Automatic License Plate Detection

CS771 Course Project
Winter Semester 2015-16

Author:
Anurag Sharma(12146)
Anurendra Kumar(12147)
K.V Sameer Raja(12332)
Shreesh Ladha(12679)

Supervisors:
Prof Harish Karnick

Abstract

Automatic License Plate Recognition is an important problem in Computer Vision and Image processing. There are many applications ranging from complex security systems to common areas and from parking admission to traffic control. Automatic license plate recognition has complexity due to diverse effects such as of light and speed of the vehicle. In this project report we explore the methods to detect number plate in a frame using machine learning methods. We first use some basic image processing techniques (which uses some properties of number plate like white background etc.) to filter out possible objects for number plate and then use trained model to deteect number plates among them.

Contents

1 Introduction

Automatic License plate recognition (ALPR) algorithms in images or video segments are generally composed of the following three steps: 1) extraction of a license plate region; 2) segmentation of the plate characters; and 3) recognition of each character. This task is quite challenging due to the diversity of plate formats and the nonuniform outdoor illumination conditions during image acquisition. Therefore, most approaches work only under restricted conditions such as fixed illumination, limited vehicle speed, designated routes, and stationary backgrounds [1].

In this project we focus on the task of extracting the license plate region from the video segments of the data.

1.1 Licence Plate Detection

The license plate of a vehicle remains as the principal vehicle identifier. The video surveillance methods rely heavily on robust Licence Plate Recognition systems. The focus of this work is on detecting licence plate on the given data and do it in an online manner.

1.1.1 Literature Review

The techniques based upon combinations of edge statistics and mathematical morphology [2–5] have been proven to give good results. Typically in these methods, gradient magnitude and their local variance in an image are computed. They are based on the property that the brightness change in the license plate region is more prominent and easily detectable than otherwise. Then, regions with a high edge magnitude and high edge variance are identified as possible license plate regions. A disadvantage is that edge-based methods alone can hardly be applied to complex images, since they are too sensitive to unwanted edges, which may also show high edge magnitude or variance [1].

2 Methodologies Used

2.1 Histogram of Gradient descriptors

HOG features have been introduced by Navneed Dalal and Bill Triggs [6] who have developed and tested several variants of HOG descriptors, with differing spatial organization, gradient computation and normalization methods.

The essential thought behind the Histogram of Oriented Gradient descriptors is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The implementation of these descriptors can be achieved by dividing the image into small connected regions, called cells, and for each cell compiling a histogram of gradient directions or edge orientations for the pixels within the cell. The combination of these histograms then represents the descriptor. For improved performance, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination or shadowing.

It is mainly used for object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

We have used HOG descriptors for converting image dataset into feature-space which can be easily used by various machine learning algorithms.

2.2 Sliding Window Technique

In the context of computer vision (and as the name suggests), a sliding window is rectangular region of fixed width and height that slides across an image.

By combining a sliding window with an image pyramid we are able to localize and detect objects in images at multiple scales and locations. While both sliding windows and image pyramids are very simple techniques, they are very useful in object detection.

2.3 Selective Search

Exhaustive search of window has to evaluate for whole of the video frame and search in very large space. To reduce possible search space , authors in [7] took hierarchical graph based segmentation as supervised data to find possible object locations.

2.3.1 Segmentation as selective search

Based on the observation that appearance and immediate nearby context are effective for object recognition.Steps for segmentation:-

- Normalized graph-based initial oversegmentation from [8]
- Group adjacent regions on texture and region level similarity.
- Consider all scales of hierarchy

Finally search with different classifiers on predicted locations.

2.4 Contour Detection

Since Number plates have fixed contour with very less varying sizes and shapes, we tried to use basic image processing techniques to improve our results. We found contours using two different methods viz. HSV and Otsu's Method. The former method transforms RGB image to HSV image and looks for regions which are above a certain threshold (this threshold corresponds to white patches in RGB image) and returns the required patches in image. On the other hand, the latter method also works on similar lines but uses histogram based techniques to detect regions of interest. This technique is generally used for foreground and background seperation.

3 Experimental Setup

3.1 Dataset Description

Our goal when generating data set is to ensure that videos conform to the basic requirement of uniform representation. The data from the camera is continuous and so it contains data across all times of the day and all days of the year. This data can then be divided into training, validation and testing sets. We have chosen 8 minutes of video randomly from every hour during the day time to capture different illumination levels in an appropriate way.

3.2 Experimental Methodology

For creating training dataset we used four videos of lengths 15min, 1 min, 45sec and 40 sec which accomodated brightness variations(various time orious time o). After extracting frames for all the videos and filtering out for the frames which contained numberplates, we purged the blurred number plate regions. Our final data set consisted of 1414 numberplate regions (each region is cropped from a frame). For generating negative samples, we used some of the contour regions of different areas which are not intersecting with the numberplate regions. This resulted in a total of 11241 negative samples. We used a 45 second video for testing real time numberplate detection.

- Take each frame of the video.
- Convert each frame from BGR to HSV subspace.
- Tune the hyper-parameters for appropriate contour detection.

Another technique we applied to improve the results

- Convert each image from BGR to grayscale.
- Do Otsu thresholding to convert to bi-model histogram. Otsu algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal,

Finally used heuristics such as area of contour, width, height and their ratios for possible number plate location. We used the state of the state of the art classifiers SVM (with different C), logistic regression and adaboost classifier to detect number plate.

4 Results and Conclusions

4.1 Using Otsu's and AdaBoost Method

4.2 Using HSV and SVM Method

C SVM	Logistic	
1	0.927	0.9841
11	0.959	0.9841
21	0.967	0.9841
31	0.969	0.9841
41	0.972	0.9841
51	0.973	0.9841

Table 4.1: SVM and Logistic

Number of tree	Adaboost
1	0.973
11	0.984
21	0.983
31	0.991
41	0.9902
51	0.9904

Table 4.2: A

We trained and tested using 5-fold cross-validation by applying classifier on extraxted positive and negative examples to check whether classifier gives error or sampling strategies gives error. Following is the recall of positive examples.

The codes enclosed with the report have been developed using help from [9] and [10].

Acknowledgement

Firstly we would like to thank Prof. Harish Karnick for assisting. We are grateful for his patience and persistence with us as his students.

Finally We would like to thank Department of Computer Science and Engineering, IIT Kanpur for giving us the resources and infrastructure required for the project.

Bibliography

[1] Christos-Nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Ioannis D Psoroulas, Vassili Loumos, and Eleftherios Kayafas. License plate recognition from still images and video sequences: A survey. *Intelligent Transportation Systems, IEEE Transactions on*, 9 (3):377–391, 2008.

[2] Fernando Martin, Maite Garcia, and Jose Luis Alba. New methods for automatic reading of vlps (vehicle license plates). In *Proc. IASTED Int. Conf. SPPRA*, pages 126–131, 2002.

[3] Bai Hongliang and Liu Changping. A hybrid license plate extraction method based on edge statistics and morphology. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 831–834. IEEE, 2004.

[4] Danian Zheng, Yannan Zhao, and Jiaxin Wang. An efficient method of license plate location. *Pattern Recognition Letters*, 26(15):2431–2438, 2005.

[5] SherrZheng Wang and Hsi-Jian Lee. Detection and recognition of license plate characters with different appearances. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 2, pages 979–984. IEEE, 2003.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[7] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2): 154–171, 2013.

[8] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[9] Pyimage. http://www.pyimagesearch.com/2015/03/23/sliding-windows-for-object-detection-with-python-and-opencv/.

[10] Opencv. http://docs.opencv.org/3.1.0/df/d9d/tutorial_py_colorspaces.html.