and recognition of license plate objects in a way of applying filters and operators for image and video processing.

A recognition of license plate number was implemented using Tesseract library. Tesseract [3] is a library for optical character recognition (OCR), which enables a machine to recognize and read text characters. The library supports many programming languages, including Java, and it is compatible with operating systems Windows, Linux and MAC OS. Text characters from more than 100 languages worldwide can be read and the output of the recognition can be written in txt, pdf, hocr or tsv file.

II. OBJECT DETECTION

Object detection from input image is a process of image processing during which suitable techniques for image and object features quality improvement are applied. The image processing is commonly divided into three steps:

1. Preprocessing
2. Feature extraction
3. Classification

Figure 1 Object detection process

Preprocessing

The purpose of image preprocessing is reduction of image imperfections and preparation of image for feature extraction. The main parts of this process are brightness and contrast normalization, noise reduction and normalization of image dimensions.

Brightness and contrast normalization

Brightness and contrast of processing image are often being affected by lighting conditions of the environment. Due to that the normalization is essential in the process of image preprocessing. Mostly used techniques for brightness and contrast normalization are: histogram normalization, global and adaptive thresholding. Histogram is a graph of brightness values in each pixel of image, which is used to find proper brightness values. Thresholding is a process of converting image into monochrome colors, which improves image contrast and is better for edge detection.

Noise reduction

During the process of creating a signal there is always created also an undesirable signal, called noise, which is necessary to reduce. In the case of video or image, the noise is created as an insufficient color information or an unbalanced brightness due to the illumination or camera features. Noise reduction is necessary for further feature extraction. It is essential to choose an algorithm which preserves important features as edges. Commonly used filters for noise reduction are Gauss or median filter.

Normalization of dimensions

In the process of image preprocessing it is also necessary to crop image or its parts and normalize image dimensions, because feature extraction is applied to image with fixed dimensions. For these purposes we use resampling. Resampling is a process of reduction or addition of image information.

Methods used for resampling are divided into two groups [4]:

1. Non-adaptive algorithms – the image content is not being considered in the process of resampling
2. Adaptive algorithms – the resampling is based on image content

There are many resampling methods such as: nearest neighbor method, bilinear interpolation, polynomial interpolation or weighed average resampling. It is necessary to choose the right method according to used feature extraction method.

Feature extraction

Input image contains many information irrelevant for object detection. These irrelevant information increase computation and time requirements of object detection systems. Feature extraction is a process of extracting only important features for object detection. The irrelevant information is removed from image. The operators applied using convolution matrices are commonly used for feature extraction. Mostly used operators are local binary patterns (LBP), Sobel operator, Roberts operator or Prewitt operator.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Classifiers

Based on processed image with extracted features it is possible to classify objects in image. The purpose of classification is to identify the desired object and to differ it from other objects or background. For classification we use classifiers, which are able to 'learn', so called machine learning. Classifiers use feature vectors as points in multidimensional space and they search for the planes, which divide detected object in the image from its surrounding. One of mainly used classifiers is Support Vector Machine (SVM).

III. LICENSE PLATE DETECTION

The input for our license plate detection process is a video file. Since video is a series of continuing image frames, the process of detection is applied also on the image frame. Video frames are being loaded in a cycle. After successful loading of the frame, the license plate detection algorithm is applied on the concrete frame. License plate is searched in image as a rectangular area with increased horizontal and vertical edges caused by license plate numbers. After successful detection of this area, the license plate numbers are segmented, and the process of optical character recognition is applied.

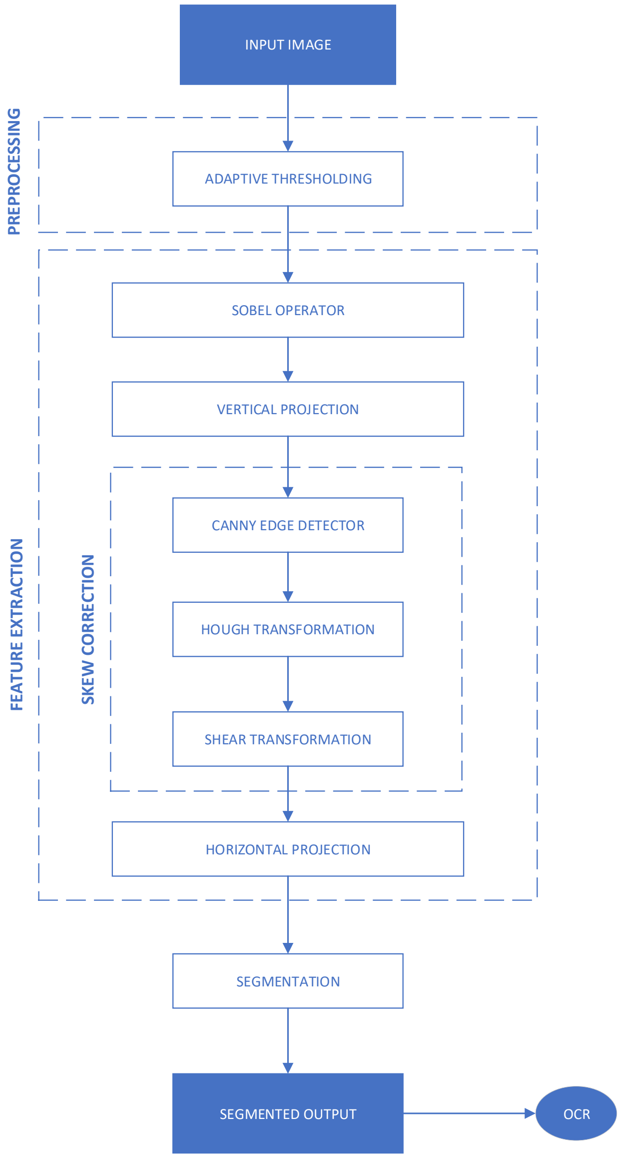

Figure 2 License plate detection process

Preprocessing

The contrast is normalized during preprocessing using adaptive thresholding with local thresholds. Adaptive thresholding creates monochrome image from input frame with clearly defined edges of license plate. Thresholding is implemented using OpenCV library function.

Figure 3 Input frame before and after thresholding

Feature extraction

The first step of feature extraction is applying of Sobel operator for finding vertical edges in the image. The value of each image pixel represents edges in that point. According to these values it is possible to compute the vertical projection of the image.

Vertical projection

Vertical projection is a graph representing pixel values with respect to y axis. Let the image with dimensions $w \times h$ be represented by function $f\ x\ y\ (\ ,\ )$, then vertical projection of the image can be computed as a sum of all pixel values $[x_i, y]$, where $i \in < 0, w-1 >$.

The relation is defined mathematically as follows [5]:

$$p_y(y) = \sum_{i=0}^{w-1} f(i, y) \tag{1}$$

Figure 4 Vertical projection graph

The resulting area with license plate is detected according to peaks in graph of vertical projection (Figure 4). Detected area is limited only vertically (Figure 7).

Skew correction

The vehicle arriving through the access ramp is not always captured concurrently with the camera. License plate angle can be computed applying Hough transformation on image. Firstly, Canny edge detector can be applied for better edge recognition, then Hough transformation evaluates skew of license plate object represented by red line (Figure 5). Both Hough transformation and Canny edge detector are implemented using OpenCV library functions.

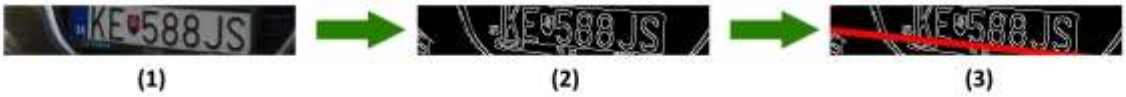

Figure 5 Canny edge detection and Hough transformation used for skew detection

If the horizontal axis of license plate is not concurrent with camera horizontal axis, the license plate is skewed. This transformation is in mathematics called shear transformation and the object skew can be corrected by applying the transformation matrix as follows [6]:

$$T_{Sy} = \begin{bmatrix} 1 & S_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad (2)$$

where S_y is a coefficient of shear in direction of y axis, $S_y = -\tan(\theta)$ and θ is an angle between red line and x axis.

Vertical projection

After skew correction of vertically detected license plate area a horizontal projection of this area can be computed. Let w be width of detected area, then horizontal projection represents the sum of all pixel values in points $[x, y_j]$ of this image part, where $j \in < y_{b0}, y_{b1} >$.

Horizontal projection is mathematically defined as follows [5]:

$$p_x(x) = \sum_{j=y_{b0}}^{y_{b1}} f(x,j) \qquad (3)$$

The black frame of license plate is detected as black and white contrast in the image. This contrast is detected using derivation of horizontal projection resulting in the graph. Boundaries of license plate are represented by peaks in this graph (Figure 6).

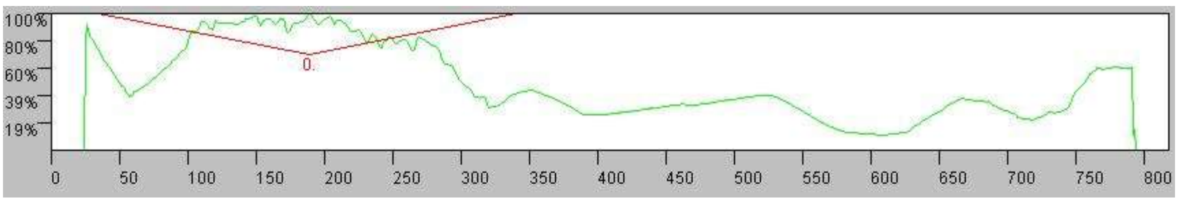

Figure 6 Horizontal projection

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Figure 7 Area detected by vertical projection (green) and horizontal projection (red)

IV. LICENSE PLATE NUMBER RECOGNITION

The result of license plate detection is described in picture (Figure 7) by red lines. Before applying proper character recognition method, it is essential to segment license plate numbers.

Segmentation is achieved by following algorithm:

1. The maximum value x_{max} horizontal projection $p_x(x)$ is computed.
2. Left x_L and right x_R boundaries of maximum value peak in graph of projection $p_x(x)$ are specified.
3. Projection $p_x(x)$ is set to zero on interval $\langle x_L, x_R \rangle$.
4. If the peak value x_{max} is above minimum peak value, algorithm follows with next step, otherwise algorithm ends.
5. License plate object is divided in point x_{max} and algorithm follows with step 1.

Process of segmentation divided license plate object into *n* segments $s_0, s_1, \dots, s_n$ (Figure 8).

Figure 8 Detected segments in license plate image

Segments are run in the cycle where each segment is converted to black-and-white image. Then the adaptive thresholding for contrast enhancement and median filter for noise reduction are applied. These filters are implemented using OpenCV library. Number recognition is implemented using Tesseract library function *doOCR(File image)*, where *image* is concrete segment.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

V. CONCLUSION

The resulting program described in this paper is the application for detection of car license plates entering through the access ramp. Cars are captured on video file, which is the input for the application. The advantage of this solution is fast detection in case of concrete image frames. With proper application of tracking algorithms provided by OpenCV library it would be also possible to reduce detection on few concrete frames and reduce computation requirements even more. Another advantage of this solution is independency on license plate country. The application can detect license plates from any country.

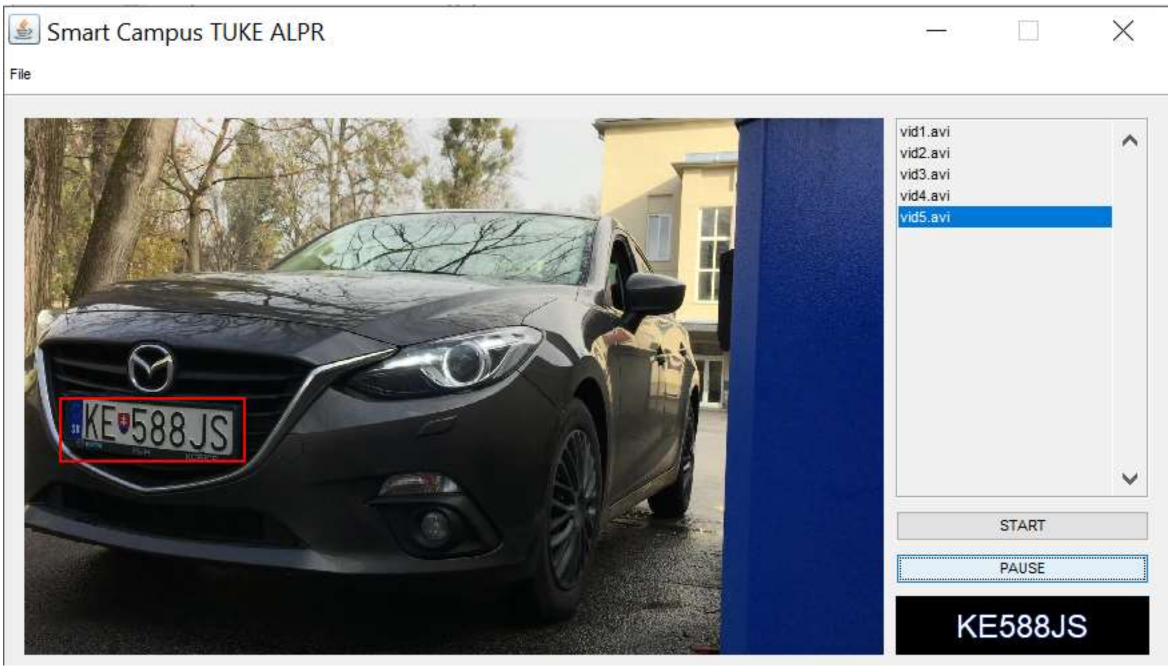

Figure 9 License plate detection using implemented application

Application was tested on five video files in 1920×1080 px resolution. Videos were captured in exterior during different lighting conditions. A time needed for detection and reading of license plate in case of whole video as well as partial frames was tracked. The results were evaluated as average values after ten repetitions of measurements. Average time for detection of one license plate was estimated on 511,4 ms. The results are presented in table below (Table 1).

Table 1 Testing results

Video	Frame count in video	Average detection duration per one frame (ms)	Detection duration per video (s)
1	271	468	127,341
2	361	506	183,323
3	361	518	187,331
4	421	511	215,296
5	1201	578	694,799

VI. ACKNOWLEDGEMENTS

This publication is the result of the Project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, Phase II., ITMS: 313011D232, supported by the Research & Inovation Operational Programme funded by the ERDF. We support research activities in Slovakia/This project is being co-financed by the European Union.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

REFERENCES

[1] Elena Šikudová, "Počítačové videnie - Computer vision," http://dai.fmph.uniba.sk/~vittek/proseminar/microsites/pgv/Elena%20%C5%A0ikudov%C3%A1%202011/videnie.html
[2] OpenCV, "OpenCV documentation," https://opencv.org/about.html
[3] Tesseract, "Tesseract documentation," https://github.com/tesseract-ocr/tesseract/wiki
[4] J. Reichl, M. Všetička, "Encyklopedie fyziky: Použité metody převzorkování," http://fyzika.jreichl.com/main.article/view/1531-pouzite-metody-prevzorkovani
[5] Martinsky, O, "Algorithmic and mathematical principles of automatic number plate recognition systems," in Technická univerzita v Brne, 2007
[6] B. Sobota, M. Šujanský, "Počítačová grafika: Transformácie v PG," http://pg.kpi.fei.tuke.sk/?q=node/6

CITE AN ARTICLE

Lippaiová, D., Michalko, M., Kainz, O., & Jakab, F. (2018). AUTOMATIC LICENSE PLATE RECOGNITION SYSTEM USING OPENCV LIBRARY. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, 7*(8), 57-64.