# Traffic Light Recognition using High-Definition Map Features

Manato Hirabayashi[a], Adi Sujiwo[a], Abraham Monrroy[a], Shinpei Kato[b], Masato Edahiro[a]

[a] *Graduate School of Information Science, Nagoya University*
[b] *Graduate School of Information Science and Technology, The University of Tokyo*

## Abstract

Accurate recognition of traffic lights in public roads is a critical step to deploy automated driving systems. Camera sensors are widely used for the object detection task. It might seem natural to employ them to traffic signal detection. However, images as captured by cameras contain a broad number of unrelated objects, causing a significant reduction in the detection accuracy. This paper presents an innovative, yet reliable method to recognize the state of traffic lights in images. With the help of accurate 3D maps and a self-localization technique in it, elements already being used in autonomous driving systems, we propose a method to improve the traffic light detection accuracy. Using the current location and looking for the traffic signals in the road, we extract the region related only to the traffic light (ROI, region of interest) in images captured by a vehicle-mounted camera, then we feed the ROIs to custom classifiers to recognize the state. Evaluation of our method was carried out in two datasets recorded during our urban public driving experiments, one taken during day light and the other obtained during sunset. The quantitative evaluations indicate that our method achieved over 97 % average precision for each state and approximately 90 % recall as far as 90 meters under preferable condition.

*Keywords:* Autonomous Vehicles, Vehicle Environment Perception, Information Fusion

## 1. Introduction

Autonomous driving systems have been suggested as a promising solution to solve transportation problems, such as serious traffic accident and personal transportation in aging societies. A survey conducted by the Cabinet Office of Japan in 2014 [1] found that the trend of rapidly aging population is a worldwide phenomenon. That survey indicated that the percentage of people aged 65 and over increased from 5.1 % in 1950 to 7.7 % in 2010 and is expected to reach 17.6 % by 2060. In addition, the survey found that this trend is expected to continue in the latter half of the century.

Personal transportation is an emerging problem in aging societies. People who live in areas without an effective public transportation system must rely on driving. However, many older people have difficulty driving due to a natural decline in their physical and cognitive abilities. The Japanese government has set graduated levels for autonomous driving [2], and the level required for an aging society is at least three. At this level, the vehicle controls all the functions of acceleration, steering, and braking under preferable condition, while driving will be handed over to the driver when the operation reaches the system's capacity limit. From this perspective, autonomous driving systems will continue to be developed and will eventually become common. Moreover, it is commonly said that V2I (Vehicle-to-roadside-Infrastructure) communication, as well as V2V (Vehicle-to-Vehicle) communication help to improve traffic safety and autonomous driving functionality. These systems share various information such as traffic light state, and the position of obstacles with roadside infrastructure or other vehicles. However, it will take considerable time until the full deployment of such communication systems. Because these require considerable extra equipment, or replacement for existing vehicles or infrastructures. In this transition period, autonomous self-driving vehicles and vehicles driven by people will share the road. Under such conditions, autonomous vehicles must be able to recognize various traffic information, such as road signs and traffic lights, and respond appropriately.

Camera sensors are typically employed in object recognition tasks. Case in point, several traffic light detection and recognition methods [3–7], and traffic sign recognition methods [8, 9] have been presented. Traffic light recognition systems for autonomous vehicles must be sufficiently fast and accurate. Generally, cameras installed in vehicles capture a wide variety of unrelated objects to the task of traffic light recognition, such as billboards, roadside trees, etc. These objects are considered "noise" in terms of traffic light recognition and can cause a significant reduction in the detection accuracy. High-resolution cameras can improve recognition accuracy; however, they require longer processing time, which is not ideal for driving situations.

In this paper, we explore, propose and evaluate the possibility to use Region of Interest (ROI) to extract only the image section related to the traffic lights. To achieve this, we use the localization result in the 3D map and extract the traffic lights' position in the 3D space. Using an intrinsically and extrinsically calibrated camera, we then project the position of these traffic lights to the image space, extract the surrounding area in the image, and feed them to a custom classifiers to finally obtain the status of the traffic light.

**Contributions:** This paper makes the following contributions.

- To provide an accurate method to extract ROIs from images with the help of a fusion technique between a camera and accurate 3D maps. This technique can help to greatly reduce the number of unrelated objects to the recognition tasks that cause wrong inference results.

- To present two different methods for the classification of the color state of the traffic lights in the ROIs. The first one is based on a morphological interpretation of the image, while the second is based on a deep CNN. Both, methods are fast to execute due to the reduced area from the ROIs.

- To evaluate the proposed methods in detail with datasets collected during actual on-road driving experiments. We compared recognition tendencies for each method and evaluated effects of distance from the target traffic lights for the correct deployment of autonomous driving systems in public road.

**Organization:** The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the system model and assumptions. Section 4 presents the implementation of the proposed traffic light color state recognition method. The proposed method is discussed and evaluated in Section 5, and conclusions and suggestions for future work are presented in Section 6.

## 2. Related Work

**Traffic light recognition methods:** Traffic light recognition is a critical step in the deployment of advanced driver assistance systems (ADAS) and autonomous driving systems. Several works have been done regarding this topic. Mori *et al.* proposed a method to detect traffic lights in real time from video camera images [3]. Their method extracts contours from an image, searches for circular shaped objects to identify the traffic lights in the image. These algorithms are known as morphological processing, on which our work is also based.

Omachi *et al.* surveyed the effects of normalized RGB images obtained from a vehicle-mounted camera in their traffic light detection method [4]. They defined threshold conditions and edge detection techniques to detect the traffic lights in the images. Charette *et al.* employed spot light detection on gray scale images in conjunction with template matching to detect traffic lights [5].

Fairfield *et al.* exploited a prior map of 3D traffic light locations to estimate a position of traffic lights in images captured by a vehicle-mounted camera [10]. Once this was estimated, they applied blob-segmentation to recognize the state of the traffic lights. Similarly, Jang *et al.* exploited a prior map of 3D traffic light, extract its ROIs, and fed them into detection and classification modules prepared for appropriate type of traffic light using map information [7].

Typically, images being used for the recognition include several other objects that can result in reduced recognition rate. In this paper, we present a novel method that tackles this problem by confining processed region with the help of accurate 3D maps, greatly reducing the number of noisy objects. While [10] and [7] adopted a similar strategy to reduce the number of noisy objects, the map we use contains 3D pose of traffic lights. Furthermore, our method uses accurate camera pose in 3D maps estimated from driving environment. Exploiting these features, the ROI size can be decided dynamically, leading to an effective reduction of noisy objects, even if the recognition targets are at a far distance from ego-vehicle. Finally, in order to increase the recognition accuracy, we integrated a deep convolutional neural network to our recognition module.

**The techniques used in the proposed method:** The system in this work was implemented on top of Autoware [11], which is an Autonomous-driving framework for urban scenarios. It is built on top of the Robot Operating Systems (ROS) [12]. ROS provides libraries, tools and optimized inter-process communication functionality, while Autoware compiles sensing, localization, perception and control modules for autonomous cars. Exploiting these features, we were able to obtain raw data from sensors, perform localization on 3D maps, and calculate the positions of traffic lights projected on images. ROS allows the transformation between several coordinate frames, with the help of TF library [13]. This library eases the tracking of multiple coordinate systems over time and allows efficient transformations between them. Case in point, it helped us to compute and track locations of traffic lights in both 3D map and camera coordinates frames, once we registered all the frames and transformations between them.

## 3. System Model and Assumption

The proposed traffic light color state recognition method can be divided into

I) The extraction of the regions that include a traffic light from a camera image and

II) Color state recognition using the extracted regions.

To clarify the problem addressed in this paper, we first present an overview of 3D maps and a ROI extraction technique. Then, we discuss the morphological and deep learning traffic light state classification methods.

### *3.1. 3D Map*

The system assumes that two 3D maps described below are given in advance. First one is an accurate 3D point cloud map. This map is the aggregation of measured points with 3D coordinate values and represents shape information around driving environment. Combining this map with the position estimation method called NDT, described in the following section, we can obtain the ego-vehicle's position on the map coordinate system. The second one is a 3D feature map. This one is similar to a HERE HD Live Map [14]. The map we used in this research mainly contains the following information for individual traffic lights:

- ID for each of the bulbs composing each traffic light,
- Horizontal and vertical angle that the traffic light bulb is facing,
- ID of the pole where the traffic light bulb is installed,
- Class of the traffic light bulb, such as pedestrian, traffic,
- ID of the nearest lane for the traffic light.

Due to the fact that the map contains independent road features, i.e. points in the space, vectors defining direction, traffic lights bulbs, etc. and knowing that this feature map was built on top of the point cloud, we developed a system to form complete instances of traffic lights and relate them to the vehicle driving path by comparing the lights' facing against the direction of our ego-vehicle.

### *3.2. ROI Extraction*

The ROI extraction is achieved with the help of a camera, a LIDAR sensor, a localization method, and a map. A LIDAR is a sensor that emits ultraviolet, radiant, or infrared laser rays, similar to radio waves used in conventional radar sensors. In a similar fashion, it measures the distance between the sensor and objects by analyzing the flight time of the emitted rays. Compared to radio waves, laser wavelengths are an order of magnitude shorter, these enables a LIDAR to measure smaller objects and acquire detailed shape information. We employed a 360° LIDAR sensor to estimate the sensor position accurately in a given 3D point cloud map by comparing the measured shape information to the map. Once the location is known on the point cloud map, the corresponding 3D feature map is used to obtain the positions of the traffic lights. The ROI extraction process is as follows:

Step 1) estimate the LIDAR position in a 3D map,

Step 2) estimate the camera position in a 3D map,

Step 3) project the 3D traffic light position coordinates to a camera image, and

Step 4) extract the ROI according to the projected traffic light position.

To estimate the LIDAR position in a 3D map (Step 1), the shape of the surrounding environment is measured first. The LIDAR position on the 3D map is acquired by comparing the shape to a 3D point cloud map using Normal Distributions Transform (NDT) [15, 16]. During experimentation, we found that NDT can estimate the LIDAR position accurately. The range of localization error is within centimeter order; therefore, we make the assumption that localization precision is acceptable on 3D maps.

The proposed method also requires the LIDAR and the camera 3D positional relationship (also referred to be as "extrinsic parameters"). We employ the method proposed in [17] to acquire this relationship. This work obtains the extrinsic parameters from multiple LIDAR sensors using multiple planes. According to the paper results, the calibration result contains certain errors, but these can be almost negligible.

Having in mind that both localization and calibration contain errors, we designed our method to be resilient to a certain degree of error. However, as discussed in Section 5, the excessive calibration errors will lead to a reduced recognition accuracy, because ROI extraction would fail to capture traffic lights in the ROIs properly.

The preliminarily calculated positional relationship between the LIDAR sensor and the camera is used to estimate the camera position in the map (Step 2). This relationship has six degrees of freedom that represent translations and rotations in 3D space. This extrinsic parameter remains constant as long as the sensors are fixed to the vehicle body. The camera position on the 3D map is calculated using the location obtained by NDT (Step 1) and the relative position between the LIDAR and camera sensor as follows:

$$p_{cam} = R(\alpha, \beta, \gamma) \cdot p_{LID} + T(x, y, z) \tag{1}$$

where

$p_{cam}$ denotes the camera position in the 3D map coordinate system,

$p_{LID}$ denotes the position of the LIDAR sensor in the 3D map coordinate system,

$(x, y, z)$ denotes relative translations,

$(\alpha, \beta, \gamma)$ denotes relative rotation angles around each axis,

$T(x, y, z)$ is a translation matrix, and

$R(\alpha, \beta, \gamma)$ is a rotation matrix.

To project the 3D traffic light position coordinates to a camera image (Step 3), traffic light coordinates in the 3D camera coordinate system $(s_x, s_y, s_z)$ are obtained by transforming traffic light coordinates in the 3D map coordinate system. These coordinates are projected onto image plane coordinates $(u, v)$ using the camera's focal length $(f_x, f_y)$ and the center coordinates of the image $(c_x, c_y)$ as follows.

$$u = s_x \frac{f_x}{s_z} + c_x, \; v = s_y \frac{f_y}{s_z} + c_y \tag{2}$$

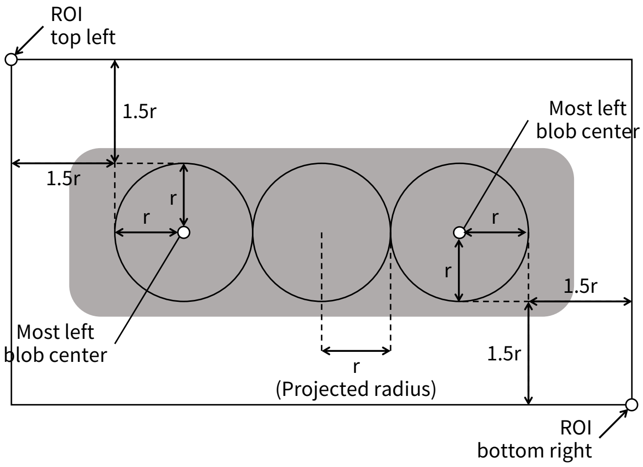


Figure 1: Proposed ROI extraction definition

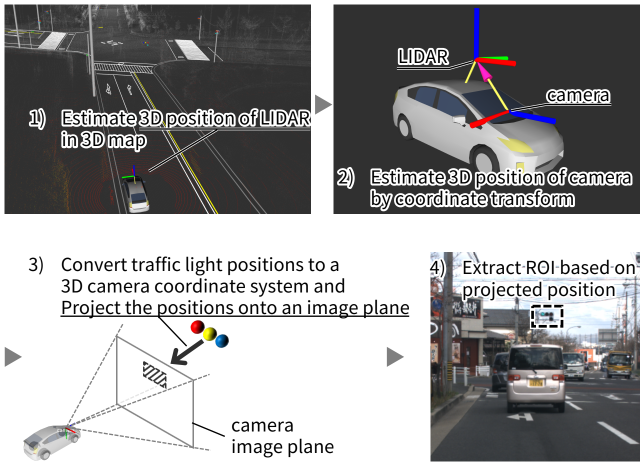

1) Estimate 3D position of LIDAR
in 3D map
LIDAR
camera
2) Estimate 3D position of camera
by coordinate transform
3) Convert traffic light positions to a
3D camera coordinate system and
Project the positions onto an image plane
camera
image plane
4) Extract ROI based on
projected position

Figure 2: ROI Extraction Overview

The ROI is extracted from images (Step 4) according to the coordinates calculated in Step 3. Note that the ROI extraction process tolerates a certain level of numerical errors that occur in Step 1-Step 3 and extrinsic calibration error mentioned above. Figure 1 shows the specification of ROI with some margins. In the proposed system, the radius of each traffic light blob in the real world is assumed as 30 cm. By exploiting 3D pose of traffic lights, and vehicle-mounted camera on the 3D map, the radius of traffic light projected on image plane can be estimated ($r$ in Figure 1). To tolerate numerical errors, we adopt $1.5r$ margins from the projected edge of traffic light. An overview of the ROI extraction technique used in the proposed system is shown in Figure 2.

### *3.3. Morphology Processing*

Morphology processing manipulates and analyzes the brightness values of pixels in the obtained ROI as explained in Section 3.2.

First, images in RGB color space are converted to HSV color space. HSV color space is more closely related to human chromatic sensation than RGB color space; thus, this conversion makes determining color value thresholds easier. A mask image is generated using H, S, and V threshold conditions set for each traffic light's color to extract regions that include a colored light. The generated candidate regions can contain areas that do not correspond to a traffic light because some pixel blocks that satisfy the threshold conditions might not actually be traffic lights but other objects in images. We assume color traffic lights projected onto an image plane are round; therefore, an additional threshold condition using the degree of circularity is applied to each candidate region. The degree of circularity $R$ is expressed as follows:

$$R = \frac{4\pi \times S}{L^2} \tag{3}$$

where $S$ and $L$ denote the area and perimeter of the region, respectively. The degree of circularity represents the extent to which a region is circular (values closer to 1 indicate a region better depicting a circle). If several candidate regions remain after applying the degree of circularity threshold, the region with highest degree of circularity is selected and a mask to filter areas outside the region is applied. Table 1 shows the threshold values we decided experimentally for the morphology processing method. This mask image is overlaid on the input ROI to obtain the pixel values that are assumed to represent a traffic light. This recognition process infers the color state of a target traffic light by searching the most dominant color in the pixel values. Figure 3 shows the work flow up to this point.

### *3.4. Deep Learning based detector*

A deep CNN might be applied to obtain the location and class of the traffic signals. We used Single Shot MultiBox Detector (SSD) [18] for this purpose. SSD, was originally developed for

Table 1: Threshold values used in Morphology processing. Note that “H” is represented in the 0 to 360 range cyclically, while “S”, “V”, and “R” are in 0 to 1 linearly

| Light color | H | | S | V | R |
|---|---|---|---|---|---|
| | lower | upper | | | |
| Red | 340 | 50 | 0.37 | 0.55 | 0.75 |
| Yellow | 50 | 70 | | | |
| Green | 80 | 190 | | | |

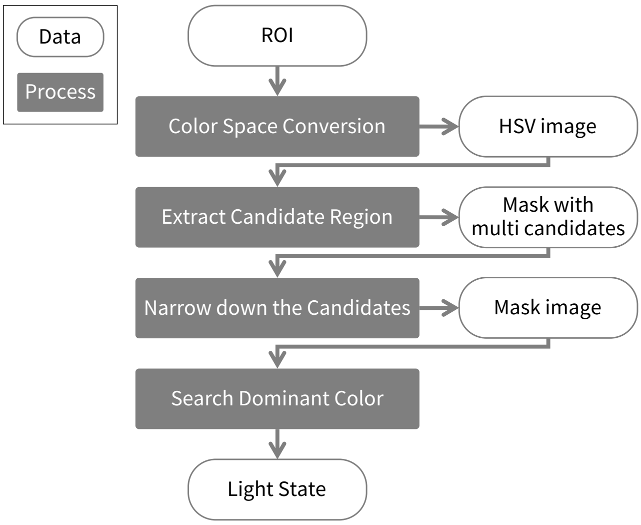


Figure 3: Morphology recognition flow

object detection, being a fully convolutional network, its execution time is fast. Moreover, SSD detection accuracy is comparable to other state-of-the-art object detector algorithms (e.g., the Faster R-CNN [19]), and work effectively with lower resolution images because it exploits hierarchical feature maps. While the shape of traffic lights that appear in the extracted ROI (Section 3.2) are almost similar, the SSD can distinguish each traffic light color state as a different object because color element weights are increased when sufficient samples are provided in the training phase. For these reasons, we applied this object detection algorithm to traffic light color state recognition.

As discussed in Section 1, traffic light recognition techniques in autonomous driving require fast processing speed in order to recognize traffic lights in front of a car driving at typical speeds, e.g., 60 km/h in Japan. High recognition accuracy is also required because the results have significant influence on vehicle control (e.g., deceleration). The size of the ROI image (Section 3.2) is reduced when traffic lights appearing at greater distances are projected to the image plane. It is preferable for a traffic light recognition algorithm to work with low resolution input in order to plan vehicle behavior well in advance. Therefore, relative to the previous discussions, we consider the SSD algorithm suitable for our purpose. The recognition flow with SSD is shown in Figure 4. Note that as SSD is an object detection algorithm, it outputs bounding boxes (e.g., location of objects) and classes, while our purpose is just classification inside the ROI. Therefore, we ignore all bounding boxes resulted from the SSD and adopt a class with highest detection score as a recognized state for the ROI.

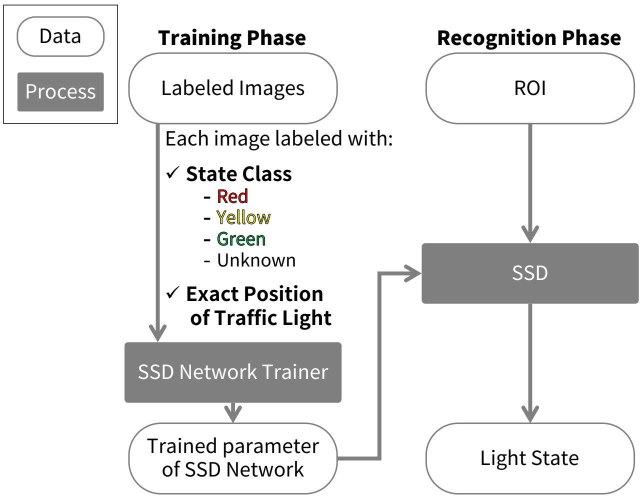


Figure 4: Flow of SSD recognition

## 4. Implementation

This section presents schemes for the traffic light color recognition described in Section 3.

### *4.1. Autoware Implementation*

We implemented all functions described in Section 3 using Autoware[1] [11], which is an open source research and development platform for autonomous driving based on the ROS [12].

[1] https://github.com/CPFL/Autoware

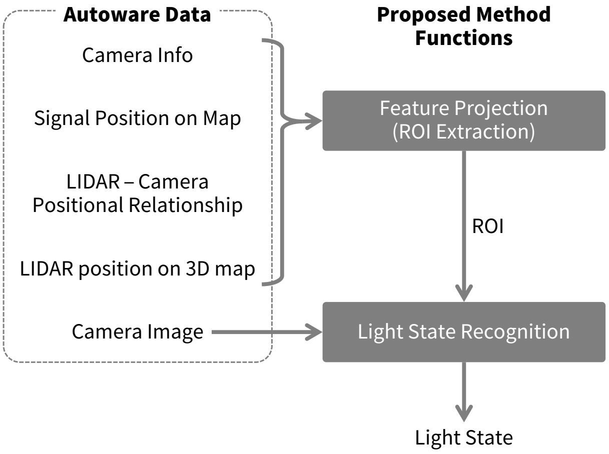


Figure 5: Overview of Data Connections between Autoware and the Proposed Method

Autoware modularizes functions required for autonomous driving as individual processes. ROS provides an efficient inter-process communication. It allows to send and receive user-defined data structures in a flexible manner.

The proposed traffic light recognition module employs the modules integrated in Autoware for localization, 3D Maps to achieve the expected functionality. Figure 5 shows an overview of the connections between data acquired by the Autoware modules and the proposed method's functionality.

As shown in the right side of Figure 5, the proposed method is divided in two parts (i.e., "Feature Projection" and "Light State Recognition"), which we implement as individual ROS nodes. The "Feature Projection" process performs ROI extraction using a 3D map and localization result (Section 3.2). This process receives data from Autoware, such as camera information (image size and focal length), traffic light coordinates in a 3D map, the positional relationship between the LIDAR and camera sensors, and the estimated LIDAR position in a 3D map. Then, it outputs ROI information, which indicates where traffic lights are supposed to exist in a given image.

The "Light State Recognition" process corresponds to the morphology processing (Section 3.3) and deep learning (Section 3.4) methods. These nodes receive the ROI information obtained by the Feature Projection process and images captured by the camera. The target region is extracted from the camera image according to the ROI information followed by the recognition process. Note that the format of the input and output data is uniform for re usability among all recognition methods that comply with the defined messages.

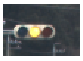
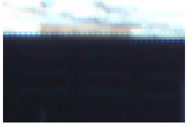
### *4.2. Color State Training and Recognition by SSD*

The Caffe [20]-based implementation[2] of SSD (Section 3.4) was used as base for our work. As shown in the "Training Phase" (Figure 4), the traffic light color states and traffic light position in each image are used as training data when SSD learns the network parameters. Note that the appearance of noise, such as other vehicle in front and roadside trees due to vibration while driving and occlusions are contained in an ROI acquired in a real autonomous driving environment. Therefore, in order to create training dataset, images obtained using vehicle-mounted camera during actual on-road driving experiments were corrected and labeled with color state and traffic light position information. Some training data examples are shown in Table 2. Despite the fact that the images used for training are ROI extracted from captured images during on-road experiments, we provided exact bounding box coordinate to enclose traffic light in ROI as well as color state to SSD training phase. This is similar to other general training schemes for object detection methods. It aims for SSD to learn features of traffic light without margin area included in extracted ROI. Figure 6 shows the training datasets and a breakdown of the color states considered in this paper.

Although a previous study [18] reported that an SSD algorithm can achieve a higher detection rate with low resolution input compared to state-of-the-art detection algorithm, such as Fast-RCNN and YOLO [21] algorithms, the smallest image resolution considered by that study was 300×300 pixels. However, a traffic light recognition system for autonomous driving must frequently deal with smaller resolution images (Table 2) for the reasons described in Section 3.4. We discuss evaluations of recognition accuracy transition using multiple input resolutions in Section 5.

[2] `https://github.com/weiliu89/caffe/tree/ssd`

Table 2: Training Data Examples

| Image | Size<br>(width × height) | State | Exact Light Position<br>(x, y, width, height) | Remarks<br>(Not used in training) |
|---|---|---|---|---|
| | 74 × 50 | Green | (15, 23, 57, 38) | – |
|  | 52 × 35 | Yellow | (5, 12, 36, 24) | – |
| | 76 × 50 | Red | (11, 19, 57, 36) | – |
| | 109 × 70 | Unknown | (31, 23, 93, 46) | Turned off due to camera frame rate and signal blinking frequency |
|  | 68 × 46 | Unknown | (0, 0, 68, 46) | Occluded by vehicle in front |

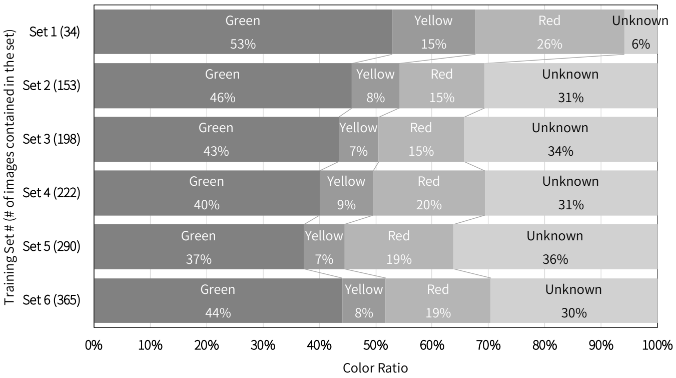


Figure 6: Breakdown of the training data set

### *4.3. Inter-frame Filter*

As traffic lights are designed for manual human visual assessment, we can assume that the color state does not change faster than a camera's frame rate. Moreover, general traffic lights in Japan change color state in order of green, yellow, and red. By considering these assumptions, we modify filters to reduce false recognition occurrences in a few frames [22]. These filters were designed to consider the order of color state changes. The final output recognition results do not change until a fixed number of the equal recognition results have been acquired by getting through the filter. Note that we created different filters for the morphology processing and deep learning recognizers (Section 3.3 and 3.4, respectively) because each method demonstrates different recognition tendencies.

Table 3 details the inter-frame filter for the morphology recognizer. With the morphology recognizer, the existence of all light colors (green, yellow, and red) is estimated independently. Here, a threshold-based assessment is primarily used in this process to estimate which color pixel blocks correspond to a traffic light. As discussed in Section 3.3, some blocks may fulfill the HSV and degree of circularity threshold conditions and eventually become a final candidate. In Table 3, combinations of the estimated existence for each light color in a current ROI are represented by the "GYR result of current frame". For example, "110" indicates that the input ROI contains regions that satisfy the conditions for luminous green and yellow light. The inter-frame filter returns the current color state by comparing the previous recognition result to the recognition result for the current frame.

Table 4 shows the inter-frame filter for the SSD recognizer. SSD is an end-to-end process that takes an image as input and outputs the recognition result; thus, its recognition results are limited to only four patterns: green, yellow, red, and unknown. In addition, the SSD recognizer can improve recognition ac-

Table 3: Inter-frame filter for morphology processing method

Each digit of "GYR result of current frame" turns 1 if there is a region that fulfills conditions for corresponding color (green, yellow, and red) in a current ROI

| | GYR result of current frame | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Previous result | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Green | Green | Unknown | Yellow | Yellow | Green | Green | Yellow | Unknown |
| Yellow | Yellow | Red | Yellow | Red | Unknown | Unknown | Yellow | Unknown |
| Red | Red | Red | Unknown | Red | Green | Red | Green | Unknown |
| Unknown | Unknown | Red | Yellow | Red | Green | Red | Yellow | Unknown |

Table 4: Inter-frame filter for machine learning recognition method

| | Result of current frame | | | |
|---|---|---|---|---|
| Previous result | Green | Yellow | Red | Unknown |
| Green | Green | Yellow | Yellow | Green |
| Yellow | Unknown | Yellow | Red | Yellow |
| Red | Green | Red | Red | Red |
| Unknown | Green | Yellow | Red | Unknown |

Table 5: Evaluation dataset

| | Daytime | Sunset |
|---|---|---|
| Driving duration | 596s | 328s |
| Image resolution | 1368 (w) × 1096 (h) | 1368 (w) × 1096 (h) |
| # of frames | 8946 | 4929 |
| # of extracted frames | 3347 | 1505 |
| # of target signals | 8 | 5 |
| # of state change | 6 | 4 |

curacy for an individual frame via learning whereas the morphology recognizer cannot. Thus, compared to the morphology recognizer's inter-frame filter, the SSD recognizer's inter-frame filter more upholds recognizer suggestion for current frame as the final result (e.g., the SSD recognizer's suggestion for current frame can be the final result more straightforwardly compared to morphology recognizer's suggestion).

## 5. Evaluation

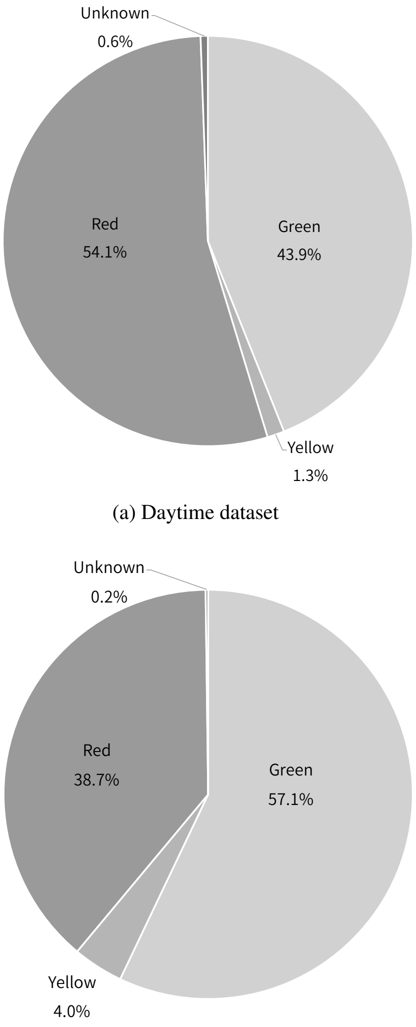
To quantify the performance of the proposed traffic light recognition method, the effects of the following factors were evaluated: 1. ROI extraction, 2. Distance from recognition target, 3. GPU usage, and 4. Number of images in the training dataset for the SSD recognizer.

### *5.1. Experimental Setup*

An Intel Core i7-6700K operating at 4.0GHz with four cores was used as the host processor. The GPU environment for program acceleration was an NVIDIA GeForce GTX 980Ti (CUDA version 8.0). We collected images captured by a vehicle-mounted camera on public roads as the evaluation dataset. Image collection was performed on several roads during the morning and at sunset. We created the evaluation dataset by extracting frames in which target traffic lights appear. An overview of the evaluation dataset is shown in Table 5. A breakdown of color states in each evaluation dataset is shown in Figure 7.

### *5.2. Effect of ROI Extraction on Recognition Accuracy*

Figure 8 compares the recognition accuracy obtained by the two recognition methods with the daytime and sunset datasets. "Accuracy" represents the number of recognition results that agree with the ground truth states of the evaluation dataset (expressed as a percentage). ROI extraction improves accuracy for nearly all dataset and recognition method combinations. There-



(a) Daytime dataset

(b) Sunset dataset

Figure 7: Proportion of ground truths in each evaluation dataset

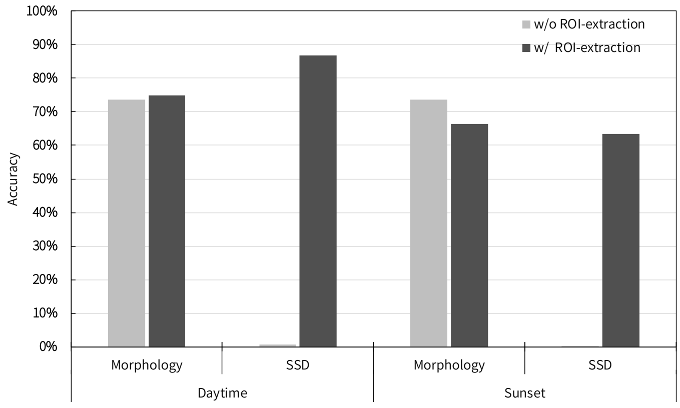
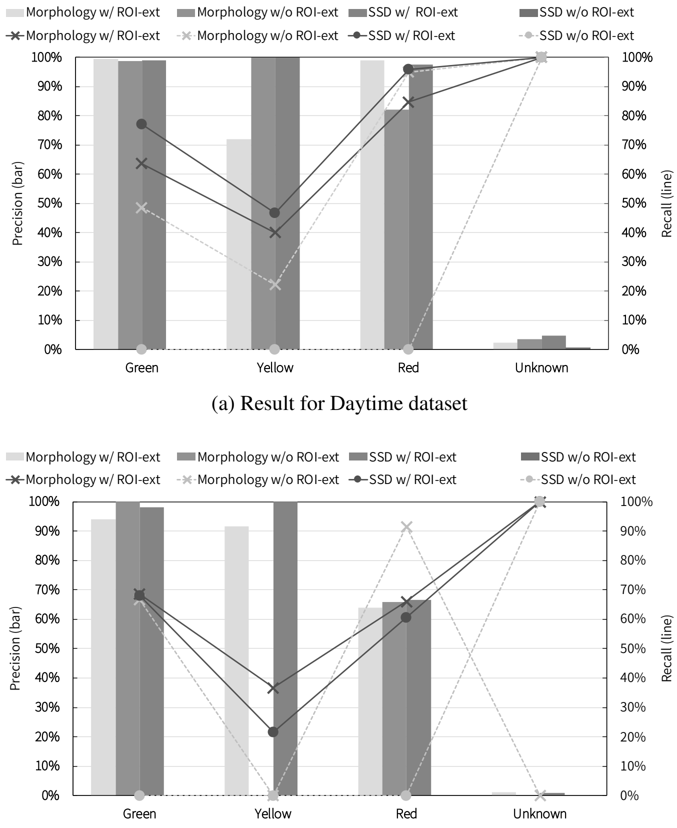


Figure 8: Effect of ROI extraction on recognition accuracy



(a) Result for Daytime dataset

(b) Result for Sunset dataset

Figure 9: Recognition precision and recall by color for each data set

fore, we conclude that confining a processed region by extracting an ROI benefits traffic light recognition. The accuracy of the SSD recognizer with the evaluation dataset was up to 86.9%, which is significant improvement. Note that Set 6 (Figure 6) was used for the following SSD recognizer evaluations unless otherwise noted.

Figure 9 shows the recognition precision (bars) and recall (lines). Here, “Precision” is the proportion of the number of correct results to all recognizer outputs, and “Recall” is the proportion of the number of correct recognitions to the total number of all images whose ground truth corresponds to the state in the evaluation dataset. Tables 6 and 7 show detailed information about the recognition results expressed as confusion matrices.

As shown in Table 6(d) and 7(d), the SSD recognizer without ROI extraction outputs “Unknown” for all inputs. We assume this was caused by using clipped out images to train the SSD network (Table 2). Note that the SSD recognizer without ROI

Table 6: Confusion matrices for each recognition method (daytime dataset)

(a) Morphology, w/ ROI-extraction

| | | Prediction | | | | |
|---|---|---|---|---|---|---|
| | | Green | Yellow | Red | Unknown | Recall |
| Ground Truth | Green | 936 | 0 | 17 | 518 | 63.6% |
| | Yellow | 6 | 18 | 0 | 21 | 40.0% |
| | Red | 0 | 7 | 1536 | 269 | 84.8% |
| | Unknown | 0 | 0 | 0 | 19 | 100.0% |
| | Precision | 99.4% | 72.0% | 98.9% | 2.3% | |

(b) Morphology, w/o ROI-extraction

| | | Prediction | | | | |
|---|---|---|---|---|---|---|
| | | Green | Yellow | Red | Unknown | Recall |
| Ground Truth | Green | 715 | 0 | 363 | 393 | 48.6% |
| | Yellow | 9 | 10 | 13 | 13 | 22.2% |
| | Red | 0 | 0 | 1721 | 91 | 95.0% |
| | Unknown | 0 | 0 | 0 | 19 | 100.0% |
| | Precision | 98.8% | 100.0% | 82.1% | 3.7% | |

(c) SSD, w/ ROI-extraction

| | | Prediction | | | | |
|---|---|---|---|---|---|---|
| | | Green | Yellow | Red | Unknown | Recall |
| Ground Truth | Green | 1132 | 0 | 44 | 295 | 77.0% |
| | Yellow | 11 | 21 | 0 | 13 | 46.7% |
| | Red | 0 | 0 | 1735 | 77 | 95.8% |
| | Unknown | 0 | 0 | 0 | 19 | 100.0% |
| | Precision | 99.0% | 100.0% | 97.5% | 4.7% | |

(d) SSD, w/o ROI-extraction

| | | Prediction | | | | |
|---|---|---|---|---|---|---|
| | | Green | Yellow | Red | Unknown | Recall |
| Ground Truth | Green | 0 | 0 | 0 | 1471 | 0.0% |
| | Yellow | 0 | 0 | 0 | 45 | 0.0% |
| | Red | 0 | 0 | 0 | 1812 | 0.0% |
| | Unknown | 0 | 0 | 0 | 19 | 100.0% |
| | Precision | 0.0% | 0.0% | 0.0% | 0.6% | |

Table 7: Confusion matrices for each recognition method (sunset dataset)

(a) Morphology, w/ ROI-extraction

| | | Prediction | | | | |
|---|---|---|---|---|---|---|
| | | Green | Yellow | Red | Unknown | Recall |
| Ground Truth | Green | 589 | 0 | 217 | 53 | 68.6% |
| | Yellow | 38 | 22 | 0 | 0 | 36.7% |
| | Red | 0 | 2 | 385 | 196 | 66.0% |
| | Unknown | 0 | 0 | 0 | 3 | 100.0% |
| | Precision | 93.9% | 91.7% | 64.0% | 1.2% | |

(b) Morphology, w/o ROI-extraction

| | | Prediction | | | | |
|---|---|---|---|---|---|---|
| | | Green | Yellow | Red | Unknown | Recall |
| Ground Truth | Green | 573 | 0 | 215 | 71 | 66.7% |
| | Yellow | 0 | 0 | 60 | 0 | 0.0% |
| | Red | 0 | 0 | 534 | 49 | 91.6% |
| | Unknown | 0 | 0 | 3 | 0 | 0.0% |
| | Precision | 100.0% | 0.0% | 65.8% | 0.0% | |

(c) SSD, w/ ROI-extraction

| | | Prediction | | | | |
|---|---|---|---|---|---|---|
| | | Green | Yellow | Red | Unknown | Recall |
| Ground Truth | Green | 585 | 0 | 141 | 133 | 68.1% |
| | Yellow | 11 | 13 | 36 | 0 | 21.7% |
| | Red | 0 | 0 | 353 | 230 | 60.5% |
| | Unknown | 0 | 0 | 0 | 3 | 100.0% |
| | Precision | 98.2% | 100.0% | 66.6% | 0.8% | |

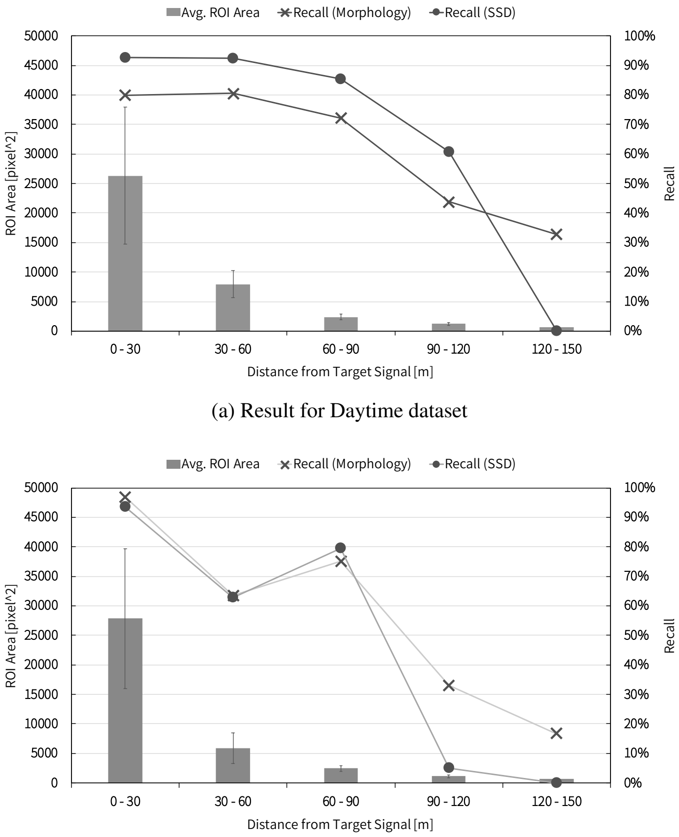
(d) SSD, w/o ROI-extraction

| | | Prediction | | | | |
|---|---|---|---|---|---|---|
| | | Green | Yellow | Red | Unknown | Recall |
| Ground Truth | Green | 0 | 0 | 0 | 859 | 0.0% |
| | Yellow | 0 | 0 | 0 | 60 | 0.0% |
| | Red | 0 | 0 | 0 | 583 | 0.0% |
| | Unknown | 0 | 0 | 0 | 3 | 100.0% |
| | Precision | 0.0% | 0.0% | 0.0% | 0.2% | |

extraction is omitted from subsequent discussion because all its outputs were “Unknown”. As a result, we cannot discuss its recognition tendency.

Figure 9(a) shows that all recognizers demonstrated a similar tendency, i.e., recall was lowest for the yellow state and increased in order of green and red states. The evaluation dataset inherently contains fewer yellow state images than the other states; thus, a moderate false recognition may have caused a significant reduction of the recall proportion. For the SSD recognizer, lower recall for the yellow state was presumably due to the fact that the training dataset included fewer yellow state images, as shown in Figure 6. We observe that the morphology recognizer without ROI extraction produced lower precision for the red state than the morphology recognizer with ROI extraction for the daytime dataset. In addition, by comparing the “Red” columns in Tables 6(a) and 6(b), we observe that the morphology recognizer without ROI extraction was predisposed to output red for any input. This tendency seems to show that not confining the processed region by ROI extraction caused false recognition because the recognizer considered non-traffic light regions (e.g., the tail lights of other vehicles) as red traffic lights.

In contrast, as shown in Figure 9(b), the recall of all recognizers was reduced with the sunset evaluation dataset. This may have been caused by pixel value saturation resulting from strong backlight from the sun. Even if sunlight did not enter the camera directly, significant HSV changes caused by automatic white balance correction triggered by an overly bright backlight could be another reason for such reduced recall. Regardless of whether ROI extraction was employed, the morphology recognizers are predisposed to incorrectly recognize the green state as the red state, which is suggested by a comparison of the "Green" rows in Table 7(a) and 7(b). This implies that the morphology recognizers have less flexibility relative to environmental changes, such as strong backlight. On the other hand, Table 7(c) shows that the SSD recognizer achieved higher precision for nearly all states than the morphology recognizer even though the input images were the same. It is conceivable that, if appropriate data are included in the training dataset, the SSD recognizer can also identify traffic lights using other information, such as shape and the order of colors when HSV values change slightly due to backlight. Thus, collecting a feasible training dataset is important for an SSD recognizer.



(a) Result for Daytime dataset

(b) Result for Sunset dataset

Figure 10: Recognition recall shift relative to distance from the target traffic light

### 5.3. *Effect of Distance from Target on Recognition Recall*

In Figure 10, the horizontal axis is the distance from the recognition target traffic light, the left vertical axis is the area (square pixels) of the extracted ROI, and the right vertical axis is recognition recall. Here, both recognizers (morphology and SSD) took ROI images as input.

As shown in Figure 10(a), the ROI area and recognition recalls for both recognizers increase with decreasing distance from target traffic light. Intervals closer than 120 m from the target, both recognizers yielded similar curb and converged to their own maximum recall value. In contrast, the morphology recognizer outperformed the SSD recognizer in terms of recall in the interval from 120 m to 150 m from the target. The mean area of the ROIs in this interval in the daytime dataset was approximately 673.5 square pixels, which means that each ROI had approximately 25 pixels per side. In this case, the target traffic lights were represented by only a few pixels and the available feature of the traffic lights was limited to color information only. The SSD recognizer seems to learn color information and features such as traffic light edge information. The recall inversion in the interval from 120 m to 150 m from the target was presumably caused by the lack of edge information and by confining the determination criteria to only color information.

In contrast to Figure 10(a), the morphology recognizer often outperformed the SSD recognizer relative to recalls for the sunset dataset (Figure 10(b)). A possible reason for this is error in traffic light projection to an image plane when extracting the ROI. Many traffic lights in the sunset evaluation dataset emerged after curves in the road. If such traffic lights are projected to an image plane while the vehicle was driving such curves, the entire traffic light was not captured in the ROI due to LIDAR-camera calibration error. In this case, the morphol-

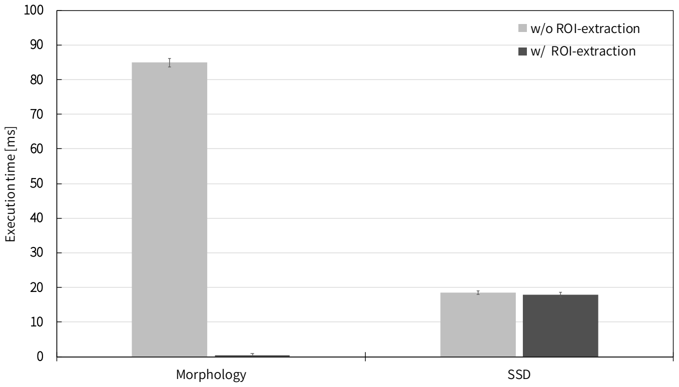


Figure 11: Effect of ROI extraction on recognition time

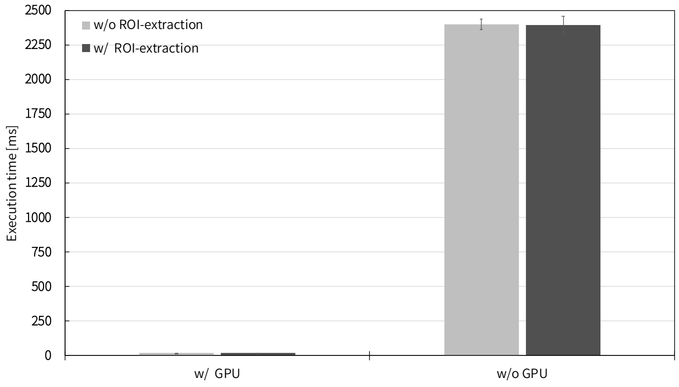


Figure 12: Effect of GPU on recognition time

ogy recognizer could recognize the traffic light state as long as glittering light was contained in ROI, while the SSD recognizer probably could not achieve the same result because it uses additional information (other than color). In the interval from 30 m to 60 m from the target, the recalls of both recognizers were reduced. This presumably occurred because the bright west sun broke through cloud cover and entered the camera in this distance interval. Moreover, some glittering lights were not contained in the ROI due to the ROI extraction error mentioned previously, which made it difficult for both recognizers to recognize traffic lights. Having more data presenting these cases might improve our model.

### *5.4. Effect of ROI extraction on Recognition Speed*

Figure 11 shows the recognition time of each recognizer with and without ROI extraction. The average recognition time of the morphology recognizer without ROI extraction was approximately 84.9 ms (approximately 0.51 ms with ROI extraction). For the SSD recognizer, the average recognition time was approximately 18.5 ms and 17.9 ms without and with ROI extraction, respectively. Note that the ROI extracting processing time is not included in these recognition times. The morphology recognizer improved recognition speed by approximately 166.5 times by using ROI extraction because it contains a process that depends on the input image's resolution, i.e., color conversion from RGB to HSV and searching contours. In contrast, the SSD recognizer demonstrated similar recognition times regardless of whether ROI extraction was employed. Presumably, this was due to the fact that the SSD recognizer takes fixed size images as input, which makes processing time independent of the input image's resolution because the input images are resized to fixed size.

### *5.5. Effect of GPU on SSD Recognition Speed*

The execution time of the SSD recognizer with GPU acceleration is compared to the execution times obtained without a GPU in Figure 12. The execution time of the SSD recognizer with ROI extraction and GPU acceleration was approximately 17.9 ms (18.5 ms without ROI extraction). On the other hand, if GPU acceleration was invalid, the execution times of the SSD recognizer for each setting were approximately 2392.2 ms and 2397.9 ms. As mentioned in the previous section, the SSD execution time is not dependent on the input image resolution because resizing to fixed input resolution is performed. From this perspective, the SSD recognizer is an inherently fast representative algorithm. However, the experimental results indicate 133.6 times worse execution time when GPU acceleration is not valid. Therefore, GPU acceleration is absolutely necessary in terms of applying SSD to autonomous driving. In addition, the acceleration rate varies significantly according to the type of GPU used [23]. Thus, we must select appropriate GPUs for practical use in consideration of energy consumption and sufficient processing speed.

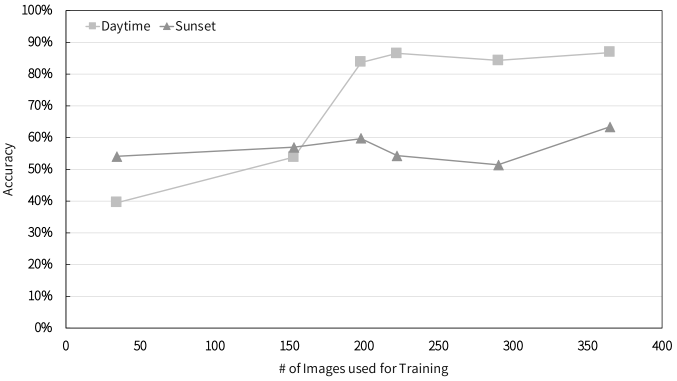


Figure 13: Recognition accuracy shift relative to the number of images in the training dataset

### 5.6. *Effect of the Number of Images in the Training Data on Recognition Accuracy*

Variations in recognition accuracy relative to the number of images in the training data for each dataset obtained by the SSD recognizer are shown in Figure 13. The six sample points correspond to the recognition accuracies of the results obtained with the training dataset shown in Figure 6. Roughly, a training dataset with more images results in greater accuracy. Accuracy improvement appeared to converge up to Set 5 (Figure 6), which contains 290 images. The training datasets for Sets 1 to 5 include images extracted from images recorded during daytime driving. In contrast, Set 6 contains both daytime images and images of traffic lights exposed to bright west sun light. Recognition accuracy improvement with both evaluation datasets was observed when we applied the training results obtained using Set 6 to the SSD recognizer. Thus, it follows that data diversity relative to how the recognition target appears in the training dataset and data quantity contribute to recognition accuracy.

### 5.7. *Discussion*

Fairfield *et al.* presented a method to recognize the color state of traffic lights that exploits a prior map of the 3D traffic light locations and camera images [10], which is a similar technique we have presented. They report that recognition precision and recall of their method are 99 % and 62 %, respectively. In contrast, the recognition precision of the proposed method achieved greater than 97 % for each color state (red, yellow, green) in the daytime dataset (Table 6(c)). Moreover, the recognition recall was approximately 90 % if the recognition targets were within 90 m (Figure 10(a)). While simple comparison does not make an absolute sense because evaluated datasets are different, the proposed method roughly achieved comparable level recognition precision and improved recognition recall by 1.4 times under favorable condition.

One limitation of our method is that it might not recognize the color state of traffic lights if the 3D positions are not previously known. Case in point, newly installed traffic lights not contained in the 3D map. The solutions to tackle this problem include a concept such as “connected vehicles”, which stands for vehicles that have the internet connection. If the information of newly installed traffic lights is updated on 3D maps in servers, connected vehicles can download the latest 3D maps information and recognize the color state of the newly installed traffic lights.

Another limitation is that the proposed method cannot work if features of traffic lights are not available due to pixel value saturation resulting from strong backlight from the sun. This is a general limitation for recognition methods using camera sensors. In order to avoid traffic accidents caused by this limitation, installing hierarchical connections between recognition modules and vehicle control modules is one solution. For example, even if pixel value saturation incidentally happens in a vehicle-mounted camera, the vehicle can stop behind other vehicles at an intersection by the help of other modules, such as collision avoidance modules, that uses other detection mechanisms such as LIDAR sensors. In this case, collision avoidance modules should be nearer hierarchy level to control modules than traffic light recognition modules.

## 6. Conclusion

This paper presented an image ROI extraction method for traffic lights based on current location on 3D maps. We also proposed two methods to recognize traffic light states on the extracted ROIs.

Two datasets, daytime and sunset, were obtained during public driving experiments. These were used to train a model and evaluate our methods. The experimental results indicate that ROI extraction improves recognition accuracy for both recognition methods; thus, providing a more reliable recognition. The SSD recognition method outperformed the morphology processing method in terms of recognition precision and recall in almost all states of the evaluation datasets. The SSD recognizer achieved more than 97 % average precision (i.e., red, yellow, and green) under favorable conditions. Moreover, if the recognition targets were within 90 m, the SSD recognizer achieved approximately 90 % recognition recall.

The proposed recognition methods can tolerate calibration and localization errors at some extent. However, when the error is too high, the classification is incorrect. To obtain better results, a different and more accurate calibration method should be considered [24–26]. Finally, increasing the number of images in the training dataset will improve the average precision and resilience to lighting conditions.

## Acknowledgment

This work was supported by JST CREST Grant Number JP-MJCR1683, Japan.

[1] Cabinet Office Japan, Annual Report on the Aging Society [Japanese full version] FY 2014, Japanese Government, 2014.

[2] Ministry of Economy, Trade and Industry, Japan, METI Journal, Japanese Government, 2014.

[3] Y. Mori, K. Eguchi, Traffic Light Recognition from the Video Image of Home Video Camera, Tech. rep., Aichi institute of Technology (2012).

[4] M. Omachi, S. Omachi, Traffic Light Detection with Color and Edge Information, in: Proc. of the IEEE International Conference on Computer Science and Information Technology, 2009, pp. 284–287.

[5] R. de Charette, F. Nashashibi, Traffic Light Recognition using Image Processing Compared to Learning Processes, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 333–338.

[6] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, M. M. Trivedi, Traffic light detection: A learning algorithm and evaluations on challenging dataset, in: Proc. of the IEEE conference on Intelligent Transportation Systems, 2015, pp. 2341–2345.

[7] C. Jang, S. Cho, S. Jeong, J. K. Suhr, H. G. Jung, M. Sunwoo, Traffic light recognition exploiting map and localization at every stage, Expert Systems With Applications 88 (2017) 290–304.

[8] A. Møgelmose, D. Liu, M. M. Trivedi, Traffic Sign Detection for U.S. Roads: Remaining Challenges and a case for Tracking, in: Proc. of the IEEE conference on Inteligent Transportation Systems, 2014, pp. 1394–1399.

[9] Y. Yu, J. Li, C. Wen, H. Guan, H. Luo, C. Wang, Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data, ISPRS Journal of Photogrammetry and Remote Sensing 113 (2016) 106–123.

[10] N. Fairfield, C. Urmson, Traffic Light Mapping and Detection, in: Proc. of the IEEE International Conference on Robotics and Automation, 2011, pp. 5421–5426.

[11] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, H. Tsuyoshi, An Open Approach to Autonomous Vehicles, IEEE Micro 35 (6) (2015) 60–68.

[12] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, ROS: an open-source Robot Operating System, in: Workshop on Open Source Software of the IEEE International Conference on Robotics and Automation, Vol. 3, 2009, p. 5.

[13] T. Foote, tf: The transform library, in: Proc. of the IEEE International Conference on Technologies for Practical Robot Applications, 2013, pp. 1–6.

[14] HERE Global B.V., HERE HD Live Map, `https://here.com/en/file/8596/download?token=EFFgFGKC` (2016).

[15] P. Biber, W. Straßer, The normal distributions transform: A new approach to laser scan matching, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003, pp. 2743–2748.

[16] E. Takeuchi, T. Tsubouchi, A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 3068–3073.

[17] M. He, H. Zhao, J. Cui, H. Zha, Calibration method for multiple 2d LIDARs system, in: Proc. of the IEEE International Conference on Robotics and Automation, 2014, pp. 3034–3041.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single Shot MultiBox Detector, in: Proc. of the European Conference on Computer Vision, 2016.

[19] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in: Proc. of the Advances in neural information processing systems, 2015, pp. 91–99.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, in: Proc. of the ACM international conference on Multimedia, 2014, pp. 675–678.

[21] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, in: Proc. of the IEEE Conference

on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[22] I. Orlov, Trafficlight, `https://github.com/igororlov/TrafficLight`.

[23] M. Hirabayashi, S. Kato, M. Edahiro, K. Takeda, S. Mita, Accelerated Deformable Models on GPUs, IEEE Transactions on Parallel and Distributed Systems 27 (6) (2016) 1589–1602.

[24] G. Pandey, J. R. McBride, S. Savarese, R. M. Eustice, Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information, in: Proc. of the AAAI Conference on Artificial Intelligence, 2012, pp. 1293–1300.

[25] A. Geiger, F. Moosmann, Ö. Car, B. Schuster, Automatic Camera and Range Sensor Calibration using a single Shot, in: Proc. of the IEEE International Conference on Robotics and Automation, 2012, pp. 3936–3943.

[26] D. Scaramuzza, A. Harati, R. Siegwart, Extrinsic Self Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 4164–4169.