Monocular Vision-based Prediction of Cut-in Maneuvers with LSTM Networks

Yagiz Nalcakan[1,2] and Yalin Bastanlar[1]

***Abstract*— Advanced driver assistance and automated driving systems should be capable of predicting and avoiding dangerous situations. In this study, we propose a method to predict the potentially dangerous lane changes (cut-ins) of the vehicles in front. We follow a computer vision-based approach that only employs a single in-vehicle RGB camera and we classify the target vehicle's maneuver based on the recent video frames. Our algorithm consists of a CNN-based vehicle detection and tracking step and an LSTM-based maneuver classification step. It is computationally efficient compared to other vision-based methods since it exploits a small number of features for the classification step rather than feeding CNNs with RGB frames. To evaluate our approach, we have worked on a publicly available dataset and tested several classification models. Experiment results reveal that 0.9325 accuracy can be obtained with side-aware two-class (cut-in vs. lane-pass) classification models.**

***Index Terms*— Vehicle Behavior Prediction, Vision-based Maneuver Classification, Maneuver Prediction, Driver Assistance Systems**

I. Introduction

Prediction of intended maneuvers of surrounding vehicles is an important research area that supports the development of Advanced Driver Assistance Systems (ADAS). Also, it is one of the challenging problems on reaching fully driverless vehicles (SAE level 4). Moreover, statistical data show that unexpected maneuvers of drivers on highways may lead to deadly accidents. According to U.S. Department of Transportation, National Highway Traffic Safety Administration's (NHTSA) 2018 report on ”Driving Behaviors Reported For Drivers And Motorcycle Operators Involved In Fatal Crashes” [1], one of the top-3 reasons of fatal crashes is failure to keep the vehicle in proper lane. Therefore, early prediction of risky lane-change maneuvers of surrounding vehicles can help drivers to avoid fatal crashes on the road.

Detection and distance measurements for surrounding vehicles can be obtained via different sensors including radar, camera, and LiDAR. Each of these sensors has pros and cons compared to each other. For example, although LiDAR can detect much smaller objects and generate more detailed images compared to the others, it is still an expensive sensor. Radar has advantages on extreme illumination and weather conditions but its field of view is generally narrow and its output is noisy requiring cleaning [2]. A camera, which is the sensor we use in this study, is cheap, easily accessible and it enables us to obtain a variety of information (such as color, speed, distance, depth, etc.) at the same time if accompanied with powerful computer vision techniques.

In our study, we focus on vehicles in front and we only employ a single in-vehicle forward-looking RGB camera. This brings simplicity to our approach compared to other studies in the literature that use camera, radar, and LiDAR sensors ([3], [4], [5]). Since there is no benchmark dataset for the classification of potentially dangerous cut-in maneuvers in traffic, we have prepared a classification dataset with the videos of the publicly available Berkeley Deep Drive dataset [17], which consists of videos that are collected via the front camera of the vehicles on highways of various cities. We have cut and labeled 875 video clips containing vehicle maneuvers belonging to cut-in or lane-pass classes. These video clips cover two seconds of action. In our experiments, we represented this duration with varying number of frames (15, 30, 45 or 60). As number of frames increases, it becomes a dense representation but also it requires more computation. We made labeled clips and source code of our methods publicly available[1].

We classify the maneuver of the vehicles in front whether they are cutting-in to ego-vehicle's lane or keeping their own lane (Figure 1). In our experiments, we evaluated several models for binary classification of cut-in and lane-pass maneuvers, as well as 3-class models to discriminate left-hand side and right-hand side cut-ins. There is no doubt that the proposition of classifying maneuvers into two or three is an oversimplification of the real life cases. However, this scheme is enough from the viewpoint of predicting risky overtaking maneuvers and it enables us to compare our results with the state-of-the-art lane-change detection methods.

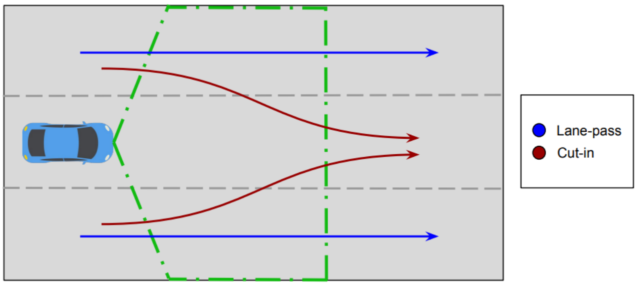

Fig. 1. Lane-pass and cut-in maneuvers (green area indicates considered safety field for ego vehicle.)

*This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), Grant No: 2244-118C079.

[1] Y. Nalcakan and Y. Bastanlar are with Department of Computer Engineering, Izmir Institute of Technology, 35430, Urla, Izmir, Turkey. {yagiznalcakan, yalinbastanlar}@iyte.edu.tr

[2] Y. Nalcakan is also with TTTech Auto Turkey Software, Izmir, Turkey.

[1]https://github.com/ynalcakan/cut-in-maneuver-prediction

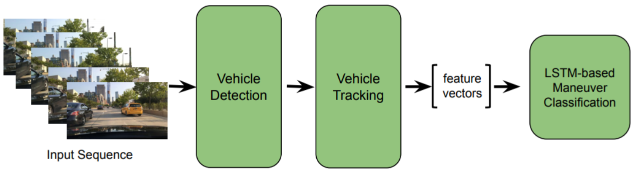

Fig. 2. Overview of the proposed approach

Our approach consists of three steps (Figure 2). The first step is a CNN-based vehicle detection, where we employ YOLOv4 [6] to detect vehicles in each frame of the sequence. The second step is tracking the detected vehicles using DeepSort [7]. In the third step, extracted features from the detected and tracked bounding boxes of vehicles in front are fed into an LSTM network to be classified.

Using expensive sensor setups or complex processing pipelines, limits the availability and robustness of previous methods. Thus, we state the contributions of our work as:

1) Our approach employs very simple equipment (a standard field-of-view RGB camera) and does not require a calibration procedure or any other specific adjustment depending on the ego-vehicle.
2) Our approach is computationally cheap compared to previous work that feed CNNs with video frames and use complex architectures ([12], [13], [14], [16]) or the ones that project the scene on the ground plane ([3], [15]). Instead we exploit a small number of features extracted from input sequence and feed an LSTM with those. We are able to produce a classification decision every two seconds.

The remainder of this paper is structured as follows: the related works are reviewed in Section II. Section III has detailed information about our method on maneuver prediction and experimental results can be found in Section IV which is followed by the conclusion in Section V.

II. Related Work

Most of the work done in the field of maneuver classification is on lane change prediction. There are few studies on detecting risky cut-in maneuvers, which we focus on. Therefore, we will discuss the studies on lane change prediction and later on we will compare our results with their performance.

We would like to discuss the recent studies in the literature by dividing them into two according to their methodologies, as trajectory-based and vision-based maneuver classifications. Basically, trajectory-based maneuver classification is about projecting the trajectory of each surrounding vehicle on the ground plane using on-vehicle camera, radar or external sensors (i.e. surveillance cameras) and classifying the maneuver with this trajectory information. Vision-based classification is about extracting features from an image sequence and classifying the maneuver using these features.

A. *Trajectory-based Maneuver Classification*

In a study conducted before the deep learning era, Kasper *et al.* [5] proposed to model driving maneuvers using Object-oriented Bayesian Networks (OOBN). According to their experiments, the combined use of lane-related coordinate features and occupancy grids are very effective to classify driving maneuvers. Deo *et al.* [3] proposed an approach to classify maneuver trajectories by using hidden markov model (HMM) and variational Gaussian mixture model (VGMM). First, they extracted different maneuvers like lane-pass, overtake, cut-in, and drift-into-ego-lane from highway recorded videos, radar and LiDAR data. Then they classified all trajectories using VGMM. Their method reached 0.842 accuracy on all maneuvers and 0.559 accuracy on overtake and cut-in maneuvers. Scheel *et al.* [8] used the trajectories of the right lane change, left lane change, and follow maneuvers as input to an attention-based LSTM network, and they reported accuracy of prediction of each maneuver separately as 0.784 for left lane change, 0.962 for follow, and 0.679 for right lane change. Altché *et al.* [9] proposed an LSTM-based method to predict future vehicle trajectories on NGSIM dataset [10] in which two layer LSTM achieved better RMSE results compared to other similar approaches in two and three seconds prediction horizons.

B. *Vision-based Maneuver Classification*

With the increasing popularity of deep learning methods in vision, recent studies of vision-based maneuver classification generally use convolutional neural networks (CNNs) to get visual information regarding the scene. Usual practice is using a CNN as a feature extractor by feeding video frames into CNN and using an RNN or an LSTM as a classifier. In [12], features are extracted by a CNN on region-of-interest (ROI) and width, height, center coordinate values are added to the feature vector. Then, classification of the lane change maneuvers is performed by an LSTM. Their best model achieved 0.745 accuracy. Another alternative in the same study [12] was converting movements of objects into contours in an RGB image and feeding CNNs with this motion history image. However, performance was worse.

Another study that first crops ROIs from the original frames [14] exploited two modes of input video, which are high frame rate video itself and its optical flows. They compared two-stream CNNs and spatiotemporal multiplier networks. In a follow-up study [13], authors also included slow-fast network (the one uses videos of high and low frame rate) into the comparison which achieved 0.908 accuracy and performed slightly better than other alternatives.

In [11], authors compared the results of RNN and LSTM on a lane change classification dataset. The center coordinates of the target vehicle are the only features included and the LSTM achieved the best F1-scores as 0.94 for lane-keeping, 0.78 for left lane-changing, and 0.94 for right lane-changing. Lee *et al.* [15] proposed a method that can be used for adaptive cruise control, which uses a front-faced radar and camera outputs to infer the lane change maneuvers. In their method, they convert the traffic scenes to a simplified bird's eye view (SBV) and those SBVs are given as input to a CNN network to predict lane keeping, right cut-in and left cut-in intentions. Yurtsever *et al.* [16] proposed a deep learning-based action recognition framework for classifying dangerous lane change behavior in video captured by an in-car camera. They used a pretrained Mask R-CNN model to segment vehicles in the scene and a CNN+LSTM model to classify the behaviors as dangerous or safe.

We extract bounding boxes of surrounding vehicles with well-performing computer vision techniques. This step exists in previous vision based methods as well. However, unlike previous methods ([12], [13], [14], [15], [16]) we do not feed a feature extractor CNN with RGB frames of the sequence. Instead, we feed an LSTM with a feature vector consisting of bounding box data, which is very fast. With a 15-frame model we can make an estimate for every two seconds, which makes it deployable for real-time systems.

To the best of our knowledge, there is no cut-in/lane-pass classification accuracy reported in the literature but we can compare our results with lane-change classification accuracies (that does not differ if a vehicle is entering or existing ego-lane). Previous 3-class (left lane-change, right lane-change, no lane-change) classification studies ([11], [12], [13], [14]) reached 0.90% accuracy even with complex models. We achieved 91.35% and 93.25% accuracies with the best 3-class and 2-class models, respectively (cf. Section IV.A).

III. Methodology - Dataset

A. Dataset Description and Labeling

A subset of the Berkeley Deep Drive Dataset [17] was used in this study. This dataset consists of 100K driving video that labeled for 10 different tasks (road object detection, instance segmentation, drive-able area, etc.). The videos were collected from the front camera of the vehicles at various times of the day in New York, Berkeley, San Francisco, and Tel Aviv. We focused on lane change actions that happened at highways. 875 video sequences containing vehicle maneuvers belonging to cut-in and lane-pass classes (Figure 1) were cut from approximately 20K videos. The final distribution contains 405 cut-in and 470 lane-pass samples which are divided into train-validation-test datasets using 60%-20%-20% split ratio.

Figure 3 shows the principle while labeling cut-in samples in our dataset. At the starting frame of the sequence, the target vehicle is on the other lane and there is no indication whether a cut-in will occur or not. The lane change event occurs as the target vehicle enters the safety field (the polygon that is indicated with green lines in Figure 3). The sequence is cut when the vehicle is entered to the ego-lane with its full body (no need to be aligned in the center).

B. Vehicle Detection

As we extract our features using the bounding box of the target vehicle, it is very important to collect the information regarding the surrounding vehicles with high accuracy. Therefore, as vehicle detection network, we used YOLOv4 [6] which is widely used in the area. Each frame of the image sequence is fed into the pre-trained YOLOv4 and bounding boxes of the target vehicles are sent to the vehicle tracking step together with the same input image (Figure 4).

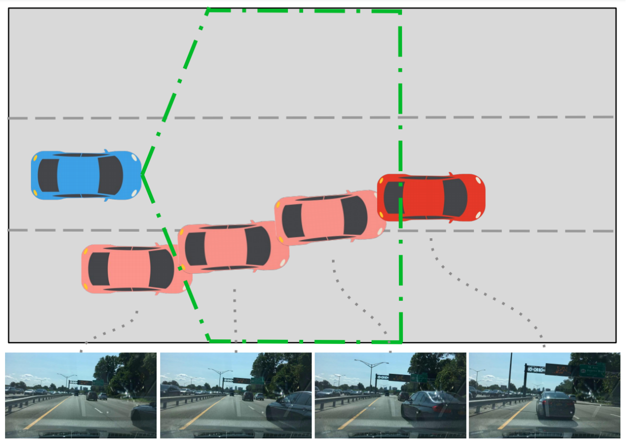

Fig. 3. Start and end points of maneuvers that are labeled as cut-in

C. Vehicle Tracking

Detected vehicles per frame (bounding boxes) can have position errors caused by YOLOv4. To minimize those errors, we added a vehicle tracking step to the pipeline. Tracking is done via DeepSort [7] which is known for its ease of implementation with YOLO. Corrected bounding boxes of target vehicles are given as input to the feature extraction step.

D. Feature Extraction and Network Architecture

Many visual cues may allow us to predict whether any of the surrounding vehicles makes a cut-in maneuver or not. In our work, we obtain cues from the detected bounding boxes of surrounding vehicles. Specifically, we extract the center (x,y) coordinates, width and height values of the bounding box. Collected features are given to a single-layer LSTM to obtain the classification result. We tried four different sequence lengths (15, 30, 45, 60). Shorter sequence length means more sparse representation (3 out of 4 frames are neglected in 15-frame sequences) of the action but faster maneuver prediction due to decreased processing time.

As hyperparameters of the LSTM, various hidden unit sizes, batch sizes, activation unit types etc. are evaluated to find the best performing LSTM architecture. Evaluated hyperparameters are given in Table I and proposed framework can be seen in Figure 4.

TABLE I
EVALUATED LSTM HYPERPARAMETERS

Hidden Units	Batch Sizes	Optimizer	Activation
60 128 256 512	5 10	Adam RMSProp AdaDelta	ReLU Sigmoid Tanh

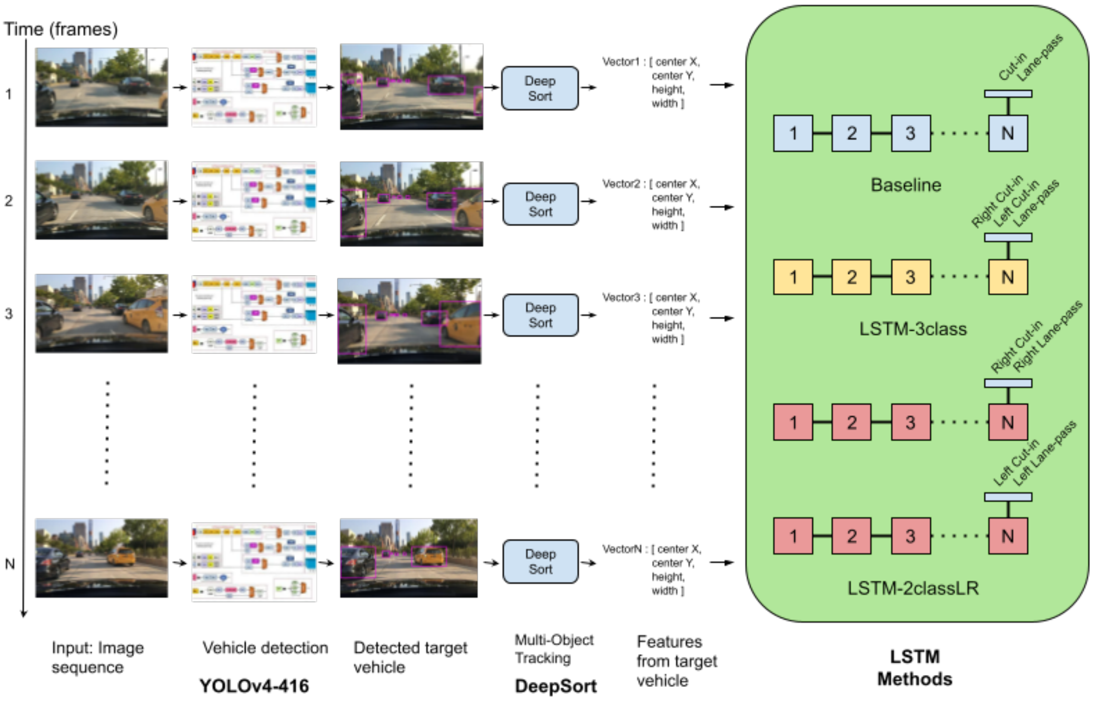

Fig. 4. Pipeline of the proposed approach. Following the steps for extracting the bounding boxes of target vehicles (TVs), baseline LSTM method uses feature vectors of TVs and classifies maneuvers as cut-in or lane-pass. LSTM-3class method classifies into 3: right cut-in, left cut-in or lane-pass classes. As a third alternative, LSTM-2classLR has two separate LSTMs for left-hand side and right-hand side TVs. We conducted experiments with varying sequence lengths (15, 30, 45 and 60 frames) all representing two seconds of the video.

IV. Experimental Results

A. Maneuver Classification Results

We evaluated our approach with different methods where classification strategy varies. As a baseline method, a single layer LSTM model is trained with four features (center (x,y) coordiantes, width and height of the target vehicle's bounding box) for a side-agnostic 2-class classification, i.e. each sample is a cut-in or a lane-pass. In LSTM-3class method, training and evaluation are done by classifying samples as left-hand side cut-in, right-hand side cut-in and lane-pass. This second strategy is closer to several lane-change prediction studies in the literature, where maneuvers were classified as left lane-change, right lane-change and no lane-change. By examining the target vehicle's center coordinates, it is straightforward to extract if it is on the left or on the right of the ego vehicle. Thus, as a third method (LSTM-2classLR), we train two networks one responsible for left-hand side maneuvers and the other for right-hand side, each performs a 2-class classification (cut-in/lane-pass).

For all the models mentioned above, hyperparameters were optimized by a random search algorithm and the top-5 models having the highest accuracy on the validation set were evaluated with the test set. Average performance of these five models, rather than the top-1 model, are presented in Table II. Since most previous studies reported accuracy, precision and recall, we also evaluated our methods with these metrics to be able to make a comparison.

Our baseline method, which classifies the sequences without separating if the maneuver is on the right or on the left, reached an accuracy of 0.9216 with 30-frame sequences and slightly lower accuracy for other sequence lengths. Taking into account the side of the cut-in maneuver (3 classes: right cut-in, left cut-in, lane-pass) caused a very slight decrease in the performance, achieving 0.9135 accuracy. However, when we train two separate networks for the right-hand side and left-hand side (LSTM-2classLR), the classification accuracy increased from 0.9216 to 0.9325 accuracy. The best values were obtained with 45-frame sequences, but for other lengths as well accuracies were increased compared to the baseline and LSTM-3class methods.

We should also note that all the evaluated models exceed the previously reported performances in the literature which are 3-class lane-change prediction accuracies of 0.7450 in [12] and 0.9080 in [13].

B. Adding Distance-to-ego-lane Feature

Extraction of the new features from vehicle detection module could improve our classification performance. Considering that the lane lines can provide information, we decided to add distance-to-ego-lane feature to our feature vector. In LSTM-distance model (Table III), we add the distance between the target vehicle's bounding box center and the ego-vehicles lane. Distance-to-ego-lane feature is extracted using simple computer vision techniques. First, ego lane lines extracted by applying Sobel filter and white color filtering methods to the safety field. Then, if the target vehicle is on the right-hand side, distance between its center and the

TABLE II

CLASSIFICATION RESULTS ON TEST DATA OF DIFFERENT METHODS FOR VARYING SEQUENCE LENGTHS

Method	Sequence Length	Accuracy	Precision (Cut-in)	Recall (Cut-in)	Precision (Lane-pass)	Recall (Lane-pass)
Baseline	15	0.9189	0.9043	0.9233	0.9342	0.9185
	30	**0.9216**	0.8986	**0.9313**	**0.9418**	0.9145
	45	0.9189	**0.9420**	0.8911	0.8987	**0.9470**
	60	0.8657	0.8406	0.8900	0.9114	0.8763
LSTM-3Class	15	0.9094	0.8708	0.8930	0.9392	0.9233
	30	0.9026	0.8556	0.8980	0.9443	0.9077
	45	**0.9135**	**0.8825**	0.8985	0.9342	**0.9250**
	60	0.9094	0.8512	**0.9173**	**0.9570**	0.9028
LSTM-2ClassLR	15	0.9283	0.9289	0.9005	0.9279	0.9500
	30	0.9255	**0.9358**	0.8914	0.9186	**0.9541**
	45	**0.9325**	0.8878	**0.9537**	**0.9674**	0.9314
	60	0.9271	0.9192	0.9162	0.9394	0.9440

right lane marking is calculated (yellow line in Figure 5), vice versa for the left-hand side target vehicle. Results when this feature is added (LSTM-distance) are given in Table III for varying sequence lengths.

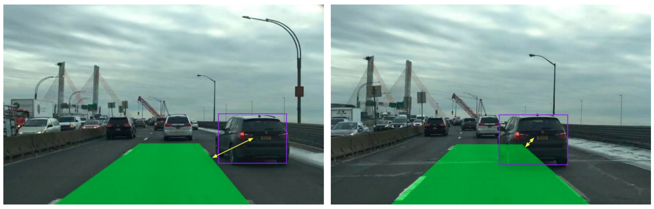

Fig. 5. Extraction of the distance-to-ego-lane feature. Yellow line shows the shortest distance between center coordinates of the target vehicle's bounding box and right lane marking of the ego lane. Left image: just before the cut-in maneuver starts, right image: a further frame of the maneuver.

TABLE III

CLASSIFICATION RESULTS WHEN DISTANCE FEATURE IS ADDED

Method	Sequence Length	Accuracy	Precision (Cut-in)	Recall (Cut-in)	Precision (Lane-pass)	Recall (Lane-pass)
LSTM-distance	15	0.8950	0.8900	0.8680	0.8988	0.9169
	30	**0.9092**	**0.8933**	**0.8935**	**0.9210**	**0.9210**
	45	0.8354	0.7888	0.8367	0.8815	0.8474
	60	0.8822	0.8833	0.8476	0.8815	0.9109

Addition of distance feature does not seem to increase the classification accuracy of the baseline method. It is most probably due to the fact that the decrease in this feature is proportional to the decrease in center (x,y) coordinates. Thus, our neural network is already able to make the necessary interpretation without seeing the distance-to-ego-lane feature.

C. Computational Efficiency

Execution times[1] of our LSTM models for 30-frame input sequences can be seen in Table IV. As can be seen, vehicle detection and tracking steps of our pipeline take much more time than the classification step, which is not more than 2 milliseconds. Computation times are directly proportional to the number of frames. Thus, the total time is 2 seconds for 15-frame sequences and 8 seconds for 60-frame sequences. Please note that, in the proposed approach, we process two seconds of video regardless of the number of frames in the sequence (15, 30, 45 or 60). If we use 15-frame sequences, we are able to produce a classification result (cut-in/lane-pass) for the scene every two seconds. As can be seen in Table II, results of 15-frame sequences are very close to the best results. From this point of view, we can argue that the proposed approach can be considered for real-time applications.

As mentioned in [6], compared to other object detection methods, YOLOv4 is one of the most real-time applicable object detection methods. We employed YOLOv4 for our pipeline, however any state-of-the-art vehicle detection method can be used.

In the studies we compared, the target vehicle bounding boxes were either taken from a dataset directly or obtained with CNNs. As mentioned above, we also employ CNN for vehicle detection, however different from the previous studies, we obtain features directly from the target vehicle bounding box and feed an LSTM. This is computationally cheaper than feeding CNNs with video frames to extract features ([12], [13], [14], [16]). Thus, the classification time is longer for the methods in the literature.

TABLE IV

EXECUTION TIME COMPARISON OF EVALUATED LSTM METHODS

Method	Vehicle Detection and Tracking (sec/seq)	Classification (msec/seq)	Total (sec/seq)
Baseline	4.002	2.11	4.0041
LSTM-3class		1.29	4.0032
LSTM-2classLR		1.95	4.0039
LSTM-distance		1.47	4.0034

[1] All evaluations are done on a PC with Ubuntu 16.04, i7-7700K CPU, 16 GB RAM and an Nvidia GeForce GTX 1080 GPU.

V. Conclusion and Future Work

The approaches developed for ADAS and autonomous vehicles should be as simple and affordable as possible. Therefore, methods that work with monocular vision, as in our study, may be preferable. In this work, we proposed a simple approach to predict possible dangerous lane-change maneuvers namely cut-ins. Side-aware method, LSTM-2classLR, achieved a promising result (0.9325 accuracy) using just the center coordinates, width and height of the target vehicle's bounding box. Furthermore, when we represent last two seconds of video with 15 frames, computation time is also two seconds. Thus, we conclude that a system implementing our approach can refresh its decision every two seconds.

As future work, we plan to increase the number of maneuver classes (e.g. a vehicle drifting in ego-lane and braking). We are also planning to evaluate an improved version of our method on recently published Prevention Dataset [18].

Acknowledgement

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), Grant No: 2244-118C079.

References

[1] Insurance Information Institute, Facts + Statistics: Highway safety. (2021) [Online]. Available: https://www.iii.org/fact-statistic/facts-statistics-highway-safety

[2] Sivaraman, S., and Trivedi, M. M. (2013). "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis". IEEE transactions on intelligent transportation systems, 14(4), 1773-1795.

[3] Deo, N., Rangesh, A., and Trivedi, M. M. (2018). "How would surround vehicles move? A unified framework for maneuver classification and motion prediction". IEEE Transactions on Intelligent Vehicles, 3(2), 129-140.

[4] Garcia, F., Cerri, P., Broggi, A., de la Escalera, A., and Armingol, J. M. (2012, June). "Data fusion for overtaking vehicle detection based on radar and optical flow". In 2012 IEEE Intelligent Vehicles Symposium (pp. 494-499). IEEE.

[5] Kasper, D., Weidl, G., Dang, T., Breuel, G., Tamke, A., Wedel, A., and Rosenstiel, W. (2012). "Object-oriented Bayesian networks for detection of lane change maneuvers". IEEE Intelligent Transportation Systems Magazine, 4(3), 19-31.

[6] Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). "Yolov4: Optimal speed and accuracy of object detection". arXiv preprint arXiv:2004.10934.

[7] Wojke, N., Bewley, A., and Paulus, D. (2017, September). "Simple online and realtime tracking with a deep association metric". In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649). IEEE.

[8] Scheel, O., Nagaraja, N. S., Schwarz, L., Navab, N., and Tombari, F. (2019, May). "Attention-based lane change prediction". In 2019 International Conference on Robotics and Automation (ICRA) (pp. 8655-8661). IEEE.

[9] Altché, F., and de La Fortelle, A. (2017, October). "An LSTM network for highway trajectory prediction". In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) (pp. 353-359). IEEE.

[10] U.S. Federal Highway Administration - US Highway 101 dataset. (2005) [Online]. Available: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[11] Laimona, O., Manzour, M. A., Shehata, O. M., and Morgan, E. I. (2020, October). "Implementation and Evaluation of an Enhanced Intention Prediction Algorithm for Lane-Changing Scenarios on Highway Roads". In 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES) (pp. 128-133). IEEE.

[12] Izquierdo, R., Quintanar, A., Parra, I., Fernández-Llorca, D., and Sotelo, M. A. (2019, October). "Experimental validation of lane-change intention prediction methodologies based on CNN and LSTM". In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 3657-3662). IEEE.

[13] Biparva, M., Fernández-Llorca, D., Izquierdo-Gonzalo, R., and Tsotsos, J. K. (2021). "Video action recognition for lane-change classification and prediction of surrounding vehicles". arXiv preprint arXiv:2101.05043.

[14] Fernández-Llorca, D., Biparva, M., Izquierdo-Gonzalo, R., and Tsotsos, J. K. (2020, September). "Two-Stream Networks for Lane-Change Prediction of Surrounding Vehicles". In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC) (pp. 1-6). IEEE.

[15] Lee, D., Kwon, Y. P., McMains, S., and Hedrick, J. K. (2017, October). "Convolution neural network-based lane change intention prediction of surrounding vehicles for acc". In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) (pp. 1-6). IEEE.

[16] Yurtsever, E., Liu, Y., Lambert, J., Miyajima, C., Takeuchi, E., Takeda, K., and Hansen, J. H. (2019, October). "Risky action recognition in lane change video clips using deep spatiotemporal networks with segmentation mask transfer". In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 3100-3107). IEEE.

[17] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., ... and Darrell, T. (2020). "Bdd100k: A diverse driving dataset for heterogeneous multitask learning". In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2636-2645).

[18] Izquierdo, R., Quintanar, A., Parra, I., Fernández-Llorca, D., and Sotelo, M. A. (2019, October). "The prevention dataset: a novel benchmark for prediction of vehicles intentions". In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 3114-3121). IEEE.