Counting vehicle using javascript framework library with internet of things implementation concept to lighten up object detection

J Budiarto¹, J Qudsi¹ and M Tajuddin¹
¹STMIK Bumigora Mataram, Indonesia

E-mail: jian.budiarto@gmail.com

Abstract. Nowadays, the population growth of the metropolitan city in Indonesia has increased. This problem could make road density in this city moving faster. Government is one of the stakeholders should give information about road density. However, people cannot obtain information completely to suggest the best road to achieve. The people need information about traffic condition with real-time, automatically, and easy to use with the internet. This research proposed a browser approach application to lighten up counting vehicle. The JavaScript framework library was implemented on this proposed application by processing image and video. This research contains four steps methodology: identification, analysis, training and application’s development. The result is the application should be able to inform vehicles number on 95% of accuracy. CPU and memory usage based on browser approach is less than desktop application. A browser approach gives free memory 5154 Megabyte, and CPU usage is 19.8%. The approach application was contributed to solving lack of traffic information with low processing.

1. Introduction
Increasing of the population in the metropolitan city was moving faster. This increasing population affected the congestion problem in several areas. This problem occurred because the population did not balance than road capacity. Ina congestion condition, people want to get real-time information about road density. The government must provide technology development to serve people needs quickly, effective and efficient. Unfortunately, the government had some problem with image/video processing.

Currently, the development of image/video processing technology is increasing quickly. Image processing has been developed at several platform technologies and multilevel programming language. This platform not only developed at desktop GUI but also on mobile technology and browser website. Usually, programmers use some programming language such as Visual Basic, Python, C language (C/C++/C#), Android, etc.

Nowadays, development image processing at a browser platform is more implemented by the programmer. The reason is that this platform execution process is faster than desktop GUI and mobile android. This technology is needed to lighten up image processing from traffic video cameras. JavaScript Framework Library should be applied to count the vehicle by image processing. An image will be retrieved periodically and showed at browser application. This research use Trackingjs [1] become a framework library for the application.
Both highway and traffic flow are essential facilities to improve production and product marketing. Unfortunately, several problems affected road congestion and decreased economic growth. The increasing number of the vehicle must be anticipated to decide the policy quickly.

This research proposed a browser approach application to lighten up counting vehicle. Several Research had used a different method for each purpose [2, 3, 4, 5, 6, 7, 8, 9]. To solve a lack of traffic information, we developed an application to detect a vehicle with IoT concept.

Srinivas [2] conducted research using image processing to calculate the number of vehicles on the road. A camera device is placed in the test location and connected to a computer. The computer is assigned to take pictures in a period. His research used Canny Edge Detection to determine decisions on traffic control. The study was also able to detect emergency vehicles for priority. Subawe [3] conducted research to track and count vehicle from a video. The research use Pixel Based Adaptive Segmentor for Background Subtraction. This research used visual surveillance and cars as an object to track and count. Irawan [4] investigated the regulation of traffic signals based on the number of vehicles pass at an intersection. The results of this research are a method to calculate and analyze the waiting time of traffic light.

Taghvaeeyan [5] develop a portable magnetic sensor for vehicle counting, classification, and speed measurement. This sensor consists of wireless anisotropic magnetic devices are placed to the highway. By using this device, a system is able to analyze traffic flow immediately. Zhang [6] conducted research to guide driver moved from congestion if a road-lane did not balance with other road-lane at an intersection. This research used a method backpressure based routing algorithm.

Pornpanomchai [7] build vehicle detection and counting system using image processing. This system requires a video stream and capture to a video frame. Background image without vehicles and the frame with moving vehicles is collected to find the differentiation. Choudhury [8] build vehicle detection technique that can be implemented for traffic surveillance systems. The system uses Haar-like features for vehicle detection, which is generally used for face detection. Arunmozhi [9] used three methods for object detection approaches Histogram of Oriented Gradients (HOG), Haar-like features and Local Binary Pattern (LBP). This method is analyzed using camera images. Finally, this research proposed a combination of all the three different feature descriptors and Ada Boost cascade classification.

This research aimed to count the number of the vehicle using JavaScript Framework Library with Internet of Things (IoT) concept to lighten up object detection. This concept explains the interaction between a camera sensor and application server. The delivered data of image processing and sensor activity could be controlled using the internet.

2. Methodology
This research was implemented several steps which describe at the following paragraph.

2.1 Survey and identification
In this step, we verify the road conditions with the research environment. This environment needs an ideal research time which not have many number of vehicles. This survey aims to identify traffic disruption such as traditional market and road improvement. This disruption could be a problem in taking a picture.

This research location is Bung Kario Street, Cilinaya, Cakranegara, Nusa Tenggara Barat. This street is an essential way for business and education center. The object on taking a picture is car vehicle. The vehicle must have enough range from another object. A reason to give space each object is to identify more natural by the application.

In this survey, all vehicle especially car is counted on a 10 minutes periodic time. This result is shown by Table 1.
Table 1. The result of the survey to find suitable research time.

| Time       | Distance | Vehicles / 10 minutes |
|------------|----------|-----------------------|
| 08.00-09.00| 20 meter | 38                    |
| 11.00 – 12.00| 20 meter | 34                    |
| 14.00 – 15.00| 20 meter | 18                    |
| 17.00 – 18.00| 20 meter | 23                    |

The result of this survey shows that ideal research time is 14.00-15.00. The reason is on this period did not have many numbers of vehicles.

2.2 Training

This research use Go Pro Hero 5 camera for taking a picture. Speed setup to take a picture is 60 frames each second. This speed is enough to produce a high-quality image. Using this speed, camera able to record every vehicle movement. In this step, we need 135 images as a positive image. This positive image is collected from video extraction using VLC software. This result is shown in Figure 1.

![Figure 1. Positive image.](image1)

Before training step, this image is converted to grayscale image format. This converter uses Matlab application to processing. Image type is converted from jpg/jpeg extension to pgm extension. After conversion, Matlab makes a rectangle label on top of the images. This labeling uses a toolbox Training Image Labeler from Matlab. This process aims to create a focus area of the training image. This labeling process is shown in Figure 2.

![Figure 2. Image labeling process.](image2)
Training image step applies Haar Cascade method to train images. The result is struct-data in XML format. This classification needs 10 steps for processing.

2.3 Implementation training’s file on a browser application
Implementation training’s file on browser application needs several compatible files. File XML from training step still did not support browser application yet. So that, a tool required to convert XML files to JSON files. This research uses Convert-tjs [10] to convert XML training and ready to implement with JavaScript Framework.

These JSON files are implemented to browser application using programming language HTML+javascript. The result of this implementation is the applicationable to recognize vehicles on the rectangle box. This detection was shown in Figure 3.

![Detection result on the browser.](image)

Figure 3. Detection result on the browser.

3. Main Results
In each testing method, there are any several pre-processing tests like: prepared 20 test image, arranged XML training and develop source code. Each testing method will describe at the following paragraph.

3.1 Testing on Matlab application
The source code is developed using Matlab to recognize car object. This source code is shown in Figure 4.
The result of this test indicates a rectangle box around the object. This rectangle appears because the object has several similarity features with training data. Rectangle box will be displayed if the object has stable features. Length of the minimum object to be the strong characteristics is 100 pixels. The result of this program able recognizes 17 of 20 testing image or 85% successful percentage.

### 3.2 Testing on a browser application

The source code is developed on browser application using Javascript Framework Library (Trackingjs). This source code is shown in Figure 5.

```javascript
tracker.on('track', function(event) {
    var biggest_width = 0
    s_x = 0;
    s_y = 0;
    s_w = 0;
    s_h = 0;
    event.data.forEach(function(rect) {
        if(biggest_width < rect.width){
            s_x = rect.x;
            s_y = rect.y;
            s_w = rect.width;
            s_h = rect.height;
            biggest_width = rect.width;
        }
    });
    if(s_x > 0){
        window.plot(s_x, s_y, s_w, s_h);
    }
});
window.plot = function(x, y, w, h) {
    var rect = document.createElement('div');
    document.querySelector('.demo-container').appendChild(rect);
    rect.classList.add('rect');
    rect.style.width = w + 'px';
    rect.style.height = h + 'px';
    rect.style.left = (img.offsetLeft + x) + 'px';
    rect.style.top = (img.offsetTop + y) + 'px';
};
```

**Figure 4.** Testing code using Matlab.

**Figure 5.** Browser testing code.
Testing process used programming language HTML+javascript on browser had been done. Detection box could detect object aligned with JSON training. The result of this testing is shown that the applicationable recognizes 19 from 20 image testing or 95% accuracy percentage.

This program starts with Play Button on the browser screen. This program can play video only an MP4 format file. This file video is obtained from the internet. This program able to recognize vehicle picture while the video is playing. This program is shown in Figure 6.

![Figure 6. Recognize from video](image)

This research considers the consumption of resources especially CPU and memory usage. The expected result of this research is the use of minimum resources. In this research, researcher use 2.53 GHz CPU clock, 4 Giga Bytes Memory and 1 Gigabyte swap. CPU and memory usage is shown in Figure 7 and Figure 8.

![Figure 7. The CPU usage of the browser approach.](image)
CPU usage in the browser approach application is 19.8%. This result shows that the browser approach uses low consumption processing.

Figure 8. Memory usage of the browser approach.

Figure 8 shows that free memory in the browser approach is 5154 Mega Byte. This result indicates that the browser approach uses low memory consumption.

3.3 Comparison of browser approach with desktop application

An existing python application is compared resource consumption with the browser approach. This program is developed using python 2.7 and Open CV library. Open CV is an image processing library which able to use only in desktop application. This program is shown in Figure 9.

Figure 9. Desktop application using python programming.

CPU and memory usage of the desktop application is shown in Figure 10 and Figure 11.
Figure 10. Usage python program.

Figure 10 shows that desktop application using python programming use the high resource. The CPU usage is 69.19% of computer CPU.

Figure 11. Memory usage of python program.

Figure 11 shows that free memory in the python program is 4907 Mega Byte.

4. Conclusions
A light car detection application can be obtained using a browser approach. A browser implements Javascript Library Framework (Trackingjs) as image processing. This research used two testing method such as testing using Matlab and testing on a browser. Result using Matlab application has identified 17 from 20 car image (85%). Test using browser application has identified 19 from 20 car image (95%). In the testing step, there are many different accuracy results because of the platform issue.

The comparison of CPU and memory usage in browsers is lower than desktop applications. The desktop application gives free memory 4909 MegaByte, and CPU usage is 69.19%. Meanwhile, the browser approach gives free memory 5154 Megabyte, and CPU usage is 19.8%. The proposed approach can solve the government's problem to provide traffic information with low processing.

In future work, integrated with another completed training from another research will increase accuracy. Different accuracy in this research is caused by the platform issue also will be explained.
Acknowledgment

The authors would say thanks to DRPM of Ministry of Research, Technology and Higher Education (RISTEKDIKTI) of the Republic of Indonesia for funding this research with Research Grant number SP DIPA-042.06.1.401516/2018.

References

[1] E. Lundgren tracking.js Online? : https://trackingjs.com
[2] P. Srinivas and Y. L. Malathilatha Image Processing Edge Detection Technique used for Traffic Control Problem
[3] M. B. Subaweh and E. P. Wibowo 2016 Implementation of pixel based adaptive segmenter method for tracking and counting vehicles in visual surveillance International Conference on Informatics and Computing (ICIC) pp 1–5
[4] M. Z. Irawan, T. Sumi and A. Munawar 2010 Implementation of the 1997 Indonesian highway capacity manual (MKJI) volume delay function Journal of the Eastern Asia Society for Transportation Studies pp 350–360
[5] S. Taghvaeeyan and R. Rajamani 2014 Portable roadside sensors for vehicle counting, classification, and speed measurement IEEE Transactions on Intelligent Transportation Systems pp 73–83
[6] R. Zhang, Z. Li, C. Feng and S. Jiang 2012 Traffic routing guidance algorithm based on backpressure with a trade-off between user satisfaction and traffic loading IEEE Vehicular Technology Conference (VTC Fall) pp 1–5
[7] C. Pornpanomchai, T. Liamsanguan and V. Vannakosit 2008 Vehicle detection and counting from a video frame International Conference on Wavelet Analysis and Pattern Recognition pp 356–361
[8] S. Choudhury, S. P. Chattopadhyay and T. K. Hazra 2017 Vehicle detection and counting using haar feature-based classifier 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON) pp 106–109
[9] A. Arunmozhi and J. Park 2018 Comparison of HOG, LBP and Haar-Like features for on-road vehicle detection IEEE International Conference on Electro/Information Technology (EIT) pp 0362–0367.
[10] C. S. Costa 2018 Converts old/new type of Open CV Haar Cascade xml data to trackingJs online? : https://github.com/cirocosta/gulp-converter-tjs