Journal of Computational Design and Engineering, 2022, 9(2), 296–309

https://doi.org/10.1093/jcde/qwab080
Journal homepage: www.jcde.org

RESEARCH ARTICLE

Traffic control hand signal recognition using convolution and recurrent neural networks

Taeseung Baek[1] and Yong-Gu Lee[1,2,*]

[1]School of Mechanical Engineering, Gwangju Institute of Science and Technology (GIST), 123 Cheomdangwagi-ro, Buk-gu, Gwangju 61005, Republic of Korea and [2]Artificial Intelligence Graduate School, Gwangju Institute of Science and Technology (GIST), 123 Cheomdangwagi-ro, Buk-gu, Gwangju 61005, Republic of Korea

*Corresponding author. E-mail: lygy@gist.ac.kr

Abstract

Gesture understanding is one of the most challenging problems in computer vision. Among them, traffic hand signal The lack of available datasets is also a serious problem. Most classifiers approach these problems using the skeletons of target actors in an image. The variance in the time length of gestures mixed with random pauses and noise is handled with a recurrent neural network (RNN). Furthermore, . We constructed a hand signal dataset composed of 100 thousand RGB images that is made publicly available. We achieved correct recognition of the hand signals with various backgrounds at 91% accuracy. A processing speed of 30 FPS in FHD video streams, which is a 52% improvement over the best among previous works, was achieved. Despite the extra burden of deciding the validity of the hand signals, this method surpasses methods that solely use RGB video streams. Our work is capable of performing with nonstationary viewpoints, such as those taken from moving vehicles.

Keywords: autonomous driving; traffic control hand signal; deep learning

1. Introduction

Korean road traffic law clearly states that when a traffic signal). This regulation imposes an interesting situation for self-driving cars. Nevertheless, because permitted self-driving vehicles must follow the road traffic law, they must also be able to understand the hand signals from the traffic controller. Because self-driving cars move at high speeds, the recognition of hand signals must be

Received: 5 August 2021; **Revised:** 11 December 2021; **Accepted:** 15 December 2021

performed in real time. Many methods that rely on the computation of skeletons require high computational load and are not . The use of depth sensors can reduce this computational load, but this approach requires expensive devices. Therefore, we propose a method that requires only an RGB camera. The proposed method does not require the computation of skeletons, and it is designed to achieve a speed greater than 30 FPS. The increase in speed is achieved by simplifying the computational model.

However, these methods required the preprocessing of video streams to extract skeletons, and this extra burden reduced the overall processing time. Furthermore, previous works were applied to videos taken indoors or videos with a limited number of backgrounds. Because we cannot expect that the recognition of hand signals will be conducted in a controlled environment, these datasets are not suitable to generalize the trained neural network to real-world problems. Certain works use different types of sensors than video cameras.

however, these works always assume that the actor is giving a meaningful commanding hand signal. In real situations, a police officer may not give any commanding signals, or he or she may be directing a hand signal to other cars and not the observer.

The detection accuracy is significantly affected by the ability to distinguish between situations in which signals are given and situations when no intentional signals are made.

Korean police officers direct their hands to the targeted driver before giving the driving direction order. Distinguishing meaningful combinations of basic commands in hand gestures will enable the classifier to successfully understand the hand signal given by the police officer.

Furthermore, and amplitude. Therefore, hand-signal classification must be capable of handling inaccurate hand signals.

In this work, an RGB detector and recurrent neural network (RNN) were employed for hand signal recognition. The flag sequence algorithm was selected to assess the validity of the hand signals. The proposed method does not require the extraction of skeletons, which significantly reduces the processing time. A dataset for hand signals was collected outdoors with various backgrounds in consideration of real scenarios. Our work requires no additional sensor types in addition to RGB monocular cameras. This simple approach greatly reduced the required processing time.

2. Related Work

Human action recognition has been actively investigated. In ActionXpose (Angelini *et al.*, 2020), human poses monitored in CCTV video were classified with long-short term memory (LSTM) and one-dimensional (1D) convolution neural network (CNN). Strong attention was given to the work that introduced time-based three-dimensional (3D) CNN (Ji *et al.*, 2013; Tran *et al.*, 2015). However, these methods are not suitable for processing complex actions composed of multiple small actions, similar to Korean traffic hand signals. Certain works used sensors attached to the actors.

The works by these four research groups (Zhang *et al.*, 2011; Iravantchi *et al.*, 2019; Neacsu *et al.*, 2019; Skaria *et al.*, 2020) target either the motions of the fingers or the movements from the elbows to the fingers. Therefore, they are not suitable for detecting traffic hand signals that require monitoring of both arms. Attaching intrusive measurement sensors to the actor (police officer) imposes many weaknesses. .

Hand signals must be interpreted by the driver or a self-driving . All police officers need to be equipped with costly accelerometers and communication devices.

These techniques are non-intrusive and require no additional costs other than the cost of the digital camera attached to the car. No additional equipment is necessary on the actor side. The classifier can be realized in a self-driving car with low-cost digital cameras.

Wiederer *et al.* (2020) employed 2D body pose estimation, RNNs, attention networks, and graph CNNs. They achieved 87.37% accuracy. Li and Yang (2018) applied the L-K optical flow method and key-frame-based image pyramid. They achieved 95% accuracy. Furthermore, convolution pose machines and LSTM (by He *et al.*, 2020; accuracy of 93.29% with 17.2 FPS), the n-frame cumulative difference and cumulative block intensity vector (by Sathya & Geetha, 2015; accuracy of 96.24%), the use of frame partitioning to detect the entry and exit of the hands in the prescribed region (by Varshney *et al.*, 2020), motion frequency images (MFIs) and motion history images (MHIs) (by Wang & Chong, 2014; accuracy of 81.44% with 17.2 FPS) have been employed in research. However, these studies have limitations. Extra work is needed to construct the skeleton from body pose estimations, and this shortcoming cancels the benefit of not using depth sensors. Some works cannot identify Korean traffic hand signals that are composed of various combinations of motions. High accuracy was only achieved for simple backgrounds or when the distance between the police officer and the camera was constant.

Some works utilize depth information in addition to color information. The time variant movements of joints from the

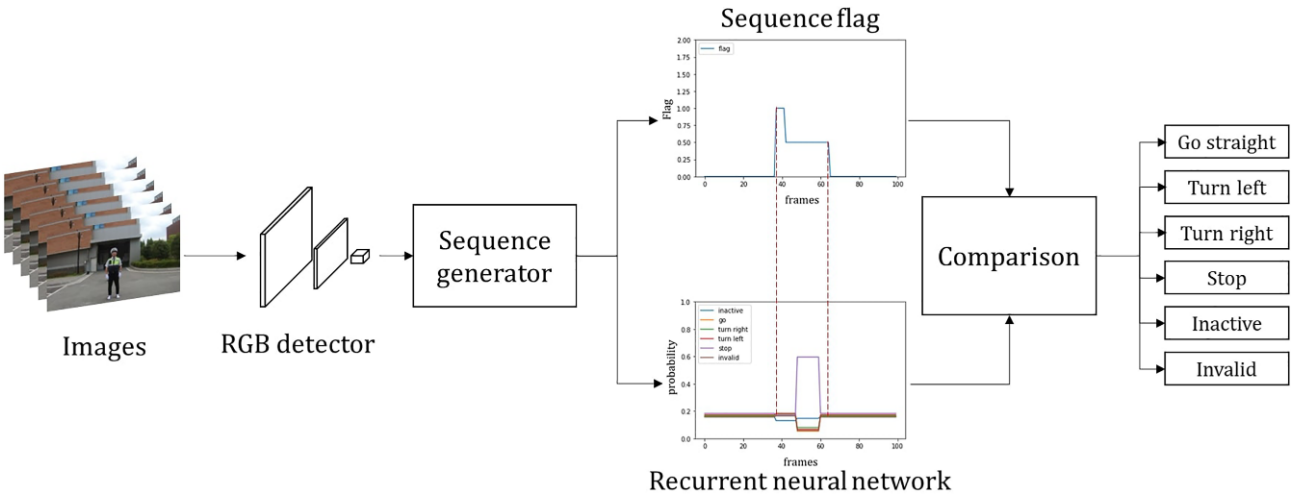

Figure 1: Overview of the proposed hand signal classifier. The RGB detector localizes the traffic controller, and the controller's arm directions are calculated from the input video stream. Arm directions are codified and concatenated by the sequence generator alongside the sequence flag that denotes whether the controller is gazing toward the camera. The flag and output of the RNN are compared to classify the hand signal.

extracted skeletons were analysed through support vector machines (Le *et al.*, 2012) or neural networks (Linqin *et al.*, 2017; Ma *et al.*, 2018; Wang & Ma, 2018). A dynamic descriptor from the MHI (Guo *et al.*, 2017) was also utilized. For the case of a classification accuracy of 97% (Linqin *et al.*, 2017), researchers have not demonstrated whether their work can be successful in natural outdoor conditions. Because they applied the dataset obtained inside a building, it is unclear whether their work can achieve high accuracy outdoors. For the case of a classification accuracy of 96.67% (Ma *et al.*, 2018), it is unclear whether their method can be extended to data obtained outside since their training data were obtained from a virtual city traffic scene. Wang and Ma (2018) did not use traffic hand signal data to identify motion recognition; instead, they applied datasets for gesture recognition from ChaLearn (Wan *et al.*, 2016).

Most of the classification methods that we have discussed

In general, these methods employed direct 3D information from depth sensors and computed skeletons as features. Depth sensors are costly, and the computation of skeletons imposes recognition speed degradation. The recognition speeds of methods that rely on the calculation of skeletons range from 1 FPS (Guo *et al.*, 2017), 17.2 FPS (He *et al.*, 2020), and 17.23 FPS (Wang & Chong, 2014). These numbers are certainly not real-time value in the computer vision community, where 30 FPS is the consensus of real-time performance. In addition to using hand signals to convey messages typically given by traffic control signals, other means of sending commanding signals exist. For example, we can use voice commands. However, voice commands can convey much richer contexts, and several weaknesses need to be solved. Although voice recognition has recently received much attention, mainly due to the advancement of natural language processing, the current state of voice recognition is very weak in outdoor situations with many disturbing sound sources. Hand signals can work at distances as far as 100 meters, but voice commands operate only in much closer spaces. Due to these limitations, we believe that hand signals are more suitable for sending traffic controls.

pealing to add depth sensors. The most widely utilized depth sensors are stereo cameras, time of flight (ToF) cameras, and LIDAR. To equip a car with a stereo camera, an additional cost of $1300 is needed. In the case of LIDARs, a 16-channel LIDAR costs $1200 and a 32-channel LIDAR costs $15 000. Compared to these price tags, monocular RGB cameras are much cheaper. The ToF cameras on handheld devices are modularized and more affordable, but they are not suitable because of their low resolutions and shorter working distances (~10 meters). Second, our method does not require the computation of skeletons. Instead, we only use 2D bounding box detections. Most previous works rely on the use of body skeletons that require a dedicated stage of computations. Third, the time needed for the classification of hand signals is much faster than that of previous works.

3. Method

Figure 1 illustrates an overview of the proposed method. The input video is processed frame by frame using a detection algorithm based on CNN. The police officer is localized, and the pose of the arm is detected. The sequence generator concatenated the directions of the poses into a sequence. The sequence is sent to the RNN for classification. Concurrent to the timestamps of the generated sequence, a stream of flags denoting whether the gazing direction of the traffic controller is facing toward the camera is also generated. The two sequences, namely the flag sequence and the hand direction sequence, are compared to classify the hand signal. We will discuss the nature of hand signals, preparation of the datasets, and detailed operational steps in the following section.

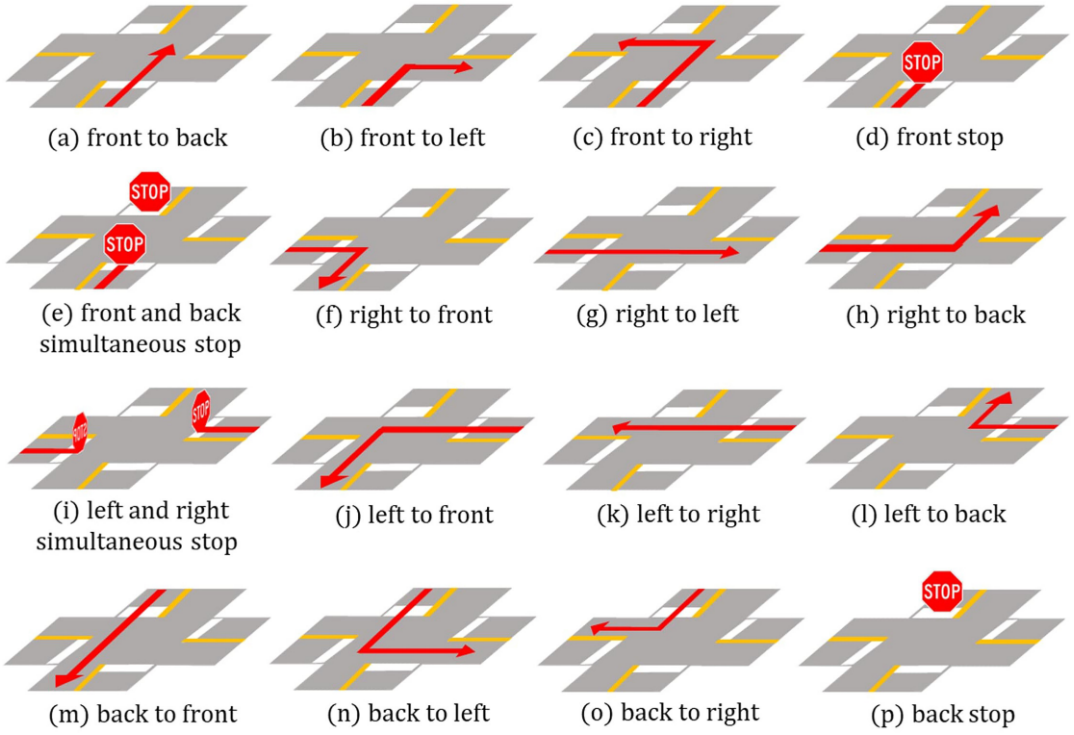

Figure 2: Sixteen traffic hand signals dictated by the Korean Central Police Academy. The hand signal is seen from the point of view of the police officer. The signals to the driver's point of view can be seen as go-straight, turn-left, turn-right, stop, and signals designated to the other driver.

3.1. Characteristics of police traffic control gesture

In this section, we will discuss the traffic control gestures in the

Understanding these gestures is crucial for designing the feature space in the neural network. Hand traffic signals are performed with arms to control the traffic flow of cars. The paper published by the CPA formally describes the standard rules for traffic control gestures. According to the paper, there are sixteen gestures to be used for regulating traffic flow. Figure 2 illustrates the meaning of each gesture.

The preamble is presented by an officer who leads their arm to the designator. Figure 3a exemplifies this two-way logic for the left-turn command signal.

Figure 4 shows the seven arm directions employed by the police officer. The arm motions are classified as (1) frontward, (2) backward, (3) toward-left, (4) toward-right, (5) toward-diagonally, (6) upward, and (7) downward. Based on this classification, the command illustrated in Fig. 3a can be codified as (7)-(1)-(3)-(7). All other commanding signals can be defined using this coding rule. We can apply the same rule to obtain the codified sequence for other cases. For example, Fig. 3b is designated to the driver in a different targeting direction. This command signal can be codified as (7)-(3)-(2)-(7). In general, the command signal that is meaningful to the driver must start with the officer correctly hailing the driver. This correct handshaking between the officer and the driver has a vital role in recognizing the hand signal. A total of 16 cases codified by the proposed rule are given in Table 1. In the table, the motion of directing the arm downward, which is always performed when returning to the neutral standing position, is omitted because it has no effect on the hand signal.

In Table 1, "Signal" refers to the 16 signals given in Fig. 2, and the "Sequence of arm directions" are described by the designated arm directions. The downward direction is omitted for brevity. "Validity" refers to the meaning of the command signal conveyed to the driver, who is directly facing the direction of the toes of the police officer. The cases shown in Table 1 (a) through (e) are valid hand signal commands conveyed to the driver who is facing the police officer: (a) go-straight, (b) turn-right, (c) turn-left, (d) stop, and (e) stop. The hand signals given from (f) to (p) are directed to the side and back of the police officer and are deemed "invalid" to the driver positioned in front of the police officer. As previously described, the driver should be attentive to the commanding sequence that starts with the front forward direction. All other command sequences that start with other directions should be disregarded. We should pay attention to the commanding sequence directed in the forward direction. An important question is what happens if the driver is facing the police officer toward their side or back. In these situations, the

(a) Valid signal for front drivers

(b) Invalid signal for front drivers

Figure 3: Left-turn command as seen by the police officer. (a) A command is given to the viewer, and (b) the officer is giving a left-turn command to the driver who is located to the right of the police officer.

driver must first recognize the intended direction of the police officer. Subsequently, they must pay attention to the following directions of the arms of the police officer only if the previous directions were designated to the driver.

In summary, all valid signals to the driver, irrespective of their position in relation to the police officer, can be understood by a sequence of arm directions given at signal (a) ∼ signal (d). We also note several interesting points. The downward state is a dummy state, and no meaning is given. When the downward state changes to the frontward state, it triggers a change from the dummy state to the valid signaling state. In contrast, if the downward state changes to other states (except the frontward state), it means that the command is given to the other driver. The command sequence then becomes invalid.

Furthermore, irrespective of the previous state, if the direction of the arm changes to the downward state, it implies termination of the hand signal.

In summary, we described the basic flow of logic that states that the validity of hand signals can be judged by looking at the directions of the arms. Table 6 summarizes the results; these concepts will be applied in the sequence flag in Section 3.4.3. We selected six hand signals in total: four meaningful signals (go-straight, turn-right, turn-left, and stop; +4), with an invalid signal that is directed to a target other than the observer (+1) and an inactive signal when the officer is in rest (+1).

3.2. Dataset

In this section, we explain the two datasets employed in the experiment – the dataset for direction detection and the dataset for the action classifier.

3.2.1. *Dataset for RGB detector*

Many previous types of research on hand signals involved the collection of datasets. Eight Chinese traffic hand signals were

contrast to the notion that most previous researchers have utilized RGBD cameras to collect their datasets, we collected our dataset with only RGB cameras. The traffic controller works in visible weather conditions. Under less favorable weather conditions, such as nighttime and snowy and rainy conditions, the use of a lighted traffic wand is recommended. For these reasons, we have established a dataset for only clear and cloudy weather. We recorded our video clips at 14 intersections. Hand signals are mostly performed in low-speed city traffic conditions. Therefore, we collected data from cameras attached to tripods. Table 2 summarizes the detailed requirements of the recordings.

shown in Table 3, each video belongs to one hand signal gesture of 16 possible classes. All frames of the training video were carefully labeled with a class based on the hand direction and bounding boxes tightly surrounding the police officer. Figure 5 illustrates the images correctly labeled according to the arm directions. Training the neural network with these labels resulted in incorrect identification of the arm direction without 3D information. We selected the uniforms of Korean police officers.

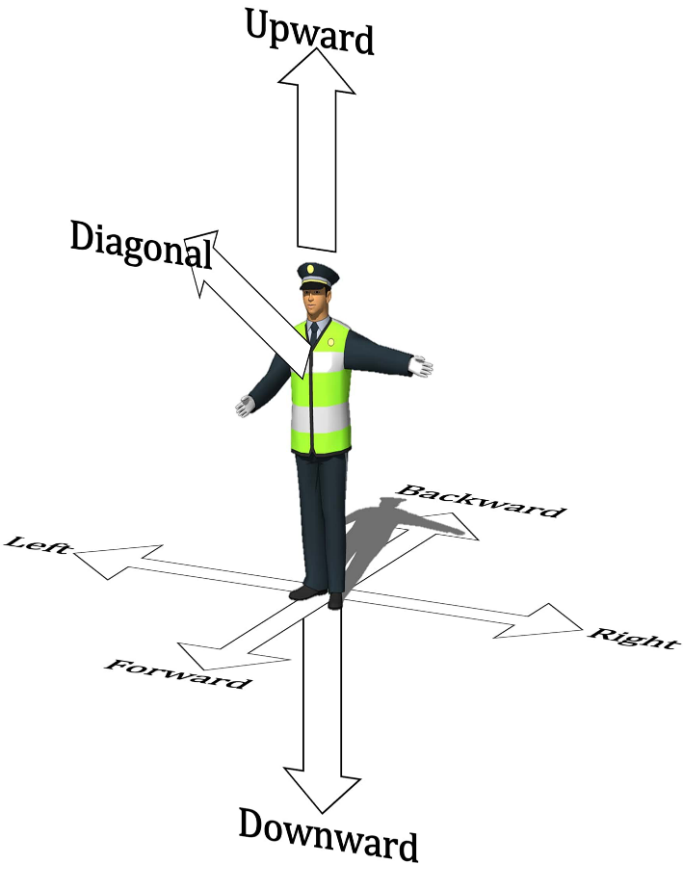

Figure 4: Seven arm directions employed by the police officer. The directions are portrayed as they are seen by the viewer in front of the police officer.

Table 1: Sequences of traffic control hand signals.

Signal	Sequence of arm directions	Validity
(a) front to back	frontward – upward – backward	go straight
(b) front to left	frontward – right	turn right
(c) front to right	frontward – left	turn left
(d) front stop	frontward – diagonal	stop
(e) front and back simultaneous stop	frontward – diagonal	stop
(f) right to front	left – frontward	invalid
(g) right to left	left – upward – right	invalid
(h) right to back	left – backward	invalid
(i) left and right simultaneous stop	left and right	invalid
(j) left to front	right – frontward	invalid
(k) left to right	right – upward – left	invalid
(l) left to back	right – backward	invalid
(m) back to front	backward – upward – frontward	invalid
(n) back to left	backward – right	invalid
(o) back to right	backward – left	invalid
(p) back stop	backward	invalid

Diverse backgrounds were applied to ensure that the training was general in nature. The Korean traffic control signal dataset is available from the following Uniform Resource Locator (URL) "data.go.kr/data/15075814/fileData.do" run by the Korean government.

3.2.2. *Dataset for RNN classifier*

The time-varying motions of the arm directions are sent to the RGB detector. The labels of the images represent the input to the RGB detector. The labels are codified and concatenated as a sequence. For example, the "front-to-back" hand signal is converted and assembled as follows: First, the entire sequence of labels is collected as a stream, e.g. "downward – frontward – upward – downward." The RNN for the classification processes the sequence. The labels are then translated by the dictionary shown in Table 4 as 00…11…55…00. The dictionary consists of eight directions and code pairs. Using this procedure, we were able to obtain 1600 sequence data points from the test dataset. To enrich the dataset, we generated an additional 7200 sequence data for the basic motions, as shown in Table 5.

Table 2: Environments and specifications for the dataset.

Environments	Specification	Environments	Specification
Actors	9	Camera	GoPro HERO6
Weather	Sunny, cloudy	FPS	30
Time	Daytime	Resolution	3840∗2160
Period	10 days	Field of view	118.2°
Distance to actor	2 m, 3 m, 5 m, 10 m	Height from ground	1.2 m

Table 3: Dataset for traffic control hand signals.

Signal	Test video	Labeled video	Labeled image
(a) front to back	149	100	8526
(b) front to left	172	100	5813
(c) front to right	149	100	5895
(d) front stop	148	100	5775
(e) front and back simultaneous stop	150	100	8978
(f) right to front	164	100	6134
(g) right to left	142	100	6761
(h) right to back	152	100	6474
(i) left and right simultaneous stop	152	100	6850
(j) left to front	151	100	5966
(k) left to right	129	100	6810
(l) left to back	126	100	6180
(m) back to front	137	100	6527
(n) back to left	135	100	6244
(o) back to right	141	100	6182
(p) back stop	110	100	5542
Total	2317	1600	104 657

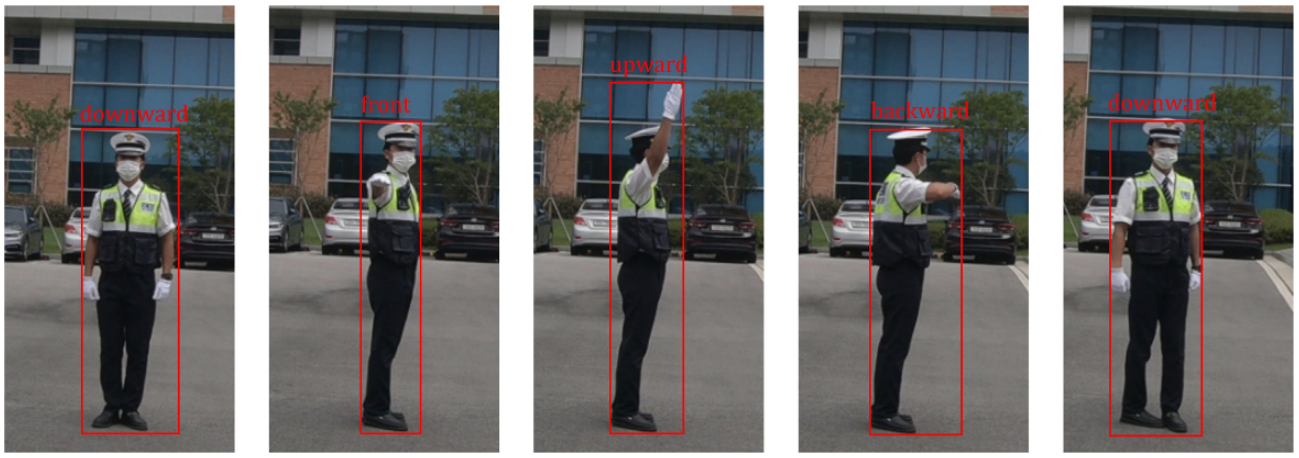

Figure 5: Stopped motions of the "front to back" traffic signal. Note that the labeling shows the direction of the arm.

Table 4: Labels for each direction of arm.

Label	Direction
0	downward
1	frontward
2	backward
3	left
4	right
5	upward
6	diagonal
7	left and right

3.3. RGB detector

The RGB detector is based on YOLOv4 (Bochkovskiy *et al.*, 2020). A self-driving car must be capable of making driving decisions while recognizing hand signals in real time. For the hand signal classifier to make accurate and quick decisions, high performance of the RGB detector in terms of speed and accuracy is necessary. YOLOv4 exceeds the performance of its predecessor by incorporating modern deep learning advancements that include a bag of freebies and a bag of specials. Furthermore, because it uses spatial pyramid pooling, YOLOv4 is resilient to input size variations instead of other object detection networks.

Table 5: Sequence for RNN.

Signal	Change of directions	Example sequence	Number of sequences
go straight	1–5–2	1111155555522222	1200
turn right	1–4	1111111444444444	1200
turn left	1–3	1111111111133333	1200
stop	1–6	1111666666666666	1200
invalid	Do not start with 1	3333333333222222 4444444555553333	1200
inactive	0	0000000000000000	1200
	Total		7200

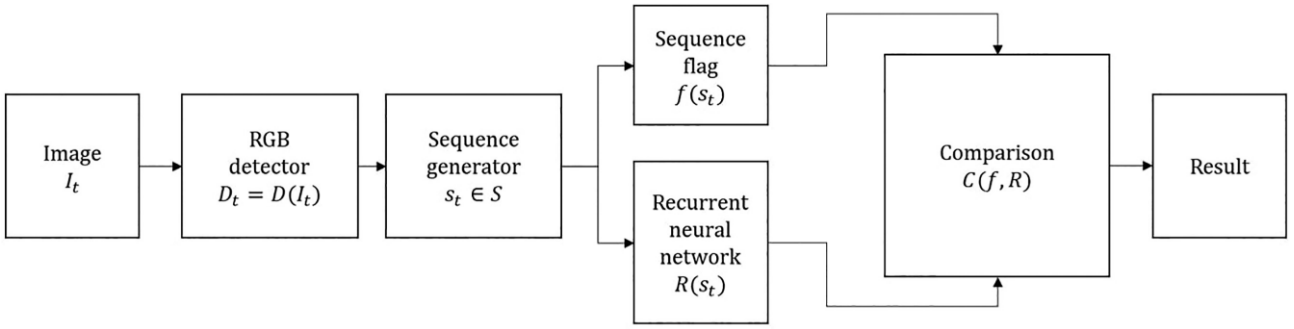

Figure 6: Flowchart of the proposed method.

YOLOv4 is being applied in many detection applications; we chose it because of its real-time speed. We randomly split the image dataset into 8:2 for training and testing and trained the detector with eight classes for the traffic controller and one class for the pedestrians. The training for the detector was conducted on four Tesla V100s. The training time consumed 7 days, and the testing time took 3 hours.

The hyperparameters are listed as follows: 80 epochs, IoU threshold of 0.5, and a batch size of 64 were employed for the training in four Tesla V100s. The resulting accuracy was 91.3%.

3.4. Algorithm

Figure 6 shows a flowchart of the algorithm. The algorithm assumes that the traffic controller gives a traffic hand signal.

r

The RGB detector D inputs image I_t at time stamp $t \in \mathbb{N}$ and outputs $D_t = D(I_t)$. The output result $D_t \in (0, 7)$ denotes one of the arm directions shown in Table 4.

r

The sequence generator is responsible for codifying the result of the RGB detector. The codified sequence is then further processed by the RNN. The sequence of length l, $s_t = \langle D_{t-l}, D_{t-l+1}, \cdots, D_{t-1} \rangle$, is assembled by stacking the output of the RGB detector. The sequence examples listed in Table 5 correspond to $l = 16$. The sequence set is denoted by symbol S. If there are no values before frame D_t of set s_t, we assigned a value of 0 to complete the sequence of length l. The sequence generator stacks the newly detected code from the RGB detector to the end of the sequence and pops the beginning of the stack to ensure that the length of the sequence is l. The sequence is input to the RNN with a time-matched sequence flag.

Previous works have not discussed how to discern the active state of hand signals or the beginning and end of hand signals. They assumed that the hand signals were always directed to the observer and that the video clips always conveyed a valid hand signal. We believe these assumptions can be a severe weakness of the classifier because, in natural traffic conditions, the police officer can be giving signals to a third party or can be giving no hand signals. To address such situations, our method is only attentive when the police officer has selected the viewer and when the hand signals are actively meaningful. This task is achieved by identifying the beginning and end of the command signal and the assessment that tests for the validity (sequence flag) of the hand signal. A sequence flag decides whether there is a hand signal given by the officer, or it can also assess the validity of the hand signal. Furthermore, the starting and ending moments of the hand signal can be obtained with the sequence flag. These moments are used by comparison with the detection result of the RNN, R(s_t), to increase the accuracy of the hand-signal classification.

Sequence flag f is determined by two previous subsequences referred to as the first subsequence $s_t' = \langle D_{t-n-m-l}, D_{t-n-m-l+1}, \cdots, D_{t-n-l-1} \rangle$ and second subsequence $s_t'' = \langle D_{t-m-l}, D_{t-m-l+1}, \cdots, D_{t-l-1} \rangle$. $n < l$ is the length of the first subsequence, and $m < l$ is the length of the second subsequence. The current sequence can be written as $s_t = \langle s_t', s_t'', D_{t-l}, D_{t-l+1}, \cdots, D_{t-1} \rangle$. The two subsequences reside before the current subsequence. $M_0' \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ and $M_0'' \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ are the mode of s_t' and mode of s_t'', respectively. Using the two modes and Table 6, Algorithm 1 is applied.

From Lines 1–2, we compute the modes of M_0' and M_0''. Lines 3–9 are used if the mode of the first subsequence M_0' is zero. When M_0'' is zero, it refers to the case when the officer is putting their arms downward, and it corresponds to the inactive signal f_{inactive}

Table 6: Types of signals related to the change in arm direction.

Type of signal	Change in direction
Valid	downward (0) – frontward (1)
Invalid	downward (0) – not frontward (2–7)
Inactive	downward (0) – downward (0)
End	not downward (1–7) – downward (0)

state. When M_o'' is one, it refers to the case when the officer is pointing in the frontal direction from the resting state, which means that the valid signal is being initiated. This situation is represented by the valid signal f_{valid}. If mode M_o'' is not zero or one, it denotes invalid signal f_{invalid}. Lines 10–14 are executed when M_o' is a value other than zero. This situation happens when there is a hand signal in progress. When M_o'' is zero, it implies that the police officer is putting his hands downward after giving a hand signal. This state is represented as f_{end}. If M_o'' changes to a signal other than zero, it means that a hand signal is in progress. This state is represented as f_{middle}. Line 15 returns the computed flag f.

Algorithm 1 Sequence flag

1: **Input**: s_t', s_t''
2: **Output**: validity flag f
3: $M_o' \leftarrow$ {mode of first subsequence s_t'};
4: $M_o'' \leftarrow$ {mode of second subsequence s_t''};
5: **if** $M_o' == 0$:
6: **if** $M_o'' == 0$:
7: $f \leftarrow$ {inactive signal f_{inactive}}
8: **else if** $M_o'' == 1$:
9: $f \leftarrow$ {valid signal f_{valid}};
10: **else**:
11: $f \leftarrow$ {invalid signal f_{invalid}};
12: **else**:
13: **if** $M_o'' == 0$:
14: $f \leftarrow$ {end of signal f_{end}};
15: **else**:
16: $f \leftarrow$ {middle signal f_{middle}};
17: **return** f

3.4.4. *Recurrent neural network*

An RNN is one of the deep learning models composed of connected hidden nodes that form a directed cycle. An RNN is used to classify continuous sequential data, such as speech and natural language. An RNN is also widely employed for classifying human motions (Wah Ng & Ranganath, 2002; Masood *et al.*, 2018; Cifuentes *et al.*, 2019). There are works that recognize police hand signals using RNNs with changes in skeletons or key points (Chen *et al.*, 2017; Lai & Yanushkevich, 2018; Shin & Kim, 2020). We also applied an RNN to classify the arm directions of police officers represented as sequence s_t. We compared the results trained with four kinds of RNNs and selected the result with the best accuracy. The four RNNs are the vanilla RNN (V-RNN), LSTM (Gers *et al.*, 1999), Bidirectional-LSTM (Bi-LSTM) (Schuster and Paliwal, 1997) and gated recurrent unit (GRU) (Chung *et al.*, 2014). The many-to-one structure is applied to classy the hand signals that were first translated to the codified 1D sequence. Figure 7 shows the layer structure.

4. Experiments

4.1. Result of RNNs

We trained and compared four types of RNNs (V-RNN, LSTM, Bi-LSTM, and GRU). The length of the input sequence length was set to 48, where $l = 48$. Any sequences shorter than this length were zero-padded by the data loader. We also set the step size, batch size, and epoch size to 32, 64, and 250, respectively. The process of hand signal sequence classification is a simple 1D sequence classification. Therefore, we decided that the number of layers was the most critical factor. We changed and tested the number of layers with 2, 4, 7, and 10. To avoid overfitting problems, early stopping was applied.

The sequence data shown in Table 5 were partitioned as train:validation:test = 6:2:2. The amount of training time increased proportionally to the number of layers employed in the neural network. On average, 3 hours were needed for processing using 10 layers. The testing time was approximately 10 minutes.

Table 7 lists the results of the training. Since the structure of the V-RNN is more straightforward than the structures of other RNNs, V-RNN performed fastest, but the test accuracy dropped as the number of layers increased. For all numbers of layers, overfitting occurred where the loss of validation became greater than that of the training. The number of epochs that triggered the early stop was 122, 58, 50, and 51 as the number of layers increased. These epochs were consistently higher than other types of RNNs. LSTM showed less overfitting than the V-RNN. However, the instability of the training increased as the number of layers increased.

The number of layers needed for early stopping, which were 29, 43, 56, and 76, was significantly less than that of V-RNN. Bi-LSTM showed more stable training than LSTM. The number of required epochs was smallest at 22, 20, 32, and 20. For the GRU, the number of needed epochs, which were 46, 41, 56, and 73, was slightly higher than that of LSTM and Bi-LSTM, but the training stability exhibited the greatest increase. In terms of test accuracy, the GRU with four layers was the best. Thus, we chose the four layers of the GRU as the sequence classifier for classifying the hand signals. The time-domain graph of the probability shown in Fig. 8 was obtained by applying the GRU to the hand signal video. Figure 8a shows an inactive signal when the traffic controller gives no hand signals. The hand signal probability remains low. Figure 8b shows the results by computing the video of the traffic controller that issues the go straight signal. At the initiation of the go straight hand signal, we observed that the probability of a go straight signal suddenly jumps and continues until the end of the hand signal. For other types of hand signals, we observed the probability of the corresponding hand signal rise and drop, as plotted in Figs 8c, d, and e. The invalid signal shown in Fig. 8f shows a slight rise in the hand signal probability. This result is different from Fig. 8a, which shows no change in the hand signal probability.

4.2. Comparison

To increase the accuracy of our classification, we compared two signals: the sequence flag $f(s_t)$ and the output $R(s_t)$ from the RNN. Both signals are derived from s_t, which is the codified output from the sequence generator. $f(s_t)$ classifies the codified sequence into three classes: inactive, valid, and invalid. $f(s_t)$ also catches the start, in-the-middle, and end of the signal. $R(s_t)$ is responsible for classifying the sequence into the relevant classes: go straight, turn left, turn right, stop, and inactive. The flag is

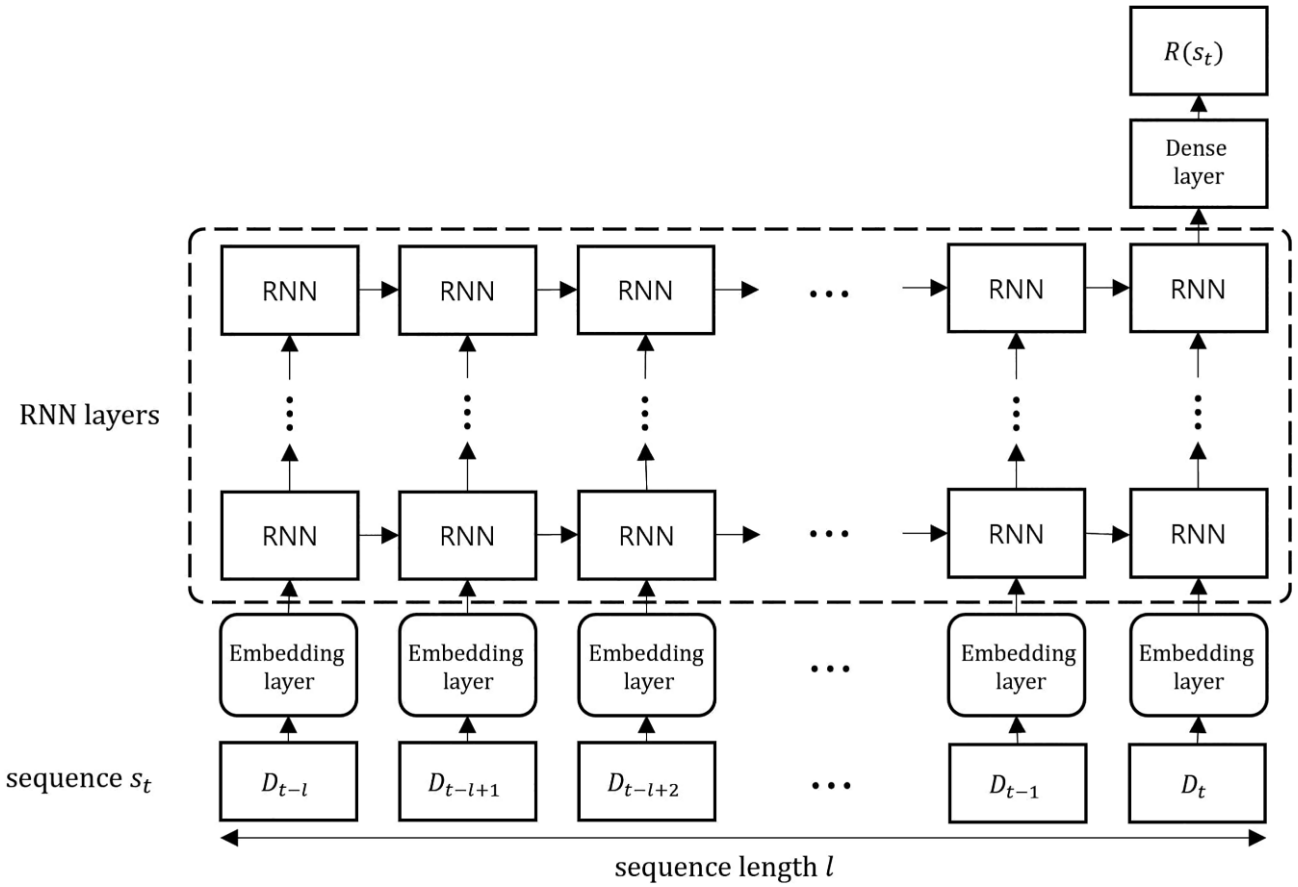

Figure 7: RNN layers.

Table 7: Results for RNN algorithms.

Methods	2 layers		4 layers		7 layers		10 layers	
	Evaluation accuracy	Test accuracy	Evaluation accuracy	Test accuracy	Evaluation accuracy	Test accuracy	Evaluation accuracy	Test accuracy
V-RNN	95.46	95.50	94.17	90.29	93.35	87.06	94.25	86.73
LSTM	**96.35**	93.53	**96.76**	95.47	**96.11**	92.88	95.46	92.23
Bi-LSTM	95.06	95.47	95.46	95.79	95.79	96.12	95.30	**96.76**
GRU	96.11	**96.76**	96.03	**97.41**	95.06	**97.09**	**96.19**	95.47

used to understand whether the hand signal is directed toward the viewer or whether it is directed to another driver. The RNN will process the sequence only if the direction is toward the viewer.

Three values are returned by $f(s_t)$. If s_t is a valid hand signal, one is returned, and $f(s_t) = 1$. For the case when the hand signal is inactive, $f(s_t)$ returns zero, and $f(s_t) = 0$. For the invalid case, 0.5 is returned, and $f(s_t) = 0.5$. Special attention must be made for 0.5. This finding indicates that the hand signal is not directed toward the viewer, and the signal is in some switching state. Thus, at least one $f(s_t) = 1$ is required to know that the hand signal is directed toward the viewer. This result also means that the hand signal is being initiated. After initiation, the returned value becomes $f(s_t) = 0.5$, and $f(s_t) = 0$ will denote the end of the hand signal. Based on $f(s_t)$, $R(s_t)$ is computed to classify the hand signal class type. Figure 9 shows the overall picture of the operation. Figure 9a denotes an inactive signal, no changes in the flag, and the plot of the hand signal probability. This signal is an inactive hand signal. In Fig. 9b, the plot of the flag rises to one, remains at 0.5, and drops to zero. This result denotes that a valid hand signal is monitored and that the width of the signal triggers the computation of the hand signal probability. The hand signal probability plot shows that the stop signal becomes highest during the monitoring time band. We can conclude from the plot that the stop signal is being sent to the driver. In Fig. 9c, the flag value continues to plot at 0.5 but never reaches one and drops to zero at frame 65. During the time band, we observe a series of turn-right, go-straight, and turn-left signals. These signals are not directed toward the driver and are deemed invalid signals. By comparing the flag with the RNN signal, we can classify the hand signal directed toward the viewer.

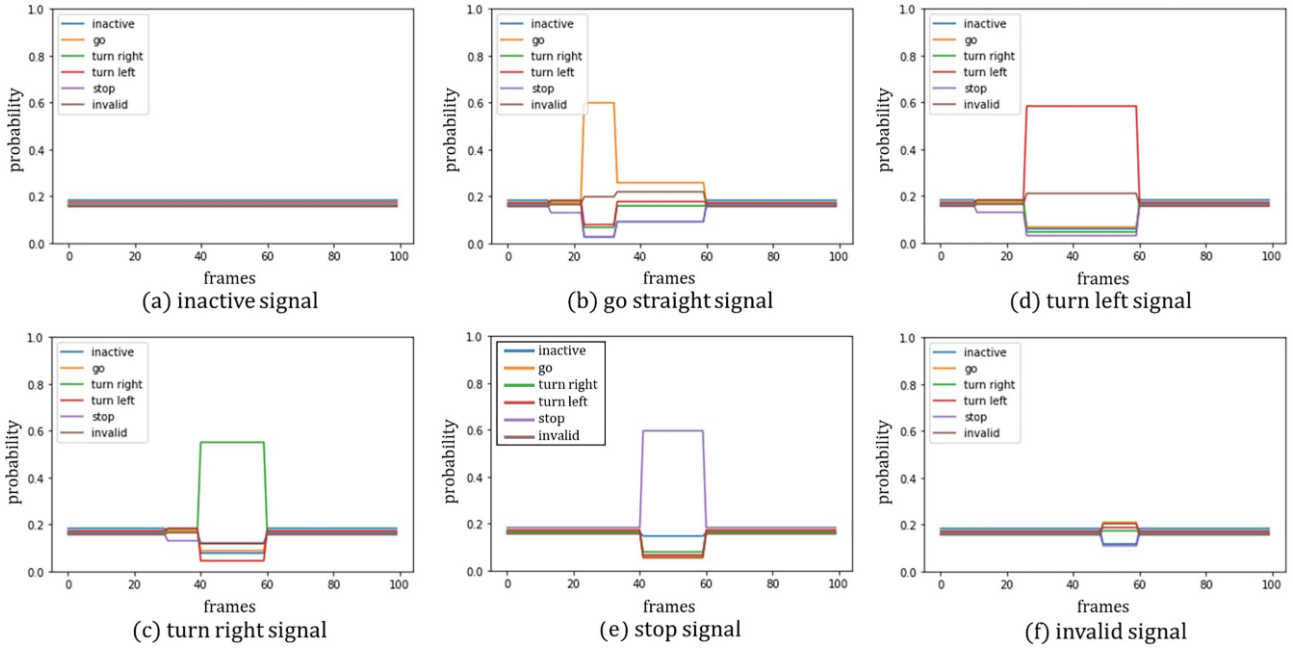

Figure 8: Results of GRU.

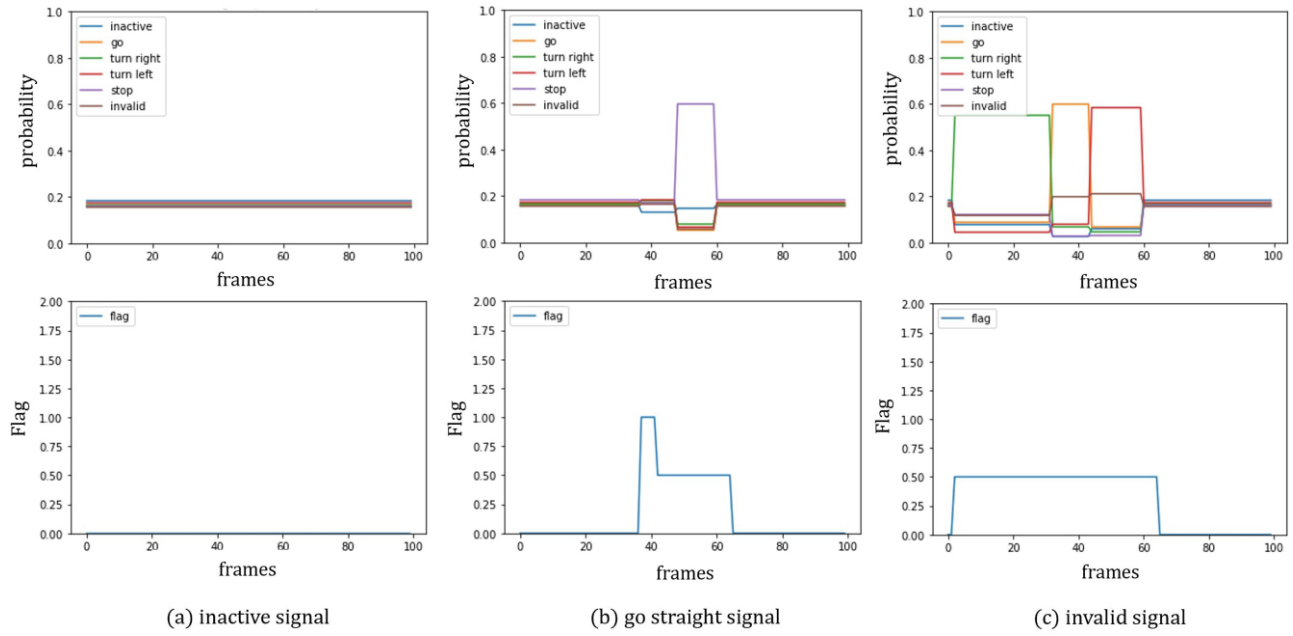

Figure 9: Comparison of flag and RNN.

4.3. Results of the proposed method

We conducted tests on videos that were not used in training. The test videos are videos shown in Table 3 that were not labeled. The flowchart shown in Fig. 6 was applied to the test video. The accuracy of the algorithm was 90.8%, and Fig. 10 shows the confusion matrix. Table 8 summarizes the evaluation result for each hand. We had performed the test on RTX8000 GPU with a frame rate of 32.5 with FHD video. Since the inactive signal is a signal that does not give any instructions, it is regarded as an invalid signal.

4.4. Discussion

We have proposed a method for classifying the hand signals given by a traffic controller. We have discussed the features of hand signals and the gathering and construction of a dataset for hand signals. We compared four types of RNNs for the best

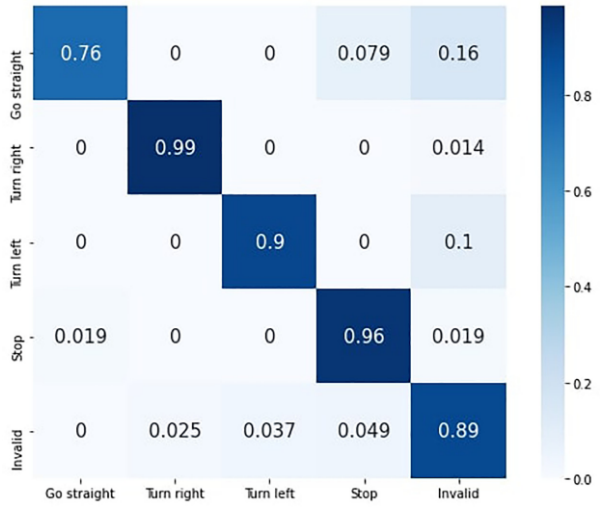

Figure 10: Confusion matrix.

result and presented the use of two sequences (flag and probability of hand signals) to increase the accuracy of the proposed method.

Previous works on hand signals relied on depth information. More specifically, the skeleton computed from the police officers was utilized to obtain feature values, such as the relative joint angles. The time transient features were then trained using RNNs. These works required additional sensors to compute the depth and needed significantly higher computations to capture the features, which yielded performances that reached merely 17 FPS.

Table 9 shows detailed comparisons of our method with previous works. Studies (Le *et al.*, 2012; Linqin *et al.*, 2017; Ma *et al.*, 2018) that applied depth information and were performed indoors generally showed high accuracy. Studies (Wang & Chong, 2014; Wiederer *et al.*, 2020) that were performed outdoors generally showed low accuracy or a low FPS. This finding is attributed to the notion that outdoor scenes are unpredictable and possess various illuminations with unseen backgrounds. For more challenging cases that were performed outdoors, without consideration of the validity of the detected hand signal and only using the raw RGB data, far less accuracy or low processing frames per second than those in our work were exhibited. We also emphasize that our dataset comprises 1600 sequences, which is the largest compared with previous works.

The proposed method uses only RGB cameras and relies on a high-speed, one-stage object detector. We simplified the algorithm to the simple features of a hand signal. The first feature is that the arm points in the right direction. Because the officer will be directing his arm toward the targeting driver, we devised a logic that concentrates on the direction of the arms. Our approach classified the directions of the arm into seven classes and employed supervised learning to train the one-stage detector to classify the seven cases. Subsequently, directed motions of the arms were then applied to classify the hand signals into six categories. This classification also used the one-stage detector to identify the sequence of directions of the arms to codify and detect the relevant six categories. The codified representation of the hand signal constitutes the second feature.

The first feature is essential because it saves considerable computational time. The expensive classification process of identifying the hand signal is only activated when the traffic controller is attentive to the driver. The classification process is dormant most of the time, saving a considerable part of the computer clock cycle. We propose a unique concept that is referred to as the sequence flag to identify the state of hand signal motion. The sequence generator creates a sequence including two prior subsequences evaluated in terms of the predefined modes; that is, the direction of the arms is used to classify the attentive state of the hand signal as inactive, valid, or invalid. The sequence flag is combined with the RNN classification results to monitor only the valid hand signals directed toward the viewer. We tested four types of RNNs to conclude that the GRU

Table 8: Results for the proposed method.

Hand signal	Go straight	Turn right	Turn left	Stop	Invalid
Accuracy	0.76	0.99	0.90	0.96	0.89
Sensitivity	0.99	1	1	1	0.64
Specificity	1	0.94	0.96	0.94	0.99
F1-score	0.99	0.88	0.92	0.89	0.78

Table 9: Comparison.

Author	Method	Accuracy	FPS	Amount of dataset	Outdoor condition	Validity
Wiederer *et al.* (2020)	Skeleton (RGB)	87.37%	-	250 sequences	○	○
Li and Yang (2018)	Key-frame extraction (RGB)	95%	-	5000 images	○	X
He *et al.* (2020)	Skeleton (RGB)	93.29%	17.2	20 videos	○	○
Sathya and Geetha (2015)	CBIV (RGB)	96.24%	-	300 videos	X	X
Wang and Chong (2014)	MFI and MHI (RGB)	81.44%	17.2	-	○	X
Linqin *et al.* (2017)	Skeleton (depth)	97%	2.7	800 sequences	X	X
Ma *et al.* (2018)	Skeleton (depth)	96.67%	-	-	X	X
Guo *et al.* (2017)	Skeleton (RGB)	95%	1	1834 frames	○	○
Le *et al.* (2012)	Skeleton (depth)	99%	0.34	5000 frames	X	X
Ours (2021)	2D bounding box (RGB)	90.8%	32.5	1600 sequences, 104 654 frames	○	○

with four layers performed the best. The complete system that we developed demonstrated a test accuracy of 91%. In Fig. 10, we observed that the results for identifying turn right, turn left, and stop hand signals were somewhat higher.

In comparison, the go straight test results were low. We believe that these results are attributed to the relatively small envelope of motions given when the officer is directing their arms upward and then giving the hand gestures for the next commanding direction. Other signals typically use fully stretched arm motions that are directed upward before giving the directional order using hand motions. In contrast, the go straight signal gives directional order with half stretched arm motions. Such a small envelope of motions may have caused the detector to misclassify them. We were able to verify that all signals were wrongly identified as an invalid class. This result is due to the sequence flag process. The detector has not correctly identified the correct designation of the targeting observer using arm motion. When the hand-signal classification is prematurely terminated with the sequence flag changing to the invalid state, the ongoing classification process is abandoned, and the system returns invalid hand signals. In the proposed method, the errors from the detector can be accumulated, which decreases the accuracy. The trained detector is capable of distinguishing pedestrians and traffic controllers. However, the training data do not have more than one traffic controller on a scene. Therefore, when multiple traffic controllers are on a scene, the detector would probably not work or the detection accuracy may decrease. However, this problem is inherent in all vision-based supervised classifying methods. Nevertheless, we argue that the proposed method is favorable because it requires no additional hardware other than the RGB camera and because the computational performance is twice that of previous works.

In future works, the dataset will include videos of actual road conditions. Current datasets were collected using performed actors and do not include data obtained from real situations. The collection of videos in the field would include situations that were not anticipated in the current dataset. Such movements in the field would consist of very distant police officers approaching the car, and variations in videos caused by the actual deceleration, shaking, and turning of car wheels. Furthermore, various adverse weather conditions, such as fog, rain, and snow, can degrade the image quality. It would also be interesting to use object detectors other than YOLOv4. More accurate detectors would improve the detection of the hand directions and therefore enhance the accuracy of the classification results obtained in addition to the detected hand directions. However, it is generally noted that more accurate detectors require more time and may produce implementations that would not be running in real time. There should be a balance between accuracy and responsiveness.

5. Conclusion

In this study, we proposed a deep learning-based architecture

We utilized the notion that hand signals can be decomposed into a more

. The former is referred to as the sequence flag,

. Because we require no stereo or depth cameras, the cost is low. Second, the proposed method only uses 2D images, the operations are simple, and the computation is high speed. We boast 30 FPS for FHD videos.

his result is twice as fast as previous studies. Last, we use the sequence flag that identifies the initiation and end of the hand signal. Because we can turn off the more expensive hand signal probability pipeline and only use it when experiencing hand signals, many computational cycles can be reserved for other

Even 1% of failures can cause serious injuries or deaths in the lives of humans. We believe that reaching 100% accuracy would not be possible even for human intelligence. Therefore, there should be a backup in the event of such failures. For example, we should enforce a manual, human-intervened emergency brake system. In emergency situations, nonspecialists may need to give hand signals. Understanding hand signals given by laypersons can be challenging and requires further study. We plan to extend our work to understand hand signals given by laypersons.

Acknowledgments

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2018-0-01290, Development of open informal dataset and dynamic object recognition technology affecting autonomous driving and No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST)).

Conflict of interest statement

None declared.

References

Angelini, F., Fu, Z., Long, Y., Shao, L., & Naqvi, S. M. (2020). 2D pose-based real-time human action recognition with occlusion-handling. *IEEE Transactions on Multimedia*, *22*, 1433–1446. https://doi.org/10.1109/TMM.2019.2944745.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *ArXiv:2004.10934 [cs.CV]*, http://arxiv.org/abs/2004.10934.

Chen, X., Guo, H., Wang, G., & Zhang, L. (2017). Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. In *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*(pp. 2881–2885). https://doi.org/10.1109/ICIP.2017.8296809.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv:1412.3555 [cs.NE]*. http://arxiv.org/abs/1412.3555.

Cifuentes, J., Boulanger, P., Pham, M. T., Prieto, F., & Moreau, R. (2019). Gesture classification using LSTM recurrent neural networks.In *Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society*

(*EMBC*)(pp. 6864–6867). https://doi.org/10.1109/EMBC.2019.8857592.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM. In *Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks (ICANN)*(pp. 850–855). https://doi.org/10.1049/cp:19991218.

Guo, F., Tang, J., & Wang, X. (2017). Gesture recognition of traffic police based on static and dynamic descriptor fusion. *Multimedia Tools and Applications*, *76*, 8915–8936. https://doi.org/10.1007/s11042-016-3497-9.

He, J., Zhang, C., He, X., & Dong, R. (2020). Visual recognition of traffic police gestures with convolutional pose machine and handcrafted features. *Neurocomputing*, *390*, 248–259. https://doi.org/10.1016/j.neucom.2019.07.103.

Iravantchi, Y., Goel, M., & Harrison, C. (2019). BeamBand: Hand gesture sensing with ultrasonic beamforming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*(pp. 1–10). https://doi.org/10.1145/3290605.3300245.

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*, 221–231. https://doi.org/10.1109/TPAMI.2012.59.

Lai, K., & Yanushkevich, S. N. (2018). CNN+RNN depth and skeleton based dynamic hand gesture recognition. In *Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)*(pp. 3451–3456). https://doi.org/10.1109/ICPR.2018.8545718.

Le, Q. K., Pham, C. H., & Le, T. H. (2012). Road traffic control gesture recognition using depth images. *IEIE Transactions on Smart Processing & Computing*, *1*, 1–7.

Li, C., & Yang, S. (2018). Traffic police gesture recognition for autonomous driving. In *Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC)*(pp. 1413–1418). https://doi.org/10.1109/CompComm.2018.8781046.

Linqin, C., Shuangjie, C., Min, X., Jimin, Y., & Jianrong, Z. (2017). Dynamic hand gesture recognition using RGB-D data for natural human-computer interaction. *Journal of Intelligent & Fuzzy Systems*, *32*, 3495–3507. https://doi.org/10.3233/JIFS-169287.

Ma, C., Zhang, Y., Wang, A., Wang, Y., & Chen, G. (2018). Traffic command gesture recognition for virtual urban scenes based on a spatiotemporal convolution neural network. *ISPRS International Journal of Geo-Information*, *7*, 37. https://doi.org/10.3390/ijgi7010037.

Masood, S., Srivastava, A., Thuwal, H. C., & Ahmad, M. (2018). Real-time sign language gesture (Word) recognition from video sequences using CNN and RNN. In V. Bhateja, C. A. Coello Coello, S. C. Satapathy, & P. K. Pattnaik (Eds.), *Intelligent engineering informatics*(pp. 623–632). Springer.

Neacsu, A. A., Cioroiu, G., Radoi, A., & Burileanu, C. (2019). Automatic EMG-based hand gesture recognition system using time-domain descriptors and fully-connected neural networks. In *Proceedings of the 2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*(pp. 232–235). https://doi.org/10.1109/TSP.2019.8768831.

Sathya, R., & Geetha, M. K. (2015). Framework for traffic personnel gesture recognition. *Procedia Computer Science*, *46*, 1700–1707. https://doi.org/10.1016/j.procs.2015.02.113.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*, 2673–2681. https://doi.org/10.1109/78.650093.

Shin, S., & Kim, W.-Y. (2020). Skeleton-based dynamic hand gesture recognition using a part-based GRU-RNN for gesture-based interface. *IEEE Access*, *8*, 50236–50243. https://doi.org/10.1109/ACCESS.2020.2980128.

Skaria, S., Huang, D., Al-Hourani, A., Evans, R. J., & Lech, M. (2020). Deep-learning for hand-gesture recognition with simultaneous thermal and radar sensors. In *2020 IEEE Sensors*(pp. 1–4). https://doi.org/10.1109/SENSORS47125.2020.9278683.

Statutes of Republic of Korea.(2009). Article 5 (Obligations to abide by signals and instructions). In *Road Traffic Act*.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*(pp. 4489–4497). https://doi.org/10.1109/ICCV.2015.510.

Varshney, N., Upadhyay, P., Arora, U., Singhal, S., & Mittal, S. (2020). Real time model for hand gesture recognition of traffic policeman. *International Journal of Advanced Science and Technology*, *29*, 7. http://sersc.org/journals/index.php/IJAST/article/view/25282.

Wah Ng, C., & Ranganath, S. (2002). Real-time gesture recognition system and application. *Image and Vision Computing*, *20*, 993–1007. https://doi.org/10.1016/S0262-8856(02)00113-0.

Wan, J., Li, S. Z., Zhao, Y., Zhou, S., Guyon, I., & Escalera, S. (2016). ChaLearn looking at people RGB-D isolated and continuous datasets for gesture recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*(pp. 761–769). https://doi.org/10.1109/CVPRW.2016.100.

Wang, B., & Yuan, T. (2008). Traffic police gesture recognition using accelerometers. In *Proceedings of the 2008 IEEE Sensors*(pp. 1080–1083).

Wang, X., & Chong, L. (2014). A recognition method of traffic directing gesture based on multi-feature extraction and sparse coding. *Journal of Computer Information System*, *10*, 2445–2453.

Wang, G., & Ma, X. (2018). Traffic police gesture recognition using RGB-D and faster R-CNN.In *Proceedings of the 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*(pp. 78–81). https://doi.org/10.1109/ICIIBMS.2018.8549975.

Wiederer, J., Bouazizi, A., Kressel, U., & Belagiannis, V. (2020). Traffic control gesture recognition for autonomous vehicles. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*(pp. 10676–10683). https://doi.org/10.1109/IROS45743.2020.9341214.

Zhang, X., Chen, X., Li, Y., Lantz, V., Wang, K., & Yang, J. (2011). A framework for hand gesture recognition based on accelerometer and EMG sensors. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, *41*, 1064–1076.https://doi.org/10.1109/TSMCA.2011.2116004.