



Real-Time Image-Based Automotive Sensing: A Practice on Fine-Grained Garbage Disposal

Wenhao Huang
Kazuhiro Mikami
gerhua@sfc.keio.ac.jp
quantan@sfc.keio.ac.jp
Keio University
Japan

Yin Chen
ychen@reitaku-u.ac.jp
Reitaku University
Japan

Jin Nakazawa
jin@sfc.keio.ac.jp
Keio University
Japan

ABSTRACT

This research presents a real-time automotive sensing system for the data of urban garbage disposal. The proposed solution is implemented on an edge computing device mounted on garbage truck where a deep learning based image processing algorithm is implemented to automatically counted the number of collected garbage bags from garbage collection video. A MQTT-based data server was developed to enable data publication from sensing device to server and data accumulation and to facilitate application development. Our system has the functions of high concurrency and low transmission delay, offline reconnection, breakpoint transmission and client authentication. This work is to provide a real-time, low-cost, reliable and replicable system for the implementation of a wide-spread sensing network for automotive edge computing and smart city applications.

CCS CONCEPTS

• **Information systems** → *Location based services*; • **Human-centered computing** → *Empirical studies in ubiquitous and mobile computing*.

KEYWORDS

MQTT, automotive sensing, edge computing, garbage collection, smart cities

ACM Reference Format:

Wenhao Huang, Kazuhiro Mikami, Yin Chen, and Jin Nakazawa. 2023. Real-Time Image-Based Automotive Sensing: A Practice on Fine-Grained Garbage Disposal. In *The International Conference on the Internet of Things (IoT 2023)*, November 07–10, 2023, Nagoya, Japan. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3627050.3627058>

1 INTRODUCTION

The generation and management of garbage in cities are closely linked to our daily lives. Along with the economic development and increase of global population, the overall amount of garbage disposal is constantly increasing and has been affecting the environment and is considered to be critical to the sustainable development

of the planet. For example, according to the report of [13], in fiscal year 2021 the per capita daily average value of garbage disposal in Japan was 901g, which leads to 2.1 trillion JPY cost on administration, and it is predicted to continually increases in future. To fulfil the sustainable development goals and for better recycle and save management cost, it is necessary to collect fine-grained information about garbage collection and real-time collection status of garbage trucks to optimize the policy of garbage management and recycling.

Collecting urban data in a citywide scale plays a fundamental role in the research, development and implementation of smart cities [1]. Managers of garbage disposal facilities could use these sensing data to understand the current garbage collection status in real time. They could also use the recorded historical data to understand the cumulative fine-grained garbage disposal in various regions, and allocate vehicle resources, human resources and costs reasonably. At the same time, it can also promote environmental management within the city.

Different from many of the other countries, the garbage collection in Japan is conducted periodically in a door-to-door way, making it difficult to develop a fine-grained sensing system for its garbage disposal amount. In Japan, house waste is first classified by inhabitants and put in to specified garbage bags (e.g., burnable, nonburnable, plastics), which will be placed in front of their house or a nearby garbage station according to a collection schedule made by local governments. The garbage collection workers will drive garbage trucks to collect these bags around the city in every collection day (typically every weekday). In this way, most of the smart garbage collection solutions that rely on sensors mounted on garbage bins will not work [12].

To overcome the challenges of garbage sensing, Mikami et al. proposed DeepCounter [12], an automotive sensing system where deep learning based image processing technology is used to automatically count the number of collected garbage bags from the video taken through a camera mounted on the rear of a garbage truck in order to generate a fine-grained spatio-temporal distribution on the amount of disposed garbage in cities. While the researchers of DeepCounter validate the preliminary feasibility of the proposed sensing approach, only an off-line evaluation using garbage collection video data recorded from garbage trucks was conducted.

In this study, we implemented a practical sensing system on garbage trucks and evaluated its performance during daily garbage collection in Japan. Considering the massive deployment of the system in the future and the real-time communication of a large amount of data, we adopt the MQTT protocol as the basis of information interaction of the system. MQTT is a Client Server



This work is licensed under a Creative Commons Attribution International 4.0 License.

IoT 2023, November 07–10, 2023, Nagoya, Japan
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0854-1/23/11.
<https://doi.org/10.1145/3627050.3627058>

publish/subscribe messaging transport protocol. It is light weight, open, simple, and designed to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium [2].

The purpose of this study is to develop a real-world installation of the proposed systems. In particular, the following requirements are essential.

- 1) The sensing device needs to be mounted on trucks and is able to on-site processing of the proposed sensing algorithm integrating deep learning based object detection.
- 2) The sensing data of garbage disposal should be published wireless to a data server in real-time.
- 3) The received data can be subscribed and recorded for future application study.

Notice that the previous work of DeepCounter [12] mainly conducted off-site evaluation using pre-recorded garbage collection video to validate the feasibility of the idea. Compared with [12], this paper makes the following contributions.

- At first, to evaluate the performance of the proposed system in real-world installation, we conducted development to implement a sensing device that can be mounted on garbage trucks with required functions of GPS reception, mobile communication, and data publication via the MQTT protocol. We also developed a data server based on MQTT broker where the data can be accumulated and utilized for application development.

- Secondly, for better sensing performance, we used the YOLOv5s [16] model instead of a customized tiny-SSD (Single Shot Detector) model used in [12] for object detection and DeepSORT [19] for object tracking.

- Moreover, to validate the performance of the developed system, we deployed the sensing device to 13 garbage collection trucks. Moreover, to demonstrate the collected data, a web visualization application of fine-grained garbage disposal data was developed in conjunction with Kepler.gl [11] and OpenStreetMap (OSM) [3] to monitor garbage collection in real time and analyze historical data.

This paper is organized as follows. Section 2 describes related works. Section 3 describes our proposed system architecture. Section 4 describes the application implementation and evaluation of our system. Section 5 and section 6 present the performance evaluation and lessons learned in our development. Finally, section 7 concludes this paper.

2 RELATED WORKS

2.1 Smart Garbage Collection Sensing

There has been some research on smart sensing for garbage collection and management. In different countries and regions, garbage and waste management methods are also different. Much of the current research was carried out by measuring the amount of waste and garbage collected in bins based on low cost and low power IoT sensors. In [14], the weight and volume of waste thrown in the waste bins are collected by economical sensors and then sent to cloud server using a micro-controller and GPRS. This data is used to find the waste collection schedule to maximize the collection. In [17], the researchers propose a framework based on Kafka and

Spark aiming at collecting, monitoring and processing streams of data received in real-time by IoT sensor devices while measuring the waste level of waste bins in a distributed environment. Katura et al. proposes a waste monitoring and collection planning system based on narrow-band Internet of Things (NB-IoT) that includes smart bins equipped with ultrasonic sensor that detect the fill level and type of waste, as well as a web-based platform for real-time monitoring and decision-making support [10].

The sensing system proposed in this paper are deployed in an on-board garbage collection sensing system, DeepCounter [12]. This is an automotive sensing system where deep learning based image processing technology is used to automatically count the number of collected garbage bags from the video taken through a camera mounted on the rear of a garbage truck in order to sense a fine-grained spatio-temporal distribution on the amount of disposed garbage in cities. Table 1 summarized the implementation details of the systems developed in this work and those of [10] [12] [14].

Table 1: Comparison of related works about smart garbage collection sensing.

	Sensing Devices	Sensing Methods	Sensing Objects	Data Transmission Media
This Work (Japan)	EDGEPLANT T1 Jetson TX2	Camera	Garbage bags thrown in truck	4G/LTE
Work in [12] (Japan)	NVIDIA Jetson TX2	Camera	Garbage bags thrown in truck	None (Off-site)
Work in [14] (Greece)	Arduino	Ultrasonic sensor	Waste fill level of bins	GPRS
Work in [10] (China)	STM32	Ultrasonic and Gas sensors	Waste fill level of bins	BC95 NB-IoT

2.2 Real-Time Image Processing-based Automotive Sensing

Recently, with the development of technologies related to smart cities, the research on automotive sensing has been extensively invested in the collection of various information and data in cities. It utilizes the mobility of automobiles to realize a wide sensing area with relatively few sensors [1].

Real-time image processing-based automotive sensing is a rapidly evolving field that involves the use of advanced computer vision techniques to analyze visual data captured by cameras mounted on vehicles in order to improve safety, efficiency, and driver assistance, as well as to provide multi-dimensional sensing data that can be sensed by the vehicle's movement. Sivaraman et al. proposed a synergistic approach to integrated lane and vehicle tracking for driver assistance [15]. Without specific hardware and software optimizations, their fully implemented system runs at near-real-time speeds of 11 frames per second. Dong et al. present the techniques

to simultaneously detect the fatigue and distracted driving behaviors using vision and learning based approaches [6]. They use facial features to detect the open/close of eyes, yawning and head posture. The random forest is adopted to analyze the real-time driving conditions.

In addition, some researches make use of the characteristics of automotive sensing to carry out the sensing of real-time passenger flow in urban transportation. Huang et al. proposed a bus crowdedness sensing system which exploits deep learning-based object detection to count the numbers of passengers getting on and off a bus and thus estimate the crowdedness of city buses in real time [9].

3 SYSTEM ARCHITECTURE

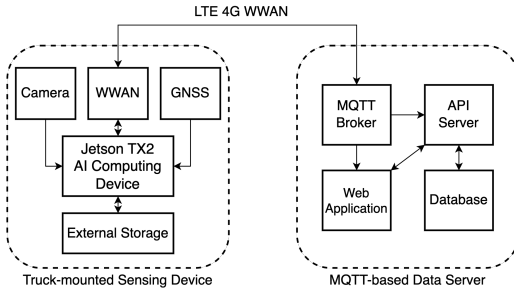


Figure 1: Real-time Sensing System Architecture

3.1 Sensing Device

In this paper, our sensing device uses an embedded AI computing device based on Jetson TX2, which utilizes a CPU+GPU heterogeneous mode to deploy a YOLOv5s [16] V4.0 network model and DeepSORT [19] for real-time sensing of fine-grained garbage disposal. The system architecture of the sensing device is shown in Fig. 1. The computing device is connected to some devices, including the drive recorder camera, the USB video capture module, the WWAN module, the GNSS module and the external storage devices. The drive recorder camera is used to get the video data of garbage collecting process. The WWAN module is designed to enable the embedded device to connect to the Internet. The GNSS module is to get the truck location, speed and time information. The external storage devices are used to store historical sensing results and the historical video of garbage collecting process.

The real-time detection of garbage disposal is to estimate the number of garbage bags thrown into the container of a garbage truck. The video of collecting process will be processed by a Detection-Tracking-Counting (DTC) [12] algorithm running on the embedded AI computing device. The DTC algorithm consists of three processes: object detection, tracking and counting. The DTC algorithm first detects the location of targeted garbage bags from each frame, then tracks each detected bag via the relative locations in successive frames and finally counts it as collected when the location of a tracked bag satisfies a pre-defined condition. The practical sensing sample of garbage bag counting by DTC algorithm is shown in Fig. 3. In this figure, as the number of garbage bags thrown into the garbage collection truck increases, so does the sensing count in the upper left corner of the figures.

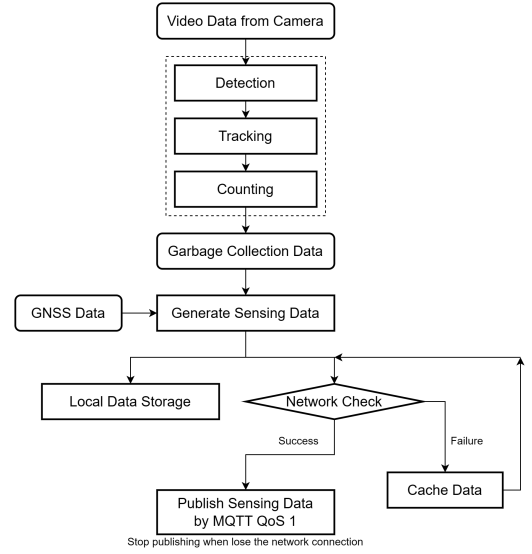


Figure 2: The workflow of embedded sensing device.



Figure 3: A practical example of garbage counting by Detection-Tracking-Counting (DTC) algorithm. Using object detection and tracking, the number of garbage bags put into the garbage collection truck is counted.

After the data of garbage disposal volume is obtained, the sensing data messages to be published are generated by combining with the spatio-temporal geographic information. The sensing data message consists of seven parts, including the SoC serial number, time and date, latitude, longitude, speed, garbage disposal volume, and garbage truck number. The sample messages are shown in Fig. 4

```

1422821088608,2023-02-15 14:51:50,35.217779542,139.709131196,0.023,0,Yokosuka/003
1422821068521,2023-02-15 14:51:51,35.217818902,139.70919277,0.101,0,Yokosuka/002
1422821068514,2023-02-15 14:51:51,35.217720887,139.709309268,0.014,0,Yokosuka/001
1422821088717,2023-02-15 14:51:51,35.217715182,139.709254878,0.03,0,Yokosuka/004
  
```

Figure 4: Sample messages of sensing data. (SoC serial number, time and date, latitude, longitude, speed, garbage disposal volume, and garbage truck number)

In this system, the embedded sensing device will publish a message every second to the MQTT broker. QoS level setting is very important in MQTT protocol communication. The Quality of Service (QoS) level is an agreement between the sender of a message and the receiver of a message that defines the guarantee of delivery for a specific message. There are 3 QoS levels in MQTT: QoS 0,

at most once; QoS 1, at least once; QoS 2, exactly once [8]. In the message transmission of this system, because the outdoor cellular network is used, the network connection is often unstable, as well as duplicate data has no impact on information collection and processing, so we choose QoS 1 to ensure that the information is published at least once. In the case of QoS 1, messages may publish multiple duplicate data due to unstable connection. In this system, we control whether to publish data by obtaining the MQTT connection state in real time, so as to control the sensing data cache when the network is disconnected and reduce the occurrence of duplicate data as much as possible. The MQTT connection state represents the state of the connection between the MQTT client object and the MQTT broker, and it also reflects the network connection state of the sensing device. If MQTT client returns a connection error or the message publishing process cannot obtain connection, the process will not be interrupted and try to reconnect. Meanwhile, the recently collected garbage sensing data will be temporarily stored in a cache before being transmitted. After confirming that the MQTT connection is connected, the cached data will be reissued. We ensure data integrity through the above workflow and methods. As for the data transmission time of this system, we expect it can be managed below 100ms. To evaluate the real-time performance of data transmission, it will be calculated in conjunction with the delay on the server-side subscription.

Some measures have also been taken to ensure the security and confidentiality of the transmitted data. To prevent unauthorized clients from publishing unauthorized data and subscribing data, the system uses usernames and passwords to authenticate devices and clients. In terms of authentication, the SHA256 hashing algorithm is utilized to hash passwords and improve security. After the sensing data is published, it will be processed by MQTT-based data server.

3.2 MQTT-based Data Server

The data server is a combination of a MQTT broker, a database and an application interface server. The workflow of the data server is shown in Fig. 5. We use EMQX [7] as the MQTT broker for real-time data Pub/Sub [4], saving historical sensing data and data processing for web applications. The historical data API will be invoked to access the database only when historical data is requested by the web application. The design of the data server relies on the MQTT broker for the primary sensing data exchange, reduces the coupling of devices and functions, minimizes the frequent invocation of application interfaces, and facilitates further development, system maintenance, and service migration.

The workflow is mainly divided into real-time data processing and historical data processing, among which the real-time data is divided into the real-time data storing process and the web virtualization application directly subscribe to the real-time sensing data. First, the message subscription API will connect to the MQTT Broker to subscribe the messages of all topics and parse sensing data. To avoid overwhelming the database server due to a large number of connected devices, the data is cached every second and then inserted into the database in a single commit operation to reduce the pressure on the database server and prevent it from reaching its data processing capacity limit. An index was built in

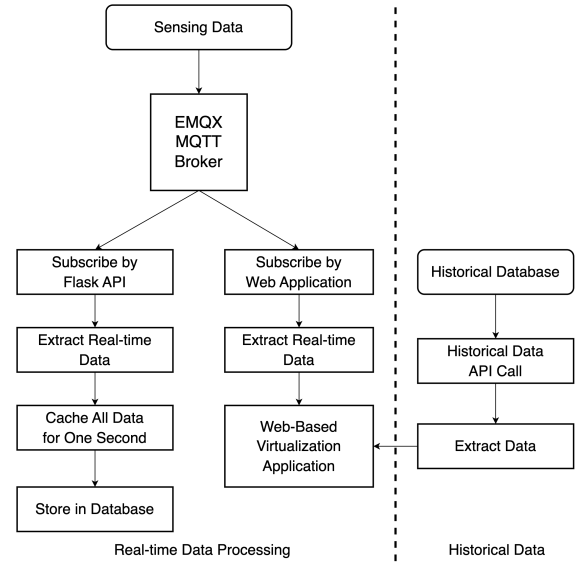


Figure 5: The workflow of the MQTT-based data server.

the database based on timestamps using B+ trees to optimize the performance of database queries.

To demonstrate the collected garbage disposal data, we developed a data visualization platform for real-time sensing data and historical data. The system is visualized based on Kepler.gl [11] and OpenStreetMap [3]. Kepler.gl is a powerful open source geospatial analysis tool for large-scale data sets [11]. The demonstration application directly obtains the latest data through subscription to MQTT broker and obtains the historical data from the database by invoking the developed web application API. The demonstration application displays the real-time locations of garbage trucks and the collected fine-grained garbage disposal. Please refer to Section 5 for specific demonstration examples.

4 EXPERIMENTAL IMPLEMENTATION

In our experiment system, we have deployed the sensing devices into 13 trucks in three cities in Kanagawa, Japan by March, 2023. The system is installed in the garbage collection trucks in Fujisawa, Kamakura, and Yokosuka, as shown in Fig. 6(a), Fig. 6(b) and Fig. 6(c).

For edge sensing devices installed in the garbage collection trucks, the key hardware components are shown in Table 2. For the power of the sensing devices, we utilize the 24V on-board power supply from the vehicle's battery and use the accessory signal to control the power on/of the sensing device. When the engine turns on, the system will start. After the engine turns off, the sensing device will be shut down if the engine is not turned on back before a countdown finished. We use the video taken by a camera mounted on the rear of a truck as shown in Fig 6(c), which is part of the driving recording system of the truck. For wireless communication, there are several wireless communication candidates, for example the NB-IoT, which provide low-power, low-cost and wide area communication service. We choose 4G/LTE as our communication protocol for the following reasons. First, the 4G/LTE is much more

Table 2: The detailed information of devices used in our system.

Parts	Details
Embedded AI Computing Device	EDGEPLANT T1 Jetson TX2 Ubuntu 18.04 Jetpack L4T 32.7.1 CUDA 10.2
Camera	Sony HDV Camera (for verification) / C4015R (on truck)
GNSS Module	u-blox NEO-M8U
WWAN Cellular Module	Sierra Wireless EM7430 4G WWAN Module
Video Capture Module	DriverGenius VDC2021
External Storage Device	SanDisk Extreme SSD

mature and reliable, ensuring the smooth deployment of the studied system. Also, there are certain 4G/LTE services in Japan which are monthly priced in a reasonable price (typically 1000 JPY per month). In these services, there are some communication limitations (e.g., a communication rate up to 128 kbps.) but it is sufficient for the use of our system. Moreover, the communication rate of these cellular services can be upgraded temporally when required, e.g., for a system update, while it is difficult if not impossible to fulfil such heavy communication tasks in the NB-IoT or other LPWAN networks. To this end, we integrate the Sierra Wireless EM7430 Cat6 WWAN module [18] and use the cellular network provided by NTT Docomo [5]. The access points and ID provided by ISP are set in the operating system. In our sensing system, we use the serial number of the processor and the garbage truck number as the unique ID as the client number of MQTT. We adopt some basic security methods including the client device authentication in the MQTT communication and closing unnecessary ports in the sensing device to prevent unauthorized external access.

For the implementation of the data server, we combine three major components a loosely coupled whole. As described in the Chapter 3, we use EMQX [7] MQTT broker for the Pub/Sub service, MySQL for database, and Flask, a lightweight web application framework written in Python, for internal data processing API and application service. Regarding the physical server implementation, the data server is currently set up in the laboratory of Keio University, using a Synology NAS and deploying all services based on Docker containers.

5 DEMONSTRATION AND EVALUATION

Table 3: Sensing performance evaluation between this work and [12]

	YOLOv5s + DeepSORT (This work)	Tiny-SSD + IoU Tracking [12]
Precision	0.915	0.42
Recall	0.690	0.80
FPS Average	16.52	6.2

In this section, we evaluate the sensing performance, the system functions and the real-time performance. Firstly, we compared the



(a) A Fujisawa garbage collection truck equipped with the sensing system.



(b) The sensing device installed behind the passenger seat of a garbage collection truck.



(c) The drive recorder camera mounted on the back of a garbage collection truck.

Figure 6: Experimental implementation in the actual environment.

sensing performance between this work and [12] in Table 3. From Table 3, we can see that the both the precision and the processing speed are improved greatly while the recall slightly decays, mainly due to the YOLOv5s [16] and DeepSORT [19] algorithms. This observation ensures the feasibility of real-world installation the developed system.

The evaluation of the system function mainly includes the evaluation of the image processing algorithm of the sensing device of the system, the collection of garbage disposal data, truck data and geographic data, and the data processing under various network connection status, including the data cache and publishing of the sensing data. All functions designed and developed so far have been validated. At first, we measured the processing performance of one sensing device including its CPU/GPU usage and memory load. We present the performance statistics of 690 sampling points measured when the sensing algorithm is running in during minutes. We can see from Fig. 7 that the average CPU load is 16%, the average GPU load is 70.1% and the average memory consumption is about 3.6GB (i.e., an average load of 94.1%). The result is expected because a deep-learning based object detection model is used to detect garbage bags, which requires high GPU and memory resources. While the current system works stably in our evaluation, how to compress the detection model to release more GPU/memory resources or transplant it to a relatively low-end platform platform such as the Jetson Nano is still a valuable direction.

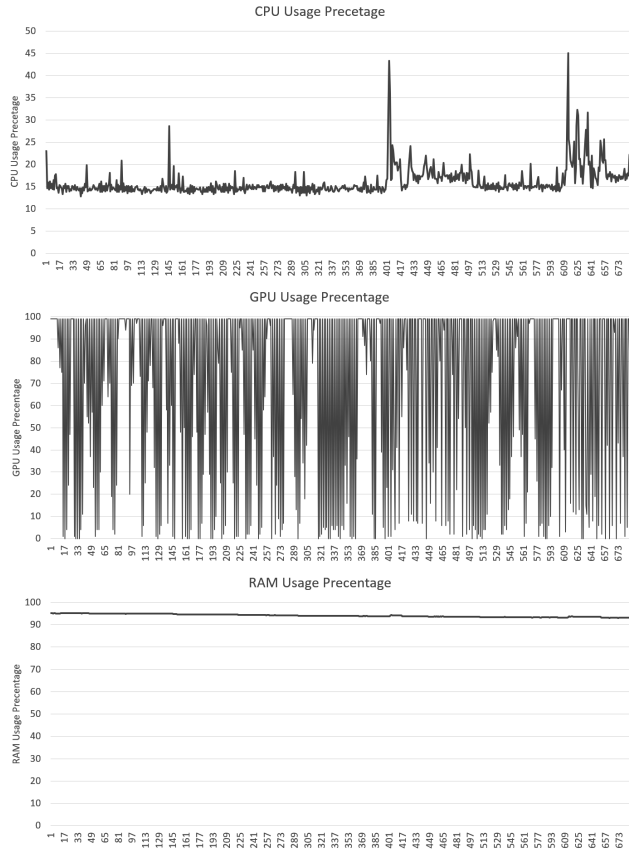


Figure 7: Evaluation of system hardware resources of one sensing device including its CPU/GPU usage and memory load



Figure 8: Evaluation of the processing time of each frame varies during the collection process. The blue part represents that the collection is in progress.

Recall that the whole processing of the sensing algorithm includes detection, tracking and counting. To evaluate the processing time, we conducted experiment using a 30-minute garbage collection video and measured the processing time of each frame. At first, the average per-frame processing time is 60.48 milliseconds, i.e., 16.52 frames per second (FPS). Also, the Fig. 8 demonstrates how the processing time of each frame varies during the collection process. In the Fig. 8, the X-axis represents the sequence number of each

frame, and the Y-axis represents the processing time of each frame. The parts included in blue squares represent the time periods of garbage collection and the others represent those without collection activity such as during movement. We can see from Fig. 8 that for the frames when garbage bags are detected and tracked, i.e., during collection, the processing time slightly increases than the others. This phenomenon can be explained as follows. When there are bags detected, the DeepSORT algorithm will start to work and thus result in more processing time. It is also notable that the increased time of DeepSORT is not significant than that of object detection. This implies that the object detection is the major bottleneck for further improvement on the overall performance.

Meanwhile, to ensure all the sensory data are published even in the cases including temporary network disconnection, the sensory data including timestamp will be temporarily stored in a cache until been transmitted. Once the MQTT connection is available, the cached data will be published in a FIFO pattern. Please notice that the cache mainly serves as a backup mechanism whose usage is very low because the typical communication requirement is about 166 bytes, i.e., sending one packet per second. Meanwhile, the communication rate of the cellular service is 128 kbps in theory. Due to this reason, the cache load is extremely low and does not place significant effect on system performance.

We have extracted three days' running data from the data background of MQTT Broker. Within three days of the garbage collection operation, the data throughput is about 250 pieces of data every two minutes (the packet size for each piece is approximately 166 bytes), and no messages have been dropped. Data integrity is further evaluated by comparing the history of the original data stored in the sensing device with the historical data in the database received from MQTT broker. All data were published and subscribed correctly, including data that was temporarily cached when there was no network connection.

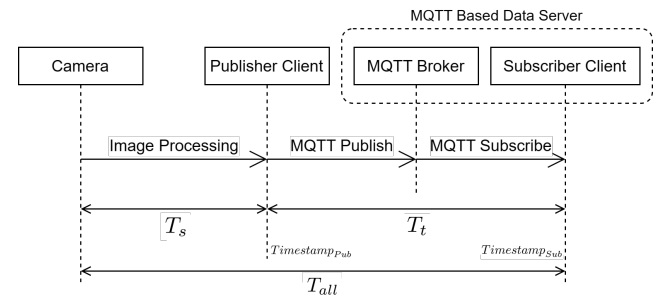


Figure 9: The time separation of the various parts of the process from data acquisition to data received by the MQTT-based server.

In terms of system real-time performance evaluation, the data processing of the sensing system is mainly divided into image processing based on deep learning and data transmission time. In order to measure the real-time performance of the system, we measure the two parts respectively and take the arithmetic mean value, and use them to evaluate the real-time performance of this sensing system. Since that the timestamp of the data received by

the data server cannot be obtained directly from the MQTT broker, we set the MQTT subscriber client for delay evaluation hosted by the same machine of the data server so that the delay from MQTT broker to subscriber is neglectable. Therefore, in our evaluation, we use the timestamp when a published data received at the subscriber for delay evaluation. As shown in Fig. 9, we illustrate the total process from video capture to data reception. T_s represents the processing time from video capture at camera to data publishing. T_t represents the time from data publishing to data reception at data server. $Timestamp_{Pub}$ represents the timestamp of when the sensing data is published. $Timestamp_{Sub}$ represents the timestamp of when the sensing data is subscribed. T_{all} represents the total processing time of the sensing data. The related equations for the real-time performance evaluation are as follows.

$$T_t = Timestamp_{Sub} - Timestamp_{Pub} \quad (1)$$

$$T_{all} = T_s + T_t \quad (2)$$

We used the frame processing speed recorded in the change of garbage disposal quantity to estimate the processing time of garbage disposal volume sensing. In the real-time evaluation of image processing of truck-mounted sensing device, the average time taken to process each frame of a 30-minute video of the garbage collection process was collected. The frame rate is 16.52 frames per second (FPS), which means that one frame takes about 60.48 milliseconds to be processed.

In the real-time evaluation of sensing data transmission by MQTT, 1800 times of data transmission time is collected. For this evaluation, data transfer time is defined as the time it takes from the time the sensing message is published to the time it is received by the data subscriber. The data transfer time is calculated based on the difference of the timestamp.

After obtaining the processing time of garbage collection volume sensing based on deep learning and the time of data transmission, the overall system performance evaluation results can be obtained as shown in Table 4. The average time required for image processing is about 60.48ms. The average communication delay in transmission is about 16ms. Thus, the average time including processing and communication in the current system is 76.48ms. It can be seen from the data that the processing of image data based on deep learning takes more time, but in general, the acquisition of real-time sensing data is realized.

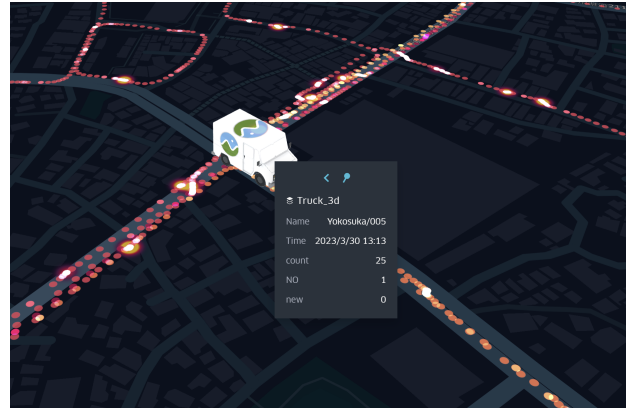
Table 4: Real-time performance evaluation results.

	T_{all}	T_s	T_t
Arithmetic Mean (Millisecond)	76.48	60.48	16.0

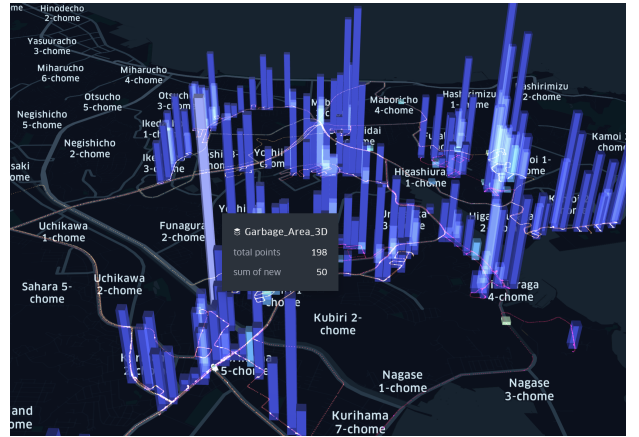
In terms of data analysis and application, as described in Chapter 3, a real-time fine-grained garbage disposal monitoring and historical data analysis platform has been developed by combining sensing data with open source geospatial analysis tool. The screenshots of the data platform are shown in Fig. 10. which are taken from real-time data and historical data at March 30, 2023. In this visualized data application, users can view the current real-time garbage



(a) Visualized fine-grained garbage disposal data for Yokosuka.



(b) Historical routes and real-time locations of garbage collection trucks can be seen from the data platform.



(c) Move the mouse over each bar shows the information regarding the overall quantity of garbage disposal details for a particular region. In this figure, "sum of new" represents the amount of newly increased garbage disposal within the range from that date to now, and "total points" represents the total amount of sensor data messages received within this range.

Figure 10: Visual fine-grained garbage disposal sensing data application.

disposal sensing data, view historical data, and reproduce the process of garbage collection by selecting the timestamp for animation play. In Fig. 10(a), it shows how the volume of garbage collected can be distributed by region by combining sensing data and maps. A 3D bar chart is used to represent the amount of fine-grained garbage handled. The higher the height of the bars, the higher the amount of garbage disposed within the range. The granularity of fine grain can be changed to visualize the amount of garbage disposed from different ranges of city areas. The continuous colored dots in Fig. 10(b) represent the path of the garbage collection trucks. The real-time location of garbage collection trucks currently in operation is showed by using a 3D truck model. In Fig. 10(c), moving the mouse over each bar shows the information regarding the overall quantity of garbage disposal details for a particular region.

6 LESSON LEARNED IN THIS STUDY

In Section 5, we have validated that the developed system is able to conduct a fine-grain sensing on urban garbage disposal in Japanese cities. All the heavy processing can be finished on the vehicular sensing device mounted on trucks, so that we can publish sensory data to data server with mobile network in real time. Moreover, the object detection consumes the most GPU and memory resources while the DeepSORT for object tracking only incurs insignificant processing delay but makes remarkable performance improvement than the previous work of [12]. This result inspires us that we should place more effects to improve the object detection process for less GPU and memory consumption. For device and system safety, it is notable that the sensing devices are deployed in an environment without sufficient physical protection. How to identify the potential security threats and develop corresponding solutions to enhance the security of such systems can be a good future direction.

To demonstrate the collected data, we developed a preliminary data visualization and observed interesting spatial distributions of the garbage disposal. It is notable that the behaviors of garbage disposal are highly related to social, geometric and environmental factors, cross-domain and comprehensive data analysis would produce many interesting and useful applications. For example, we are planning to develop an estimation algorithm to estimate future disposal volumes to help collection workers refine their collection plans. Also, we are studying how to use disposal data to change the inhabitants' behavior in garbage disposal. Since such fine-grained data of garbage disposal is not available before, it is reasonable to expect that a series of new research results in data science and application study can be born from the output of this study.

7 CONCLUSION

This paper reported our development of real-time fine-grained garbage disposal sensing system. The system consists of sensing devices mounted inside the garbage truck and the MQTT-based data server. Through the visual data platform, the proposed solution allows users to obtain real-time sensing data such as fine-grained garbage disposal, truck location, and truck operation data. Functionally, MQTT protocol and data cache ensure data integrity, real-time performance and loose coupling. Developers can expand functions based on the data interface of application server according to application requirements. The data transmission performance and

function effectiveness of the system are verified. The architecture of the system can be applied not only to the garbage disposal sensing described in this article, but also to real-time automotive sensing systems with similar requirements.

ACKNOWLEDGMENTS

This work was partly supported by the commissioned researches (No.22610, No.05401) of NICT and JSPS KAKENHI Grant Numbers JP21K17735.

REFERENCES

- [1] Yin Chen, Jin Nakazawa, Takuro Yonezawa, Takafumi Kawsaki, and Hideyuki Tokuda. 2016. Cruisers: A Public Automotive Sensing Platform for Smart Cities. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*. 767–768. <https://doi.org/10.1109/ICDCS.2016.53>
- [2] OASIS MQTT Technical Committee. 2018. MQTT Version 5.0. Committee Specification 02.
- [3] OpenStreetMap contributors. 2023. Planet dump retrieved from <https://planet.osm.org>. <https://www.openstreetmap.org>.
- [4] C. Doblander, K. Zhang, and H.-A. Jacobsen. 2016. Publish/Subscribe for Mobile Applications Using Shared Dictionary Compression. In *Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)* (Nara, Japan). 775–776. <https://doi.org/10.1109/ICDCS.2016.70>
- [5] NTT DOCOMO. 2023. NTT DOCOMO Home. <https://www.docomo.ne.jp/english/>. Last visited on September 21st, 2023.
- [6] B. T. Dong and H. Y. Lin. 2021. An On-board Monitoring System for Driving Fatigue and Distraction Detection. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*. Valencia, Spain, 850–855. <https://doi.org/10.1109/ICIT46573.2021.9453676>
- [7] EMQX. 2023. EMQX 5.0. <https://github.com/emqx/emqx>. Last visited on March 31st, 2023.
- [8] EMQX. 2023. MQTT QoS. <https://docs.emqx.com/en/enterprise/v5.0/mqtt/mqtt-qos>. Accessed: March 31st, 2023.
- [9] W. Huang, A. Tsuge, Y. Chen, T. Okoshi, and J. Nakazawa. 2022. A Bus Crowdedness Sensing System Using Deep-Learning Based Object Detection. *IEICE Transactions on Information and Systems* E105.D, 10 (2022), 1712–1720. <https://doi.org/10.1587/transinf.2022PCP0007>
- [10] M. Katura, L. Wu, C. Fu, and W. Liu. 2022. NB-IoT Based Waste Monitoring and Collection Planning System. In *Proceedings of the 2021 9th International Conference on Information Technology: IoT and Smart City*. Guangzhou, China, 213–219.
- [11] Kepler.gl. 2023. Kepler.gl. <https://github.com/kepler.gl/kepler.gl>. Accessed: April 7th, 2023.
- [12] K. Mikami, Y. Chen, J. Nakazawa, Y. Iida, Y. Kishimoto, and Y. Oya. 2018. Deep-Counter: Using Deep Learning to Count Garbage Bags. In *2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. Hakodate, Japan, 1–10. <https://doi.org/10.1109/RTCSA.2018.00010>
- [13] Government of Japan Ministry of the Environment. 2021. Results of the Survey on the Actual Situation of General Waste Disposal. https://www.env.go.jp/recycle/waste_tech/ippan/r2/index.html. Accessed: July 27th, 2023.
- [14] S. Sharmin and S. T. Al-Amin. 2016. A Cloud-Based Dynamic Waste Management System for Smart Cities. In *Proceedings of the 7th Annual Symposium on Computing for Development*. Nairobi, Kenya.
- [15] S. Sivaraman and M. M. Trivedi. 2013. Integrated Lane and Vehicle Detection, Localization, and Tracking: A Synergistic Approach. *IEEE Transactions on Intelligent Transportation Systems* 14, 2 (June 2013), 906–917. <https://doi.org/10.1109/TITS.2013.2246835>
- [16] Ultralytics. 2021. YOLOv5 Github. <https://github.com/ultralytics/yolov5>. Last visited on September 21st, 2023.
- [17] G. Vonitsanos, T. Panagiotakopoulos, A. Kanavos, and A. Kameas. 2022. An Apache Spark Framework for IoT-Enabled Waste Management in Smart Cities. In *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*. Corfu, Greece.
- [18] Sierra Wireless. 2019. Sierra Wireless AirPrime EM7430. <https://source.sierrawireless.com/devices/em-series/em7430/>. Last visited on September 21st, 2023.
- [19] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*. 3645–3649. <https://doi.org/10.1109/ICIP.2017.8296962>