Lithuanian traffic monitoring system (LTTMS) based on Android

Yan Wan¹, Tomas Jankauskas²*, Lamptey Khalid Odartey ³

School of Computer Science and Technology, Donghua University, Shanghai, China

*Corresponding author e-mail: tomastomasj@gmail.com

Abstract. The number of vehicles is growing exponentially all over the world. Therefore, spreading a real-time information to commuters is necessary. Recently, the GPS availability on smartphones and abundance of license plate recognition algorithms made a mobile phone-based traffic monitoring system highly convenient. In this paper we will introduce a transportation monitoring system for Lithuania. A country which is lacking a reliable traffic application capable of generating traffic data, informing users about the situation on the road and helping to prevent crimes.

1. Introduction

Transport is one of the most important infrastructures of any country. As the times goes by, the number of vehicles and crimes on the road is drastically increasing, traffic systems are becoming more complex. Therefore, the necessity of monitoring these activities has clearly turned into a problem of a huge importance. One of the main issues related to the transport infrastructure are the crimes and the traffic jams. A way to solve this is to develop a reliable and user-friendly traffic monitoring system that could be accessible to everyone and would successfully deal with these difficulties.

In recent years, traffic congestions and accidents have brought a significant loss of time, damage of property and pollution of the environment. As a result, we have to advance the traffic management and invoke technologies that could lead to improvement of traffic issues and would reduce the pressure on the road. With the help of collected real-time traffic information, which is gathered by its users, the system can identify current traffic operation and flow conditions and help to predict the traffic flow in the near future, help drivers choosing optimal routes.

The License Plate Recognition (LPR) system is a powerful crime fighting tool. License plate numbers help to uniquely recognize a vehicle. By identifying car’s plate number, the system can detect a trail of convicts’ movement and quickly reveal the crime. License plates have various styles, colors in different countries. Every country has exclusive license plate format, therefore, they develop the LPR system suitable for the vehicle’s plate number. Each system uses different combination of algorithms.

One of the countries that doesn’t have a highly developed traffic system is Lithuania. In 2016, there were 1.5 million registered motor vehicles and 39.7 thousand registered motorcycles and mopeds in the country. Keeping in mind that the country has only around 2.8 million people, these are significant numbers. Therefore, there is a need to develop a system that could help to manage and observe the traffic in the country. It should be convenient and free.

The main goal of this paper is to present the system for Lithuanian traffic tracking which is capable to detect and tag the license plates and traffic accidents in the real-time utilizing wireless
communications. The proposed architecture allows us to collect the traffic data that is generated by the users’ mobile devices based on Android platform.

2. Related work

The main goal of our system is to detect license plates and do the monitoring of traffic problems. Therefore, we will try to have a brief overview about previous works related to these topics.

2.1 LPR

Nowadays, many different LPR techniques have been introduced. However, even though LPR has been researched for decades, there are yet many shortcomings that reduces the recognition precision.

Sobel-based technique with morphological operations was used to identify 610 Saudi Arabian plate numbers [2]. The LPR system has achieved a high precision – roughly 95%. However, it did not cover the entire spectrum of Saudi Arabian plate numbers.

In Turkey, a technique based on edge detection and fundamental morphology operators such as erosion, filtering convolution. Smearing algorithm was used to scan the entire image in horizontal and vertical planes. This project managed to achieve 92% accuracy [3]. Despite the high precision, the system could not recognize alphabetic letters such as Y, X and Q.

An Australian LPR system, using two techniques that include neural network used for character recognition, was proposed. The system could show high accuracies with digital images and the standard Australian plate number [4].

2.2 Traffic monitoring systems

In recent years, there were some iterations to build a basic traffic monitoring systems and also, currently, we could notice efforts to implement the crowd-sourcing and the extraction of location data. To get a deeper insight about the problem and analyze possible solutions we will take a look at the systems that already exist.

Waze, a geographical GPS-based navigation application built for android devices with GPS support and a display, which provides client with turn-by-turn data and travel times submitted by the users as long as the route details. Waze is an application driven by community. The complementary map and traffic data is being gathered by its users. [5]

Google Traffic is a Google Maps’ feature that displays traffic conditions on main roads and highways and can observed on a smartphone with a Google Maps application. It analyzes the GPS locations which is generated and transmitted to Google by its users. [6]

“Beat the Traffic” offers a real-time traffic information, color-coded map with areas of traffic jams and live traffic cameras. It gathers information from smartphones and navigation systems in car on the roads and highways. Users can also send manual reports by sending information about accidents, hazards, constructions zones, etc. [7]

3. Methodology

We will shortly describe techniques and methods used to build the LTTMS system.

3.1 OpenALPR

LTTMS does the license plate recognition using an open source C/C++ library, an LPR system called OpenALPR.

There are 8 stages that OpenALPR follows to complete the license plate recognition (figure 1): 1. License plate detection, 2. Binarization, 3. Char analysis, 4. License plates’ edges computation, 5. License plate image deskew, 6. Character segmentation, 7. Optical Character recognition (OCR) 8. Post processing. For simplicity, we divided these steps into three groups: 1. License plate detection, 2. License plate character recognition, 3. Post processing.

We will define all 8 stages processed by OpenALPR system to recognize a Lithuanian plate number
3.2 OpenALPR modules

**Figure 1.** Stages of License Plate Recognition System.

**Figure 2.** LPR phases: A) LP detection; B) Binarized LP; C) Character sized regions in LP area; D) LP with precise edges; E) LP image deskew; F) LP after character segmentation

**Module 1: LP detection**

**LP area detection**

The Local Binary Patterns (LBP) algorithm is applied to classify and find possible license plate regions and the values of its dimensions: x and y plane, width and height. The main concept of LBP calculation is to compare each pixel with the neighborhood and summarize the local structure of the picture. If the intensity of the center pixel is larger or equal to the intensity value of the nearby pixel, then set the value of a center one to 1, otherwise it is 0 [8] (Figure 2 A).

**LP image binarization**

Binarization is a process that takes place in every found LP region during the license plate detection stage. During this phase, many binary images are produced for every license plate region, helping to increase the probability of detecting the characters [9] on the plate (Figure 2 B).

**Character Analysis**

The main goal of this phase is to discover the regions with the size of a character in the license plate area. First, all the connected blobs in the region of license plate have to be found. Next, blobs which are approximately the width, height of LP character and have tops/bottoms that are arranged in a straight line with other similar blobs, have be found. In every area, this analysis is performed at least a few times. If nothing was found in the region, no further process will happen. However, if potential characters are found, it is saved and the next steps take place (Figure 2 C).
Compute LP image edges

In this phase, the main idea is to discover the accurate position and edges of the license plate. First of all, all the Hough lines \([10]\) for the license plate region must be found. It is done processing the image of a license plate and doing some computations on a list of vertical, horizontal lines. Then, we need to use this list with characters height which is found during Character Analysis.

Perform LP image deskew

The deskew changes the license plate region to a standard size and orientation. It is done by using the LP edges information. In an ideal case, after the deskew stage, we should have correctly oriented LP image, without any rotation or skew.

Module 2: LP characters recognition Character Segmentation

Main goal in this phase is to isolate the characters that combine the entire LP view. A vertical histogram helps to discover gaps between the LP characters. During the analysis, character boxes are cleaned up.

OCR (Optical Character Recognition)

OCR uses Tesseract OCR engine and it recognizes every single character independently. It calculates possible characters and their confidence rates.

Module 3: Post processing

Post processing is responsible for choosing the best plate letter combinations after the OCR phase. All characters below a specific threshold are disqualified. During the post processing, the region validation is handled if requested.

3.3 Training OCR

To improve the precision of the plate detection, the OCR of OpenALPR can be trained. After gathering a large library of proper Lithuanian license plate images (~200) and processing them, we have used the binaries package provided by the OpenALPR to train the new data \([11]\).

3.4 Design of the system

There are three main components in the system: 1) android client; 2) web server; 3) database. For an android device to communicate with the web server, HTTP libraries have to be invoked. PHP scripts are used to establish their communication. Similarly, we use PHP to make connection between web client and the MySQL database.

The Lithuanian monitoring system can deal with a few problems. First of all, it can detect and report a crime done by a vehicle. Car’s license number and location is used as an evidence of illegal behavior. Application can access the camera and capture an image. The picture is cropped by manually selecting a desired region. This region is processed by OpenALPR which is responsible for the recognition of the plate. Characters of the license plate and location of the car is sent to the database. This process is represented by blue color in figure 1.

LTTMS can also be utilized to record an on-road traffic situation. It takes an image, address and definition as the values to be stored in database. All this information is sent through the web server from which with the help of PHP scripts it is delivered to our dataset (figure 1, blue arrow).

LTTMS users can also send a request (figure 1, red arrow) and require for an update of crime (LPR) or traffic report data. If the dataset is not empty, the information will be displayed to the android user (figure 1, green arrow).
3.5 XXAMP Web server

XAMPP is a simple, lightweight Apache distribution that we used to deploy and test web application on a localhost level [12]. It is suitable for experimenting MYSQL with PHP.

In our android project, we wrote a short PHP script so that XAMPP can connect a Web server to the dataset. At the moment when android user sends a source of information, XAMPP receives it and a corresponding PHP file gives the further commands, redirects data to the MySQL dataset where it is stored. In a similar manner, when an android user requests to be informed about the traffic or plate number dataset status, a PHP files help to retrieve information from the MySQL and send it to the web client. For testing purposes of the project we run two XAMPP modules: the web server Apache and MySQL database module.

3.6 MYSQL Database

MySQL is an open source relation database management system (RDMS) based on standard computer language for data management and manipulation called structured query language (SQL).

Development of the Lithuanian traffic monitoring system brought a demand to build two datasets. One of them stores Lithuanian plate numbers with a short definition of a crime and its location that complements to that specific license number (figure 4 A). The dataset is used both for storing and retrieving the record.

The other table saves traffic event information. It contains the image name of traffic accident with geographical coordinates (latitude and longitude), the description of the traffic problem (figure 4 B). Same as for crime_event table, traffic_event is utilized both for storing and getting the data.

Figure 3. A block diagram of the system’s design

Figure 4. MYSQL datasets: A) criminal activities; B) traffic problems on the road
4. Experimental results

During the experiment, we have trained the OCR to improve the recognition rate of Lithuanian license plates. A large library of high quality images of different angles was gathered and used utilizing the OpenALPR training binaries. Furthermore, we have excluded the letters that are not present in Lithuanian plate numbers (Q, W, X). This fixed the error for most cases, when O is mistakenly detected as a Q. We have tested 246 images to see how the accuracy has changed. Our Lithuanian license plate data is divided into three groups – small visualization angle, large visualization angle, plate with “O”. A noticeable improvement in the recognition rate can be seen (table 1).

**Table 1.** Recognition accuracy with trained and untrained OCR and different plate number cases

| Plate number case              | Untrained OCR (%) | Trained OCR (%) |
|-------------------------------|-------------------|-----------------|
| Small visualization angle     | 90.4              | 92.6            |
| Large visualization angle     | 84.3              | 90.7            |
| Plate with “O”, small angle   | 88.9              | 91.3            |

In addition, the communication between the application and our database (MYSQL) was tested. First of all, we have tried to examine the LPR part of the system. We have captured an image of a car with Lithuanian license plate (figure 5, 1). It was processed by OpenALPR system which has recognized the plate number. The data and the image were sent to the XXAMP web server, where following commands of PHP scripts, data was stored in MYSQL database. We have also tried to retrieve plate numbers’ information from dataset (figure 5, 2). All the steps were successful.

![Figure 5. Stages of application’s LP detection block (sending and retrieving data)](image)

Traffic problems block was tested. This activity requires three items to successfully send the report. First of all, the user has to obtain the location of the event. By pressing the ‘Get Location’ button one can open the Google map and see the marker there. Initially, it is located in the current position of the user according to the GPS. However, if it is not precisely correct, user can do the long click on the marker and drag it to a desirable spot. If the accuracy is satisfying, a ‘CONFIRM’ button is pressed. The second item is the image of the traffic accident. It can be captured by pressing the round button with a camera logo on it. It is in the main ‘traffic report’ menu. After the image is taken, it will be visible in the middle section of the menu. Finally, the description of the event. If needed, the client can include a short definition to make the report less ambiguous and understandable to other users. Similarly to plate number report, we confirmed the traffic accident record and sent it to the
traffic_report database. Later on, we could also successfully retrieve the data from the server (figure 6).

Figure 6. Traffic status block

5. Conclusion
In this paper we present an android APP in Lithuania that is capable of dealing with crimes by invoking trained OpenALPR and is able to recognize, tag traffic problems with the help of Geocoding. During the experiment we have seen a noticeable improvement in recognition accuracy in all cases: small visualization angle (90.4% vs 92.6%); large visualization angle (84.3% vs 90.7), plates with “O” (88.9% vs 91.3%). Our system was able to capture the image and depict the location on the map as well as give us the exact address and do the license plate recognition. Moreover, we have successfully sent and retrieved the data from the database on android client. However, the speed of data traffic could be increased (less than 5 seconds). At this moment, the system cannot manage a lot of data (~255 records). Therefore, we need to find some enhancements towards this problem.

References
[1] “Transport and road traffic accidents statistics”, [Online]. Available https://osp.stat.gov.lt/
[2] M. Sarfraz, M. J. Ahmed, and S. Ghazi, "Saudi Arabian license plate recognition system," in Geometric Modeling and Graphics, 2003. Proceedings. 2003 international Conference on, 2003
[3] S. Ozbay and E. Ercelebi, "Automatic vehicle identification by plate recognition," World Academy of Science, Engineering and Technology, pp. 2005.
[4] L. G. Harney and C. Priest, "Automatic number plate recognition for australian conditions," in Digital image Computing: Techniques and Applications, 2005. DICTA ’05. Proceedings 2005, 2005
[5] "Waze," [Online]. Available: https://www.waze.com/about.
[6] "Beat the Traffic used as TV Broadcast Solution," [Online]. Available: https://en.wikipedia.org/wiki/Beat_the_Traffic#TV_Broadcast_Tr affic Solution. "Beat the Traffic," [Online]. Available: http://www.beatthetraffic.com/aboutus.htm.
[7] Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004
[8] J. Sauvola and M. Pietikainen. Adaptive document image binarization. Pattern Recognition, 2000
[9] C. Wolf and J. Jolion. Extraction de texte dans des videos: ’ le cas de la binarisation. In 13eme congres francophone de reconnaissance des formes et intelligence artificielle, Jan. 2002
[10] “Hough Line Transform”, [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html
[11] “Train OCR”, [Online]. Available: https://github.com/openalpr/train-ocr
[12] ”XAMPP”, [Online]. Available: https://en.wikipedia.org/wiki/XAMPP