Automatic Vehicle Counting Approach through Computer Vision for Traffic Management

Arman Allamehzadeh[1], Mohammad S. Aminian[2], Mehran Mostaed[1] and Cristina Olaverri-Monreal[1[0000-0002-5211-3598]]

[1] University of Applied Sciences Technikum Wien, Department of Information Engineering and Security, Höchstädtplatz 6, 1200 Vienna, Austria
[2] Vienna Technical University, Institute of Transport Planning and Traffic Engineering, Vienna, Austria
{arman4u, mehran.mostaed7}gmail.com, m.s.aminian@hotmail.com, olaverri@technikum-wien.at

Abstract. Technology based on sensors or cameras that is related to the field of Intelligent Transportation Systems (ITS) can help to alleviate road congestion problems by collecting and evaluating real time traffic data. In this paper, we present an approach to monitor traffic by collecting and processing video streaming information for further analysis in traffic management centers. Results showed a 94 % rate of correct vehicle detections in a short period of time with a low rate of false detections.

Keywords: Vehicle Counting, Computer Vision, Real-time Traffic Management Systems

1 Introduction

Technology based on sensors or cameras that is related to the field of Intelligent Transportation Systems (ITS) can help to alleviate road congestion problems by collecting and evaluating real time traffic data [1]. In traffic management such data is used, for example, to suggest alternative routes to divert traffic from a congested road, or to display other road-related information to drivers to maximize traffic flow and prevent congestions [2].

Traffic monitoring systems supervise traffic volumes and are able to derive other data such as vehicle classification and weight, which is later analyzed through parameters such as speed, density and flow. Speed is defined by the distance traveled per unit of time, density is the number of vehicles per unit length of the roadway and flow is the number of vehicles passing a specific reference point per unit of time [3]. In this paper, we present an approach to traffic counting in roads that are equipped with surveillance cameras. Relying on the EmguCV library for image processing and .NET technologies, traffic information is collected, processed and sent to a server for later statistical analysis.

The final authenticated version is available online at https://doi.org/10.1007/978-3-319-74727-9_48

2 Related Work

Current traffic counting techniques mostly rely on inductive loops, piezoelectric, infrared or radar sensors. Although these sensors provide macroscopic information about traffic, they cannot document data for single vehicles without being combined and synced with cameras [4, 5, 6].

Computer vision techniques make it possible to exploit traffic cameras solely as a sensor for vehicle detection, counting, tracking and even for speed measurement. Much of road networks are already equipped with surveillance cameras. Cameras are non-intrusive and are used to capture data at high resolution. Like human eyes, cameras capture the scene with details that other sensors like radar, ultrasonic and lasers cannot detect [7]. Hence road operators are very interested in exploiting this technology as tool to derive the real-time traffic information that they need for optimal management of their networks. Data collection technology relying on cameras can have other interesting applications for security tracking, fleet control and electronic toll collection.

Real-time image and video processing to derive traffic data has been applied in many works. A broad review of the literature has been compiled in [8] and cited in this work. For example the robust system Autoscope classified vehicles in real-time and provided traffic analysis reports [9, 10]. A system for counting vehicles and measuring their speed in complex traffic scenarios was presented in [11], and more recently the authors in [12] proposed a hybrid method, based on background subtraction and edge detection for vehicle detection and shadow rejection, to classify and count vehicles in multilane highways. A further approach was proposed through a video analysis method for vehicle counting in [13]. The authors relied on an adaptive bounding box size to detect and track vehicles according to their estimated distance from the camera, given the geometrical setup of the camera. A vehicle counting method based on blob analysis of traffic surveillance video was additionally described in [14] through moving object segmentation, blob analysis, and tracking.

Relying on the techniques of this last work, we present in this paper an approach for vehicle counting in multiple lanes for further analysis in traffic management centers.

3 Technical Implementation

Our approach used the EmguCV vision library in C# to detect the vehicles and process the images. As a platform for developing, running and evaluating the code, we used Visual Studio IDE. All the data was stored in the Microsoft SQL Server. Vehicles are counted individually in the direction that they are travelling.

A login system prevents public access to the records. Black and white are used in order to reduce the size of the stored images. The system works with a live stream in real time using recorded video files. We selected the JSON format for sending the data.

3.1 Video Stream Analysis

A video stream was provided by either a camera or by a recorded video (e.g. by CCTV). After receiving the data, the image processing procedure was initiated and consisted of the following blocks:

- Background building. The background was determined by comparing sequential series of frames through OpenCV functions to create a scene with no vehicle that was then used as background.
- Comparison of two sequential frames. Comparison points for two sequenced frames were determined and a threshold was calculated that enabled the detection of a frame with distinct black/white contrast. The resulting image noise from this operation was then reduced by using Gaussian Blur and morphological image processing noise filters. To reduce the computing process, we first obtained an input frame from the camera, recorded the following frames 2 and 3, and then compared both frames to create a new one that resulted from subtracting them.
- Movement detection. Having subtracted the background from the threshold frame, we obtained a new frame and determined the contours or curve by joining all the continuous points that had the same color or intensity.

3.2 Filtering Process

In this section we provide a more detailed description of the process used to filter the frames sequentially. As stated above, we applied a threshold on the frame to differentiate between black and white. We then applied two common operators to eliminate noise:

- erosion that slid around by using a slider (kernel) with a size of 3 x 3 pixels and that delivered white if all the pixels were white, or otherwise remained black.
- dilation that slid around, and if the entire area was not black, it was converted to white.

Finally we extracted the Binary Large OBject (BLOB) that consisted of a group of connected pixels in a binary image, in order to differentiate between the "large" objects that were relevant to this work (i.e. vehicles) and the other "small" objects that were defined as noise. The image processing procedure is depicted in Fig. 1.

3.3 Traffic Counting

Relying on the approach in [15] to prevent counting the same vehicle more than once, a horizontal line was inserted in the image that had to be crossed (see Fig. 2). Vehicles were detected and counted only after they cross the line. Afterwards, the data record was sent to the database.

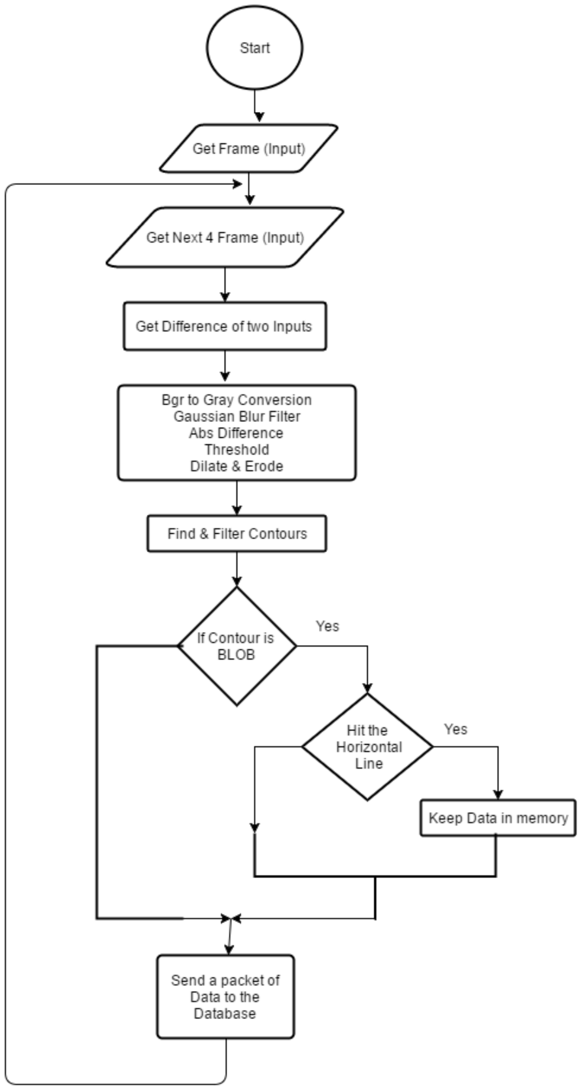

Fig. 1. Image processing procedure for vehicle detection and counting

3.4 Database Storage

After obtaining the information related to the time stamp, scene image, and plate number, we stored it in a database and repeated all of the above mentioned steps until receiving the stop command. Fig. 3 illustrates the process. The administrator is responsible for granting permission to the users to access the data for analysis as well as for configuring new cameras or removing them from the street. Users can be employees who work with a traffic management system.

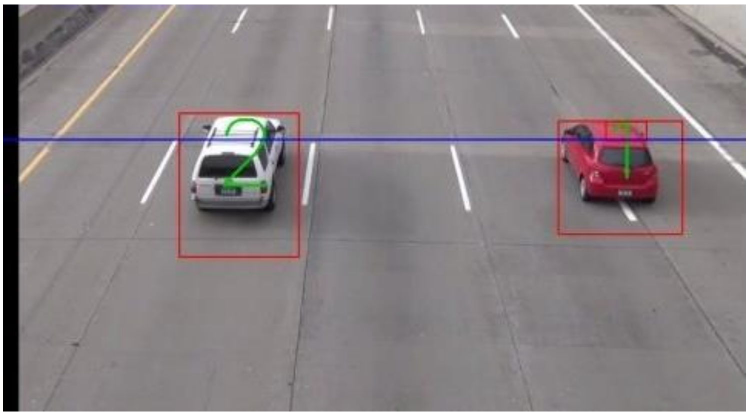

Fig. 2. View of the horizontal line that represents the threshold for counting the vehicles.

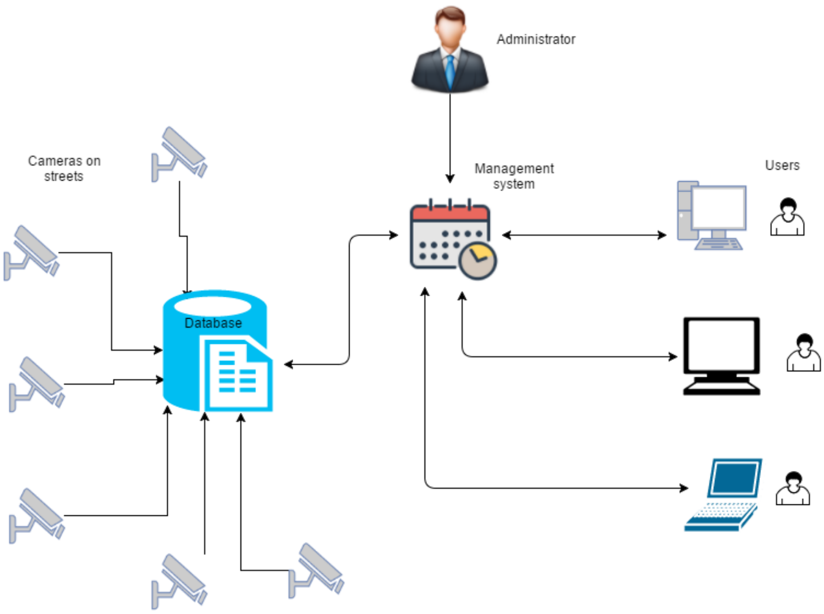

Fig. 3. Traffic data acquisition and storage process

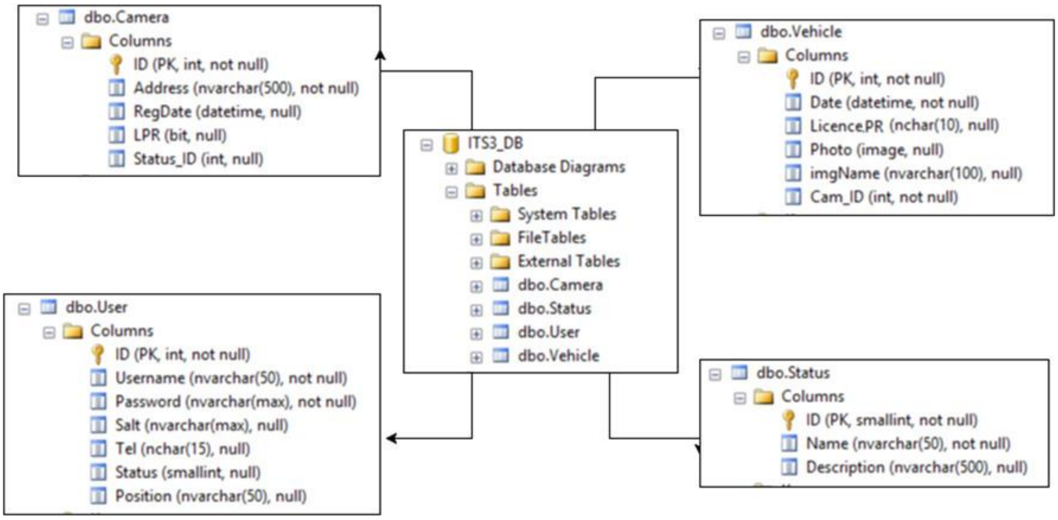

Fig. 4. Database architecture for the storage of the acquired data

The database contained 4 tables as depicted in Fig. 4 and described below:

- Camera: One or several cameras that can be defined for each street or road as well as been configured, and removed.
- Vehicle: to save the pertinent collected data (e.g. detection time, location id, etc.).
- Status: status of street camera (enabled/disabled)
- User: user data such as username, password, etc.

Licence Plate Recognition (LPR) was additionally defined as a field in the camera and vehicle tables for future use.

4 System Evaluation Results

We evaluated the implemented system by examining a received video stream provided by a video file relying on the work presented in [15] recorded under daylight conditions and using a proper camera installation. The system was evaluated in 6 different streets. Initial evaluations delivered some problems related to the duplicated detection of the same vehicle in two different scenes, but this was solved in the last version.

Final results (see Fig. 5) showed a rate of 94 % correct vehicle detection in a short period of time. Results showed a very low rate of false detection. Only in case 6 was the camera position not optimal and the number of false detections increased.

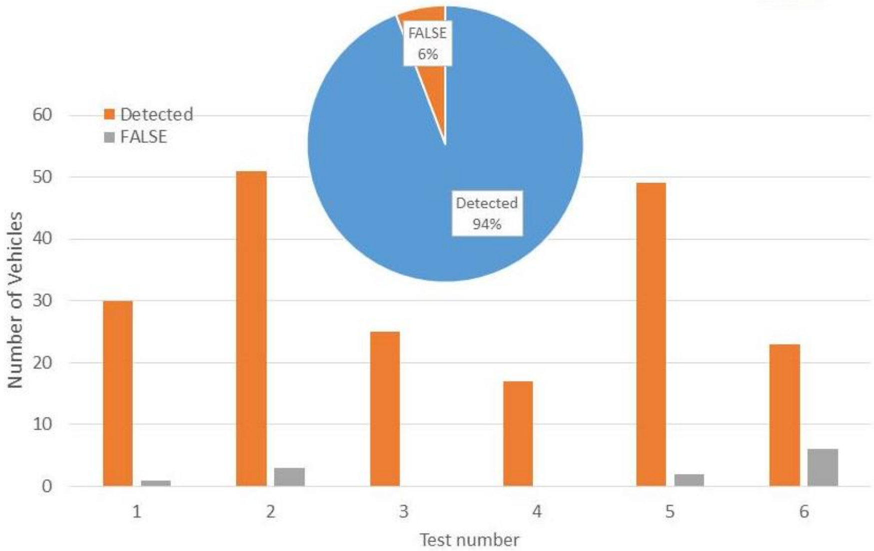

Fig. 5. Results regarding the vehicle detection rate

References

1. Olaverri-Monreal, C. Automated Vehicles and Smart Mobility Related Technologies. Infocommunications Journal, 8(2), 17-24 (2016).
2. Goncalves, J., Goncalves, J. S., Rossetti, R. J. and Olaverri-Monreal, C. Smartphone sensor platform to study traffic conditions and assess driving performance. In: Proceedings International Conference on Intelligent Transportation Systems (ITSC), 2596–2601 IEEE, Qingdao, China (2014).
3. Hall, F. L. Traffic Stream Characteristics, in Traffic Flow Theory. US Federal Highway Administration, 1-32 (1996).
4. Traffic Monitoring Guide, 2013, U.S. Department of Transportation https://www.fhwa.dot.gov/policyinformation/tmguide/tmg_fhwa_pl_13_015.pdf, last accessed 2017/04/21.
5. Pengjun Z. and McDonad M. An Investigation on the Manual Traffic Count Accuracy, Procedia - Social and Behavioral Sciences 43, 226-231 (2012).
6. Skszek, S. L. State-of-the-Art Report on Traffic Non-Traditional Counting Methods, Arizona Department of Transportation in cooperation with U.S. Department of Transportation. Arizona (2001).
7. Ahmadi-Pour, M., Ludwig, T., Olaverri-Monreal, C. Statistical Modelling of Multi-Sensor Data Fusion. In: Proceedings International Conference on Vehicular Electronics and Safety (ICVES), IEEE, Vienna, Austria (2017). In press.
8. Loce, R. P., Bernal, E. A., Wencheng W., Raja B. Computer vision in roadway transportation systems: a survey, Journal of Electronic Imaging 22(4), 041121 (Oct–Dec 2013)

9. Michalopoulos, P. Vehicle detection through image processing: the Autoscope system, IEEE Trans. Veh. Technol. 40(1), 21–29 (1991).
10. Carlson, B. Autoscope clearing the congestion: vision makes traffic control intelligent, Adv. Imaging 12(2), 54–56 (1997).
11. Ali, A., Bulas-Cruz, J. and Dagless, E. Vision based road traffic data collection. In Proc. ISATA 26th Int. Conf., Croydon, pp. 609–616 (1993).
12. Yu, M., Jiang, G. and Bokang, Y. An integrative method for video based traffic parameter extraction in ITS. In Proceedings Asia Pacific Conf. on Circuits and Systems, 136–139, IEEE (2000).
13. Bas, E., Tekalp Murat. A. and Sibel Salman F. Automatic Vehicle Counting from Video for Traffic Flow Analysis. In: Proc. Intelligent Vehicles Symp. 392–397, IEEE (2007).
14. Chen, T. H., Lin, Y. and Chen, T. Y. Intelligent vehicle counting method based on blob analysis in traffic surveillance. In: Proceedings Second Int. Conf. on Innovative Computing, Information and Control, 238–242, IEEE (2007).
15. Dahms, C. MicrocontrollersAndMore, https://github.com/MicrocontrollersAndMore, last accessed 2017/02/20.