Engineering Journal issn: 0125-8281

Article

Determining Bus Stop Locations using Deep Learning and Time Filtering

Jitpinun Piriyataravet[1,a], **Wuttipong Kumwilaisak**[1,b], **Jatuporn Chinrungrueng**[2,c], **and Teerawat Piriyatharawet**[3,d,*]

1 Department of Electronics and Telecommunication Engineering, King Mongkut's University of Technology Thonburi, Bangkok, Thailand
2 Health Innovation and Information Research Team, Assistive Technology and Medical Devices Research Center, National Science and Technology Development Agency, Pathum Thani, Thailand
3 School of Computer Science and Engineering, Nanyang Technological University, Singapore
E-mail: [a]jitpinun.max@mail.kmutt.ac.th, [b]wuttipong.kum@kmutt.ac.th, [c]jatuporn.chi@nstda.or.th, [d,*]teerawat.p@ntu.edu.sg (Corresponding author)

Abstract. This paper presents an intelligent bus stop determination from bus Global Positioning System (GPS) trajectories. A mixture of deep neural networks and a time filtering algorithm is used in the proposed algorithm. A deep neural network uses the speed histogram and azimuth angle at each location as input features. A deep neural networks consists of the convolutional neural networks (CNN), fully connected networks, and bidirectional Long-Short Term Memory (LSTM) networks. It predicts the soft decisions of bus stops at all locations along the route. The time filtering technique was adopted to refine the results obtained from the LSTM network. The time histograms of all locations was built where the high potential timestamps are extracted. Then, a linear regression is used to produce an approximate reliable timestamp. Each time distribution can be derived using data updated at that time slot and compared to a reference distribution. Locations are predicted as bus stop locations when timestamp distributions close to the reference distributions. Our technique was tested on real bus service GPS data from National Science and Technology Development Agency (NATDA, Thailand). The proposed method can outperform other existing bus stop detection systems.

Keywords: Bidirectional LSTM, bus stop determination, convolutional neural network, deep learning, Global Positioning System.

ENGINEERING JOURNAL Volume 25 Issue 8
Received 11 June 2021
Accepted 10 August 2021
Published 31 August 2021
Online at https://engj.org/
DOI:10.4186/ej.2021.25.8.163

This article is based on the presentation at The 2021 2nd International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP 2021) in Bangkok, Thailand, 20th-22nd January 2021.

DOI:10.4186/ej.2021.25.8.163

1. Introduction

An intelligent transportation system (ITS) is a sophisticated program that strives to deliver novel services related to various modes of transportation and traffic management, allowing users to be better informed and make safer, more coordinated, and intelligent use of transportation networks. Automating bus stop determination is one of the essential components in the ITS. The bus stop detection problem is the process of locating the bus stop location. This is a classification issue, in which all input points along the route are categorized as either bus stops or not bus stops. The bus stop locations are frequently tainted with mistakes, such as the bus's route being changed or the vehicle failing to stop at some bus stops because buses are maintained by separate authorities.

With the rapid advancement of Global Positioning System (GPS) tracking devices and the increased availability of GPS, several studies have proposed several approaches for analyzing GPS data for autonomous systems. The inventors of [1] suggest a system that detects messages on the road using a Kalman filter [2] and pattern matching. However, over time, road signs become less dependable. Instead of employing roadside messages, James Biagioni's system [3] proposes an approach based on kernel density estimation [4] to autonomously find bus stops and manage timetables using GPS traces data. A minimum density threshold is applied to extract a set of stop locations where any location with a density above the density threshold is regarded as a bus stop. Fabio Pinelli [5] presents a technique for analyzing variations of accessibility in the city at different times of the day and estimating all bus stop positions in bus GPS traces using many clustering processes. To recognize bus stops, Mandal et al. [6] employ density-based clustering. Their system can locate all planned bus stops, but stoppages caused by a speed breaker, congestion, or a crossroad cannot be distinguished from bus stops. The authors of [7] suggest a method that uses repeated trip data to improve the density accuracy with a clustering technique. The system is capable of dealing with erratic bus timetables and stops along the route. However, the size of the data has a big impact on clustering performance [8]. Due to a lack of data, the algorithm may not be efficient in small transportation systems with fewer daily rounds. The system proposed by Leon Stenneth et al. [9] employ a machine learning algorithm with spatial and temporal clustering technique. Using a density distribution histogram and other factors, the method captures usage pattern around bus stops and non-bus stop areas. Many techniques in [10, 11], dealing with GPS trajectory data to find the bus stop, use machine learning methods such as Decision Tree, Random Forest and Support Vector Machine. The authors in [12] provide an approach to dealing with image data provide an approach to dealing with image data. It helps in the analysis of the most ambiguous images [13] with efficient Bayesian Active Learning strategies. The authors from [14, 15] propose an incremental active learning-based technique for wearable on-device scenarios. The authors in [16] utilize deep learning [17] to recognize the bus stop, which avoids the need for GPS traces by combining photos from both the left and right sides of each bus. This poses privacy problems, as photographs from the bus side may reveal people's faces in some situations, making it vulnerable to lawsuits.

With the above-unsolved challenges, we propose a technique for bus stop discovery algorithms. This work is inspired by the National Science and Technology Development Agency (NSTDA) bus service problem. The NSTDA bus is a local transportation system that rarely updates bus stops and routes to ensure that passengers are satisfied and maximize their benefits. Intelligent bus stop classification with deep learning and time filtering Technique are the two main components of the proposed algorithm. The initial step is to create bus speed histograms at several points along a route. A deep neural network uses a bus speed histogram and an azimuth angle at each site as the input characteristics. CNN networks and fully connected networks make up the deep neural network. The outputs of a deep neural network at all points along a route are the inputs to the LSTM network. For points along a route, the LSTM network makes soft judgments about bus stop classification. The findings of the LSTM network are refined using a time filtering method. It creates timestamp histograms for all of the sites and extracts the most likely timestamps. Timestamps are then corrected using a linear regression approach. From the rectified timestamp, time distributions can be calculated and compared to a reference distribution. Locations with time distributions close to the reference distributions are predicted as bus stop locations.

This paper is organized as follows. Section 2 explains the overall architecture of a system for determining bus stop locations. Section 3 describes an intelligent bus stop classification with deep learning. Section 4 describes a time filtering algorithm. The experimental results are presented in Section 5. Section 6 contains the concluding remarks.

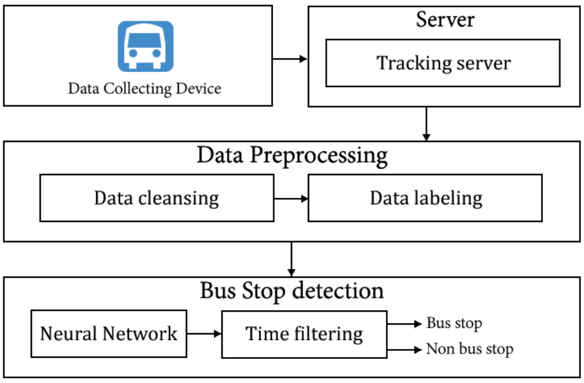


Figure 1. The system architecture of bus stop classification.

2. System Architecture

The suggested bus stop classification framework has four key components: (1) Data Collecting Device (2) Server Module (3) Data Preprocessing Module and (4) Bus Stop Determination Module. The following is a list of each component's description.

2.1. Data Collecting Device

The handheld GPS device used to collect data is known as a data-gathering device. During operation, it can connect to a server and send updated data every five seconds. The Open Traffic Data Exchange and Collaborative platform [18] is used to set up the gadget on the bus. However, because of the possible imprecise GPS, some of the acquired data may be incorrect, and the device is not always available. The GPS unavailability and inaccuracy are influenced by various factors, including air conditions and barriers, which might result in a different GPS location even in the same area.

2.2. Server Module

The data collection devices send data to a server module, which receives and collects them. The Message Queue Telemetry Transport (MQTT) protocol is used to communicate between the server and the data collection devices [18]. The coordinate position in terms of latitude and longitude and timestamps are included in each batch of data.

2.3. Data Preprocessing Module

Data cleaning and data labeling are two sub-modules of a preprocessing data module. The data cleaning sub-module is responsible for correcting non-concerned data, such as data collected while a bus is traveling off the beaten path. This is accomplished by detecting all pathways that exit the region of interest and eliminating any data that is irrelevant. The data is divided into two slots based on shuttle bus service driving in the morning (6.00-9.00) and afternoon (17.00-20.00), with data outside those two slots and data at the weekend being discarded because the bus timetable picks up passengers in the morning and drops them off after working hours. The data labeling sub-module is used to label all locations along a route either bus stop or non-bus stop locations. The classification agent is trained with the labeled data. Humans are responsible for this submodule.

2.4. Bus Stop Determination Module

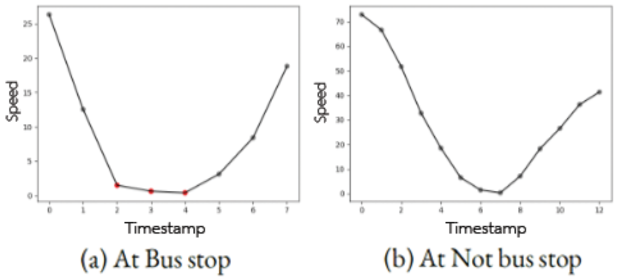
A bus stop determination module is the core process of the proposed framework, which will be explained in the next section. It uses the collected data in the server to determine the positions of bus stops along a route. The output of this module is the set of predicted bus stop locations along a route.

3. Intelligent Bus Stop Classification with Deep Learning

We define the problem of bus stop detection as a classification problem to locate the bus stop locations from sets of GPS data along the route. Directly locating the bus stop using raw GPS data can result in poor performance, especially in the case of lacking data. We establish reference points that reflect various sites along a bus route given a route.

Let $f_{cd}(v)$ be the function of the bus stop determination where the vector of features v can obtain from extracted raw GPS data by considering a speed histogram of each reference point. The bus stop determination function returns a value ranging from zero and one. The output indicates the confidence of each location, whether it is not a bus stop location.

The histogram of bus speed is constructed using all raw GPS locations lie within a radius of D_{TH} meters. The timestamp is included in each GPS location. The speed of a current data can be determined by employing a previous data and a current data. We include previous location within five seconds before arriving at the considering location in this paper. Bus speeds vary depending on the location. We can construct a speed histogram from all of the available speeds of places. There are multiple parameters in the function $fcd(v)$. It is not required to be linear and it is difficult to formulate in a close mathematical form. To derive this non-linear function, we use a learning-based approach. Speed histograms are provided as inputs to the proposed deep learning architecture to obtain the rough output. The rough output is then refined using a time filtering technique.



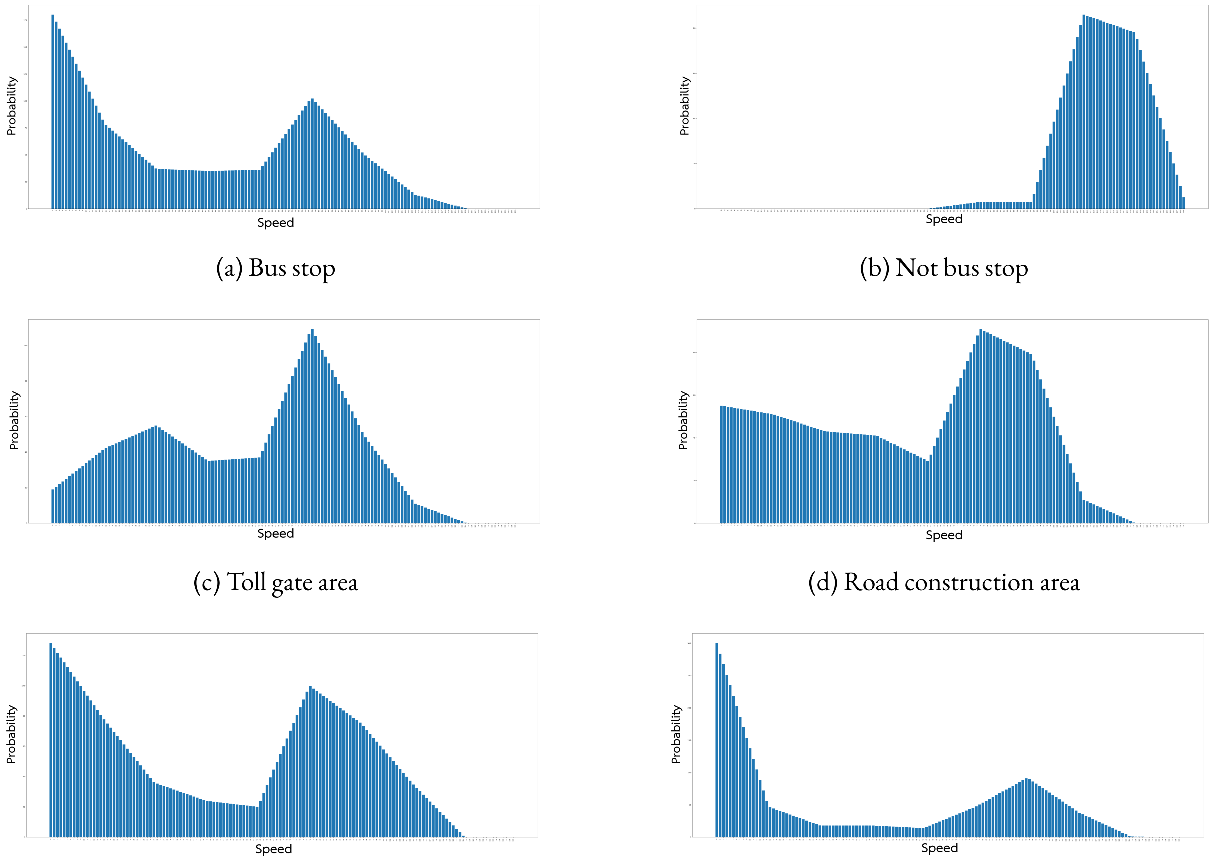
(a) At Bus stop

(b) At Not bus stop

Figure 2. The graph of bus speed in time sequences.

3.1. Bus Movement Features

Bus movement features are based on the idea that a bus's speed is zero for a complete stop at a bus stop. However, due to traffic congestion, the speed threshold cannot completely identify the bus stop position because a bus may need to stop at random locations, causing zero speed at the non-bus-stop. Furthermore, as shown in Fig.2, the sequence of bus speed in time sequences at bus stop locations may have characteristics similar to non-bus-stop sites. In general, using a group of GPS points can provide better performance in bus stop determination than only one GPS point. This is because the



(a) Bus stop

(b) Not bus stop

(c) Toll gate area

(d) Road construction area

(e) Tunnel

(f) At crossroad

Figure 3. The speed histogram.

bus will not stop at the exact GPS location due to GPS precision. Moreover, many factors, such as air conditioning and barriers, influence GPS accuracy, resulting in a different GPS location even though the position is accurate.

Figure 3 shows speed histograms in different scenarios. The histogram at the bus stop will be skewed to the right, indicating that the bus driver is driving slowly in that area. However, due to the bus driver's poor behavior, the speed at the bus stop can fluctuate between low and high, as seen in Fig.3(a). While the bus driver is driving fast, the histogram at the non-bus stop position will have a left-skewed distribution, as seen in Fig.3(b). It appears that the histogram makes it simple to define the bus stop location. Unfortunately some scenarios are challenging to identify whether they correspond to bus stop locations due to road environments, such as crossroads, toll gates, construction, or traffic congestion. As shown in Figs.3(c) and 3(d), there may be a toll gate on the highway or road construction, so the bus must stop temporarily. This introduces the low-speed portion into the histogram. Another example is when a bus enters a tunnel and is forced to slow down due to the tunnel's speed limit. It is also difficult to tell if a location is a bus stop or not when the histogram seems half split and comprises both very low and extremely high speeds, as shown in Fig.3(e). Moreover, a bus is likely to slow down before it turns at a crossroads to avoid crashes. As a result, the histogram may resemble the location of a bus stop because the speed is near zero in some situations, making it difficult to distinguish from the bus stop, as shown in Fig.3(f).

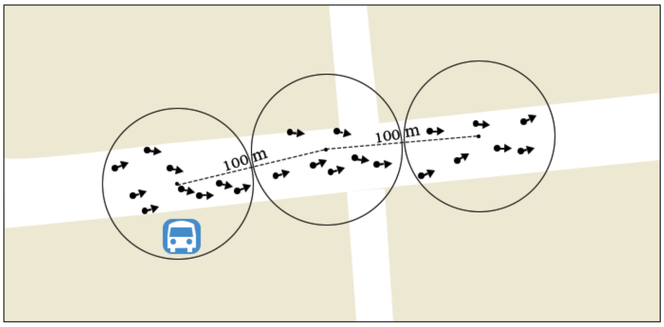


Figure 4. The reference location along the route way.

The GPS data is used to find the reference points along a journey. In this work, the ideal reference points are spaced by approximately $D_{TH} = 100 meters$. However, GPS data may not be available all of the time, and GPS location may change even though the device is at the same spot.In this instance, we use GPS data from numerous days of the same trip to obtain reference points. We choose the closest location to the present position as the next reference point from among the available locations spaced by somewhat greater than D_{TH}. This process will be repeated until we reach the bus final destination. After obtaining a set of reference points,We quantize all GPS data to these reference points after acquiring a set of reference points. To be more exact, any sites within D_{TH} of the reference point are deemed to be in the same geographic location as the reference point. Figure 4 shows an example of the derived reference points.

Each reference point's speed histogram can be constructed by discretizing the bus speed data by speed range. The frequency of bus speed data matching to each speed range is shown in each histogram interval. There are ten speed intervals in this work. However, employing merely the speed histogram element will not produce satisfactory results because the bus movement characteristics are similar at low speeds, making it difficult to distinguish between bus stop locations or some locations around intersections. Another input feature is the bus heading direction associated to each point. By combining the frequencies of the speed histogram and the heading direction, the input feature vector can form a vector with dimensions 1×11. To take advantage of temporal correlation among feature vectors, we can create a matrix by stacking feature vectors from two locations immediately before and after the present one. It will give us a matrix with a size of 5×11. These matrix features will be an input to the deep neural networks.

3.2. Deep Neural Networks

The speed histogram obtained from the observation at the location i is H_i^s. The heading direction or azimuth value is A_i. From the function of bus stop detection, the input feature obtain from raw GPS data can be expressed as

$$v = [H_i^s, A_i] \tag{1}$$

The function of probability is $P(H_i^s)$. The prediction probability of a bus stop at location i can be expressed as

$$P_{bs}(i) = f_{cd}(H_i^s, A_i) \tag{2}$$

The probability obtained from the nonlinear function is the soft decision that represents the confidence of being the bus stop at each location i. We have to deal with several parameters and input features. It might be impossible to derive a close function of a bus stop detection. Instead, the close function can be approximated by using a deep neural network. The deep neural network used for extracts the candidate locations for determining bus stop location. However, the candidate locations are predicted as the bus stops by time filtering.

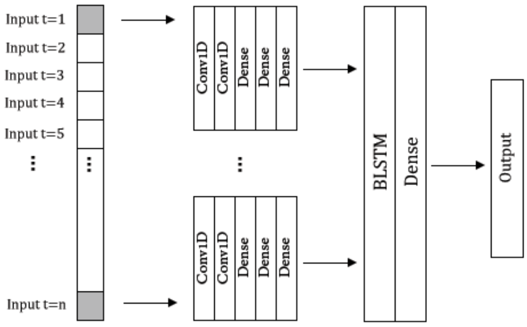


Figure 5. The proposed deep deep neural network architecture.

Our architecture consists of Convolutional Neural Networks (CNNs) [19], fully connected network (FCN), and Long Short Term Memory (LSTM) network [20]. A bus movement feature matrix with a dimension of 5x11 is fed to the deep neural network for each location. In each bus movement feature matrix, take advantage of CNNs and FCNs to extract and embedded the bus movement matrix feature. Since the correlation among the consecutive location when the bus moving, the output from this can view as the sequence data. Then, the output from the CNNs and FCNs of all locations be the input of the LSTM network. However, the extension of traditional LSTM which is bi-directional LSTM [21] can improve model performance on sequence classification problems is utilized. It can be used to obtain both forward and backward temporal data. Figure 5 shows the proposed neural network architecture. A deep neural network for bus stop prediction at each location can be described as follows.

Layer 1: Apply the CNN networks to the input matrix. The filter size 1x3. There are 64 filters. The stride is one with zero paddings. ReLU is used as a the activation function. The outputs from this layer are fed to the next CNN layer.

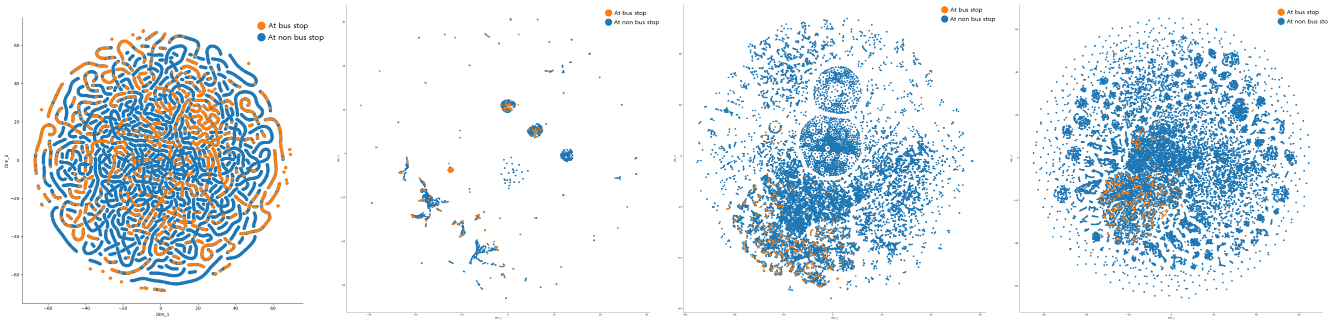
Layer 2: Apply the CNN networks to the inputs. The filter size 1x3. There are one filters. The stride is one with zero paddings. ReLU is used as a the activation function. There are eleven features in the outputs. The outputs from this layer are fed to the next FCN layer.

Layer 3: Apply the FCN networks with 64 nodes to the inputs. A ReLU is used as a the activation function. The outputs from this layer are fed to the next FCN layer.

Layer 4: Apply the FCN networks with 64 nodes to the inputs. A ReLU is used as a the activation function. The outputs from this layer are fed to the next FCN layer.

Layer 5: Apply the FCN with two nodes to the inputs. A softmax classification is used to output the prediction probabilities of a bus stop.

The outcome feature from all prediction locations can be concatenated to be another feature vector. This feature vector is an input to the LSTM network to process the entire sequences of data. The output is a rough bus stop prediction



(a) Raw GPS location

(b) Speed Histogram

(c) Stacked speed histogram

(d) The input feature of the deep neural networks

Figure 6. The t-SNE visualization from different feature vectors.

vector of all locations along the route.

4. Time Filtering

The output from the deep learning can provide erroneous bus stop locations due to the traffic congestion. A bus may have to stop temporarily at the non-bus-stop location. It is more complex and confuses the deep neural network.Accordingly, we propose the time filtering technique to solve this problem by refining the rough outcome from the deep neural network.

First, construct the time histogram for each location by time range. Each time histogram interval shows the frequency of data corresponding by time range. In this work, each time interval size is equal to 30 seconds. The time histogram size is 2880 intervals corresponding to 24 hours. Then, applying the median filter with a zero-padding size two for preserving sharp edges. This filter is used to remove outlier data. After that, extract only some intervals which representing the time distribution of the considering location among all intervals. The time histogram has 140 intervals in this work. From the observations, the driver drops passengers off around the same area location and tends to be at the same time to keep its schedule.

Therefore, let the peak of the time histogram as the center of the time distribution is suitable. However, there have variances in the peak of the time which makes the unreliable of the peak time distribution. To obtain the reliability of the peak time for all locations along the route, we collect the timestamps for approximated the time distribution the timestamps should corresponding to the center of time distribution at each location. Define the timestamps vector t_{max} which corresponding to the center of time histograms of each location as

$$t_{max} = [t_{max,1}, t_{max,2}, ..., t_{max,N}] \tag{3}$$

where $t_{max,i}$ is the timestamp corresponding to the peak time of the histograms at considering location i. However, with the fact that a bus should arrive at location i before location $i + 1$. It is suppose to have $t_{max,i} > t_{max,i+1}$ which contradicts with the fact but it's impossible due to variance. To obtain the reliability of the peak time for all locations, we approximate the timestamps of all locations by using linear regression. It can be expressed as

$$T_{max} = a_{opt} t_{max} + b_{opt} \tag{4}$$

where a_{opt} and b_{opt} are linear regression parameters. T_{max} as the predicted centers of the time histogram of each location. For each location along the route, we approximated the time distribution at the considering location, estimating from the left to the right to obtain the approximated time distribution. The approximated time distribution is compared with the reference time distribution obtained from experts. The distance metric is the Wasserstein distance [22]. We determine the considering location as a bus stop, if the distance is lower than a certain threshold.

5. Experimental Results

5.1. Bus Movement Features

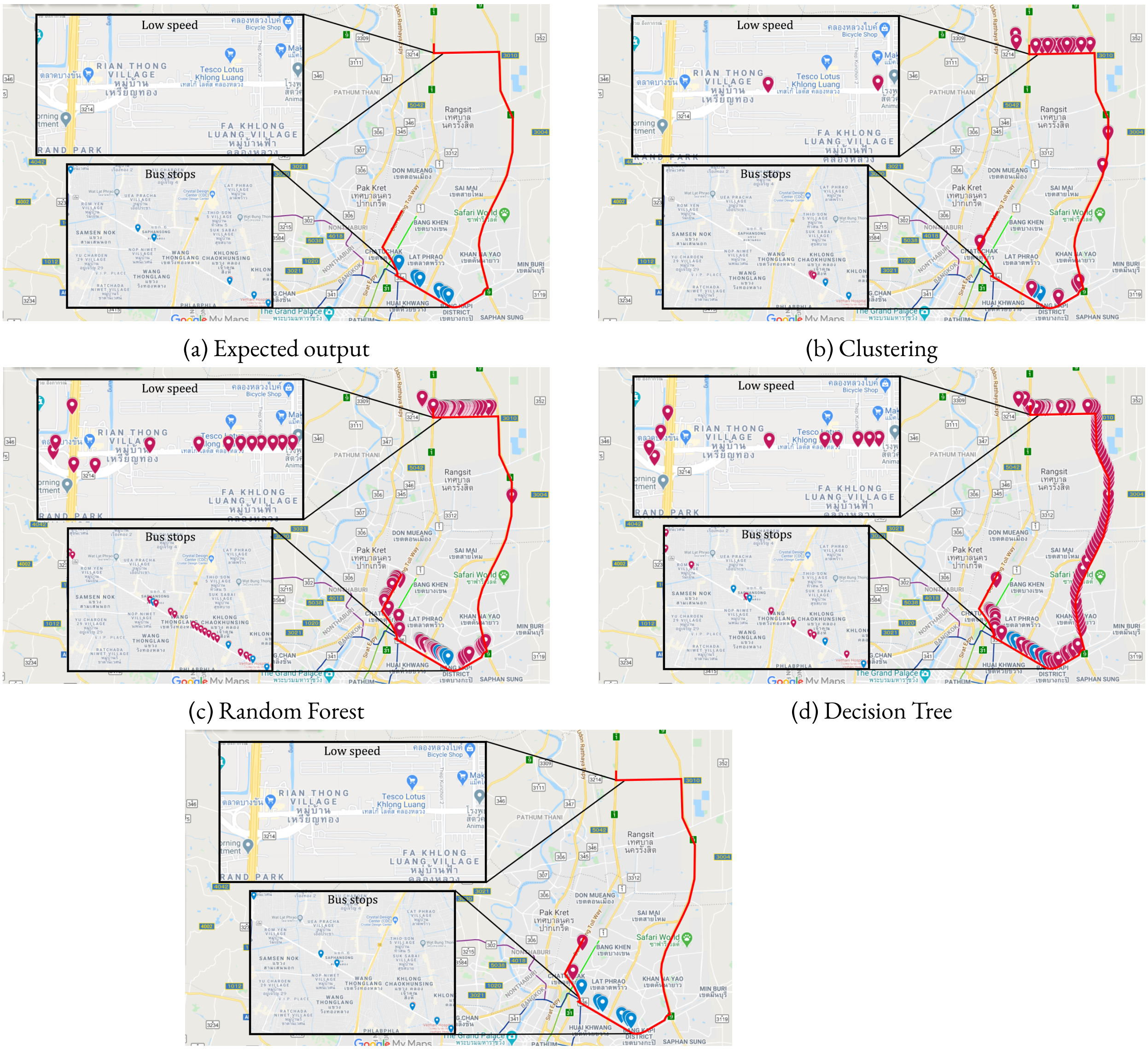
In this section, we assess the characteristics of bus movement features. The t-SNE [23] is used to project feature vectors to be two-dimensional data to interpret feature characteristics easier. Figure 6 shows the t-SNE plots of different speed features. Figures 6(a), 6(b), 6(c), and 6(d) illustrate speed features obtained from only raw GPS data of each reference point, speed histogram from only one reference location, speed histograms from five reference locations, and speed histograms from five reference locations together with bus heading direction. We can see that the speed features in Fig.6(c) and 6(d) can be used to classify bus stop locations better than other features. These features can help the deep neural network to provide better classification performance.

5.2. Training Neural Networks

A speed histogram has ten intervals. The centers of these ten intervals are at 1,2,3,4,5,10,20,40,80, and 100. The distance threshold D_{TH} has been set at 100 meters. Our training data is collected over a three-month period and includes

Table 1. Comparing the bus stop classification qualities obtained from different algorithms.

Data	Decision Tree [11]	Random Forest [10]	Clustering [7]	Our proposed method without filtering	Our proposed method with filtering
	TPR/FPR	TPR/FPR	TPR/FPR	TPR/FPR	TPR/FPR
1	**1.00**/0.034	0.25/0.029	0.40/0.032	**1.00**/0.053	**1.00/0.014**
2	0.67/0.40	0.67/0.317	0.33/0.115	**1.00**/0.194	**1.00/0.008**
3	0.4/0.309	0.60/0.071	0.40/0.032	**1.00**/0.090	0.80/**0.026**
Mean	0.69/0.248	0.507/0.139	0.377/0.06	**1.00**/0.112	**0.93/0.016**



(a) Expected output

(b) Clustering

(c) Random Forest

(d) Decision Tree

(e) Our method

Figure 7. The predicted bus stop locations from various algorithms.

approximately 90 routes. Adam Optimizer is used to train our deep learning network. A learning rate of 0.0001 is used, with an eight-batch size and 50 training iterations. The timestamp histogram when a bus stops at the actual bus stops but on other bus routes is used as the reference histogram for the time filtering technique.

5.3. Bus Stop Detection

Table 1 shows the results were obtained by comparing our proposed method to the other traditional techniques on the test data. The test data includes 15 sets of each three bus routes. It was collected in the same way as training data but it is not included in training data. It was collected along three routes by three different drivers. The results show that our proposed method outperforms all other methods in terms of True Positive Rate (TPR) [24] and False Positive Rate (FPR) [25]. It also surpasses state of the art by 0.553 TPR (the higher, the better) and 0.044 FPR (the lower, the better).

However, traffic congestion, construction roads, or unreliable traffic light make the shuttle bus cannot keep their schedules. These situations are pretty challenging to filter out. The bus stop location may have characteristics similar to non-bus stop locations with high variance in time histogram. In these situations, the time filtering technique will refine the result from the bus stop as a non-bus stop location. Although there is a chance that the time filtering technique refines the wrong outcome but it helps improve user satisfaction. As we stated above, giving the least possible false alarm is crucial because providing the non-existed bus stop location will harm the passenger much more than not telling some bus stop location to the passenger. With reliable information, passengers can avoid waiting at a location the bus will never stop.

Figure 7 visualizes results of bus stop prediction techniques. First, we compare the bus stop locations obtained from different algorithms with the ground truth. Other algorithms tend to have higher false-positive bus stops caused by traffic congestion, sudden stop, or speed reduction at the intersections. However, the proposed algorithm is robust to these challenging scenarios.

6. Conclusions

This paper propose the novel architecture of deep convolutional and bi-directional LSTM networks with time filtering technique to automatic bus stops location determination from GPS trajectories. The proposed network is designed to have two stages. The deep neural networks use the speed histogram as the input for output the rough output of bus stop prediction of all locations. Then, the time filtering technique on the time histogram to refines the results obtained from the deep neural network. The results show that our proposed network outperforms existing algorithms in terms of detection bus stops and lower false-positive.

References

[1] U. Franke and A. Ismail, “Recognition of bus stops through computer vision,” in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683)*, 2003, pp. 650–655.

[2] M. Piovoso and P. Laplante, “Kalman filter recipes for real-time image processing,” *Real-Time Imaging*, vol. 9, pp. 433–439, 2003.

[3] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, “Easytracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones,” in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. Association for Computing Machinery, 2011, p. 68–81.

[4] T. Ledl, “Kernel density estimation: Theory and application in discriminant analysis,” *Austrian Journal of Statistics*, vol. 33, no. 3, pp. 267–279, 2016.

[5] F. Pinelli, A. Hou, F. Calabrese, M. Nanni, C. Zegras, and C. Ratti, “Space and time-dependant bus accessibility: A case study in rome,” in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, 2009, pp. 1–6.

[6] R. Mandal, N. Agarwal, P. Das, S. Pathak, H. Rathi, S. Nandi, and S. Saha, “A system for stoppage pattern extraction from public bus gps traces in developing regions,” in *Proceedings of the Third ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*. Association for Computing Machinery, 2014, p. 72–75.

[7] P. Rajput, M. Chaturvedi, and P. Patel, “Advanced urban public transportation system for indian scenarios,” in *Proceedings of the 20th International Conference on Distributed Computing and Networking*. Association for Computing Machinery, 2019, p. 327–336.

[8] P. Tan, M. Steinbach, and V. Kumar, “Cluster analysis: Basic concepts and algorithms,” *Introduction to Data Mining*, pp. 487–568, 2005.

[9] L. Stenneth and P. S. Yu, “Monitoring and mining GPS traces in transit space,” in *Proceedings of the 13th SIAM International Conference on Data Mining*. SIAM, 2013, pp. 359–368.

[10] N. Garg, G. Ramadurai, and S. Ranu, “Mining bus stops from raw gps data of bus trajectories,” in *2018 10th International Conference on Communication Systems Networks (COMSNETS)*, 2018, pp. 583–588.

[11] F. Pinelli, F. Calabrese, and E. Bouillet, “Robust bus-stop identification and denoising methodology,” *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 2298–2303, 2013.

[12] Y. Gal, R. Islam, and Z. Ghahramani, “Deep bayesian active learning with image data,” in *Proceedings of the*

34th International Conference on Machine Learning, vol. 70. PMLR, 2017, pp. 1183–1192.

[13] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2017, p. 5580–5590.

[14] G. K. Gudur, P. Sundaramoorthy, and V. Umaashankar, "Activeharnet: Towards on-device deep bayesian active learning for human activity recognition," in *The 3rd International Workshop on Deep Learning for Mobile Systems and Applications*. Association for Computing Machinery, 2019, p. 7–12.

[15] S. Dasgupta, A. T. Kalai, and C. Monteleoni, "Analysis of perceptron-based active learning," *J. Mach. Learn. Res.*, vol. 10, p. 281–299, 2009.

[16] G. K. Gudur, A. Ramesh, and S. R, "A vision-based deep on-device intelligent bus stop recognition system," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. Association for Computing Machinery, 2019, p. 963–968.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[18] J. Chinrungrueng, L. Kovavisaruch, L. Narupiyakul, and K. Wongpatikaseree, "Open traffic data exchange and collaborative platform," in *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2018, pp. 1–4.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 2015.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 1997.

[21] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4, 2005, pp. 2047–2052 vol. 4.

[22] L. Rueschendorf, "The wasserstein distance and approximation theorems," *Probability Theory and Related Fields*, vol. 70, pp. 117–129, 1985.

[23] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[24] H. Wang and H. Zheng, *True Positive Rate*. Springer New York, 2013, pp. 2302–2303.

[25] S. Z. Li and A. Jain, Eds., *False Positive Rate*. Springer US, 2009, pp. 407–407.

DOI:10.4186/ej.2021.25.8.163

Jitpinun Piriyataravet received the B.E. degree in Computer Engineering from King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand, in 2018. Currently, She is studying for the M.E. degree in Electrical Engineering, with support from the Thailand Graduate Institute of Science and Technology (TGIST) under the National Science and Technology Development Agency (NSTDA). Her research interests include deep learning, Artificial Intelligence, Data analysis, image processing, and computer vision. Her current research focuses on light-weight deep learning for Real-time analysis of traffic flow information.

Wuttipong Kumwilaisak obtained his B.E.degree from Chulalongkorn University in 1995, the M.S. and the Ph.D. degree from University of Southern California in 2003, all in Electrical Engineering with support from Thai Government Scholarship. During the May-August of 2001 and 2002, he was a research intern at Ericsson Eurolab, Aachen, Germany, and at Microsoft Research Asia, Beijing, China. From April 2003 to August 2004, he was a senior engineer and project leader of the mobile platform solution team and multimedia laboratory at Samsung Electronics, Suwon, Korea. He was a postdoc fellow at Thomson Research Laboratory, Princeton, USA from March to November 2006. He has been an Associate Professor at King Mongkut's University of Technology Thonburi, the Electronics and Telecommunication Department, Thailand. His research interests are the optimization and algorithmic design for wireless communications and multimedia communication systems. His current researches include multimedia compression and processing and 3-D image processing.

Jatuporn Chinrungrueng graduated with a Bachelor degree in Electrical Engineering from Chulalongkorn University and received his Master degree in Electrical and Computer Enginerring from Cornell University. He then graduated with a Ph.D. from University of Colorado at Boulder in Electrical and Computer Engineering. His Ph.D. thesis was about clustering fMRI brain signal. He obtained a full scholarship from the Royal Thai Government. Since graduation, he has worked with National Science and Technology Development Agency as a researcher. His research interests include wireless communications, multidimensional signal processing, clustering analysis and pattern recognition, time-frequency and wavelet analysis, image and optical computing, physical activity classification and fMRI data analysis. Currently, his research work focuses on applying technologies for helping elderly people.

Teerawat Piriyatharawet received the B.E. degree in Electronics and Telecommunication Engineering, the M.E. degree in Electrical Engineering; all from King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand, in 2016 and 2020, respectively. Since August 2018, he has been with Nanyang Technological University (NTU), Singapore, and he is currently a research assistant. He has been working on camera processing and 3D related algorithms design. His research interests include image processing, image quality assessment, computer vision, and deep learning. He was a co-recipient of the 18th International Symposium on Communications and Information Technologies (ISCIT 2018) Best Paper Award in 2018.