Image Uploading for Safe Driving Applications in Vehicular Networks Based on Mobile Edge Computing Technologies

Ming-Fong Tsai[1], Chia-Yuen Lin[2]

[1] Department of Electronic Engineering, National United University, Taiwan
[2] Department Information Engineering and Computer Science, Feng Chia University, Taiwan
mingfongtsai@gmail.com, lo6615@yahoo.com.tw

Abstract

This paper proposes an image uploading system for vehicle safety monitoring applications based on mobile edge computing technologies. In this system, numerous camera nodes are placed within vehicles to collect vehicle journey data, and when emergency events such as accidents are detected, a camera with a view in the direction of the accident can upload more data than those in other directions. In addition, other vehicles within the area of the accident will also upload image data about the event, meaning that a greater amount of useful data on the accident will be collected for analysis and assistance to drivers. We propose to use mobile edge computing because of its many advantages, such as fast service response, low end-to-end delay and suitability for large data storage systems. We also propose solutions to minimize the uploading of redundant, repetitive and similar image data, a problem that has been encountered in previous systems. The results of both experiments and simulations showed that our system achieves better performance than those proposed in related studies.

Keywords: Vehicle safety monitoring, Mobile edge computing, Image uploading

1 Introduction

Traffic accidents occur at a high rate, and there is a lack of effective systems to store evidence of these accidents and to provide assistance for timely treatment for the victims. Motivated by this idea, we propose an image uploading system for vehicle safety monitoring applications. In traditional vehicular networks, data recorded during driving are normally stored in the memory card of the driving recorder. However, storing data in memory card has several disadvantages: First, the memory card of the driving recorder has the limits in storage capacity when the driving journey is longer than expected. Recorded data are often replaced by data recorded later, resulting in incomplete preservation of driving data and even worse, the failure to restore the car accident scene; second, the use of offline data storage makes users more passive in detecting possible risks during driving and factors that affect their driving safety. At present, the solution to the above-mentioned problems is to increase the memory capacity in the driving recorder and the mandatory storage combined with non-deletable mechanisms after detecting abnormal signals from moving vehicles so that the driver can be secure in retaining the evidence for accidents. However, this mechanism only solves part of the problem, but it still cannot solve the problem of offline data storage. It will make it hard for drivers to adopt driving data to protect themselves when accidents occur. Thanks to the development of Internet of Things technologies, driving data now can not only be stored on memory cards, but it can also be stored on servers for backing up driving data for later use. In large systems, mobile edge computing technology is now often used to replace traditional servers for the storage and deployment of services, due to its various advantages such as low end-to-end delay, fast response times, security and flexibility. To upload the driving information in real time directly to a server through internet connection, there are many solutions including satellite communication systems, mobile communication systems (3G/4G), vehicle communication networks, Vehicle-to-Base Station communications [1-5]. Once uploaded, the data will be analyzed and classified to at user's service so that users can access to these data anytime and anywhere they want via their smart devices [6-8]. This online solution makes the access of driving data easier and more to users. Uploading driving data to servers is crucial. However, vehicles move constantly, so their channel conditions, that is, the available bandwidth also changes over time. Therefore, transmitting data in good quality and in real time to cloud servers confront challenges.

This paper proposes a vehicle image uploading system with cameras installed around the vehicle. These cameras will collect continuous images along the driving route and will upload these images to a mobile edge computing layer via 4G mobile networks.

*Corresponding Author: Ming-Fong Tsai; E-mail: mingfongtsai@gmail.com
DOI: 10.3966/160792642020122107005

A base station is constructed as the edge node in the mobile edge network. This is feasible with a software-defined networking solution. Vehicles will upload and retrieve driving journey data within range of these edge nodes. In addition, an important issue in an image uploading system is to find effective solutions to minimize the uploading of redundant, repetitive and similar image data, an issue that has been identified in previous systems [9-10]. The image uploading systems in the vehicle network also often encounter the problem that the bandwidth of the transmission channel is constantly changing. In this study, we therefore focus on analyzing the relationship between the available channel bandwidth, the vehicle speed and the shooting rate of the camera nodes. A further important factor affecting safety driving is danger signals, which means signals that will be triggered while making turns. In a situation such as this, the device captures more images on this side than on the other sides. We also describe the selection of parameters and a method that can predict the available channel bandwidth to adjust the number of images taken by each camera to provide a high quality of service and to ensure the performance of the driving safety awareness information system. This study considered that the memory is limited in capacity that it can only memorize part of moving process and might not be sufficient for preserving evidence. When there are major accidents, cars and driving recorders might both be severely damaged, this system can still obtain all related information from cloud server.

The remainder of this paper is organized as follows: Section 2 presents some related works. Section 3 describes the architecture and components of the proposed system. Section 4 presents our adaptive control algorithms. Section 5 provides analysis of the system and its performance parameters. Section VI reports the experimental results of the system and presents a comparison of the proposed system with other related systems. The final section offers our conclusions.

2 Related Works

In view of the development of mobile communication networks and the increase in speed and bandwidth of mobile network, there are a variety of papers researching on proposing using mobile communication network to exchange data on driving safety awareness information system. However, when there are too many users share the mobile communication network at the same time, the available bandwidth allocated for each user will be limited. That is, each user's throughput will decrease as the number of users increase. The more users use the mobile communications network at the same time, the more end-to-end delay will occur for users to transmit data [11]. In addition, we also evaluate the impact of speed of moving vehicles on end-to-end delay of the user's data transmission. The experimental results show that the user's throughput is reduced by only the number of users. Since the number of mobile communication network users is not a controllable factor, avoiding unnecessary data uploading will be crucial. Therefore, we explore the development of adaptive methods for uploading driving data to reduce the waste of the network available bandwidth.

The most typical application of transmission of driving data over mobile communication networks is uploading driving data to cloud servers. The authors in [12] proposed to dynamically adjust images uploaded to cloud servers according to the vehicle movement speed. Its main contribution is that it proposed the minimum number of images required for building the streetscape when vehicles move. It refers to vehicle speed to determine single image shot and adaptively control the number of uploaded images via image overlay technology. To restore continuous streetscape images, the adaptive image uploading is adjusted according to vehicle speed to obtain the minimum number of images required for restoring the continuous streetscape image. If the vehicle moves slowly or remains still, the image of the same block will be over-photographed and the device will store a large number of repetitive images. On the contrary, if the vehicle moves at a high speed, it will cause continuous overlaying of images, that is, continuous gaps between the moving streetscape images. In view of the fact that the available bandwidth of the mobile communication network is insufficient, even when the number of images to be uploaded is controlled in accordance with the vehicle speed, it is urgent to propose a mechanism which can adjust the quality of uploaded images adapting to the moving of vehicles.

Previous studies [9-10] proposed that a dynamic adjustment mechanism on the number of uploaded images and the quality of uploaded videos according to vehicle speed and available bandwidth. Today, driving safety sensing components such as video cameras and radar devices have gradually become the standard equipment for automobiles. Authors in [13] proposed neural-like algorithms to implement a real-time detection of surrounding vehicles and judgment of relative distances. As shown in related work [14], multi-image stitching is used to form vehicle peripheral images for drivers. Authors in [15] proposed a method to analyze the vibration information through the built-in triaxial sensing element of the smart handheld device. Check the degree of jolt of the road when the vehicle is driving. Therefore, this paper discusses information such as vehicle speed, network available bandwidth and driving safety perception to optimize the adaptability and dynamically adjust the number and quality of the images before uploading them to cloud server.

3 The Proposed System

Figure 1 shows the architecture of our proposed system, including cloud server, cam node, and information system. The simulated on-board information system is mainly utilized to simulate the danger signal perception on the vehicle. Since the vehicle used in this study does not contain equipment that can obtain driving safety awareness information, it is possible to simulate the condition of the vehicle by the simulator. We also simulate the on-board system and real-time switching of the vehicle condition through road test to achieve the same effect as directly obtained danger signals from the on-board system. The cloud server primarily receives driving data from cam nodes and stores these data. The data are uploaded via 4G networks and history information of the data images are managed by cloud database management. Users can access to this data through RESTful Web services. The cam node is mainly composed of a Raspberry Pi and a webcam. Raspberry Pi and the webcam can implement a network to receive on-board signals and capture streetscapes at the same time.

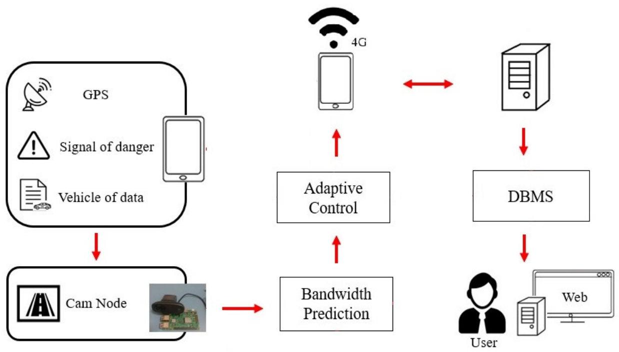


Figure 1. System architecture

3.1 Cloud Server

The cloud server uses Ubuntu Linux as an operating system to construct the full cloud-based driving safety awareness information system. The mobile edge computing server is divided into three parts. The first is the cloud database, which uses MySQL as the database system, while the library management system uses the cloud API to upload and organize vehicle information. In addition, it also provides logs with the amount of images successfully uploaded of received data for Cam Node to use. The second part is the cloud file system, which uses a server-side file management system to classify image files. The third is the cloud platform, which is mainly composed of three technologies: (i) a web server called Tomcat that provides Java Server Page and Servlet support, and includes a management and control platform to manage projects safely; (ii) HTML/JS, which is used to construct the entire cloud webpage platform, and which provides a web UI interface to search for and browse vehicle information uploaded in the past; (iii) a socket server, which is used to control the network connection of the camera node, and to provide a window for simulating the on-board information system and for uploading traffic information and images from the camera node. Drivers can access the cloud server database to view their driving data or other statistical analysis data.

3.2 Cam Node

It's a Raspberry Pi 3 hardware module running Ubuntu Linux operating system. Raspberry Pi is a very popular hardware control module suitable for today's Internet of Things applications. The Raspberry Pi 3 contains its own network Bluetooth communication module and a small easy-to-carry rack mounted on the vehicle. In the deployment of the system, we used six Raspberry Pi modules as network nodes, and each is connected to a camera webcam to control the process of images collecting along the driving route as well as transfer images to the Fog server. The cam node is further divided into three parts. The first part is the image acquisition. Images taken from the webcam are processed through OpenCV image pre-processing, and then the second part is used to classify and store the images taken by the camera. The third part is a Socket Client writing in Python programming language. This client is responsible for exchanging data with the cloud server through the 4G mobile network and the TCP/IP communications protocol.

3.3 Driving Safety Awareness Information System

In order to implement the driving safety awareness information system and strengthen retention of driving data in cloud server, this study uses Android smart devices as repeater devices. In addition to providing global satellite positioning information, the cloud storage platform can record the movement trajectory. Vehicle speed is estimated via GPS signals. Events of vehicle making turns are also indicators. When there are emergencies, the camera on that side will be told to increase the shooting rate and the quality of captured images than cameras on other sides. This helps drivers to spot and handle risks earlier.

4 Adaptive Control and Algorithms

4.1 Adaptive Parameters

In fact, the process of uploading driving data in mobile network is affected by the number of users in the same base station. This is an important issue. To ensure the quality of the recovery image on the database server, we proposed methods to automatically adjust the number of uploaded images according to network available bandwidth. Our proposal optimized the number of uploaded images and quality parameters by using three factors including vehicle movement speed, driving safety awareness and network available bandwidth.

4.1.1 Vehicle Speed

The driving data is stored and used to recover the accident scenes. In order to complete continuous image streaming on moving vehicles, the minimum number of shots required per second of cameras installed on vehicles must be automatically adjusted according to current vehicle speed. In our study, for a vehicle moving at a speed of 90 kilometers per hour, that is 25 meters per second, if the shooting distance from the camera to the road surface was 3 m, it requires a shooting rate of at least 9 images per second to ensure continuous streaming of images. Complete driving data can thus be achieved. If the vehicle speed is less than 30 kilometers per hour and its moving path per second is about 9 m, only 3 images per second are needed to achieve complete image retention. If the shooting rate exceeds three images per second at speed of 30 km/h, it will cause the cloud server to store a large number of repetitive images. The relationship between the minimum numbers of shots required for different vehicle movement speeds is shown in Table 1. In our system, six cameras are placed around the vehicle to collect images that cover the entire space surrounding vehicle. Therefore, in order to preserve the driving data of car accident, six shots are required to be installed on the vehicle.

Table 1. Relationship between the minimum numbers of shots required for various vehicle speeds (Taking 3m Shooting Range as Example)

Car Speed		Minimum number of shots per second	Total of shots per second (distance = 3 m)
< 30 km/h	< 8.33 m	3	18
30 km/h	8.33 m	3	18
45 km/h	12.5 m	5	30
60 km/h	16.66 m	6	36
75 km/h	20.83 m	7	42
90 km/h	25 m	9	54
> 90 km/h	> 25 m	10	60

In addition to increasing the quality of shooting images, we can increase the resolution and shooting ranges of images to preserve better driving data as the evidence of car accidents. In Table 1, the images are captured at distance of 3 m and image quality of 240p. If we use quality image of 480p, we will achieve a wider image that covers the street. The relationship between the minimum numbers of required shooting images for different image quality is shown in Table 2. As the image quality is higher, the network traffic for uploading videos will also increase, and a set of considerations must be devised to optimize the three-parameter algorithm of vehicle movement speed, driving safety perception and network available bandwidth.

Table 2. Relationship between the minimum numbers of shots required at various vehicle speeds and various shooting ranges

Car Speed		Minimum number of shots per second (distance = 3 m)	Minimum number of shots per second (distance = 3.5 m)	Minimum number of shots per second (distance = 4 <m)
< 30 km/h	< 8.33 m	3 (240p)	3 (480p)	3 (720p)
30 km/h	8.33 m	3 (240p)	3 (480p)	3 (720p)
45 km/h	12.5 m	5 (240p)	4 (480p)	4 (720p)
60 km/h	16.66 m	6 (240p)	5 (480p)	5 (720p)
75 km/h	20.83 m	7 (240p)	6 (480p)	6 (720p)
90 km/h	25 m	9 (240p)	8 (480p)	7 (720p)
> 90 km/h	> 25 m	10 (240p)	9 (480p)	8 (720p)

4.1.2 Driving Safety Awareness Information System

When the driver needs to turn left or right while driving on the road, he will give a left-right turning signal to remind the nearby moving vehicles. In this situation, left- and right-rear sides belong to dangerous sides. Therefore, in order to have images which are continuous of good quality in dangers sides, it is very important to increase the number and quality of uploaded images in these areas where danger may more easily to occur. This dangerous area is depicted in Figure 2. At present, many safety driving systems have been developed and commercialized, for example, lane departure warning systems, rear-end collision warning systems and traffic flow information system. The above-mentioned driving safety awareness information is for specific side of the driver's vehicle. Due to the possibility of danger occurrence, the number and the quality of uploaded images from specific sides need to be dynamically adjusted in order to preserve this driving data for later evidence. The design concept of danger signals is it exchanges information from OBD-II message from its own car or Internet of Vehicle. This study takes simulation signals to verify the applicability of the technology and tests experimental data.

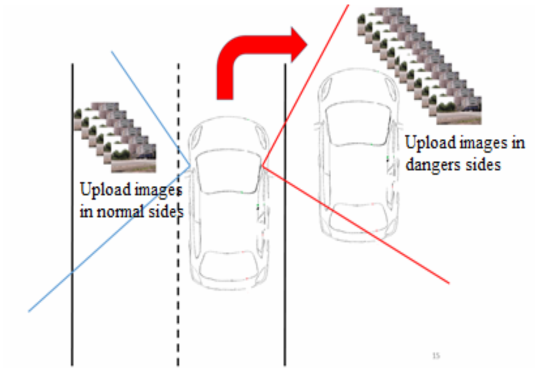


Figure 2. Illustrate danger areas when the driver turns left

4.1.3 Available Bandwidth

The channel's available bandwidth is always changing as the vehicle passes through different areas of the mobile network. In view of the fact that the number of users in mobile communication network is not controllable, and the available bandwidth of the network changes over time, problems emerge will be how to match the number and required quality of uploaded images at the above-mentioned vehicle speed and how to gain driving danger perception. The bandwidth is dynamically adjusted and utilized in real-time to avoid unnecessary repetitive data transmission to occupy network available bandwidth.

4.1.4 Predict the Available Bandwidth

The proposed image uploading mechanism adopts TCP protocol which uses sliding window method to control traffic and congestion. Available Bandwidth Estimation utilizes sending end with uploaded images in specific unit of time (for example per second). Because TCP protocol has resending mechanism to ensure image reception in receiving end, it can estimate available bandwidth through correctly received uploaded images in specific unit of time in sending end. Available Bandwidth Estimation in this study is based on TCP protocol because it has been widely applied in mobile network. In our study, we aim at achieving the best image quality for safety driving applications. We propose an algorithm that can adjust the number of images sent to the server to be greater than or equal to available bandwidth of the channel. This algorithm will adjust the number of uploaded images to help the system accommodate the change in bandwidth. Our algorithm was described in Figure 3. For example, in initial process, each camera uploads one image to the cloud server, so there will be six images uploaded in first transmission. On the cloud server (receiver), when all six images are successfully transmitted in the first transmission, the system will attempt to transmit a larger number of images in the next transmission. The server will send back the transmission log including the number of images (1) successfully uploaded to cam nodes in order to predict and adjust both shooting rate and bandwidth level. Therefore, in the second transmission, the total number of images sent to the server will be 12 images. Assume in the second transmission, only 11 images are successfully received in the first attempt, the available bandwidth of the transmission channel is not enough compared to the amount of data uploaded. The server will still send back the transmission log. However, because this difference is not significant, (2) the cam node will maintain the same both shooting rate and bandwidth level as previous transmission. In third transmission, assume all 12 images are successfully uploaded at the first try, the server will send back the transmission log (3) to cam nodes, and then the cam nodes will increase both their shooting rate and bandwidth level. Therefore, the total number of images uploaded in the next transmission will be 18 and our algorithm will repeat this cycle for the following transmission.

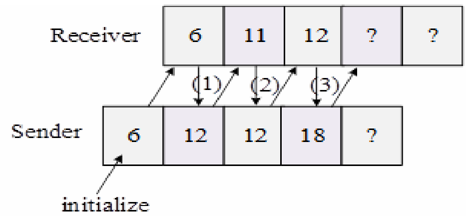


Figure 3. Adjust the number of uploaded images with available bandwidth

4.2 Flow Chart and Adaptive Control Algorithms

Figure 4 is a systematic flowchart of the proposed method. In the beginning, the system checks the connection between cam nodes. Once successfully connected to nodes, the cam nodes begin to collect data and transmit the data to the cloud server. When receiving data from cloud server, cam nodes will begin to predict bandwidth and provide adaptive control. The prediction of available bandwidth and provided adaptive control are based on method described above. Here we further take danger signals into consideration. The system will check whether the data from the vehicle contain danger signals. When not detecting danger signals, cam nodes will adaptively control the data by the algorithm described above and determine the number of images collected at each camera by included factors such as vehicle speed and image quality. When detecting danger signals, cam nodes will proceed to adaptive control with the driving safety awareness information system. This means that there will be a cam node on the side with danger signals receiving messages to gain shooting rate (larger than cams on other sides). The gained shooting rate will vary with current vehicle speed, and it will be presented in detail in the following sections. The comparisons between the proposed method and related work are shown below.

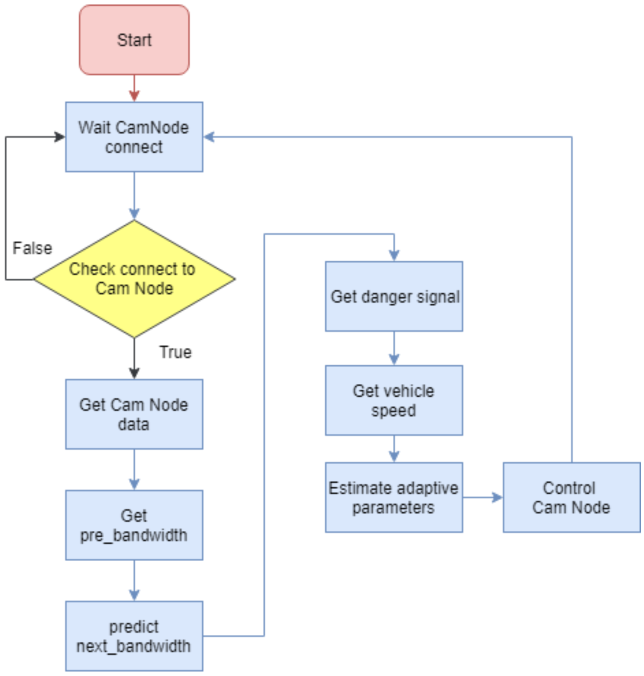


Figure 4. Algorithm flowchart

4.2.1 Vehicle Moves at Speed of 30 Kilometers Per Hour without Driving Safety Awareness Information

The selection of parameters is shown in Table 3. Assume that a vehicle is moving at 30 km/h and the available network bandwidth is 3.6 Mbps. The data size under image quality of 240p, 480p and 720p images is 36kB, 120kB and 360kB, respectively. The selection of number of shots parameters corresponding to this speed is 3.

Table 3. Environmental assumptions

	Number of shots	Picture Quality 240 p	480 p	720 p
Bandwidth: 3.6 Mbps	3	108 kB	360 kB	1080 kB
240p image: 36 kB	5	180 kB	600 kB	1800 kB
480p image: 120 kB	6	216 kB	720 kB	2160 kB
720p image: 360 kB	7	252 kB	840 kB	2520 kB
	8	288 kB	960 kB	2880 kB
Danger signal condition	9	324 kB	1080 kB	3240 kB
requires the shotting rate at least 15	10	360 kB	1200 kB	
images/second	15	540 kB	1800 kB	
	20	720 kB	2400 kB	
	25	900 kB	3000 kB	
	30	1080 kB		

4.2.2 Vehicles Moves at a Speed of 30 Kilometers Per Hour with Driving Safety Awareness Information

In this section, we examined the operation of adaptive control algorithm in the presence of driving safety awareness information. The minimum number of shots required in the case of driving safety awareness is at least 15. This requirement is only for one cam node, which in the direction of the warning signal occurs. Because available bandwidth of the network is 3.6 Mbps, selecting cases in 15 shots is the optimal solution. For good image quality, the image quality of selected by the system will be 480p.

4.2.3 Vehicles Moves at a Speed of 60 Kilometers Per Hour with Driving Safety Awareness Information

If the speed of the vehicle is increased to 60 kilometers per hour, and there are two cam nodes to capture images and upload them to the server. One cam node needs to increase the number and the quality of uploaded images as driving safety awareness information when the vehicle moves. The other cam node will select parameters depending on the cam node in the direction of the danger signal and the available channel bandwidth. Finally, the final result needs to be determined based on the available bandwidth of the network at 3.6Mbps. Therefore, the final optimization of adaptive control the number and the quality of uploaded images will be selected as six 480p images in size of 720kB and 15 480p images in size of 1800kB.

5 Implementation and Experimental Results

5.1 Data Upload Module

While the system is in operation, the Data Upload Module will continuously mount to the server waiting for connecting to socket client. Once the module attempts to connect to sever, it will first request user identification from socket client. After the verification is passed, the module will wait for a JSON type data packet in order to know content of the following packets is driving images or information before next packet is sent. Content in socket client package is designed as JSON type for transmission except for the driving images socket. Driving images will be transformed into image file from byte type and saved in fog sever after the server receives them. By getting Cam Node information from cam node key, the data upload module is capable to identify and save the entire driving information from on-board system. After the packets are broken, the driving information will be sent to Database Management System for further usage. The system is responsible to deal with the needs from different module to database. The Data Upload Module mainly requests for user identification from on-board information system and saves the uploaded data into database. The other request is in charge of Bandwidth recording and reading. The bandwidth will be saved into database while the server receives data, and the data will be send back to on-board information system for prediction of next bandwidth.

5.2 On-Board Information System

The on-board information system is divided in two parts. Part one is Cam Node with socket client interface and function of taking frames written in Python and running on Raspberry Pi is one, and part two is the Simulated On-Board Information System written in Java and running on the following version of Android 5.0 in order to deal with the circumstance when driving information is unreachable on some kinds of vehicles. Simulated on-board information system transmits driving information data to cam nodes through sockets, and thus cam nodes are able to control the number of photos should be taken per second in current driving condition and packet data for uploading. When the system boots, it continuously takes frame and control the number of frames to be taken. Figure 5 is the flowchart of on-board information system. At first, the system will garb the vehicle speed and danger signals from the simulated on-board information system via GPS module and the button implemented on it. As these two driving information is sent to cam nodes, the system will proceed to check if there are temporary driving images and information inside the local MySQL database. Then, it will check the connection status between socket servers. The driving data will be temporarily stored in the local MySQL database if the system fails to connect to the server, and these data will be uploaded to the server once the system reconnects to the server and be deleted from the local MySQL database. If the connection between system and server is normal, the system will check the records of successfully uploaded images downloaded from the server, and then use the history records to predict and adjust how many frames is adequate to be sent currently. Finally, the predicted results will be sent to Cam Node for adaptively controlling the number of images to upload.

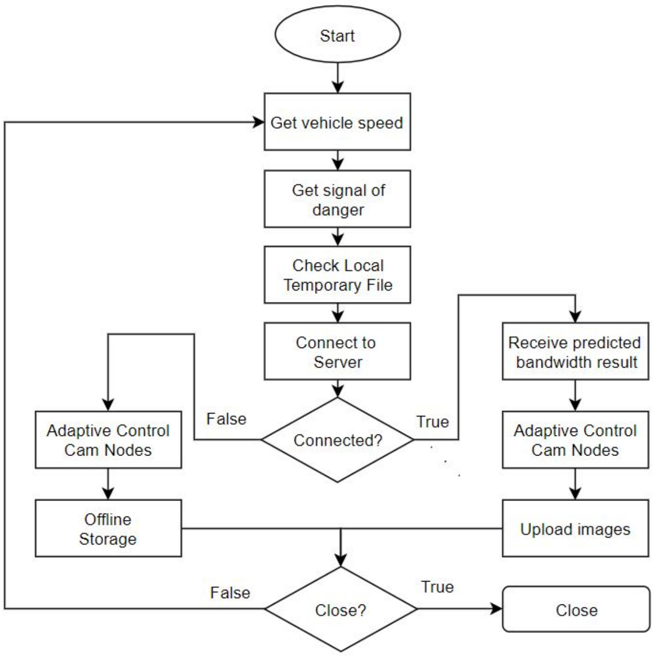


Figure 5. On-board information system flowchart

5.3 Adaptive Calculate Module

Figure 6 is the flowchart of Adaptive Calculate Module. As shown in the figure, when the module is initialized, the module will grab the vehicle speed, danger signals and the condition of local temporary data at first and obtain data from Bandwidth Prediction Module to proceed adaptive calculation. Because of the data from Bandwidth Prediction Module includes the prediction results of available bandwidth and the information used for prediction from Cam Nodes, it is necessary to provide parameters which control the amount of frames under conditions of various vehicle speed, danger signals and local status while adaptive control parameters are initialized. When there are danger signals detected in any cam nodes, adaptive control parameters will be set to 10 frames per second which meets the maximum needs. On the contrary, adaptive control parameters of those cam nodes without containing any danger signals will be set by the calculation result of three parameters (available bandwidth, danger signals and vehicle speed) grabbed from the initiation process. And these adaptive control parameters are calculated during estimate adaptive control process. After the procedure of adaptive control is initialized and after it estimates adaptive control parameters, it will then check whether the summary of adaptive control parameters of all cam nodes exceed the predicted value from Bandwidth Prediction Module. When the summary of adaptive control parameters of all cam nodes is lower than the predicted bandwidth, these parameters will be sent to On-Board Information System for adaptive uploading and pictures taking control. On the contrary, when the summary of adaptive control parameters of all cam nodes is higher than the predicted bandwidth, the module will decrease the number of pictures taken by cam nodes without containing danger signals so that those containing danger signals can preserve more pictures and upload to the database.

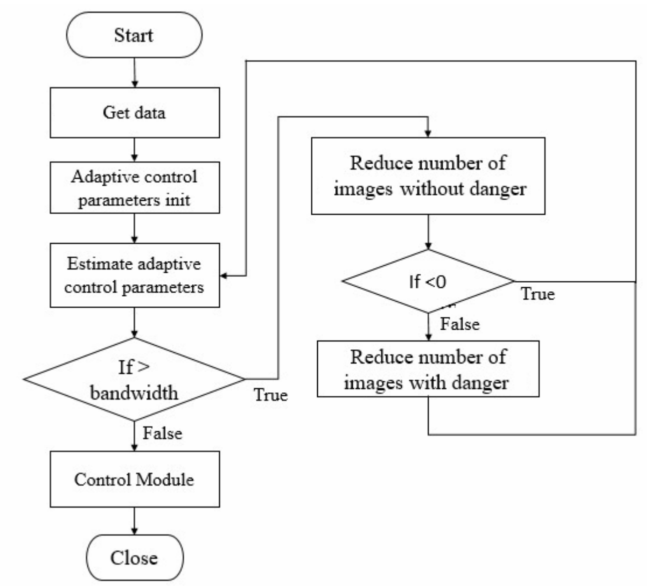


Figure 6. Adaptive calculate module flowchart

If adaptive control parameters of the cam nodes without danger signals are already decreased to 3 frames per second, the module will start to adjust adaptive control parameters of the cam node with danger signals.

5.4 Bandwidth Prediction Module

Figure 7 is the flowchart of bandwidth prediction module. In the beginning, the module will garb the number of cam nodes from Data Upload Module, transmission logs from server and the bandwidth level of the last transmission saved in local database in order to estimate bandwidth next time. The predicted bandwidth in our proposed method is the quotient of the amount of cam node and bandwidth level. If the last prediction result is same as the transmission log, the module will adjust the level higher for the following transmission, and vice versa. The predicted results provide Adaptive Calculate Module for further usage.

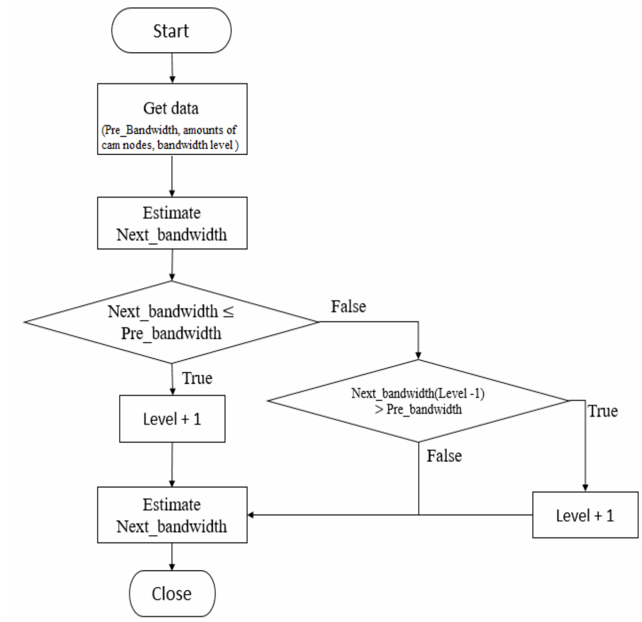


Figure 7. Bandwidth prediction module flowchart

5.5 Experimental Environment

The experimental environment chosen in this study is a route from Feng Chia University to National United University Ba Jia Campus as the left figure in Figure 9 involving 48.1 kilometers in one trip. The experiments were conducted for 6 times on the same route containing highway and general road taken 2 hours for each experiment. The smart mobile device used in the experiment is Sony Xperia Z5 which provides Wi-Fi connection for cam nodes to transmit data and was taken as the simulated on-board information system. There were six cam nodes and each of them were composed of a raspberry pi 3 model B and Logitech C920R webcam implemented on the roof of vehicle. Every cam node took pictures of 60 degrees in order to achieve round view recording as shown in the right figure in Figure 8.

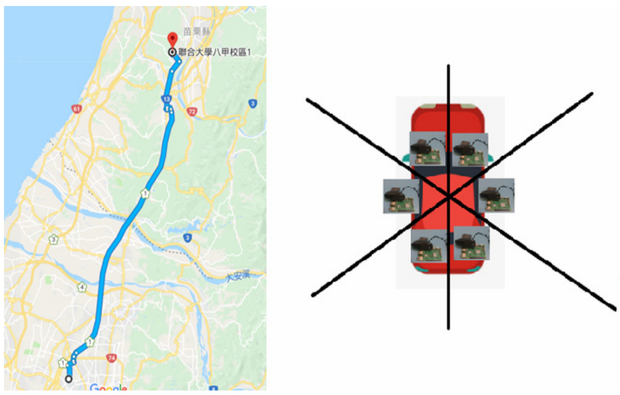
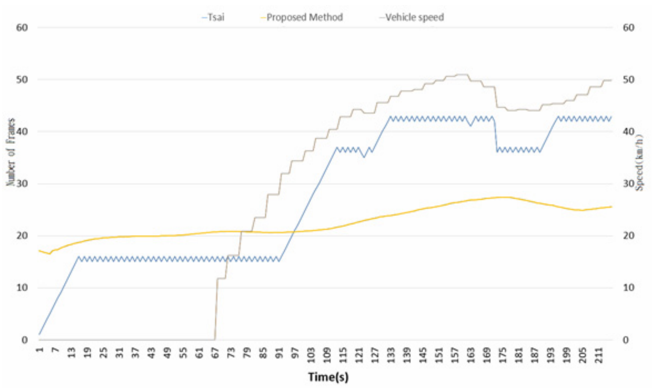


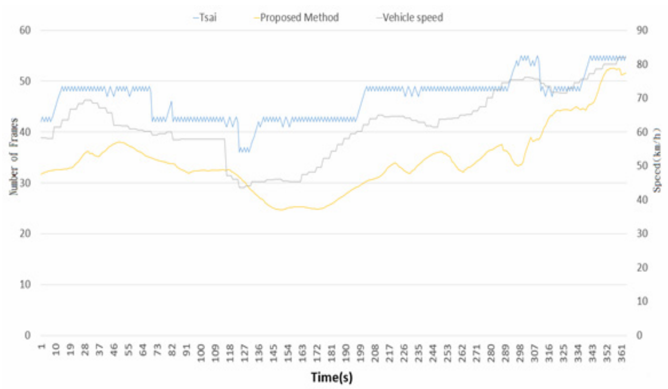
Figure 8. Testing route (From Feng-Chia University to National Untied University Ba-Jia Campus) and schematic diagram of cam node implementation

5.6 Dynamically Adjusts the Number and Quality of Uploaded Images Based on the Proposed Methods

The number of uploaded images is dynamically adjusted to adapt to vehicle speed. In this implementation, we use cam nodes and placed on six corners of the vehicle to capture images. We then proceeded to compare the proposed method with other related methods under the same deployment scenario. In Figure 9, we compared the proposed adaptive method with Tsai's method [10] which without considering driving safety awareness information system. The values of vehicle speed curve are shown in right vertical axis, while values on curves of the proposed method and Tsai method are shown in left vertical axis. Our proposed algorithm has less uploaded image than Tsai's method under the same conditions. According to Figure 9 (a), the system will make sure to upload the minimal images when vehicle speed is low even it stops. As Figure 9 (b) shows, the amount of the uploaded images changes with current vehicle speed. The actual tested speed is shown in grey curve (coordinate value on the right side). And the jitter of the chart is caused by unstable bandwidth of mobile network. The bandwidth usage of our proposed method decrease by approximately 20% compared to Tsai's. Due to the prediction of available network bandwidth in Tsai's method takes average of the summary value in previous 5 seconds, the accuracy of bandwidth prediction is not high under the circumstance in which the available network bandwidth varies in a wide range. This shows that our method is the most efficient method to save bandwidth yet maintain the same quality of service comparing with other related methods.



(a) Low speed



(b) High speed

Figure 9. Comparison the total uploaded image/s between Tsai's method and the proposed method

In Figure 10, we evaluated the proposed method on driving safety awareness information system. The values of vehicle speed curve are shown in right vertical axis, while values on curves of the proposed method and Tsai method are shown in left vertical axis. We conducted a comparison of the proposed method with methods in related studies at high risk driving condition such as making turns. The proposed method not only adjusts the amount of uploaded frames adaptively by vehicle speed on the basis of Table 1 but also increases amount of the uploaded images in order to preserve more detailed driving images while detecting danger signals. However, Tsai's method can only adjust the number of uploaded frames by vehicle speed. As a result, the proposed method is more capable to conserve significant driving images. There are some situations such as driving through the tunnel or some area with poor signal which make the network unreachable. In the proposed method, there is a resending mechanism to deal with this situation. The system is able to save images temporarily in the Raspberry Pi while the internet connection is not available. It then will resend the temporary data to Fog server as the system reconnects to the internet. In Figure 11, it shows the operation of resending mechanism in the proposed method. The values of vehicle speed curve are shown in right vertical axis, while values on curves of uploaded amounts and local temporary file are shown in left vertical axis. In the boxed area, the chart of temporary file rises while the internet connection is unreachable and vice versa. In order to verify the dynamically adjusted shooting and uploading in dangerous circumstance in the proposed method, a road test was conducted. It is obvious that the cam node image of boxed area is clearer than others in the figure below. To verify the proposed method, making turns will be considered as dangerous condition by our algorithm. As shown in Figure 12, only cam in the red marked place uploads more pictures when the vehicle turns right.

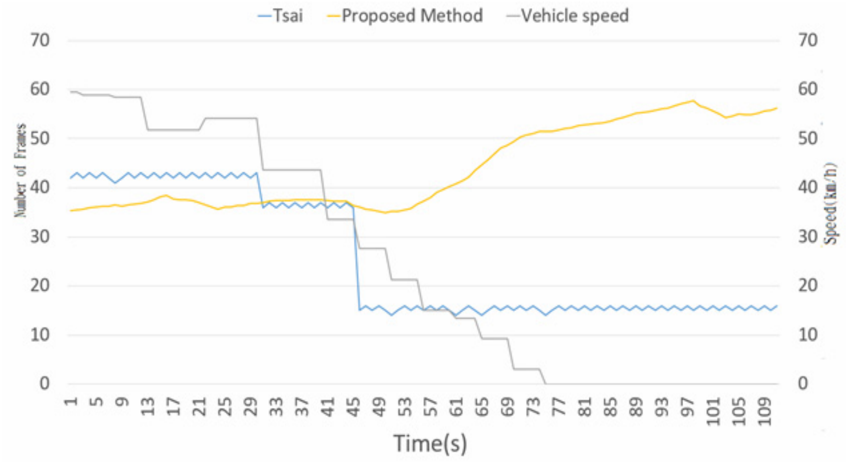


Figure 10. Comparison between Tsai's method and the proposed method in driving safety awereness information system

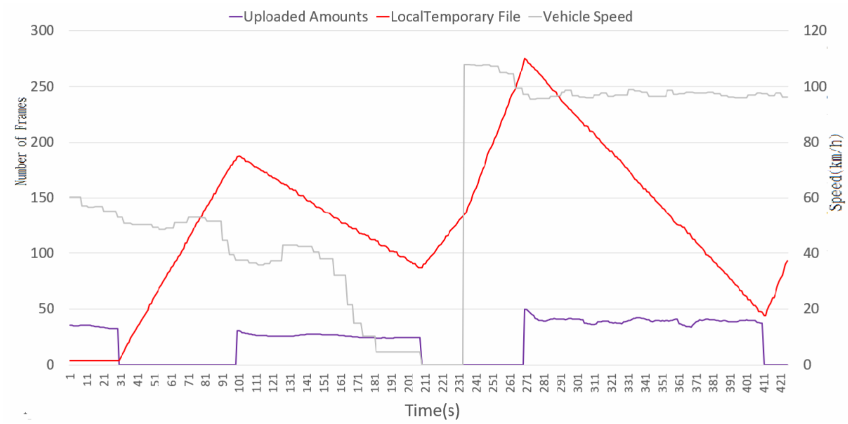


Figure 11. Resending mechanism in the proposed method

Figure 12. Road test results of the proposed method

6 Conclusion

This paper proposes an adaptive image uploading control system for vehicle safety driving monitoring applications based on mobile edge computing technologies. In this study, we propose an architecture for an image uploading system based on mobile edge computing technologies. Our system achieves high efficiency, with low end-to-end latency and fast response. Our study also evaluates parameters such as the vehicle speed, danger alert signals and available bandwidth of network. We then use these parameters to adjust the number of images taken per second, and then automatically controls the quality and amounts of image uploaded. We also calculate the minimum number of images required to restore continuous driving scenes based on the speed of the vehicle. We explore the relationship between factors affecting data transmission to the server, and this helps our system to provide better service to road users driving than other related studies, especially in dangerous situations. When combined with a driving safety awareness information system, our system can offer the most complete and timely information to drivers in emergency situations. It is hoped to propose solutions for the current complex mobile networks in our future works.

Acknowledgements

We thank the Ministry of Science and Technology of Taiwan for supports of this project under grant number MOST 109-2622-E-239-002-CC3, MOST 108-2622-E-239-004-CC3, MOST 107-2218-E-167-004 and MOST 106-2221-E-239-036. We thank co-authors and reviewers for their valuable opinions.

References

[1] C. Xu, S. Hu, W. Zheng, T. Abdelzaher, P. Hui, Z. Xie, H. Liu, J. Stankovic, Efficient 3G/4G Budget Utilization in Mobile Sensing Applications, *IEEE Transactions on Mobile Computing*, Vol. 16, No. 6, pp. 1601-1614, June, 2017.

[2] M. Brahim, Z. Mir, W. Znaidi, F. Filali, N. Hamdi, QoS-Aware Video Transmission Over Hybrid Wireless Network for Connected Vehicles, *IEEE Access*, Vol. 5, pp. 8313-8323, March, 2017.

[3] M. Tsai, 3CV3S: Cloud-Enabled Cooperative Car Video Share and Search System, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 995-1002, July, 2018.

[4] B. Letswamotse, R. Malekian, C. Chen, K. Modieginyane, Software Defined Wireless Sensor Networks (SDWSN): A Review on Efficient Resources, Applications and Technologies, *Journal of Internet Technology*, Vol. 19, No. 5, pp. 1303-1313, September, 2018.

[5] N. Sathishkumar, K. Rajakumar, A Study on Vehicle to Vehicle Collision Prevention Using Fog, Cloud, Big Data and Elliptic Curve Security Based on Threshold Energy Efficient Protocol in Wireless Sensor Network, *IEEE International Conference on Recent Trends and Challenges in Computational Models*, Tindivanam, India, 2017, pp. 275-280.

[6] S. Zheng, X. Zhang, J. Chen, Y. Kuo, A High-Efficiency Compressed Sensing-Based Terminal-to-Cloud Video Transmission System, *IEEE Transactions on Multimedia*, Vol. 21, No. 8, pp. 1905-1920, August, 2019.

[7] Z. Deng, Y. Zhou, D. Wu, G. Ye, M. Chen, L. Xiao, Utility Maximization of Cloud-Based In-Car Video Recording Over Vehicular Access Networks, *IEEE Internet of Things Journal*, Vol. 5, No. 6, pp. 5213-5226, December, 2018.

[8] Y. Zhang, M. Chen, N. Guizani, D. Wu, C. Leung, SOVCAN: Safety-Oriented Vehicular Controller Area Network, *IEEE Communications Magazine*, Vol. 55, No. 8, pp. 94-99, August, 2017.

[9] M. Tsai, T. Pham, F. Ching, L. Chen, An Adaptive Solution for Images Streaming in Vehicle Networks Using MQTT Protocol, *EAI International Conference on IoT as a Service*, Taichung, Taiwan, 2017, pp. 1-6.

[10] M. Tsai, T. Pham, C. Hsiang, L. Chen, Evaluation of the Effect of Variations in Vehicle Velocity and Channel Bandwidth on an Image-Streaming System in Vehicular Networks, *Springer Mobile Networks and Applications Journal*, Vol. 24, No. 3, pp. 810-828, June, 2019.

[11] Z. H. Mir, F. Filali, LTE and IEEE 802.11p for Vehicular Networking: A Performance Evaluation, *Journal on Wireless Communications and Networking*, Vol. 2014, Article No. 89, pp. 1-15, May, 2014.

[12] H. Chen, D. Eddy, R. Chen, C. Chou, Speed-adaptive Street View Image Generation Using Driving Video Recorder, *IEEE International Conference on Multimedia and Expo*, Seattle, WA, USA, 2016, pp. 1-6.

[13] K. Lee, K. Bong, C. Kim, J. Jang, K. Lee, J. Lee, G. Kim, H. Yoo, A 502-GOPS and 0.984-mW Dual-Mode Intelligent ADAS SoC With Real-Time Semiglobal Matching and Intention Prediction for Smart Automotive Black Box System, *IEEE Journal of Solid-state Circuits*, Vo. 52, No. 1, pp. 139-150, January, 2017.

[14] K. Nobori, N. Ukita, N. Hagita, A Surround View Image Generation Method with Low Distortion for Vehicle Camera Systems Using a Composite Projection, *IEEE International Conference on Machine Vision Applications*, Nagoya, Japan, 2017, pp. 386-389.

[15] A. Allouch, A. Koubaa, T. Abbes, A. Ammar, RoadSense: Smartphone Application to Estimate Road Conditions Using Accelerometer and Gyroscope, *IEEE Sensors Journal*, Vol. 17, No. 13, pp. 4231-4238, July, 2017.

Biographies

Ming-Fong Tsai received the Ph.D. degree from the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan. He is currently an Associate Professor with the Department of Electronic Engineering, National United University, Taiwan. His current research interests include Internet of Things and Vehicular Communications.

Chia-Yuen Lin received the B.S. and M.S. degrees from the Information Engineering from the Feng Chia University, Taiwan. His current research interests include Internet of Things and Vehicular Communications.