RESEARCH ARTICLE

A Novel Spatio–Temporal Deep Learning Vehicle Turns Detection Scheme Using GPS-Only Data

MUSSADIQ ABDUL RAHIM [1], **SULTAN DAUD KHAN** [1], **SALABAT KHAN** [2],
MUHAMMAD RASHID [1], **RAFI ULLAH** [1], **HANAN TARIQ** [3], **(Graduate Student Member, IEEE),**
AND STANISLAW CZAPP [3], **(Member, IEEE)**

[1]Department of Computer Science, National University of Technology (NUTECH), Islamabad 44000, Pakistan
[2]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[3]Faculty of Electrical and Control Engineering, Gdańsk University of Technology, 80-233 Gdańsk, Poland

Corresponding author: Mussadiq Abdul Rahim (mussadiq.ar@gmail.com)

ABSTRACT Whether the computer is driving your car or you are, advanced driver assistance systems (ADAS) come into play on all levels, from weather monitoring to safety. These modern-day ADASs use various assisting tools for drivers to keep the journey safe; these sophisticated tools provide early signals of numerous events, such as road conditions, emerging traffic scenarios, and weather warnings. Many urban applications, such as car-sharing and logistics, rely on accurate and up-to-date road map data. Map generation methods use a variety of data sources, including but not limited to global positioning systems (GPS). In this research we propose a GPS-only data trajectory analysis and a novel scheme to convert GPS trajectory data to image-based data to train a custom Convolutional Neural Network (CNN) model. The empirical results with an extensive 5-fold cross-validation show that the proposed scheme identifies turn and not turn with more than 94% recall. It outperforms the existing turn detection schemes on two major frontiers, the required data and the accuracy achieved in detecting different driving behaviors.

INDEX TERMS Advance driver assistance systems, CNN, deep learning, GPS data, naturalistic driving, spatio–temporal window analysis, turn detection.

I. INTRODUCTION

Vehicle driver behavior entails multiple aspects of a driver's trip from one place to another. The measurable driver behavior contributes to various parts of Advanced Driver Assistance Systems (ADASs) to provide essential information for decision-making. Particularly driver trajectory analysis imparts solutions to numerous problems, including but not limited to map construction, business affair analysis, and driver behavior analysis. A vehicle trajectory is an ordered sequence of timestamps and sampled locations along a moving path. With the help of location providers such as Global Positioning System (GPS), it has become possible to collect vehicle trajectory information. The vehicle trajectory analysis has been previously used for various purposes, including but not limited to anomaly detection [1],

The associate editor coordinating the review of this manuscript and approving it for publication was Venkata Ratnam Devanaboyina [ID].

lane-change detection [2], behavior detection [3], accident detection [4], traffic event detection [5], user identification [6] through behavioral analyses [7], map generation [8], and outlier detection for location based services (LBS) [9].

Fu et al. [1] proposed a two-layer hierarchical clustering analysis that enables the identification of dominant pathways and lanes. The clustering findings also allow for the detection of unique pathways. Yu et al. [5], a unique approach of trajectory description is suggested to construct a semantic model for the automated identification of traffic infraction occurrences. Rahim et al. [7] presents a driver identification system using minimal data. The approach utilizes global positioning system (GPS) data to determine an individual's driving style. Rahim et al. [6] presents an event-driven system that uses GPS as its only data source. Quehl et al. [8] proposed a method for predicting vehicle trajectories based on maps that contain information about the behavior of traffic participants in a specific region. Wu et al. [9] proposed a

probabilistic model-based method for identifying trajectory outliers. Guo et al. [10] proposed a method for predicting the business affairs of a traveler based on the trajectory analysis. To assess vehicle behaviors, they focused on the segmentation of GPS trajectory data collected in logistics transportation. Guo et al. [11] used graph theory to analyze vehicle trajectories and extract various patterns through the movement of vehicles. They presented a graph-based method for analyzing trajectories as a complicated network. Using efficient and precise methods to compare trajectories is a crucial aspect of trajectory analysis. Sousa et al. [12] provided a summary of vehicle trajectory data, including models and preparation methods. Wang et al. [13] compared the performance of six widely-used trajectory similarity metrics using an a vehicle trajectory dataset. Magdy et al. [14] proposed a comparative analysis of frequently employed trajectory similarity metrics, highlighting their merits and shortcomings. Zheng [15] discussed multiple data mining techniques used for vehicle trajectory analysis. This survey investigated the relationships, correlations, and distinctions between current approaches. It also described the methods for transforming trajectories into various data types, including graphs, matrices, and tensors.

Schlechtriemen et al. [16] proposed method for identifying traffic participants' intentions to shift lanes. In a large-scale experiment using real-world data, they trained and validated the intention prediction model using more than 160 samples. They achieved an average accuracy of 93% for lane-change detection. Mandalia et al. [17] presented a support vector machine (SVM)-based method for estimating driving intentions. SVMs were particularly successful at spotting driver lane changes early on. They achieved an average accuracy of 89.22% for 5-second non-overlapping window. Ramanishka et al. [18], of which the dataset is used in this research, proposed a computer vision based scene annotation tool to facilitate research on understanding driver behaviour from untrimmed data sequences. A 98% accuracy is achieved, while it requires extensive hardware to run proposed machine learning model and application. Mao et al. [19] discussed measures to check similarities of trajectories in an urban environment with the effect of different transforms. According to the study, using trajectory data processing techniques, it is possible to mine the patterns of human activities and vehicle movement. This research presented a segment-based dynamic temporal warping technique for mining trajectory data. Wang et al. [20] used computer vision to analyze and differentiate between pedestrian and vehicle trajectories. They constructed semantic scene models based on long-term observations of the scene's moving items. The trajectories of automobiles and people are clustered initially, followed by clustering based on spatial and velocity distributions. Wei et al. [21] analyzed left-turn behaviors of the vehicle to provide better guiding lanes at left-turns at intersections. They presented a technique for detecting irregularities in a vehicle's approach by analyzing its velocity patterns. Chan [22] studied human behavior in traffic

---

Algorithm 1 GPSTurnDetect

▷ INPUT: 1. A chronologically ordered relation of GPS trajectories G with k tuples, where $x_i \in G$ for $i \leq k$, $x_i = (t_i, lat_i, lng_i, ori_i)$ 2. Window size $size$ in feet. 3. Number of rows of output matrix H^p. 4. Number of columns of output matrix W^p. 5. Constant of significant curve change c_{change}.

▷ OUTPUT: 1. An ordered set W of spatial feature matrices where each matrix represents an spatial window. 2. A set of labels lab^p representing the class Turn/Non Turn for each spatial feature matrix.

1: $win = \{\}$
2: $B = \text{createSpatialWindows}(G, size)$
3: $c^w = \text{length of the array } B$
4: **for all** x_i in G **do**
5:     $j = \text{getSpatialWindow}(x_i.lat_i, x_i.lng_i, B)$
6:     Append j to the win array
7: **end for**
8: Append win array as a column to the G
9: $lab^p = \{\text{null: for } j >= 1 \text{ and } j < c^w\}$
10: $W = \{c^w \text{ zero matrices each sized } H^p \text{ x } W^p\}$
11: **for all** i in range 1 to c^w **do**
12:     $G^i = \{G_j: \text{for } j < k \text{ and } j.win = i\}$
13:     $\mu_{\Delta o} = \frac{\sum_{h=1}^{size(G^i)-1} G_{h+1}^i.ori - G_h^i.ori}{size(G^i)}$
14:     **for all** h in range 1 to $size(G^i) - 1$ **do**
15:        $\Delta o = G_{h+1}^i.ori - G_h^i.ori$
16:        $m_h^i = \mu_{\Delta o} - \Delta o$
17:     **end for**
18:     **if** $\frac{P^{95}(m^i)}{P^{10}(m^i)} < 0$ **then** $lab_i^p = $ 'Not Turn'
19:     **else** $lab_i^p = $ 'Turn'
20:     **end if**
21:     **for all** g_j in G^i **do**
22:        $x_j, y_j = $ Convert Geo Coordinate to Cartesian Coordinates $g_j.lat_j, g_j.lng_j$
23:     **end for**
24:     curve = Fit a straight line on x, y
25:     **if** curve.recreate $- \{x, y\} < c_{change}$ **then**
26:        **if** $lab_i^p == $ 'Not Turn' **then** $lab_i^p = $ 'Turn'
27:        **end if**
28:     **end if**
29:     $W_i = \text{TrajectoryImageGeneration}(G^i, T^s, W^p, H^p, W^p, H^p)$
30: **end for**
31: **return** W, lab^p

---

through trajectory analysis and discussed criteria for safety warnings at intersections. Nowosielski and Forczmański [23] proposed a vision-based approach for the analysis of different vehicle trajectories. Santhosh et al. [24] surveyed different vision-based approaches for abnormal behavior detection in traffic. Ahmed et al. [25] investigate various trajectory and vision-based systems for various surveillance applications. Lin et al. [26] used graph positioning with a vision-based

approach to analyze the trajectory of different vehicles, they utilized the graph feature positioning for vehicles to detect and analyze trajectories of the vehicles from video frames. Bian et al. [27] surveyed different algorithms for trajectory analysis, including various contemporary machine learning algorithms. Sekh et al. [28] proposed an unsupervised trajectory clustering method to perform a ranking of trajectories to identify abnormal behaviors through vision-based object tracking. Kan et al. [29] performed traffic congestion analysis using the GPS data and road network data. Tang et al. [30] proposed a method to analyze the GPS trajectories and construct the intersection of roads through the trajectory analysis. Deo and Trivedi [31] classified trajectories into multiple maneuvering events using a proposed neural network model for this application. Salvo et al. [32] used unmanned aerial vehicle to capture traffic and perform an analysis of macro-level traffic flow using trajectories analyzed with computer vision.

This research aims to provide a deep learning-based solution for turn detection for driver assistance systems using trajectory analysis. This research work

1) Utilizes GPS-only data for trajectory data to detect turns in naturalistic driving.
2) Creates spatio-temporal windows of GPS trajectory allowing the trajectory data to apply convolutional deep learning for improved turn detection. A spatio-temporal window consists of spatial references showing the trajectory of the vehicle moving within that window. Whereas, the temporal part is reflected by how much time the vehicle stayed on a certain point within that trajectory within a window.
3) Employs CNN model to classify turns and non-turns without any image-based data.
4) Performs significantly well compared to different research works which utilize extensive data or require much more sophisticated hardware.

This paper is further divided into sections, *Spatio-temporal Vehicle Turn Detection* section describes GPS-only turn detection and proposes algorithms for conversion of the GPS data to spatial window images and labeling of the windows. *CNN-based Classification Model* section applies deep learning modeling for the classification of GPS-based windows into turn and non-turn. *Empirical Analysis and Results* section discusses an empirical study in detail and presents an analysis. Figure 1 illustrates the overall stages of the proposed research work in graphical form.

II. SPATIO–TEMPORAL VEHICLE TURN DETECTION

GPS data-only turn detection is proposed in algorithm 1, where the algorithm takes chronologically ordered GPS data G as input wherein each tuple consists of four basic data elements t time, *lat* latitude, *lng* longitude, and *ori* orientation. The elements are indexed in subscripts such as i^{th} tuple's orientation is denoted as ori_i. The rest of the inputs include spatial window size *size* in feet, and the size of the output matrix having height H^p and width W^p. The proposed
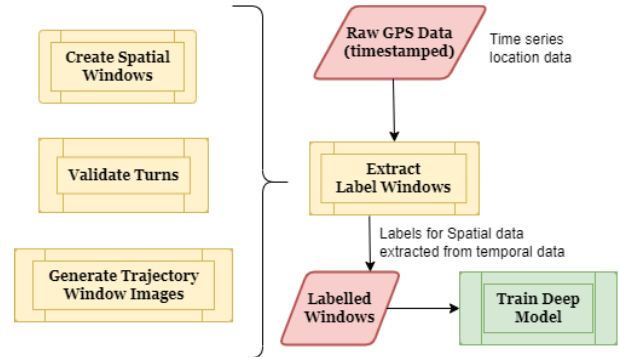

FIGURE 1. Spatio-temporal vehicle turn detection framework stages.

model uses spatial windows to detect turns through GPS-only data. For which the method createSpatialWindows is expressed in algortihm 2. Each spatial window is analyzed for a specified pattern of orientation *ori*. It uses getBoundingBox method to create a spatial fence on location for a given size. Whereas it tests whether a geographical location in a certain fence or not using inBoundingBox. It uses curve fitting as an additional measure to ensure reducing the number of false non-turns. Where a straight line is fit on the available cartesian coordinates converted from latitude and longitude available in G^i. Changes between recreation of the fitted straight line and trajectory segment of a certain window are compared to judge, whether the label needs updating or is correctly labeled. The constant c_{change} is used to validate the label at this stage. TrajectoryImageGeneration method used for a trajectory to matrix conversion in algorithm 1 is adapted from [33], which takes GPS trajectory, window sizes as input and returns a H^p x W^p-sized matrix representing trajectory in matrix form. The appendColumn method is a trivial method to append a data column to a relation, it takes two inputs, the name of the column and column data in form of a sequence. The getSpatialWindow method retrieves the identification number of a spatial window to which the given location belongs. The ground truth data is visualized in figure 2, these spatial windows form the ground truth when combined with the truth labels, these spatial windows were created using algorithm 2. The spatial windows as shown in this figure, show the trajectory of a vehicle in that particular window, whereas the higher values at certain points seen as bright yellow show that the vehicle stayed longer in the particular part, compared to the rest of the window. The labels assigned through algorithm 1 are then validated with the ground truth, as discussed in subsection *Dataset*.

A. CNN-BASED CLASSIFICATION MODEL
In this section, we provide details of the proposed deep learning model for the classification of trajectory images obtained through proposed algorithms (see sec. *Spatio-temporal Vehicle Turn Detection*). Trajectory images are binary in nature, where the white pixels represent the data points of trajectory and black pixels represent the background. Few samples of trajectory images are shown in
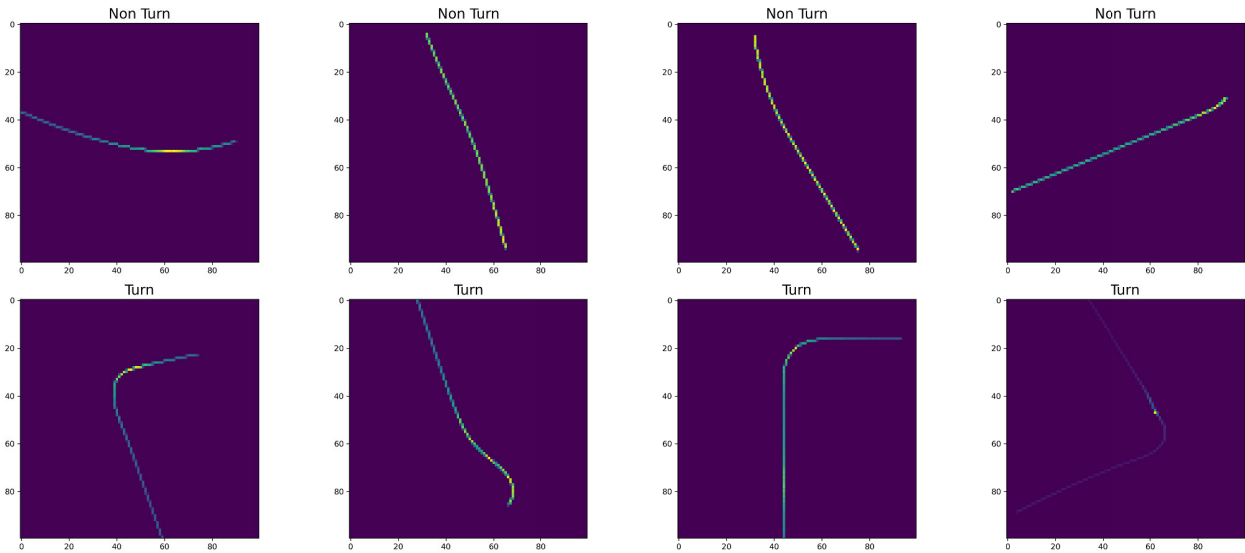
FIGURE 2. Visualization of spatial windows showing ground truth turns and non-turns.

figure 2. As obvious from figure 2, binary trajectory images are quite different from natural images. This may attribute to the following reasons: (1) Natural images contain rich texture and color information and consist of three channels: Red, Blue, and Green, while trajectory images significantly lack texture and color information. (2) Trajectory images, due to lack of texture and color information, suffer from inter-class similarities due to which different trajectory images may appear similar. Due to the aforementioned reasons, it is hard for existing deep learning models to draw decision boundaries and accurately classify images in desired categories. Deep learning models enjoyed tremendous success in various object classification, detection, and segmentation tasks in natural RGB images, however, their performance degrade when these models are directly employed to classify trajectory images. This may attribute to the fact that the receptive fields of these models are small due to which these models can not extract the rich contextual information required to classify trajectory images. This is due to the reason that we propose a shallow and effective deep learning model for the classification of trajectory images.

The detailed architecture of the proposed deep learning model is shown in Table 1. The proposed deep learning model follows a similar pipeline in [34]. Generally, the proposed deep learning network consists of a stack of six convolutional and pooling layers followed by two fully connected layers. Convolutional layers perform important operations and extract important information about low-level features, for example, edges, color, and texture as well as high-level contextual information from the input image. Generally, each convolutional layer consists of the convolutional kernel with a fixed size that convolves over an input feature map. A convolutional layer is always followed by ReLU non-linear activation function and batch normalization function. Let $K \in R^{3 \times K_x \times K_y}$ is an image that

TABLE 1. Proposed convolutional neural network for turn classification.

Layers	Number of filters	Kernel stride	Kernel size
Convolution2D	96	11 x 11	2 x 2
Max-Pooling2D	-	3 x 3	2 x 2
Convolution2D	256	5 x 5	2 x 2
Max-Pooling2D	-	3 x 3	2 x 2
Convolution2D	512	3 x 3	1 x 1
Convolution2D	512	3 x 3	1 x 1
Convolution2D	512	3 x 3	1 x 1
Max-Pooling2D	-	3 x 3	2 x 2
Fully Connected	4096	6 x 6	1 x 1
Fully Connected	4096	1 x 1	-
Fully Connected	2	1 x 1	-

is provided as an input to the stack of convolutional layers. In this case, 3 represents three channels of the input image, K_x, and K_y represents the width and height of the image. Let the output feature map $M \in R^{c \times w \times h}$, where c is the number of channels, w and h represents the width and height of output feature map. After a stack of convolutional layers, we employ pooling layers that reduce the size of the feature map by reducing the number of parameters. We employ a Max-pooling layer (with a window of fixed size) to the output of convolutional layers. Max-pooling layer chooses the maximum value from the window that represents the

most prominent feature in that particular window. Another advantage of using the Max-pooling layer is that it prevents the network from the over-fitting problem. The feature map generated after passing the input image through the stack of convolutional and pooling layers is finally provided as input to the fully connected layers. Fully connected layers take high dimension input feature vector F_v and output the classification score vector C_v that contains the scores of different classes for input feature vector F_v. For training the network, we modify the size of the input training images in order to make them compatible with the input size of the network. Since our training data are binary images, the network accepts 3-channel images. We transform a single channel binary image to a 3-channel image by copying a single-channel image three times as in [35].

III. EMPIRICAL ANALYSIS AND RESULTS

A. DATASET

Honda Research Institute (HRI) published the HRI Driving Dataset (HDD) [18], the dataset contains a large amount of data, from various sensors and post-processing, focusing on computer vision. This dataset provides labels for different driving actions, labels are provided with the video session and sensory information including Global Positioning Service (GPS) data. The data used in this research consists of different variables including GPS latitude, longitude, orientation, and label of the event. The labels for the driving events considered in this study are left turn and right turn labels out of twelve different labels available in the dataset. The data used in the experiments consists of 43 hours of human vehicle driving. For this data, the labels are mapped to the GPS data and the labeled GPS data provides the ground truth information for the turn and non-turn classes, for its use in the experiments.

B. MODEL PERFORMANCE

A set of statistical analyses provides an overview of the performance of the proposed model. These analyses are based on a set of fundamental measures including true positive, true negative, false positive, and false negative. True positive is a measure where the decision and actual both belong to the same class as the turn sample was identified as turn by the model, it is denoted as TP. Whereas, the true negative is a classification of the negative class as negative, represented as TN. False-positive and false-negative measures are the labels that do not represent the class as in the ground truth. False-positive FP is where it is incorrectly labeled as positive while false negative is where it is labeled as negative but ground truth is positive.

Receiver operating characteristic (ROC) analysis is used to elaborate the results. ROC analysis indicators used in testing the performance of this research are expressed as follows.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Algorithm 2 createSpatialWindows

$\triangleright$ INPUT: 1. A chronologically ordered GPS Trajectory G. 2. Size of geofenced bounding box *size*.
$\triangleright$ OUTPUT: Sequence of bounding boxes B for given GPS trajectory.

1: $B = \{\}$
2: $b = getBoundingBox(G_1.lat_i, G_1.lng_i, size)$
3: Append b to the B array
4: **for all** i in range 2 to size(G) **do**
5: $\quad x = False$
6: $\quad$ **for all** f in B **do**
7: $\quad\quad$ **if** inBoundingBox($G_i.lat_i, G_i.lng_i,$
$\quad\quad f$) **then**
8: $\quad\quad\quad x = True$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad$ **if** $x == False$ **then**
12: $\quad\quad b = getBoundingBox(G_i.lat_i, G_i.lng_i, size)$
13: $\quad\quad$ Append b to the B array
14: $\quad$ **end if**
15: **end for**
16: **return** B

TABLE 2. Model performance metrics as measured for all folds for CNN detecting turns.

| Fold	1	2	3	4	5	Average
Recall	0.98	0.95	0.97	0.88	0.94	0.94
Precison	0.92	0.94	0.91	0.98	0.96	0.94
F-Measure	0.95	0.94	0.94	0.92	0.95	0.94
Specificity	0.92	0.94	0.92	0.98	0.96	0.94

$$F\text{-measure} = \frac{2TP}{2TP + FP + FN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

C. DISCUSSION

An approximate total of 4500 100-feet spatial windows are used in the experiment to test the proposed approach for turn detection using GPS-only data. Where a 100-feet spatial window covers approximately 4×10^4 square feet area as 100 feet size represents the perpendicular distance from the center to any side. As discussed earlier, the labels for the generated windows are available in form of ground truth, where each GPS record is labeled with a label, any label besides turn is considered as non-turn.

Using algorithm 1, a set of predicted labels are detected for all windows. The labeled windows are then used in the experiment for automatic labeling of windows based on their appearances. A CNN model is trained and tested using a 5-fold validation approach. The labeled windows are divided into two classes *Turn* and *NotTurn*. For each class we select 500 samples randomly to ensure a balanced dataset. In each
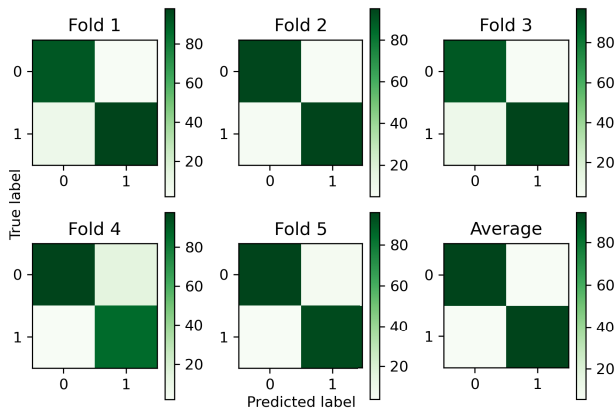
FIGURE 3. Confusion matrices for all folds and average performance of all folds. Dark green represents highest, white represents lowest value, label 0 represents turn, and label 1 represents non turn.

fold, 20% of data is split for testing purposes, and 80% of data is separated for training. The fold-by-fold precision and recall are listed in table 2. It is observed that the trained model classifies 94% of the training samples with significantly high precision. Compared to studies [2], [16], [17], [18], proposed work outperforms classification of different driving behaviors, in terms of accuracy or data requirements. Figure 3 reflects higher recall for each of the classes in different cross validation folds.

The study [2] detects lane change with accuracy as low as 58% and high as 97% with various classification algorithms. It must be noted that this study uses Magnetometer, Accelerometer, and Gyroscope sensors besides GPS for data collection. The study [16] achieved a recall of 73% and 77% to detect lane change with a two-fold cross-validation approach with a camera and several radar sensors. The study [17] achieved a recall of 87% for lane-change detection using steering angle and speed vectors. The study [18] used a camera and numerous sensors to collect data and perform behavior annotations on the video. The difference between annotations through their model versus a human annotator was 2%. The cost of such systems for real-time application will be large.

IV. CONCLUSION

Turn detection is a crucial application for the modern ADAS in vehicles. For this application, various models are available in the community, that either use computer vision or multiple data sources. That makes those either computationally or financially cost expensive. This research work proposes a deep learning-based turn detection model for GPS-only data. In this research, we propose spatial window-based algorithms to transform GPS-only data into image data for use in the deep learning process. A deep learning model is proposed for the classification of turns and non-turns and is tested on a real-time driving dataset. Five-fold testing of the deep learning classifier shows that the proposed work achieves up to 94% accuracy. Comparing the proposed research to other works

that use enormous data or far more powerful gear, it performs noticeably better.

REFERENCES

[1] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Sep. 2005, p. 602.

[2] Z. Islam and M. Abdel-Aty, "Real-time vehicle trajectory estimation based on lane change detection using smartphone sensors," *Transp. Res. Rec.*, vol. 2675, no. 6, 2021, Art. no. 0361198121990681.

[3] Z. Chen, C. Wu, Z. Huang, N. Lyu, Z. Hu, M. Zhong, Y. Cheng, and B. Ran, "Dangerous driving behavior detection using video-extracted vehicle trajectory histograms," *J. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 409–421, Sep. 2017.

[4] D. Yang, Y. Wu, F. Sun, J. Chen, D. Zhai, and C. Fu, "Freeway accident detection and classification based on the multi-vehicle trajectory data and deep learning model," *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103303.

[5] C. Yu, C. Zhang, G. Tian, and L. Liang, "Vehicle trajectory description for traffic events detection," in *Advances on Digital Television and Wireless Multimedia Communications*. Cham, Switzerland: Springer, 2012, pp. 228–235.

[6] M. A. Rahim, J. Liu, Z. Zhang, L. Zhu, X. Li, and S. Khan, "Who is driving? Event-driven driver identification and impostor detection through support vector machine," *IEEE Sensors J.*, vol. 20, no. 12, pp. 6552–6559, Jun. 2020.

[7] M. A. Rahim, L. Zhu, X. Li, J. Liu, Z. Zhang, Z. Qin, S. Khan, and K. Gai, "Zero-to-stable driver identification: A non-intrusive and scalable driver identification scheme," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 163–171, Jan. 2020.

[8] J. Quehl, H. Hu, S. Wirges, and M. Lauer, "An approach to vehicle trajectory prediction using automatically generated traffic maps," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 544–549.

[9] H. Wu, W. Sun, and B. Zheng, "A fast trajectory outlier detection approach via driving behavior modeling," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 837–846.

[10] S. Guo, X. Li, W.-K. Ching, R. Dan, W.-K. Li, and Z. Zhang, "GPS trajectory data segmentation based on probabilistic logic," *Int. J. Approx. Reasoning*, vol. 103, pp. 227–247, Dec. 2018.

[11] D. Guo, S. Liu, and H. Jin, "A graph-based approach to vehicle trajectory analysis," *J. Location Based Services*, vol. 4, nos. 3–4, pp. 183–199, 2010.

[12] R. S. D. Sousa, A. Boukerche, and A. A. F. Loureiro, "Vehicle trajectory similarity: Models, methods, and applications," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–32, Sep. 2021.

[13] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou, "An effectiveness study on trajectory similarity measures," in *Proc. 24th Australas. Database Conf.*, vol. 137, 2013, pp. 13–22.

[14] N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy, "Review on trajectory similarity measures," in *Proc. IEEE 7th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2015, pp. 613–619.

[15] Z. Yu, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, 2015.

[16] J. Schlechtriemen, A. Wedel, J. Hillenbrand, G. Breuel, and K.-D. Kuhnert, "A lane change detection approach using feature ranking with maximized predictive power," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 108–114.

[17] H. M. Mandalia and M. D. D. Salvucci, "Using support vector machines for lane-change detection," in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, vol. 49, no. 22. Newbury Park, CA, USA: Sage, 2005, pp. 1965–1969.

[18] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7699–7707.

[19] Y. Mao, H. Zhong, X. Xiao, and X. Li, "A segment-based trajectory similarity measure in the urban transportation systems," *Sensors*, vol. 17, no. 3, p. 524, Mar. 2017.

[20] X. Wang, K. Tieu, and E. Grimson, "Learning semantic scene models by trajectory analysis," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2006, pp. 110–123.

[21] F. Wei, W. Guo, X. Liu, C. Liang, and T. Feng, "Left-turning vehicle trajectory modeling and guide line setting at the intersection," *Discrete Dyn. Nature Soc.*, vol. 2014, pp. 1–7, Jan. 2014.

[22] C.-Y. Chan, "Characterization of driving behaviors based on field observation of intersection left-turn across-path scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 322–331, Sep. 2006.

[23] A. Nowosielski and P. Forczmański, "Detection of critical behaviour on roads by vehicle trajectory analysis," *Arch. Transp. Syst. Telematics*, vol. 6, no. 3, pp. 46–51, 2016.

[24] K. K. Santhosh, D. P. Dogra, and P. P. Roy, "Anomaly detection in road traffic using visual surveillance: A survey," *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–26, Nov. 2021.

[25] S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Trajectory-based surveillance analysis: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1985–1997, Jul. 2019.

[26] L. Lin, Y. Lu, Y. Pan, and X. Chen, "Integrating graph partitioning and matching for trajectory analysis in video surveillance," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4844–4857, Dec. 2012.

[27] J. Bian, D. Tian, Y. Tang, and D. Tao, "A survey on trajectory clustering analysis," 2018, *arXiv:1802.06971*.

[28] A. A. Sekh, D. P. Dogra, S. Kar, and P. P. Roy, "Video trajectory analysis using unsupervised clustering and multi-criteria ranking," *Soft Comput.*, vol. 24, no. 21, pp. 16643–16654, Nov. 2020.

[29] Z. Kan, L. Tang, M.-P. Kwan, C. Ren, D. Liu, and Q. Li, "Traffic congestion analysis at the turn level using taxis' GPS trajectory data," *Comput., Environ. Urban Syst.*, vol. 74, pp. 229–243, Mar. 2019.

[30] J. Tang, M. Deng, J. Huang, and H. Liu, "A novel method for road intersection construction from vehicle trajectory data," *IEEE Access*, vol. 7, pp. 95065–95074, 2019.

[31] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1468–1476.

[32] G. Salvo, L. Caruso, and A. Scordo, "Urban traffic analysis through an UAV," *Proc. Social Behav. Sci.*, vol. 111, pp. 1083–1091, Feb. 2014.

[33] Y. Endo, H. Toda, K. Nishida, and J. Ikedo, "Classifying spatial trajectories using representation learning," *Int. J. Data Sci. Anal.*, vol. 2, nos. 3–4, pp. 107–117, Dec. 2016.

[34] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1299–1312, May 2016.

[35] E. Felemban, S. D. Khan, A. Naseer, F. U. Rehman, and S. Basalamah, "Deep trajectory classification model for congestion detection in human crowds," *Comput., Mater. Continua*, vol. 68, no. 1, pp. 705–725, 2021.

MUSSADIQ ABDUL RAHIM received the B.S. degree (Hons.) from CIIT, Islamabad, Pakistan, in 2013, the M.S. degree in computer science from PIEAS, Pakistan, in 2015, and the Ph.D. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2020. He is currently an Assistant Professor with the Department of Computer Science, National University of Technology (NUTECH), Islamabad. His recent research interests include vehicular networks, applied machine learning for human–machine behavior studies, and deep learning-based solution of real-world problems.

SULTAN DAUD KHAN received the M.Sc. degree in electronics and communication engineering from Hanyang University, South Korea, in 2010, and the Ph.D. degree from the University of Milano-Bicocca, Italy, in 2016. He is currently an Associate Professor with the National University of Technology (NUTECH), Pakistan. His research interests include computer vision application to pedestrian and crowd analysis.

SALABAT KHAN received the Ph.D. degree from the School of Computer Science and Technology, Institute of Technology, Beijing, China, in 2020. He is currently an Associated with Shenzhen University as a Research Associate with the College of Computer Science and Software Engineering. His current research interests include secure and transparent public-key infrastructure (PKI), transparent and secure key management in VANETs, distributed ledger technology, machine learning, and blockchain.

MUHAMMAD RASHID received the Ph.D. degree in computer science from the National University of Computer and Emerging Sciences. He has over ten years of academic experience during which he has been the head of various departments and units for over eight years. He is currently working as the Head of the Department of Computer Science, National University of Technology (NUTECH), Pakistan. His research interests include virtual reality, artificial intelligence, and applied machine learning.

RAFI ULLAH received the M.S. degree in computer system engineering from GIKI, Pakistan, in 2006, and the Ph.D. degree from PIEAS, Pakistan, in 2010. He is currently working as an Associate Professor with the Department of Computer Science, National University of Technology, Pakistan. His research interests include image and video processing, digital watermarking, and EEG signals.

HANAN TARIQ (Graduate Student Member, IEEE) received the B.S. degree in electrical (power) engineering from COMSATS University Islamabad, Abbottabad Campus, Pakistan, in 2016, and the M.Sc. degree in electrical engineering from the Imperial College of Business Studies (ICBS), Lahore, Pakistan, in 2018. He is currently a Ph.D. Scholar with the Department of Electrical Power Engineering, Gdańsk University of Technology, Poland. He has been associated with research and teaching at multiple universities, since 2017. His research interests include electrical safety, power system stability, and dynamics and simulative studies of power systems.

STANISLAW CZAPP (Member, IEEE) received the degree from the Gdańsk University of Technology, Poland, in 1996, and the Ph.D. and D.Sc. degrees, in 2002 and 2010, respectively. He is currently an Associate Professor with the Faculty of Electrical and Control Engineering, Gdańsk University of Technology. He is the author and coauthor of many articles, conference papers, and unpublished studies, such as designs and expert evaluations as well as opinions. His research interests include power systems, electrical installations and devices, electric lighting, and electrical safety. He is an Expert of the SEP Association of Polish Electrical Engineers in Section 08 Electrical Installations and Devices.