IET Intelligent Transport Systems

Research Article

IET Journals
The Institution of Engineering and Technology

Trajectory tracking and prediction of pedestrian's crossing intention using roadside LiDAR

ISSN 1751-956X
Received on 21st May 2018
Revised 21st October 2018
Accepted on 11th December 2018
E-First on 9th January 2019
doi: 10.1049/iet-its.2018.5258
www.ietdl.org

Junxuan Zhao[1], Hao Xu[2], Jianqing Wu[2], Yichen Zheng[2], Hongchao Liu[1] ✉

[1]*Department of Civil, Environmental, and Construction Engineering, Texas Tech University, Lubbock, Texas, USA*
[2]*Department of Civil and Environmental Engineering, University of Nevada, Reno, Reno, Nevada, USA*
✉ *E-mail: hongchao.liu@ttu.edu*

Abstract: Trajectory tracking and crossing intention prediction of pedestrians at intersections are important to intersection safety. Recently, on-board video sensors have been developed for detection of pedestrians. However, both the detection range and operating environment of video-based systems seem to be constrained by the advancement of image-processing technologies. Additionally, on-board systems cannot alarm pedestrians to take evasive actions when at risk, a feature which is critical to saving lives. This paper summarises the authors' practice on using roadside LiDAR sensors to monitor and predict pedestrians' crossing intention, as part of an ongoing effort to develop a pioneering LiDAR-based system to systematically reduce pedestrian and vehicle collisions at intersections. The LiDAR sensors were installed at intersections to collect pedestrian data such as presence, location, velocity, and direction. A new method based on deep autoencoder – artificial neural network (DA-ANN) was used to process data and predict pedestrian crossing intention. The case study shows the proposed model is about 95% prediction accuracy and computational efficiency for real-time systems. The roadside LiDAR system has great potential to significantly reduce vehicle-to-pedestrian crashes both at intersections and non-intersection areas, either used as a stand-alone system or in conjunction with the connected V2I and I2V technologies.

1 Introduction

According to the National Highway Traffic Safety Administration (NHTSA) report in 2015, 5376 pedestrians were killed in traffic crashes in the USA [1]; additionally, nearly 129,000 pedestrians were treated in emergency rooms for non-fatal crash-related injuries [2]. Pedestrians are 1.5 times more likely than passenger vehicle occupants to be killed in a car crash on each trip [3]. In terms of crash location and time, the percentages of fatal crashes occurred at non-intersection location and during the night are 69 and 72%, respectively [4]. In recent years, advanced driver assistance systems (ADAS) have been explored and developed for improving safety for both drivers and pedestrians. Examples of ADAS include blind spot alert [5], the pedestrian protection system [6], driver drowsiness detection [7], automatic parking [8] etc. In general, crash avoidance is based on evasive actions from the pedestrian side and preventive actions from the vehicle side. Thus far, these technologies have been primarily based on the on-board sensors which have some inherent limitations such as lack of pedestrian warning mechanisms to alarm pedestrians to take evasive actions when at risk. In addition, the on-board sensing systems may become ineffective when blocked by other vehicles and roadside objects. These concerns, in conjunction with the fact that it will be a long time before all vehicles are equipped with advanced assistance systems, warrant the development of roadside systems that can accurately monitor pedestrians and at the time of hazardous events, warn the road users to take evasive actions in a timely fashion.

Technically, the most common approach for an on-board system is the use of video clips and image-processing technology to monitor pedestrian crossing behaviour and make predictions. In order to extract useful information from image data, numerous models and computing algorithms have been developed to predict whether a pedestrian is going to cross a road. Features used for making such a prediction include the location, speed, acceleration, pose, face direction, and other characteristics of a pedestrian [9–13]. The problem is, however, such a complex task requires a high-computational cost associated with detecting, classifying, and tracking target objects from a large number of candidate image regions, especially for 3D video data processing [14]. The high-performance computation requirement can be met by on-board systems but is difficult for roadside systems. In addition, it cannot be guaranteed that every pedestrian and obstacle can be observed from the vehicles under different light conditions. The performance of the video cameras is influenced by illumination condition, which means the video quality may be lower during the night compared with daytime [15]. In this regard, new infrastructure-based sensing systems need to be developed for providing real-time trajectories (e.g. presence, location, velocity, and direction) of all road users. The roadside LiDAR data will become an important data source of real-time emergency alert and crash-avoidance warning.

Light Detection and Ranging (LiDAR) sensors installed at intersections and along roadsides can be an innovative approach to solving the above issues. LiDAR sensors detect the position of objects by actively shooting laser beams and measuring the travel time after reflection. The LiDAR sensors work stably and reliably in both daytime and night-time under different environmental conditions. A 3D LiDAR sensor can scan the 360° 3D surrounding objects with high accuracy. With the advantages of reliability, accuracy, a wide covering range, and a dramatically reduced price, LiDAR sensors have become a promising component of transportation infrastructures, especially for autonomous and connected vehicle applications. The VLP-16 LiDAR sensor, a cost-efficient 16-line LiDAR manufactured by Velodyne, was used in this research as an example, while the proposed procedure and methods can be migrated to roads with other types of LiDAR sensors. With the VLP-16 sensor, 16 lasers are rotated horizontally by an internal motor with a speed of 5 to 20 rotations per second. The 360° laser point clouds are generated after each 360° 3D scan rotation. The output LiDAR data include the location information of measured points relative to the LiDAR sensor in XYZ coordinates, intensity, laser ID, azimuth, the distance between a data point and the sensor, adjusted time, and timestamp [16]. Since the data collected from roadside LiDAR sensors are 3D point clouds, existing methodologies developed for image processing do not work for the LiDAR data processing directly. Previous studies conducted by the authors have already achieved automatic background filtering, automatic lane identification, pedestrian and

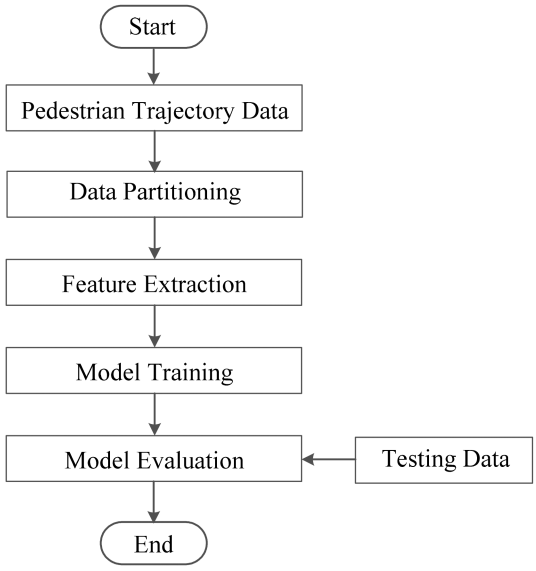

Fig. 1 *Flowchart of the proposed pedestrian crossing prediction method*

vehicle detection and tracking, and data integration from multiple LiDAR sensors [17–22]. These results lay a solid foundation for the research in predicting a pedestrian's intention for crossing. As pedestrians often follow similar paths and motion patterns at a specific location, historical pedestrian trajectory data at a site can help to learn pedestrians' crossing patterns, then be used to predict pedestrian crossings.

Here, a method based on deep autoencoder-artificial neural network (DA-ANN) was developed to predict whether pedestrians walking along the sidewalk will cross the road with a high prediction accuracy. Performance evaluation with field data proved that the DA-ANN model has a higher prediction accuracy than the traditional ANN model. In addition, the DA-ANN models trained by the proposed methods could provide prediction of a pedestrian's intention for crossing/non-crossing at different locations relative to the crossing facilities (intersections or crosswalks). The trained models can be used for predicting pedestrians' crossing intention and providing timely alerts to both pedestrians and vehicles via roadside or onboard warning systems. This study is also an important preparation for future incorporation of infrastructure-based information systems into autonomous driving environment.

This paper is structured as follows: Section 2 presents the methodology for pedestrian crossing prediction. Section 3 introduces a comprehensive case study to show the model training process and performance evaluation. Section 4 concludes the findings and future work.

2 Proposed prediction model

The proposed pedestrian crossing prediction model is based on DA-ANN, and the procedure is demonstrated by the flowchart in Fig. 1. The process starts with the pedestrian trajectory data from historical detection and tracking trajectories. The data partitioning step is to divide trajectory data into different regions along the sidewalk according to the distance to the crossing facilities and assign labels to trajectories. A deep autoencoder is then used to extract valid features and reduce the data dimensionality. The trajectory data with extracted features and corresponding crossing/non-crossing labels are the inputs to the ANN model. The DA-ANN model was first trained with a training dataset and then evaluated with a testing dataset. The traditional ANN model was also trained and evaluated with the same training and testing datasets, thus comparing the prediction performance with the proposed DA-ANN model.

2.1 Pedestrian trajectory acquisition

For processing the roadside LiDAR data, the authors have developed a data processing procedure from streaming roadside LiDAR data to real-time trajectories (e.g. presence, location, speed, and direction) of road users. The data processing procedure is shown in Fig. 2.

- *Background Filtering:* The goal of background filtering is to keep as many road users' data points as possible, and to exclude the background points at the same time. To accomplish this goal, an automatic 3D-density-statistics-background-filtering algorithm was developed. Four major steps are frame aggregation, points statistics, threshold learning, and real-time filtering [17, 18].
- *Lane Identification:* The lane identification algorithm is to automatically identify boundaries of traffic lanes. Lane boundaries are critical information for pedestrian crossing prediction. The method aggregates vehicle trajectories in a certain time interval to learn the location of vehicles' paths, and then identifies the 3D coordinates of lane boundaries [19].
- *Road User Clustering:* In the previous research, a revised DBSCAN (density-based spatial clustering of applications with noise) method was used for clustering LiDAR data points into road user objects. Two key parameters – minimal points and search radius, were adjusted according to the different point densities and distances to the roadside LiDAR sensors [20].
- *Pedestrian/VehicleClassification:* A backpropagate ANN model was trained for classifying pedestrian and vehicle clusters. Three extracted features were used to identify pedestrians and vehicles, including the total number of points in a cluster, distance to the LiDAR, and spatial distribution direction of clustered points [20].
- *Tracking:* Tracking is to identify the same object in continuous data frames. The nearest distance method was applied for object association based on the distance between an object in the previous frames to all objects in the current frames, and the time difference between the tracked two frames. After objects were matched, a discrete Kalman filter tracking method was used to track objects. The position/speed information of the object in the previous frame and the position/speed information of the matched object in the current frame were the inputs of the Kalman filter to estimate the status of the object in the current frame [21, 22].
- *Trajectory Data of Road Users:* After executing the above steps, trajectory information of pedestrians and vehicles can be obtained. Speed was calculated using the distance and time difference of two consecutive frames and the speed limit was used to filter the outliers. To get discrete direction information, the direction and distance from the object to the LiDAR sensor (whose location is at the origin) were calculated in a polar coordinate system. The outputs from the LiDAR data processing procedure were: XYZ position, total number of points, distance to LiDAR, tracking ID, frame number, velocity, direction, timestamp, and pedestrian/vehicle label of each target object. In the following procedure, only pedestrians' trajectory results were used as the initial data for pedestrian crossing prediction. The vehicles' trajectory data were not further analysed in this research.

2.2 Data partitioning

Even though pedestrians may have different motion behaviours than vehicles, their movements follow certain patterns when pedestrians are walking along the sidewalk and planning to cross the road. In general, pedestrians walk towards the road curb with a lower speed and change directions if they are going to cross, while non-crossing pedestrians maintain the speed and direction when walking along the sidewalk. In addition, the differences between crossing and non-crossing motion patterns change as pedestrians are at different distances from crossing facilities. Therefore, a region division along the sidewalk was conducted. In Fig. 3, a linear boundary is used for dividing the road area (vehicle lanes) and the sidewalk area. The red point C on the boundary represents the starting point of the crosswalk. Along the sidewalk, areas are further divided into N rectangular regions: R_1, R_2, …, R_N. The widths (w) of the rectangular regions are the same and can be

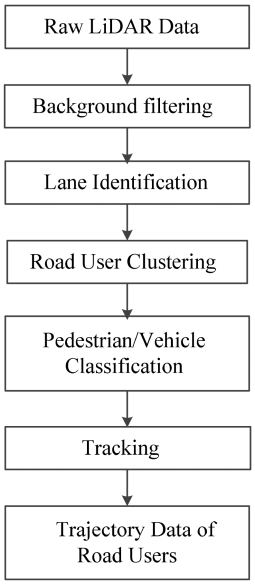

Fig. 2 *LiDAR data processing procedure for trajectory acquisition*

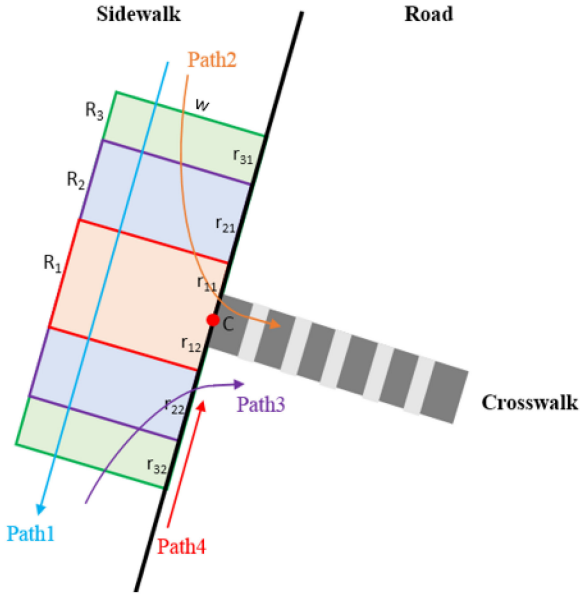

Fig. 3 *Region division (N = 3) and sample paths*

configured according to the sidewalk's width. The lengths (r) of the rectangular regions represent different distances to the crosswalk along the sidewalk direction. For example, the sidewalk only exists at one side of the crosswalk ($r_{i1} = 0$, $i = 1, 2, \ldots, N$) or the whole sidewalk is not symmetrical to the crosswalk ($r_{i1} \neq r_{i2}$, $i = 1, 2, \ldots, N$). On-site observations showed that not all crossings took place at the crosswalk, some pedestrians chose to cross before reaching the crosswalk. Therefore, the size of the rectangular regions was carefully designed so that they can cover all the pedestrians with different crossing locations, and the amount of training data (crossing and non-crossing) in each region is different. For different scenarios, both the number and the size of rectangles (i.e. the distribution of proportion of the pedestrians) may vary for the best performance. The trained prediction models are designed to use in corresponding regions.

After division, pedestrian trajectory data are grouped into different regions. It should be noted that the region R_{i-1} is included in the region R_i ($i = 2, 3, \ldots, N$), but the data belong to region R_{i-1} are not counted in the region R_i. As the location of the crosswalk is fixed, the distance between each trajectory point and the crosswalk (point C) can be calculated, which will be used as a feature in the following steps.

A label will be assigned to each trajectory point after data partitioning. There are only two trajectory point labels – crossing (label = 1) and non-crossing (label = 0). If the record of the selected trajectory segment in a specific region ends with crossing in the same region, the trajectory points in this region are labelled as crossing, otherwise, the data are given the non-crossing labels. The criteria for determining potential crossing behaviour are the distance between the pedestrian's last location in each region and the boundary of sidewalk. If the distance is less than a certain threshold, the pedestrian is considered to be crossing. This distance threshold is determined by the average speed of pedestrians and the rotation frequency (or the time duration for one data frame) of LiDAR sensors. Larger distance thresholds may result in mistaken grouping of non-crossing pedestrians into the crossing category, while smaller distance thresholds may lead to erroneous classification of crossing pedestrians into the non-crossing category. In other words, the accuracy of labelling is determined by the threshold selection, and the label information will be used as one of the inputs in the following model training and evaluation. In addition, if the trajectory is shown on the road's side, it indicates the pedestrian has already entered the road area. In this case, the data label is 1 and a note should be sent immediately to indicate the prediction is unnecessary. In Fig. 3, there are four sample paths. Path 1 goes through region 3, 2, and 1 without crossing behaviour, so the labels for the data in each region are all equal to 0; Path 2 passes through region 3 and region 2, and finally crosses the road from region 1, so the data in region 1 have the label 1 but the trajectory points in region 2 and region 3 are labelled 0. It is a similar case for Path 3, which has label 1 for the data in region 2. Path 4 shows an unsafe situation, all the data are labelled as 1 and saved in the warning dataset. The proposed data partitioning method helps the following model training procedure to obtain better prediction accuracy in each region.

2.3 Deep autoencoder – artificial neural network model

The backpropagation (BP)-neural network is a multilayer feed-forward artificial neural network [23]. It is composed of an input layer, hidden layer(s), an output layer, and neurons in each layer. The input data are fed into the input layer. Then, the activity of each hidden layer is determined by the inputs and the weights that connect the input layer and hidden layer. A similar process occurs between the hidden layer and output layer. The transmission from one neuron in one layer to another neuron in the next layer is independent. The output layer produces the estimated outcomes. The comparison information (error) between target outputs and estimated outputs is given back to the input layer as a guide to adjust the weights in the next training round. Through this iteration process, the neural network gradually learns the inner relationship between input and output by adjusting the weights for each neuron in each layer to reach the best accuracy. When the minimal error is reached, or the number of iterations is beyond the predefined value, the training process is terminated with fixed weights [24].

To train a good ANN model, features play an important role. However, some of the features may be redundant or correlated, which results in long processing time and overfitting/underfitting issues in the model. Deep autoencoder is an efficient method for finding low-dimensional representations of the input dataset based on deep learning. The extracted features can represent the characteristics of the data in a more efficient and abstract way. Thus, deep autoencoder can be considered as a pre-training process for the ANN model.

The deep autoencoder is a specific deep neural network, it has three layers: input, hidden, and output. Autoencoder is an unsupervised learning process that tries to learn representations of the inputs in a way that makes it possible to reconstruct the more representative inputs based on some intermediate representations and encoding/decoding processes [25]. A basic deep autoencoder backpropagates the reconstruction error through the network and updates its weights. The error minimisation is usually done by stochastic gradient descent [26]. The deep autoencoder part in Fig. 4 shows a learning procedure for a deep autoencoder. The input data are fed into the encoder and a compressed feature vector

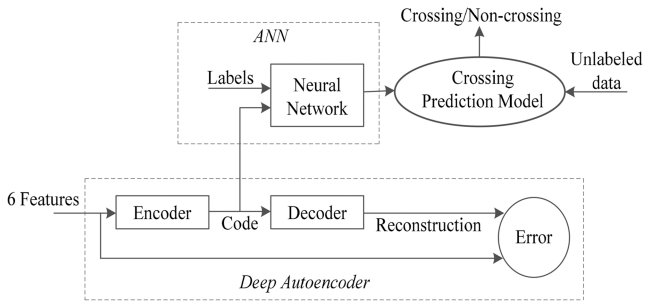

Fig. 4 *DA-ANN training and evaluation process*

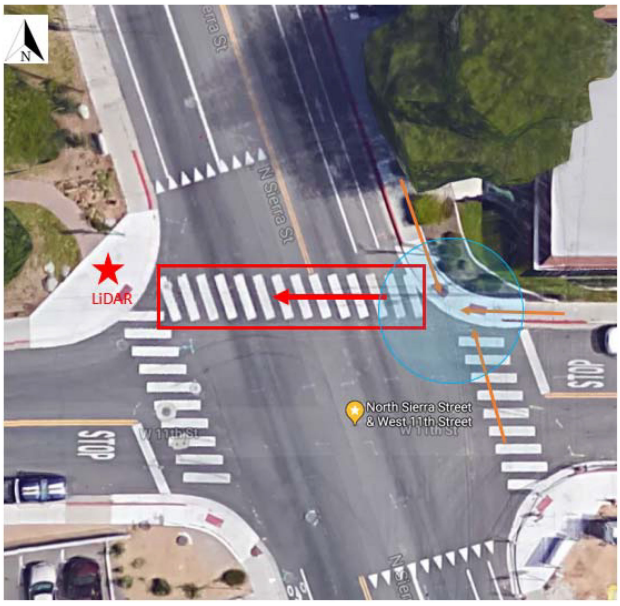

Fig. 5 *Google Map of the selected intersection*

(Code) is then calculated. The obtained Code is an abstract representation of the input data with lower dimensions. To validate the quality of input reconstruction, the decoder is applied to the Code. If the output of the decoder is similar to the input, the Code from the encoder can be used for representing the input data. Otherwise, the deep autoencoder adjusts the network's parameters to reach minimal reconstruction error.

In this research, six features and one label (crossing/non-crossing label) are used for DA-ANN model training and evaluation. Five features (XYZ position, velocity, direction) are obtained directly from trajectory acquisition. The 6th feature (distance to the crosswalk) and label information are calculated based on the layout of the sidewalk and crosswalk. In summary, as shown in Fig. 4, the inputs of the deep autoencoder are the six features and the outputs are compressed features (Code). This Code and label information are then fed into the neural network to train a crossing prediction model. Given an unlabeled trajectory to the trained prediction model, a crossing/non-crossing label will be assigned.

3 Case study

3.1 Model training

At the test site, a VLP-16 LiDAR sensor was permanently installed on the top of a pedestrian signal (approximate 7 ft height above the ground) at the intersection of North Sierra Street and 11th Street in Reno, Nevada, with 10 Hz rotation frequency (0.1 second for one data frame). The Google Map of the data collection site is shown in Fig. 5. The case study only analysed the pedestrian trajectories on the east side of the North Sierra Street. The selected crosswalk is marked with a red rectangle. Three main pedestrian approaching directions were from southbound, northbound, and westbound. The data collection site was selected for sufficient pedestrian crossings to train and evaluate the performance of the proposed methods.

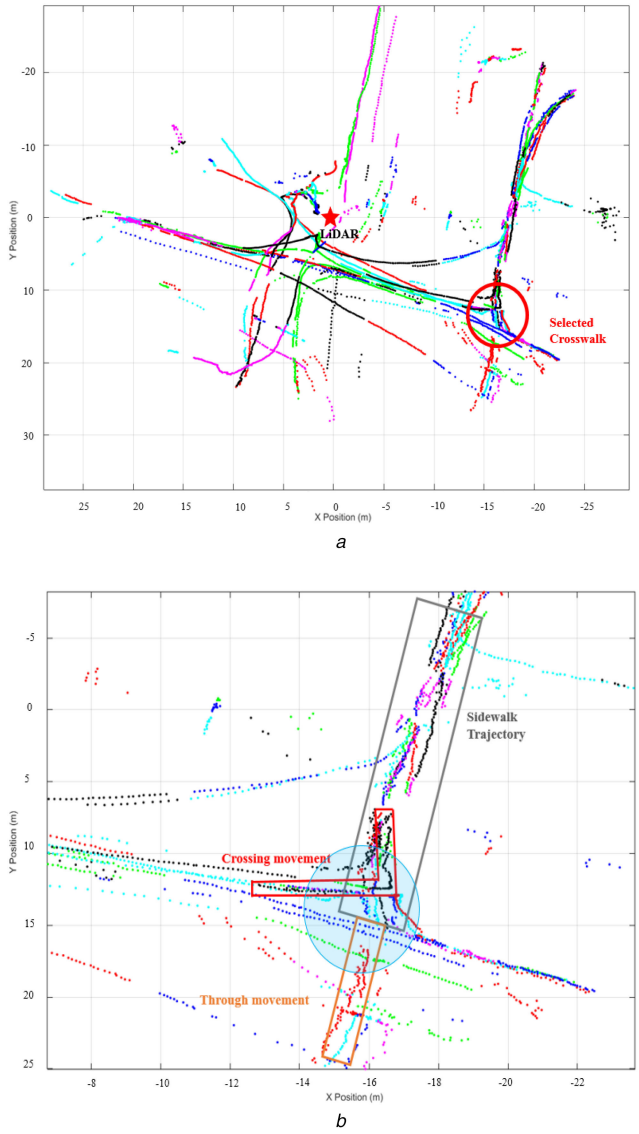

Fig. 6 *Pedestrian trajectories*
(a) Example pedestrian trajectories,
(b) Detailed example pedestrian trajectories

Totally, 9000 frames data streamed from the LiDAR sensor were processed with the steps of background filtering, lane identification, road user clustering, pedestrian/vehicle classification and tracking. As a result, pedestrian trajectory data including XYZ position, total number of points, distance to LiDAR, tracking ID, frame number, velocity, direction, timestamp, and pedestrian/vehicle label were obtained. In terms of tracking evaluation, the authors compared extracted trajectories of pedestrians and vehicles from the proposed algorithms with the trajectories from the recorded video. It shows that approximately 95% tracking accuracy can be achieved within about 30 m (in one direction) detection range from the LiDAR sensor. The main reasons for tracking failure were the limited number of data points collected for those objects at a far distance and occlusion issues. In Fig. 6*a*, the LiDAR sensor is at the origin with coordinate (0,0) and the starting location of the selected crosswalk is marked with a red circle. Different colours of trajectories refer to different tracking IDs, that is, different pedestrians. The through and crossing movements of pedestrians are clearly shown in Fig. 6*b*.

The next step is to group trajectories into different regions. Fig. 7*a* shows the data before data partitioning. Different pedestrian trajectories were represented by different colours; In Fig. 7*b*, the black point C on the road-sidewalk boundary represents the intersecting position of the sidewalk and the chosen crosswalk. The sidewalk area was divided into three ($N = 3$) regions:

Region 1 (Green): 4 m × 4 m($r_{11} = r_{12} = 2$ m)
Region 2 (Blue): 10 m × 4 m ($r_{21} = r_{22} = 5$ m)
Region 3 (Black): 20 m × 4 m ($r_{31} = r_{32} = 10$ m)
Note: Length × Width

The width of the selected sidewalk is about 2.5 m. The width of the rectangular region was chosen as 4 m in order to make sure the pedestrians walking along both edges of the sidewalk can be included. The lengths of the three regions were 4, 10, and 20 m (2, 5, 10 m at one side) based on the collected data. On-site observations showed that some pedestrians crossed approximately 2, 5, 10 m ahead of the crosswalk.

For the labelling, since the time for one data frame is 0.1 second and the authors used 1 m/s as the average pedestrian speed, the walking distance of a pedestrian during one frame is about 0.1 m. Considering the walking direction and potential occlusion issues, the authors chose 0.3 m as the distance threshold in the research. In Fig. 7*b*, data with crossing labels are marked with black dots. If the trajectory points are in the road area (vehicle lanes), then a red square marks the data point, indicating that this pedestrian is in an unsafe location (already hit the vehicle lane). In the LiDAR XYZ coordinate system (LiDAR is located at the origin, the unit is meter), the coordinate of the crossing point C is (-13, 15) and the equation of the road boundary is $y = 5x + 80$. The distance between each trajectory point and the crosswalk (point C) can be calculated and added to the dataset.

In the model training step, two autoencoders were stacked together with a softmax layer to form a deep autoencoder – artificial neural network model (as shown in Fig. 8). The autoencoder is composed of an encoder followed by a decoder. The encoder maps an input to a hidden representation, and the decoder reverses this mapping to evaluate the hidden representation. The extracted features generated from the first autoencoder are the input data for the second autoencoder [27]. In this research, the sizes of the hidden layer for autoencoder1 and autoencoder2 were 6 and 5, respectively, after trial and error. The original training data had six features in total. After training the autoencoder1, the outputs with six reconstructed features were fed into the autoencoder2. After using the autoencoder2, the number of data features was reduced to five. Finally, a neural network with softmax output layer was trained for crossing/non-crossing classification. Since the input data were scaled during the training process, it did not tell which five features were ultimately selected.

According to the region division, those qualified data with labels were grouped into three region categories. A pedestrian crossing prediction model was trained for each region. For Region 1, 136 data samples were divided into two classes: 81 crossing data and 55 non-crossing data. For Region 2, there were 56 crossing samples and 148 non-crossing samples. For Region 3, the number of crossing data (3) was too small compared with non-crossing data (184). So, training model 3 for the Region 3 was excluded from the training and evaluation, due to the extreme unbalanced data. For these training datasets, data were further divided into three categories for Training (70%), Validation (15%), and Testing (15%) steps. In the end, two DA-ANN pedestrian crossing prediction models (model 1 and model 2) were trained for Region 1 and Region 2.

3.2 Model evaluation

To validate and evaluate the performance of the trained pedestrian crossing prediction models, a testing dataset collected from the same intersection that had not been seen by the trained models was applied to the obtained models. For model 1, the testing dataset included 37 crossing and 21 non-crossing samples. For model 2, the crossing and non-crossing data were 23 and 64, respectively. The trained prediction models were directly applied to the new dataset and the results are shown in Table 1 and Fig. 9. Prediction accuracies for model 1 and model 2 were 96.6 and 94.3%. All the crossing pedestrians in both testing datasets were predicted correctly. The following causes were found for the non-crossing pedestrians to be mistakenly predicted as crossing:(i) The trajectories of non-crossing pedestrians were near the road curb;

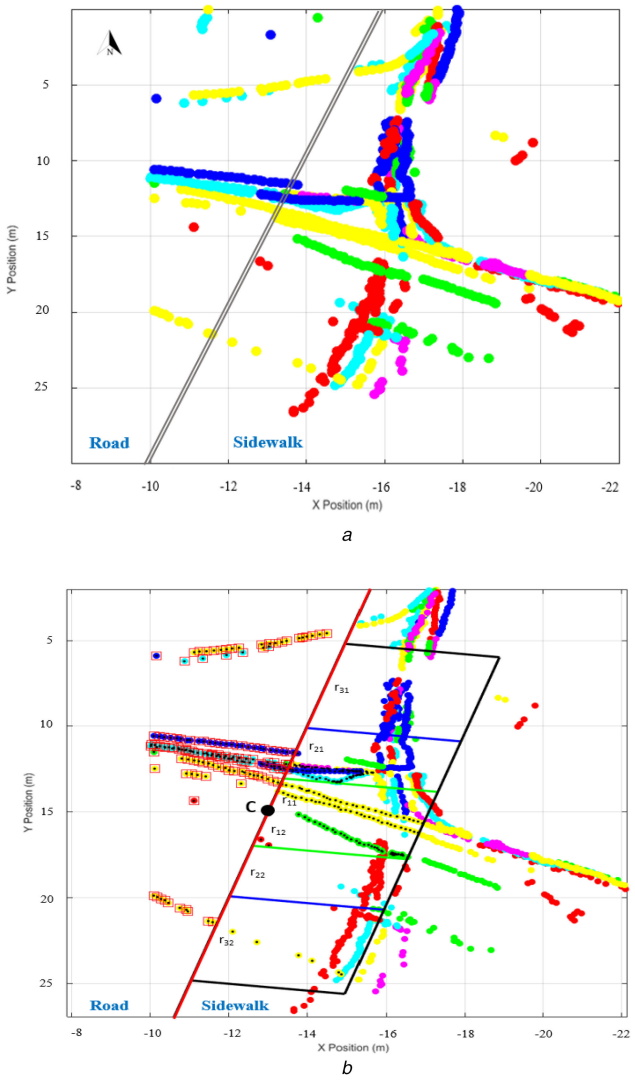

Fig. 7 *Pedestrian trajectories with data partitioning*
(a) Pedestrian trajectories before data partitioning,
(b) Pedestrian trajectories after data partitioning

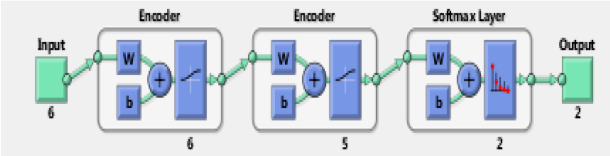

Fig. 8 *Stacked deep network*

Table 1 Performance evaluation of trained DA-ANN models

Testing dataset	Testing dataset 1	Testing dataset 2
total number of data	58	87
crossing data	37	23
non-crossing data	21	64
predicted crossing	37	23
predicted non-crossing	19	59
accuracy	96.6%	94.3%

and(ii) The velocity was lower than the regular walking speed of non-crossing pedestrians.

The model was trained offline using historical pedestrian trajectory data. To examine the model's property in real-time calculation, the authors recorded the computation time on a regular Dell desktop (Intel Core i7-4790 CPU (3.60 G Hz) and 16 GB of RAM), it took only about 0.28 milliseconds to process one single

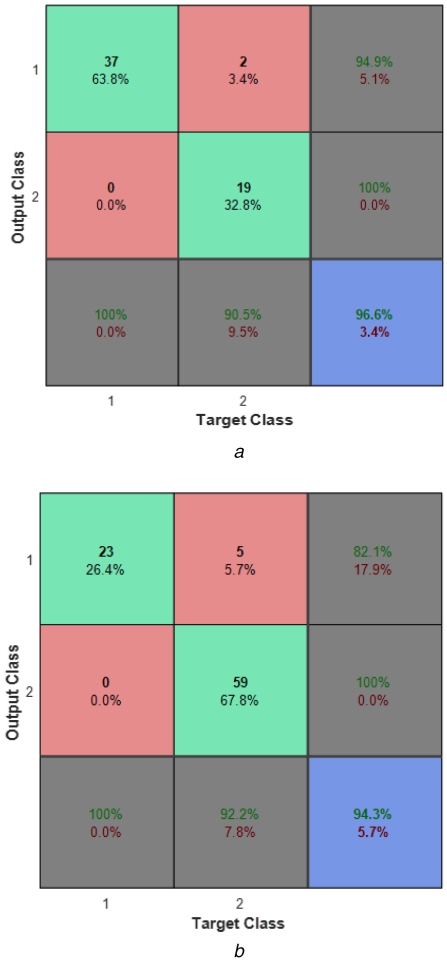

Fig. 9 *Confusion matrix for models*
(a) Confusion matrix for model 1,
(b) Confusion matrix for model 2

trajectory data point and make prediction, which was ideal for real-time applications.

To further evaluate the performance of the trained DA-ANN prediction models, the authors also used the same training datasets and the same testing datasets to train two basic backpropagate ANN models. The determined ANN model structure and functions are summarised as follows:

- *Feature selection*: XYZ position, velocity, direction, and distance to the crosswalk.
- *The number of hidden layers*: 1
- *The number of hidden neurons*: 7
- *Training function*: Scaled conjugate gradient backpropagation.
- *Transfer function*: Softmax transfer function.
- *Performance function*: Mean squared error performance function.

As there is no direct approach to determining the number of hidden layers and hidden neurons, the authors started with some values obtained by empirical equations/references [28], and then adjusted the values to select the optimum. Usually a network with more than one hidden layer is extremely hard to train. The six features used in model training came from the original trajectory data without any pre-training. The testing results of the basic ANN models are shown in Table 2. Not all the crossing pedestrians were predicted correctly, and this results in a potential safety risk if the traditional ANN results are used for warning. The prediction accuracy comparisons between DA-ANN and basic ANN models are listed in Table 3. The result shows that the DA-ANN models have improved the prediction accuracy about 6% compared with the basic ANN model in the two regions.

Table 2 Performance evaluation of trained basic ANN models

Testing dataset	Testing dataset 1	Testing dataset 2
total number of data	58	87
crossing data	37	23
non-crossing data	21	64
predicted crossing	34	13
predicted non-crossing	18	64
accuracy	89.7%	88.5%

Table 3 Performance comparison of trained models

Accuracy	Model 1	Model 2
DA-ANN	96.6%	94.3%
basic ANN	89.7%	88.5%
improve	+6.9%	+5.8%

4 Conclusions

This paper presented a new method based on deep autoencoder – artificial neural network to predict pedestrian crossings along the sidewalk. It is an innovative way to predict pedestrians' crossing intention in different regions from crossing facilities using roadside LiDAR sensors. The major steps include pedestrian trajectory data extraction, data partitioning, feature extraction, model training, and evaluation. A comprehensive case study was presented to demonstrate the effectiveness of the proposed algorithm using real data, which also showed the trained DA-ANN model outperforms the basic ANN models.

The study could be a valuable input for future development of real-time cooperative systems in which all road users will be connected via dedicated short-range communications (DSRC) unit and/or Wifi/4G technologies. Currently, it can be used as a fundamental element to develop infrastructure-based systems where warnings can be delivered to road users from roadside equipment. It could also be used to improve pedestrian signals by providing added functions such as automatic trigger of the pedestrian phase and real-time phase extension.

As part of an ongoing project, current and future works along this line include:(i) testing the algorithm at busier intersections with more pedestrian volumes;(ii) adding more LiDAR sensors to get better quality trajectory data;(iii) developing and testing communication means to synchronise all road users; and(iv) field testing of the entire system.

5 Acknowledgments

This work was supported by the SOLARIS Institute, a Tier 1 University Transportation Center (UTC) [Grant No. DTRT13-G-UTC55] and matching funds by the Nevada Department of Transportation (NDOT) [Grant No. P224-14-803/TO #13]. The authors gratefully acknowledge this financial support. This research was also supported by engineers with the Nevada Department of Transportation, the Regional Transportation Commission of Washoe County, Nevada, and the City of Reno.

6 References

[1] National Highway Traffic Safety Administration: '*Traffic safety facts 2015 data – pedestrians*' (US Department of Transportation, National Highway Traffic Safety Administration, Washington, DC, 2017). Publication no. DOT-HS-812-375. Available at https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812375

[2] Centers for Disease Control and Prevention: '*WISQARS (Web-based injury statistics query and reporting system)*' (US Department of Health and Human Services, CDC, Atlanta, GA, 2015). Available at http://www.cdc.gov/injury/wisqars

[3] Beck, L.F., Dellinger, A.M., O'Neil, M.E.: 'Motor vehicle crash injury rates by mode of travel, United States: using exposure-based methods to quantify differences', *Am. J. Epidemiol.*,2007, **166**, pp. 212–218

[4] National Highway Traffic Safety Administration: 'Traffic safety facts: 2013 data: pedestrians', 2015. Available at https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812124

[5] Singh, S., Meng, R., Nelakuditi, S., *et al.*: 'Sideeye: Mobile assistant for blind spot monitoring'. 2014 Int.Conf. on Computing, Networking and Communications (ICNC), Honolulu, HI, Feb. 2014, pp. 408–412
[6] Mammeri, A., Zuo, T., Boukerche, A.: 'Extending the detection range of vision-based driver assistance systems application to pedestrian protection system'. Global Communications Conf. (GLOBECOM), IEEE, Austin, TX, Dec. 2014, pp. 1358–1363
[7] Čolić, A., Marques, O., Furht, B.: 'Design and implementation of a driver drowsiness detection system: A practical approach'. 2014, Int.Conf. on Signal Processing and Multimedia Applications (SIGMAP), Vienna, Austria, Aug. 2014, pp. 241–247
[8] Ma, S., Jiang, H., Han, M., *et al.*: 'Research on automatic parking systems based on parking scene recognition', *IEEE. Access.*, 2017, **5**, pp. 21901–21917
[9] Gandhi, T., Trivedi, M.M.: 'Pedestrian collision avoidance systems: A survey of computer vision based recent studies'. Intelligent Transportation Systems Conf., 2006. ITSC'06, Toronto, Canada, Sep. 2006, pp. 976–981
[10] Quintero, R., Almeida, J., Llorca, D.F., *et al.*: 'Pedestrian path prediction using body language traits'. Intelligent Vehicles Symp.Proc., Dearborn, MI, June 2014, pp. 317–323
[11] Brouwer, N., Kloeden, H., Stiller, C.: 'Comparison and evaluation of pedestrian motion models for vehicle safety systems'. 2016 IEEE 19th Int.Conf. on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, Dec. 2016, pp. 2207–2212
[12] Keller, C.G., Gavrila, D.M.: 'Will the pedestrian cross? A study on pedestrian path prediction', *IEEE Trans. Intell. Transp. Syst.*, 2014, **15**, (2), pp. 494–506
[13] Rehder, E., Wirth, F., Lauer, M., *et al.*: 'Pedestrian prediction by planning using deep neural networks'. Proc. Int. Conf. Robotics and Automation, Brisbane, Australia, Sep. 2018, pp. 1–5
[14] Shi, J., Mueller, H.C., Marx, M.: 'Pedestrian detection and localization using antenna array and sequential triangulation'. IEEE Conf.Intelligent Transportation Systems, Vienna, Austria, Sep. 2005, pp. 126–130
[15] Mukhtar, A., Xia, L., Tang, T.B.: 'Vehicle detection techniques for collision avoidance systems: a review', *IEEE Trans. Intell. Transp. Syst.*, 2015, **18**, pp. 2318–2338
[16] Velodyne LiDAR: '*VLP-16 in VLP-16 manual: user's manual and programming guide*'(Inc.:San Jose, CA, USA,2016)Available at http://velodynelidar.com/docs/manuals/VLP-16%20User%20Manual%20and%20Programming%20Guide%2063-9243%20Rev%20A.pdf
[17] Wu, J., Xu, H., Zheng, J.: 'Automatic background filtering and lane identification with roadside LiDAR data'. 2017 IEEE 20th Int.Conf.Intelligent Transportation Systems, Yokohama, Japan, March 2018, pp. 1–6
[18] Wu, J., Xu, H., Sun, Y., *et al.*: 'Automatic background filtering method for roadside lidar data', *Transp. Res. Rec., J. Transp. Res. Board*, 2018, **2018**, https://doi.org/10.1177/0361198118775841
[19] Wu, J., Xu, H., Zhao, J.: 'Automatic lane identification using the roadside LiDAR sensors', *IEEE Intell. Trans. Syst. Mag.*, 2018, https://doi.org/10.1109/MITS.2018.2876559
[20] Zhao, J., Xu, H., Wu, D., *et al.*: 'An artificial neural network to identify pedestrians and vehicles from roadside 360-degree LiDAR data'. Proc. of the 97th Transportation Research Board Annual Meeting, Washington, DC, January 2018
[21] Wu, J., Xu, H.: 'An automatic procedure for vehicle tracking with a roadside LiDAR sensor'. Proc. of the 97th Transportation Research Board Annual Meeting, Washington, DC, January 2018
[22] Sun, Y., Xu, H., Wu, J., *et al.*: '3-D data processing to extract vehicle trajectories from roadside LiDAR data', *Transp. Res. Rec., J. Transp. Res. Board*, 2018, **2018**, https://doi.org/10.1177/0361198118775839
[23] Theodoridis, S., Koutroumbas, K.: '*Pattern Recognition*' (Academic Press, USA, 2009, 4th edn.)
[24] Wang, S.C.: '*Interdisciplinary computing in Java programming*', vol. **743** (Springer Science & Business Media, Germany, 2012)
[25] Bengio, Y.: 'Learning deep architectures for AI', *Found. Trends Mach. Learn.*, 2009, **2**, (1), pp. 1–127
[26] Hinton, G.E., Salakhutdinov, R.R.: 'Reducing the dimensionality of data with neural networks', *Science*, 2006, **313**, (5786), pp. 504–507
[27] Huang, W.B., Sun, F.C.: 'Building feature space of extreme learning machine with sparse denoising stacked-autoencoder', *Neurocomputing*, 2016, **174**, pp. 60–71
[28] Panchal, G., Ganatra, A., Kosta, Y.P., *et al.*: 'Review on methods of selecting number of hidden nodes in artificial neural network'. *Int. J. Comput. Theory Eng.*, 2016, **3**, (2), pp. 332–337