Experimental validation of lane-change intention prediction methodologies based on CNN and LSTM

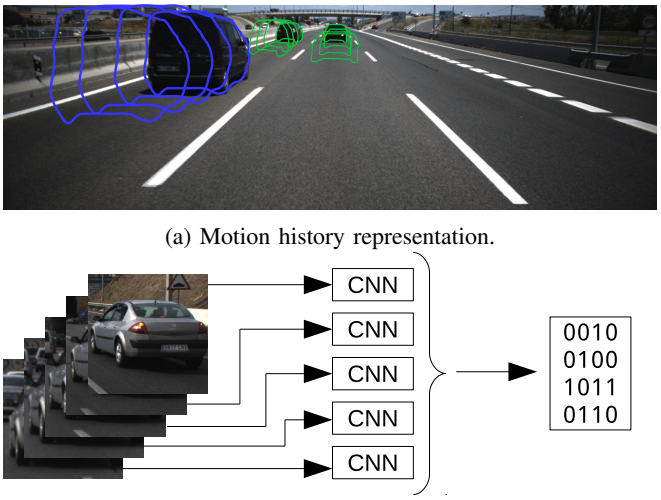
R. Izquierdo, A. Quintanar, I. Parra, D. Fernández-Llorca, and M. A. Sotelo

***Abstract*— This paper describes preliminary results of two different methodologies used to predict lane changes of surrounding vehicles. These methodologies are deep learning-based and the training procedure can be easily deployed by making use of the labeling and data provided by the PREVENTION dataset. In this case, only visual information (data collected from the cameras) is used for both methodologies. On the one hand, visual information is processed using a new multi-channel representation of the temporal information which is provided to a CNN model. On the other hand, a CNN-LSTM ensemble is also used to integrate temporal features. In both cases, the idea is to encode local and global context features as well as temporal information as the input of a CNN-based approach to perform lane change intention prediction. Preliminary results showed that the dataset proved to be highly versatile to deal with different vehicle intention prediction approaches.**

I. Introduction and Related Work

One of the most risky scenarios for autonomous vehicles in highways are the lane change maneuvers of surrounding vehicles. Endowing self-driving cars with the ability of predicting potential hazards due to these type of maneuvers is of utmost importance. Most of the current approaches to deal with the lane change prediction problem are learned-based. Accordingly, a considerable number of labeled samples from real traffic scenarios is needed. In this paper, we make use of the PREVENTION dataset [1] which provides a large number of accurate and detailed annotations of vehicles categories, trajectories and events (including left/right lane changes, among others). More than 356 minutes, 4M vehicle detection and 3K trajectories are available, with data collected from LIDAR, radar and camera sensors, from surrounding vehicles up to a range of 100 meters.

Two different deep learning-based methodologies are proposed using visual information. On the one hand, a new multi-channel representation of the temporal and context information is proposed. As can be observed in Fig 1a, the contours of the detected vehicles (like the motion history) are temporally integrated at frame t. We use different channels for the vehicle from which the prediction is inferred, and the rest of the vehicles. This new representation is fed to a convolutional neural network (CNN) architecture which is trained from scratch. On the other hand, as can be observed in Fig 1b a more standard approach is also used by applying convolutional operations (from a trained network) to each Region of interest (ROI) of a temporal sequence for each vehicle, attempting to include local context information in a canonical frame for each vehicle. Sequence folding/unfolding and a flatten layer are used followed by a Long Short-Term Memory network (LSTM). Both approaches can be easily trained thanks to the data and labels included in the PREVENTION benchmark.



(a) Motion history representation.

(b) Encoded ROI sequence.

Fig. 1: Temporal image integration.

Although a considerable number of works have been proposed in the literature to deal with the ego vehicle lane-change prediction problem, we limit our study to approaches focused on lane-change prediction of other vehicles.

As suggested by [2] vehicle motion modeling and prediction approaches can be classified into three different levels. First, physical-based, where predictions only depend on the laws of physics. Second, maneuver-based, where the future motion of a vehicle depends on the maneuver that the driver intends to perform. Finally, intention-aware, where predictions take into consideration inter-dependencies between surrounding vehicles. Note that, in some kind of vicious circle, predicting when a lane-change will happen can be addressed using the estimated trajectory from any of the three different motion model levels, and predicting the motion of surrounding vehicles can be more precisely estimated if lane-change intention prediction is available.

As highlighted in [3], most of the works related to lane-change intention prediction can be classified as probabilistic- or deterministic-based. In [4] a Support Vector Machine and a Bayesian filter are used to predict the lane change taking into account the lateral position and the heading error of the

R. Izquierdo, A. Quintanar, I. Parra, D. Fernández-Llorca and M. A. Sotelo are with the Computer Engineering Department, Universidad de Alcalá, Alcalá de Henares, Spain ruben.izquierdo@uah.es

vehicle with respect to the road. In [5], a simple classification approach using a Naive Bayesian Classifier with Gaussian Mixture as distribution model was proposed using only three features: lateral speed, the preceding vehicles speed and the lateral position with respect to the lane center. The same authors proposed in [3] a probabilistic regression approach using Random Decision Forest and a Mixture of Experts. Predicting lateral motion using neural networks was proposed in [6] and [7], which can be further used to predict lane-change intentions using deterministic classifiers such as SVM [7], [8]. The use of vehicle tracks to infer maneuver classes and future trajectories is a common approach that can be applied using LSTM neural networks [9] and convolutional social pooling [10]. The use of simplified representations of complex interactions and trajectories of the traffic participants combined with a CNN to infer lane-change intention was proposed in [11]. A compact, binary and simplified birds-eye view is used with one channel for the vehicles and one channel for the lanes. Previous frames are stacked to account for temporal information. As in our approach, this is an attempt to generate a simple representation of complex interactions. However, it does not innately take into account appearance information that can be relevant when inferring future maneuvers of surrounding vehicles (e.g., turn signals).

The rest of the paper is organized as follows. Section II provides a thorough description of the methodology followed to extract the input-output data from the PREVENTION dataset. Section III describes a new CNN-based approach to predict lane changes using image sequences encoded in single frames. A different time-integration approach based on GoogleNet, LSTM, and ROI selection method is explained in section IV. Section V presents significant results of the models proposed in sections III and IV. Finally, section VI concludes the paper, providing some insights into future developments.

II. Methodology - Labeling

In this section, the methodology employed to prepare the data and label it is detailed. As said in the previous section, data used for this work have been extracted from the PREVENTION dataset, making the most of nearly 6 hours of front video records conveniently labeled and tracked. First, the data used and the extraction process are described. Then, the lane change labeling and ground truth generation are in-depth handled, followed by the ID association process, tracking and filtering tasks. Finally, finished and enhanced lane change structures are provided, which are to be used in sections III and IV.

A. Data Extraction

Data used for this work have been extracted from the novel PREVENTION dataset, ready to download from `https://prevention-dataset.uah.es`. At the moment only image information and labels from both CNN and manual inputs are employed. The image is acquired by a Grasshopper3 camera, mounting 12.5 mm fixed focal length lens. The

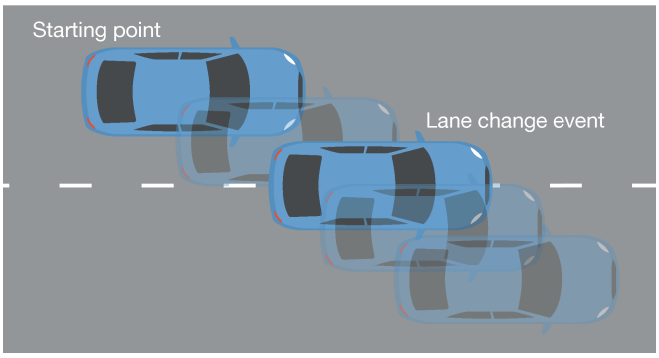


Fig. 2: Complete lane change labeling: starting and lane change event points.

Field of View (FOV) covered by the camera is 48°, featuring a SONY WUXGA (1920x1080) CMOS Bayer array sensor, which can be triggered up to 163 Hz. In the dataset, cameras are triggered at the LiDAR spinning rate, around 10 Hz.

A top-level segmentation is applied to the scene, distinguishing between *cars*, *trucks*, *buses*, *motorcycles*, *bicycles*, and *pedestrians*. For this purpose, it is necessary to focus only on vehicles, so pedestrians and bicycles detections have not been used. The labels given in the dataset have been generated using the Detectron framework [12], taking advantage of the top-class state-of-the-art Mask-R-CNN [13] model beside a ResNet-101 [14] backbone used as instance segmentation engine. Contours and bounding boxes are provided as raw output detections, as well as a temporal integration of the detections.

Moreover, these detections are filtered out, featuring only those with a confidence value over 0.5. After applying that filter, a non-maximal suppression algorithm is executed. Before finishing data preprocessing, a Hungarian Matrix script uses the modified Intersection over Union (mIoU) as the inverse of the distance (Eq. 1 where A_1 and A_2 are the evaluated areas) to time-based associate detections, assigning coherent IDs to them.

$$\text{mIoU} = A_1 \cap A_2 / \min\{A_1, A_2\} \tag{1}$$

B. Data Labeling and Ground Truth Generation

There is a file included in each driving record downloaded from the dataset called `lane_change.txt`, where each line has four values $[id, type, frame, val]$ that indicate various characteristics of the lane change performed by a vehicle, finishing in that frame and identified with a defined ID. The parameter val is used for time-lapse events annotations, rather, it indicates the initial frame of the lane change. Lane changes can be *left* (3) or *right* (4), starting when it is becoming clearly that the vehicle is making a lane change maneuver, it does not matter whether the turn signal is being used or not.

In order to enhance the already labeled data, turn signal information is now included, indicating whether the turn

signal has been activated at some point during the lane change or not (1 or 0, respectively). Consequently, the new file structure of `lane_change.txt` is [id, $type$, $frame$, val, $signal$]

Figure 2 shows labeling basis: the starting point identified in $frame$ is the frame where the driver has clearly shown motivation to begin the lane change: activating the turn signal or modifying its reasonably straight-shaped trajectory within the ego-lane to leave it are both indicators of that behavior. The lane change event in val is labeled as the frame where the middle of the rear bumper is located just over the lane pavement markings, rather, the vehicle is half in the side lane: this fact is important to understand the labels. All the files mentioned previously are available in the PREVENTION dataset as it, in the next subsection new filtering tasks are deployed.

C. Tracking and Fine-grain Filtering

As raw data was pointless as it to focus in the main actors in the scene, accurate filtering is needed to erase some detections, hence a new feature has been included in the labeling tool [15], helping to erase detections with just a click. That deletion process is, in fact, just a change in the stored ID, modifying the sign of the number so that negative IDs will not be taken into account. If necessary, it would be easy to recover that *erased* detections. Analyzing the recordings, it is clear that there are some problems related to the automatic tracking based in mIoU and Hungarian Matrix, especially regarding oncoming traffic. That issue was causing oncoming IDs to merge with some other detections tracked in the *fast lane*, apart from getting bad lane change annotations, as the vehicle ID differ during the lane change maneuver. It is necessary to face a trade-off between increasing the confidence value to improve the precision or lowering it to raise the recall, the second option has been chosen, alongside the manual adjustment of faulty annotations. Note that other lane changes that were irrelevant, far from the recording car or very tricky to understand because of their label have been conveniently removed as well. Some statistics about the number of lane changes and detections after filtering are provided in table I.

TABLE I: Detections & Lane Change Statistics

Record #	**1**	**2**	**3**	**4**	**5**
Left LC	22	34	35	104	97
Right LC	50	43	41	150	138
Unique IDs	2669	4674	4401	13757	15018
IDs changing lanes	2163	2249	3033	8224	7469
Detections while LC	8826	9842	10777	49308	41534
Frames while LC	2222	7467	3107	8417	7781
Mean frames of LC	**40.6078**				
Mean time of LC	**3.76 s**				

III. CNN & Motion History

This section proposes a new methodology to predict vehicle intentions using CNN's and video data encoded in a single image. Many types of intentions can be predicted about drivers' intentions such as left/right lane change, left/right turn, overtake, cut-in, cut-out, etc. However, these predictions can be simplified to 3 maneuvers in highway scenarios; *left*, *none* and *right* lane change. The more relevant points to take into account when predicting intentions are context and temporal inter-dependencies. This methodology includes in a standard three channels image three points: context, motion history, and prediction target selection.

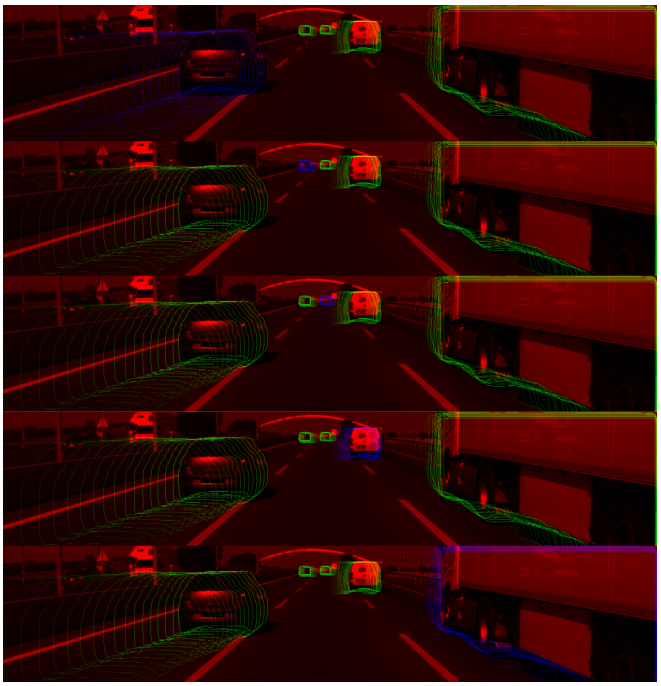


Fig. 3: Context and encoded movement histories. Red channel is used to store the scene appearance as a grayscale image. Blue channel stores the prediction target movement history. Green channel stores surrounding vehicles movement histories.

A. Motion History

Stacking raw images as input for CNN decision or prediction problems is nowadays still computationally unfeasible. To work around this problem the motion history of the involved agents in the scene is added to the image preserving the original size and depth. To generate visual relevant information the contours of the vehicles are represented in the image using different intensity values to encode the time step information. Up to 10 past contours of vehicles are included in the image as their history. Every contour is represented with the same intensity value than the contour of the previous time step incremented by 10. The current time step contour is represented with an intensity value of 200.

Other of the presented issues is the multi-agent prediction problem. There can be more than one vehicle in an image, therefore, intentions must be predicted for each vehicle. The predictions must but treated as a collective problem due

to the strong interaction between vehicles. To do so, the prediction target history is encoded in a channel and the surrounding vehicles histories in other. Fig. 3 shows how context information and movement histories are combined to generate time-integrated understandable information. Note that images in fig. 3 are focused on one vehicle at each time. First image corresponds with a *right* lane change, fourth with a *left* lane change and all other as *none*.

B. CNN Architecture

The trained state-of-the-art models do not represent a good choice to make transfer learning due to the artificial nature of the input data. Considering this, a CNN has been designed from scratch. Table II summarizes the architecture of the CNN, which basically consist of an input layer to standardize the images removing dataset mean value and dividing by the standard deviation. Then, five blocks of 2D Convolution, Batch Normalization, and ReLU layers reduce the data size and increase the data depth. Finally, the image is classified by an ending block composed of dropout, fully-connected and soft-Max layers.

TABLE II: Network Architecture

Layer	**Parameters**
imageInput	600×1920×3, $\mu = 0$, $\sigma = 1$
convolution2D + BN + ReLU	31×31×8, stride 0
convolution2D + BN + ReLU	27×27×16, stride 3
convolution2D + BN + ReLU	21×21×32, stride 3
convolution2D + BN + ReLU	15×15×64, stride 3
convolution2D + BN + ReLU	9×9×128, stride 3
dropout	$p = 0.3$
fullyConnected	2×19×128×3
softmax	$n = 3$
classification	*cross-entropy*

C. Training Hyper-Parameters

More than 1.4M detections are provided in the dataset. There are 120K detections only during the 714 lane change events. Analyzing the data, 23K (9K left and 14K right) are detections of lane change maneuver and 97K are detections of surrounding vehicles which are not performing a lane change. The images used to train, validate and test the CNN are only those which occur during the lane change maneuvers. The number of *none* samples have been reduced to be equal as the *left* or *right* samples to prevent imbalance class problems. Training, validation and test subsets are generated with a rate of 0.6, 0.2 and 0.2 respectively. The three subsets are disjoint time intervals. Training hyper-parameters are listed: mini-batch size = 64, epoch = 16, initial learning rate = 0.001, learning rate after 8^{th} epoch = 0.0001, gradient threshold = 1 and cross entropy loss function.

IV. GoogleNet & LSTM

The LSTM (Long-Short Term Memory) is a kind of cell specialized to fit and learn temporal patterns in data sequences. The problem with this kind of networks is that they can deal with neither images nor videos due to the big size of input data (more than 1M of values per image in our case). Trained CNNs can be used to reduce the input data size while retaining relevant information. To do so, GoogleNet [16] CNN trained on ImageNet [17] is used as image encoder obtaining the output of the last pooling layer *pool5-7x7_s1*. This layer produces a 1024×1 feature vector which can be concatenated in a column order with consecutive features vectors as an encoded video sequence.

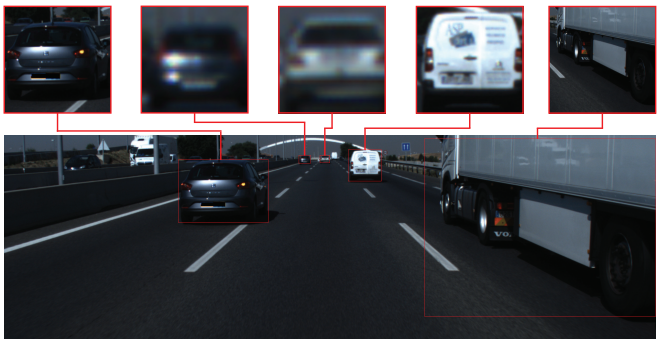


Fig. 4: Vehicle ROI selection. ROIs are generated to be processed as an independent image for each vehicle.

Fig. 5: Different ROI generation methods. From left to right: image size based, detection size based, and double detection size based.

A. ROI Generation

One of the problems when trying to recognize maneuvers in images or video is that could be multiple vehicles performing different actions at the same time. Each agent in the scene must be analyzed individually. To do so, the image is split in as many ROIs as vehicle there are in the image. Figure 4 shows how the image is transformed into different ROIs to be evaluated as separated elements. Fig. 4 shows all the vehicle detection in the image represented by a red rectangle and the generated ROI for each detection.

Three different approaches have been carried out to select the ROIs. As the GoogleNet input size is square, width and height of the ROIs are equals in any case to avoid spatial distortions. Moreover, the ROIs must be resized to fit in the GoogleNet input size which is 224×224. In the three methods described below the center of the ROI is set with the center of the vehicle detection coordinates.

- Image size based: the width and height of the ROI are set with the original image height (600px). This

method provides a mobile window following the vehicle movement over the image. As far as the height of the vehicle bounding box is smaller than the image height some context information is added in the ROI.

- Detection size based: the ROI size is set with the greater value between the height or width of the vehicle's bounding box. Minimum context information is added in the ROI, just the area needed to complete the square around the vehicle.
- Double detection size based: the ROI size is set with the double of the greater dimension of the vehicle's bounding box. If the ROI dimensions exceed the image size them are limited to the more restrictive value, in this case, the image height. This approach tries to focus on the vehicle and add some context information.

Fig. 5 shows these three types of ROI selection methods. The first method adds a lot of surrounding information. This could be a problem when vehicles are close ones to each other. The second one is more clear about where is the focus, independently if there are vehicles close or not. However, context information is completely missed. The third one is a trade-off between the two first methods.

B. Extended Feature vector

The ROI generation process misses the relative image location and real vehicle size. These values are added to the features vector in order to provide more complete information to the LSTM layer. The center (X, Y) and the dimensions (W, H) of the bounding box are appended to the features vector generated by the GoogleNet. For an easier understanding and homogeneous data interpretation, the values have been ranged between 0 to 1 according to the values in the feature vector. This process transforms the top-left corner to (0,0) and the right bottom to (1,1). Indirectly, movement information is added in the frames, such as speed, acceleration, and direction.

C. Network Architecture

The network is composed of an input layer which includes googleNet architecture and the original ROI parameters. The ROI parameters concatenation is only applied when the extended feature vector is used. Then, an LSTM layer with 2000 cells performs the "core" of the classification based on the feed sequences. Finally, dropout, fullyConnected, and softmax layers block classify the sequence. This structure is summarized in table III.

TABLE III: Network Architecture

Layer	Parameters
GoogleNet encoder	1024×1
ROI parameters concatenation	4×1
LSTM	2000 cells
dropout	p=0.5
fullyConnected	2000×3
softmax	$n = 3$
classification	*cross-entropy*

D. Training Hyper-Parameters

The same parameters used in III were used to establish the training, validation and test sets, consequently, the same samples are used in each subset. The LSTM data input for a single detection is composed concatenating the 10 previous feature vectors of the detection. Training hyper-parameters are listed: mini-batch size = 1024, epoch = 16, initial learning rate = 0.0001, gradient threshold = 1 and cross entropy loss function.

V. Results

This section presents and discuss the results achieved deploying the approaches described in sections III and IV to predict lane change maneuvers of surrounding vehicles. For a better understanding of training times and inference rates the details of the computer used to carry out this experiments are given. PC with Kubuntu 18.04LTS, i7-7700K CPU, 32GB of RAM and NVIDIA GF-1080Ti GPU using Matlab 2019a.

A. CNN & History

The CNN and the movement histories were trained using 22140 images, two subsets of 7380 images were used for validation and test. The total training process took 27h 56m in 16 epochs and 5520 iterations using a mini-batch size of 64. Table IV shows the confusion matrix for the test set, which is completely independent of the training process. As it can be seen the best classified class is the *none* lane change class. It can be explained because it is easier to recognize the *none* lane change status. In the other hand, *left* and *right* lane changes are in a high proportion confused with the *none* status.

TABLE IV: CNN & Motion History Confusion matrix

	Target Class			
Output Class	***none***	***left***	***right***	**Precision**
none	2452	888	935	57.4%
left	163	848	179	71.3%
right	145	124	1646	86.0%
Recall	88.8%	45.6%	59.6%	67.0%

B. GoogleNet & LSTM

Results for six experiments using the GoogleNet as image encoder and LSTM layer topology are presented. Three ROI generation methods and two features vectors (extended or not) experiments have been conducted. Table V shows classification mean accuracy results for the test subset in the six experiments. The time took in the training process of these experiments was close to 4 minutes in all of them.

TABLE V: GoogleNet & LSTM Accuracy

	Feature vector	
ROI method	GoogleNet	GoogleNet + ROI
Fixed Size	0.5965	0.5721
Vehicle Size	0.7363	0.7448
Double Vehicle Size	0.7441	**0.7454**

The fixed size selection method results are significantly lower than the vehicle-based selection methods. The use of the extended feature vector improves the results slightly when using vehicle size detection methods. As it can be seen the best results are achieved by the double vehicle size ROI selection method using the extended feature vector. Confusion matrix is presented for the best configuration in table VI to provide more detailed results.

TABLE VI: GoogleNet & LSTM Confusion Matrix

	Target Class			
Predicted Class	***left***	***none***	***right***	**Precision**
left	1787	433	461	66.7%
none	102	876	167	76.5%
right	111	111	1372	86.1%
Recall	89.3%	61.7%	68.6%	74.4%

VI. Conclusions and Future Work

As conclusions, two different methodologies have applied to predict lane changes using the novel PREVENTION dataset. Preliminary results are presented to validate the utility of the dataset comparing two lane change prediction algorithms. Prediction for *right* lane changes are, generally better than *left* lane change prediction. This could be explained because many of the *right* lane changes are produced after overtaking the ego-vehicle. Usually, this type of lane changes are produced in areas close to the ego-vehicle and better image representations could make easier the prediction task.

Comparing the two algorithms proposed to predict a lane change maneuvers, the one using GoogleNet and an LSTM works better than the trained CNN. Many reasons could explain it, e.g. the number of samples is not enough to properly train a CNN from scratch, number and/or size of convolution layers are not the optimal values, or simply, GoogleNet is better trained and it can encode the relevant information in a better way.

The use of the extended feature vector and three different ROI selection method have been evaluated when using the GoogleNet plus LSTM algorithm. The double-vehicle-size ROI selection method reveals to be the best choice. The use of the extended feature vector which includes the original ROI parameters slightly increases the classification performance.

As future works, more complex analysis of training and input configuration values can be conducted to improve the results up to their maximal potential. LiDAR and radar information, which are available in the PREVENTION dataset can be incorporated in the prediction algorithms adding 3D positioning to improve their results.

ACKNOWLEDGMENT

This work was funded by Research Grants SEGVAUTO S2013/MIT-2713 (CAM), DPI2017-90035-R (Spanish Min. of Economy), BRAVE Project, H2020, Contract #723021 and FPU14/02694 (Spanish Min. of Education) via a predoctoral grant to the first author. This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737469 (AutoDrive Project). This Joint Undertaking receives support from the European Unions Horizon 2020 research and innovation programme and Germany, Austria, Spain, Italy, Latvia, Belgium, Netherlands, Sweden, Finland, Lithuania, Czech Republic, Romania, Norway.

References

[1] R. Izquierdo, A. Quintanar, I. Parra, D. Fernandez-Llorca, and M. A. Sotelo, "The prevention dataset," https:prevention-dataset.uah.es, 2019.

[2] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, p. 1, 2014.

[3] J. Schlechtriemen, F. Wirthmueller, A. Wedel, G. Breuel, and K.-D. Kuhnert, "When will it change the lane? a probabilistic regression approach for rarely occurring events," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1373–1379.

[4] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 797–802.

[5] 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 *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 108–114.

[6] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 1307–1312.

[7] R. Izquierdo, I. Parra, J. Muñoz-Bulnes, D. Fernández-Llorca, and M. Sotelo, "Vehicle trajectory and lane change prediction using ann and svm classifiers," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[8] W. Yao, Q. Zeng, Y. Lin, D. Xu, H. Zhao, F. Guillemard, S. Geronimi, and F. Aioun, "On-road vehicle trajectory collection and scene-based lane change analysis: Part ii," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 206–220, 2017.

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

[10] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476.

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

[12] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[13] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[15] R. Izquierdo, I. Parra, C. Salinas, D. Fernndez-Llorca, and M. A. Sotelo, "Semi-automatic high-accuracy labelling tool for multi-modal long-range sensor dataset," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 1786–1791.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: http://arxiv.org/abs/1409.4842

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.