Two-stream neural architecture for unsafe maneuvers classification from dashcam videos and GPS/IMU sensors

Matteo Simoncini[1,2,*], Douglas Coimbra de Andrade[1], Samuele Salti[3],
Leonardo Taccari[1], Fabio Schoen[2] and Francesco Sambo[1]

***Abstract*— In this paper, we propose a novel deep learning architecture for the end-to-end classification of unsafe maneuvers from dashcam data; the proposed model is based on an innovative two-stream architecture capable of processing both video and GPS/IMU signals as input streams. A wide experimentation on a well known naturalistic driving dataset (SHRP2 NDS) shows that the two sources of information complement each other in the classification task and proves the effectiveness of the proposed approach. As a by-product of this research, we propose and make available a novel classification of safety-critical events based on the unsafe maneuver leading to them, which is representative of the real distribution of car crashes and near crashes.**

I. Introduction

Car crashes are becoming a major problem of the modern era: in 2016 in the US over 7 millions car crashes were reported, with over 2 millions injuries and roughly 35.000 fatalities [1]. This is increasing the interest of the car industry and the scientific literature in real-time car accident anticipation [2], [3] and Advanced Driver-assistance Systems (ADAS) [4]. For the same reason, vehicle owners are increasingly installing dashboard cameras (dashcams) to provide evidence in case of traffic-related accidents and incidents [5]. Such cameras are often equipped with an Inertial Measurement Unit (IMU) and a Global Positioning System unit (GPS) that record additional data such as speed, position, acceleration and angular velocity. Such sensors often act also as triggers to start video capturing: for example, once a sharp change in acceleration is detected, video footage and sensor data for some seconds before and after the event are recorded and stored locally, or sent to a server [6].

Motivated by the growing interest in traffic safety and the rising adoption of dashcams, in this paper we focus on crash and also near crash events (collectively called *safety-critical events*), and specifically on automatic recognition from dashcam footage and companion sensor streams of the maneuvers leading to them, performed either by the subject vehicle or by other vehicles or entities on the road: we name this problem *unsafe maneuver classification*. We see this as a first step towards the more general *unsafe maneuver detection* problem, while being at the same time practically useful on its own: this technology could become, for instance, the building block of a driver coaching platform, effectively contributing to the ultimate aim of preventing car accidents on the road.

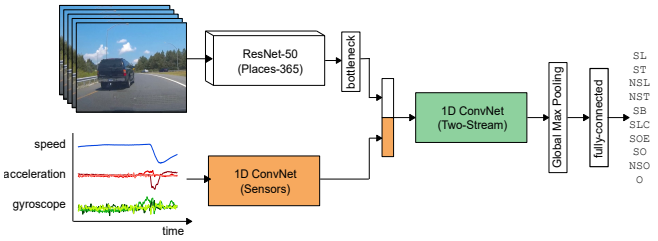

Fig. 1: Two-stream architecture for unsafe maneuver classification. In white, the *video feature extractor*, a ResNet-50 applied to each frame of the video, in orange, the *sensors feature extractor*, a 1D ConvNet and in green, the *two stram classifier*, a 1D ConvNet.

Dashcam data have been used to tackle similar problems in applications like insurance, fleet management and self-driving vehicles, to detect and classify car crashes [6], [7], [8] or driving maneuvers [9], [10], [11]. Instead, we consider both crashes and near crashes. Moreover, we consider maneuvers performed both by the subject (*subject vehicle* `SV` maneuvers, sometimes also referred to as *ego-vehicle maneuvers*) and by other vehicles (*non-subject vehicle* `NSV` maneuvers). Finally, we consider both maneuvers involving multiple vehicles (*e.g.,* improper lane change, turning) and single-vehicle maneuvers (*e.g.,* loss of vehicle control, vehicle over the edge of the road). To the best of our knowledge, no work has addressed unsafe maneuver classification in such a general way.

To ground the definition of our task on the real distribution of road events, we contribute a taxonomy of unsafe maneuvers based on the outcomes of a large-scale Naturalistic Driving Study (NDS) [12], *i.e.,* a collection of real dashcam data acquired over an extended period of time (years, in this case) from a large amount of volunteers driving across multiple states. We then propose to address the resulting classification problem with an end-to-end Deep Learning approach and a new two-stream architecture, leveraging both the video stream and the information coming from the GPS/IMU sensors. Figure 1 presents a schema of the proposed architecture. This choice of sensor data matches what is typically provided by a dashcam, which has no access, for instance, to throttle position, turn signal and brake, which could be useful for ego-vehicle maneuver detection. Moreover, although additional information, *e.g.,* driving

[1]Verizon Connect Research, Florence, Italy
[2]DINFO, University of Florence, Italy
[3]DISI, University of Bologna, Italy
*Email: `matteo.simoncini@unifi.it`

style, ego-vehicle model, could be used to assess unsafeness and liability of the detected maneuvers, we deiced to not leverage them as they might be unavailable during inference. The contributions of this work are thus the following:

- We introduce the new unsafe maneuver classification problem, aimed at classifying maneuvers that lead to safety-critical events
- We provide a taxonomy of unsafe maneuvers based on a Naturalistic Driving Study (NDS) that can be used in classification and detection tasks
- We propose a new two-streams convolution-based Deep Learning architecture, leveraging both video and sensors data, and perform extensive tests to show the impact of the two input streams on the classification results.

The remaining of the paper is structured as follows. In Section II we present a review of the scientific literature related to our task. In Section III we present the unsafe maneuver taxonomy we propose. Section IV describes the two-stream architecture while Section V reports extensive results assessing the performance of the proposed architecture.

II. Related work

To the best of our knowledge, the problem proposed in this paper has not been addressed before. The most closely related works in the literature are in the field of *Accident detection / anticipation* and *Driving maneuver detection*.

Accident detection / anticipation In the context of accident anticipation, Chan *et al.* [2] proposed a system for anticipating traffic accident from dashcam videos. They used an object detection algorithm to extract the objects in the scene and computed the features of a pre-trained VGG neural network [13] on their locations. Then, they introduced a Dynamic Spatial Attention (DSA) system in combination with a Long short-term memory (LSTM) network and a custom loss to predict the car crash as earlier as possible. They evaluated their performance on the novel DAD dataset that, however, is formed mostly of accidents not involving the ego-vehicle. Suzuki *et al.* [3] improved the previous architecture by using a Quasi-Recurrent Neural Network (QRNN) and an adaptive custom loss. They also evaluated their performance on the broader NIDB dataset, which is composed mostly of ego-vehicle accidents.

In the context of accident detection, Yao *et al.* [7] addressed the problem in an unsupervised way, by training a network to predict the position of objects in the scene in the next frame and by detecting anomalies with respect to the actual position. Taccari *et al.* [6] designed a system based on object detection and Random Forest to classify safety-critical events into crashes, near-crashes and safe events.

All the aforementioned approaches heavily rely on object detection to perform the classification or the prediction. Clearly, this approach does not generalize to events involving only the subject vehicle (*e.g.,* loss of control), where no other vehicle is visible. Recently, some works have investigated the driving attention (*i.e.,* the driver eye fixation) prediction task [14] in the context of safety-critical events, under the hypothesis that such information can provide useful insights for accident detection and prediction [15], [16], [17]. Zhu *et al.* [8] leveraged this idea to detect safety-critical events in driving videos, addressing it as an anomaly detection problem. They used the eye fixation salience map to extract anomaly candidates from the full clips and used an architecture based on isolation forests to extract the spatio-temporal safety-critical regions. They also proposed a mechanism based on image segmentation to compute a narrative (*e.g., car hit motorbike* or *car hit ego-vehicle*). While this approach has shown potential, it cannot readily be applied at scale as the one we propose in this paper due to the limited availability and adoption of solutions to capture driver attention [15], [16].

Driving maneuver detection In the context of ego-vehicle maneuver detection, Peng *et al.* [9] considered both video and GPS/IMU as inputs. Video frames were fed to a pre-trained VGG network while handcrafted features were extracted from the GPS/IMU data. The two streams were then fed to an LSTM model. The authors proposed to process only frames sampled on a uniform spatial basis (*i.e.,* a frame per meter) instead of a temporal one, which they proved to be beneficial for ego-maneuver detection. Their approach doesn't extend to general maneuver detection, though, since maneuvers performed by other vehicles while the subject is not moving are ignored. Zekany *et al.* [10] proposed a method to classify subject maneuvers from videos, using a pretrained model (DeepV2D) to extract depth from video and the camera motion information (and, thus, the trajectory performed by the subject). Then, they leveraged Dynamic Time Warping (DTW) distances between trajectories to perform the classification. However, in our case, we're not interested in detecting the subject maneuver alone, but rather in classifying the maneuver with respect to its context.

In the context of other vehicle maneuvers detection, Deo *et al.* [11] designed a framework based on the detection of road scene objects and applied tracking and motion detection.

III. Unsafe Maneuver Taxonomy

One of the aims of this work is to define a taxonomy for safety-critical events based on the maneuver that led to the dangerous situation. Most of the existing work based their results on manual classification of crowd-sourced datasets (*e.g.,* acquired from YouTube videos) [2], [7], [8]. However, this is not ideal for several reasons. First, the variety of maneuvers leading to a dangerous situation in a road environment is extremely broad, *e.g.,* short distance to another object, colliding trajectories, violation of right of way or other road laws [7] and not all these maneuvers are equally likely to be depicted in videos uploaded on-line: for instance, a driver may decide not to upload videos where he is at fault. Second, the definition of safety-relevant events, and by extension of unsafe maneuver leading to it, is not unambiguous and is both related to the environmental condition of the road scene and the perception of danger of the driver [18]. To counteract this problem, multiple reviewers should be used and clear definitions to label events should be agreed upon. Third, in

contrast with previous work focusing only on ego-vehicle maneuvers [9], [10] or other vehicles maneuvers [11], [19], our aim is to define a broad categorization that considers all possible reasons leading to safety-critical events, *i.e.,* ego-vehicle maneuvers, other vehicle maneuvers but also poor road condition, the presence of objects in the roadway, animals, *etc*.

Therefore, to create a taxonomy that is representative of the real distribution of safety-critical events while addressing the above concerns, we propose to base it on a large Naturalistic Driving Study (NDS), in particular the SHRP2 dataset [12]. The SHRP2 dataset is a collection of more that 8800 safety-critical events, gathered by more than 3300 drivers between 2010 and 2013. These events have been manually annotated with event-, driver- and environment-related variables, for a total of 75 labels [12]. Multiple round of reviews were performed to validate the annotations, and careful and unambiguous definitions of all the labels attached to maneuvers and events are provided: this greatly reduces the inherent ambiguity of the derived taxonomy.

In particular, each safety-critical event in the dataset has been labeled with the start and the end of the event and the so-called *precipitating event*, *i.e., "The state of environment or action that began the event sequence under analysis"*, answering the question *"but for this action, would the crash or near-crash have occurred?"* [12], for a total of 64 different annotations. By using these annotations as our starting point, we define a set of classes, by aggregating similar SHRP2 annotations and by manually relabelling the ones not falling perfectly into a category, as described in Table I. It is worth mentioning, as highlighted in [7], that the distribution of the safety-critical events has *a long tail*, thus it is intrinsically an umbalanced problem, *e.g.,* `SB` is the most common maneuver by far. Moreover, some of the precipitating events have too few examples to constitute a statistically significant sample size. To cope with this problem, we created the classes `SO` and `NSO` containing the remaining unsafe maneuvers performed respectively by the subject vehicle and by other vehicles. The labels obtained for the SHRP dataset according to our taxonomy are available at `https://github.com/mattsim/shrp2-unsafe-maneuver`.

TABLE I: Unsafe maneuvers taxonomy

Class	Description
`SL`	*Subject lane change*. The subject performs an improper lane change, potentially from an adjacent lane, an acceleration or deceleration lane or from a parallel parking spot, drawing dangerously close to another vehicle in another lane, being it in front of the vehicle, behind the vehicle and/or with potential sideswipe threat. Alternatively, the subject invaded the lane of a car coming in the opposite direction.
`ST`	*Subject turn*. The subject performs an improper turn, potentially at an intersection, from a driveway or from a perpendicular parking spot, invading the lane or space of another vehicle proceeding in the same or opposite direction of the vehicle.
`NSL`	*Non-subject lane change*. As `SL` but with another vehicle being the one performing the unsafe maneuver.
`NST`	*Non-subject turn*. As `ST` but with another vehicle being the one performing the unsafe maneuver.
`SB`	*Subject brakes*. The subject vehicle brakes to avoid the collision with another vehicle in the same lane and going in the same direction, potentially performing an evasive maneuver.
`SOE`	*Subject over edge*. The subject vehicle runs over the edge of the road or collides with road boundaries.
`SLC`	*Subject lost control*. The subject vehicle looses control due road condition, excessive speed or other causes.
`SO`	*Subject other maneuver*. Other unsafe maneuvers performed by the subject vehicle.
`NSO`	*Non-subject other maneuver*. As `SO` but with another vehicle being the one performing the unsafe maneuver.
`O`	*Other*. Collision or near collision with animals, pedestrian, pedal-cyclist or other objects.

IV. Methodology

To address the unsafe maneuver classification problem we propose a novel two-stream architecture that can leverage both the appearance (*i.e.,* the RGB images) and the GPS/IMU information. We demonstrate the first stream to be crucial to predict the unsafe maneuvers performed by the other vehicles, while the fusion of the two is key to predict the ego-vehicle unsafe maneuvers. Thus, the proposed architecture is formed by three main modules: a video features extractor, a sensors feature extractor and a classifier combining the two streams. A schematic representation can be found in Figure 1.

A. Video features extractor

The video information is processed using a pretrained ResNet-50 [20] on the Places-365 dataset [21], from here on also referred as *backbone*. ResNet is a widely used architecture based on residual connections that has shown superior performances on the Imagenet image classification challenge [20]. Pretrained weights are known to provide superior results with respect to starting to train from random weights [22]. Moreover, a similar network has shown good results on tasks closely related to our problem [3], [9].

Formally, each video V is a sequence of frames $\{V_{t_0}, V_{t_1}, \ldots, V_{t_T}\}$ with $\{t_0, t_1, \ldots, t_T\}$ the video frames timestamps, V_{t_i} the 3-channel RGB frame at time t_i of size $W \times H$. Such sequential formulation is converted to a tensor representation in order to be fed to the neural architecture, thus, each video is represented as a tensor of size $T \times 3 \times H \times W$. The backbone is applied to each frame and, as a result, the output is a tensor of size $T \times V_{out}$, which is then reshaped (via transposition) to a size of $V_{out} \times T$, with V_{out} the number of channels of the last convolutional filter of the network. In the case of ResNet-50, $V_{out} = 2048$.

B. GPS/IMU features extractor

We consider seven type of GPS/IMU measurements: *speed*, *three-axis accelerations* and *three-axis angular velocity*, since they are the most common and broadly available. Such signals have, in general, different sampling frequencies. Moreover, in a general setting, the sensors providing them might not be aligned between each other and with the video

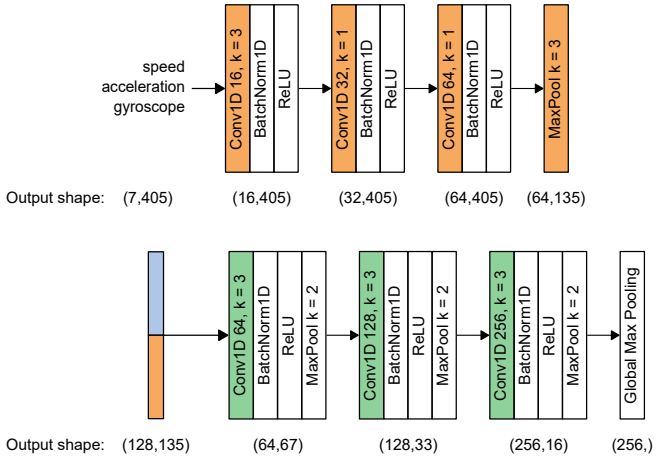

Fig. 2: Detailed overview of the sensors module (top row) and the two-stream module (bottom row), as an expansion of what presented in Figure 1. In the example $\theta = 3$, $f^s = 16$, $N^s = 3$, $f = 64$, $N = 3$ and $B = 64$ with an input of size $T = 135$.

frame timestamp. To cope with these problems, we resample the signals before processing them, so that they have the same number of samples, and this number is a multiple θ of the number of video timestamps. We do not immediately downsample the sensors to the same framerate of the videos to retain as much information as possible from the original signals for processing. Yet, resampling them at a multiple rate of the video framerate makes it easier to temporally align the extracted features with the video features after having processed the sensor streams. Therefore, this module aim is twofold: to extract some high level representation of the data, similar to what is done for the video stream; to temporally align the video and the sensors information.

Formally, each signal s is a sequence of $\theta \cdot T$ samples $\{s_{\hat{t}_0}, s_{\hat{t}_1}, \dots, s_{\hat{t}_{\theta T}}\}$ with $\hat{t}_{\theta i} = t_i \ \forall i \in \{1, \dots, T\}$ and with $s_{\hat{t}_i} \in \mathbb{R}^7$. Similarly to what we propose in Section IV-A, we represent each signal as a tensor of size $7 \times \theta T$ and we feed this tensor to a 1D convolutional neural network formed by several stacked 1D convolution filters. The network applies N^s convolutional operations. First, a convolution with kernel size θ and with f^s filters is applied, followed by $N^s - 1$ 1D convolutions with kernel size 1 and with twice the filters of the previous layer. Each convolution is followed by a 1D batch normalization and a ReLU activation function (see [20] for details). Finally, a max-pooling layer of size θ is applied, which temporally aligns the video and the sensors streams as required. A schematic representation of the described module can be found in Figure 2. The output is a tensor of size $s_{out} \times T$ with $s_{out} = f^s \cdot 2^{N^s - 1}$.

C. Two-stream classifier

This module first combines the outputs of the video and sensors feature extractors, *i.e.,* two tensors of size $V_{out} \times T$ and $s_{out} \times T$ respectively, by concatenation on the temporal dimension. However, simple concatenation may not be the best strategy to combine features as typically $V_{out} \gg s_{out}$. Further processing them so that V_{out} and s_{out} have comparable size may help in correctly leveraging the GPS/IMU information at the classifier stage.

Indeed, we observed experimentally that applying a *bottleneck layer* to the video features improves the overall performance. This layer is formed by a fully-connected layer of size B, followed by a 1D batch normalization and a ReLU activation function. Thus, the resulting concatenation is a tensor of size $(B + s_{out}) \times T$.

Such tensor is then fed to a 1D convolutional network, formed by N stacked 1D filters. Each operation is formed by a 1D convolution with kernel size 3, a 1D batch normalization and a max-pooling of size 2 and stride 2. The number of filters applied in each layer is doubled with respect to the previous one, with the first convolution having f filters, while the temporal span of the data is halved. In this way, the network is forced to learn higher level representations of the underlying data.

The output of the aforementioned filters is a tensor of size $\left(f \cdot 2^{N-1}\right) \times T'$, where T' is the temporal span after all the max-pooling layers, which is fed to a 1D Global Max Pooling layer. In our tests, such layer performed better than the commonly used Global Average Pooling layer. One possible reason is that the unsafe maneuver will occur only in a few, or even a single, temporal sample among the processed T' and, thus, the network performs better if it bases its classification only on it, without taking into account features related to safe driving. Finally, a fully-connected layer is applied, with a *softmax* activation function to perform the classification.

A schematic representation of the described module can be found in Figure 2. It is worth noticing that the proposed architecture can be used on data streams with arbitrary resolutions and number of frames, since it is formed mainly by convolutions and spatial or temporal pooling layers. The only fully connected layers are the final classifier and the bottleneck, which however do not require a fixed input size as they act after global pooling operations.

V. Experimental results

All the experiments were conducted on the SHRP2 NDS dataset [12], described in Section III. The dataset is composed of videos at 15 fps with resolution 480×356, while the GPS-related sensors have a sampling frequency of 1 Hz and the IMU of 10 Hz. The videos have variable length, going from a minimum of 150 to a maximum of 692 frames, however the vast majority are 30 seconds videos of 450 frames. To align the dataset, we capped all the videos to 450 frames length, by removing the initial frames or zero-padding the last frames when needed, and we aligned the GPS/IMU information and the event start and end to the crops. We discarded every video with missing speed or accelerometer data, while considering a constant zero signal in case of missing gyro data. Finally, we removed a few corrupted videos, ending up with a dataset of 8497 events, that we stratified split into train, validation and test with a $80/10/10\%$ proportion.

Since the backbone model has been trained on images with shape 224×224, we adjusted our input frames accordingly, maintaining the aspect ratio and having the smaller side of 224 pixels. However, to be able to perform data augmentation, we resized every video to 346×256 and picked a random 314×224 crop during training. At inference time, we only considered the central crop.

As for the GPS/IMU data, the accelerometer and the gyroscope occasionally present miscalibration artifacts. To cope with them, we first rescaled such data to have zero mean in each example and then used a robust scaling strategy, scaling the $25th$ and $75th$ percentiles of each sensor in the range $[-0.5, 0.5]$. Furthermore, Gaussian Butterworth Noise was applied, only during training, as data augmentation.

All tests were conducted minimizing the cross-entropy loss with class weight, to cope with class unbalance, and with Adam optimizer [23] with an initial learning rate $lr = 10^{-3}$, reduced to $lr = 10^{-4}$ after 30 epochs and to $lr = 10^{-5}$ after 40 epochs. Moreover, we used a weight decay $wd = 5 \cdot 10^{-3}$. The training process took 20 minutes on a V100 GPU by precomputing and storing locally the backbone outputs.

Finally, as evaluation metric we used the mean average precision (*mAP*), which is equivalent to computing the mean area under the precision-recall curve for each class and, thus, takes into accounts both precision and recall and is robust to class unbalance. We set up two sets of tests. First, in Section V-A, we evaluated the architecture on a small clip around the event, showing how performance of the model changes with the different architecture parameters and the relative benefits of the various choices made in its definition. Second, we tested our architecture on the full videos, that contain a good portion of safe driving before the safety-critical event, showing the capability of the proposed architecture to focus on the safety-critical part of the clip.

A. Clip around the event

The first set of experiments were conducted on a small clip containing only the safety-critical event, in order to prove the capability to distinguish different types of unsafe maneuvers of the proposed architecture. To do this, we considered the $2/3$ of the $[eventStart, eventEnd]$ segment as event reference point, and we considered the clip that goes from σ_{start} frames before to σ_{end} frames after such point. We empirically found that $\sigma_{start} = 90$ (6 seconds) and $\sigma_{end} = 45$ (3 seconds) let us exclude most of the footage of safe driving, while retaining the entire relevant maneuver.

The proposed architecture, as described in Section IV, has many hyperparameters to be tuned. Specifically, θ, N^s, f^s, N, f and B. Intuitively, we expect joint dependencies between the parameters in their effect on the network performance. For instance, according to the best practice of gently increase the number of filters while decrease the dimension of the data, a larger B might require a larger f. Fine tuning the parameters one by one might thus lead to a suboptimal solution. For this reason, we decided to use a Random Search [24] strategy on the parameters space, optimizing the validation *mAP*. Parameters and the range use during Random Search are reported in Table II. We run a total of 60 experiments and found the best results with the following setting: $\theta = 3$, $f^s = 32$, $N^s = 2$, $B = 32$, $f = 32$ and $N = 4$, with a *mAP* on the validation set of 0.653 and a *mAP* on the test set of 0.635.

TABLE II: Hyperparameters search space

Variable	Parameters	Variable	Parameters
θ	$\{1, 3\}$	B	$\{32, 64, 128\}$
N^s	$\{1, 2, 3\}$	N	$\{3, 4, 5, 6\}$
f^s	$\{16, 32, 64\}$	f	$\{32, 64, 128, 256\}$

The confusion matrix on the test set is reported in Table III. Thanks to the presence of the class weight in the training loss function, all the classes, including the minority ones, have overall reasonable precision, recall and AP, with two notable exceptions. The SL class tends to be confused with NSL and SB. We believe that the first case is due to a natural ambiguity of the two classes, *e.g.,* in a narrow road, maybe without a lane separation line, it is sometimes hard to say whether the subject invaded the other lane or the opposite, while the second one is mostly due to the classifier focusing on the wrong vehicle to detect the safety-critical event. The ST class tends to be confused with NST and 0. The first case is composed by events in which both vehicles are turning, into same or opposite direction and, thus, could be disambiguated only considering the road laws. In the second one, instead, the network does not seem to focus on the pedestrian or pedal-cyclist in the scene, possibly due adverse lightning conditions.

TABLE III: Confusion matrix for the best configuration on small clip around event settings

		Predicted maneuvers										**total**	**recall**
		SL	ST	NSL	NST	SB	SOE	SLC	SO	NSO	O		
True maneuvers	SL	**12**	1	8	0	4	1	1	0	2	1	30	.40
	ST	1	**15**	0	4	0	2	0	2	0	1	25	.60
	NSL	11	2	**84**	22	20	0	0	1	3	6	149	.56
	NST	2	8	6	**52**	5	2	1	6	1	14	97	.54
	SB	11	1	30	7	**239**	3	2	3	10	5	311	.77
	SOE	2	2	0	3	0	**86**	6	1	0	7	107	.80
	SLC	0	1	0	1	0	3	**12**	0	2	0	19	.63
	SO	0	1	0	1	0	2	0	**13**	0	0	17	.76
	NSO	0	0	1	1	2	1	0	1	**13**	0	19	.68
	O	5	12	5	4	1	0	2	3	1	**41**	74	.55
precision		.27	.35	.63	.55	.88	.86	.50	.43	.41	.55	848	-

To prove the effectiveness of the proposed approach, we compared the model described above, referred as VS, with several variants:

- VS+L: A baseline classifier, working on the same features, but deploying a 2-Stacked LSTM layer, proposed in related works [2], [9].
- VS−B: The proposed architecture, without the the bottleneck layer.
- V: The proposed architecture, without the GPS/IMU stream.
- S: The proposed architecture, without the video stream.

Results are reported in Table IV. Clearly, processing the two streams is key to achieve high performance, thus

validating our hypothesis that the two sources of information complement each other in the solution to our problem. In this sense, while the video stream generally performs worse than the sensor one, it is better at detecting maneuvers involving other vehicles (*e.g.,* NST, O) while the opposite is true for the ego-vehicle ones (*e.g.,* SL, ST, SOE, SLC). Furthermore, the introduction of the bottleneck layer is beneficial for the classification. Finally, even when deploying both feature streams, there is a large gap in mAP between the proposed classifier based on convolutions and pooling layers and the LSTM layers typically used in the state of the art.

TABLE IV: Results of the proposed approach and baselines

model	average precision (AP)										mAP
	SL	ST	NSL	NST	SB	SOE	SLC	SO	NSO	O	
VS	.28	.54	.61	.60	.91	.92	.60	**.68**	.62	.59	**.635**
VS-B	.20	.48	**.66**	**.61**	**.93**	.91	.55	.57	**.70**	**.66**	.627
VS+L	.17	.46	.57	.47	.89	.90	**.64**	.67	.31	.62	.569
S	**.35**	**.55**	.57	.30	.80	**.94**	.59	.65	.59	.23	.556
V	.10	.27	.51	.42	.84	.59	.51	.31	.07	.51	.414

B. Full video

As second set of experiments we applied the same architecture to the full 450 frames (30 seconds) videos, to prove the ability of the model to focus on the safety-critical event among several other safe maneuvers. We retained the optimal architecture parameters from the previous experiment and the same *train/validation/test* split, but the architecture was retrained from scratch. As a result, we obtained a validation *mAP* of 0.648 and a test *mAP* of 0.634. Such results are comparable with the ones described in Section V-A, showing the capability of the proposed architecture to automatically focus on the relevant safety-critical part of the video, an important trait in a practical deployment of our solution.

VI. Conclusions

In this paper, we have introduced the novel unsafe maneuver categorization problem. We have also proposed a taxonomy of its classes based on a NDS and, thus, on the real distribution of safety-critical events. Then, we have shown how a novel neural architecture, processing both videos captured by a dashcam as well as GPS/IMU sensor data, can be used to tackle it. After a broad calibration phase, exhaustive tests have shown the capability of the proposed architecture to distinguish the various unsafe maneuver types and also to correctly identifying the time interval in which an unsafe event occurred within a recording mostly composed of normal behaviour. The problem we propose in this paper has not been investigated before in the scientific literature; we hope our study will call for further research on this important topic, which we believe is conducive to improve driving safety and accident prevention.

References

[1] National Highway Traffic Safety Administration, "Police-reported motor vehicle traffic crashes in 2016," *Washington, DC: National Highway Traffic Safety Administration*, 2018. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812501

[2] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 136–153.

[3] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident db," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3521–3529.

[4] P. Kaur and R. Sobti, "Current challenges in modelling advanced driver assistance systems: Future trends and advancements," in *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2017, pp. 236–240.

[5] S. Park, J. Kim, R. Mizouni, and U. Lee, "Motives and concerns of dashcam video sharing," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 4758–4769.

[6] L. Taccari, F. Sambo, L. Bravi, S. Salti, L. Sarti, M. Simoncini, and A. Lori, "Classification of crash and near-crash events from dashcam videos and telematics," in *2018 21st International Conference on Intelligent Transportation Systems*. IEEE, 2018, pp. 2460–2465.

[7] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised traffic accident detection in first-person videos," *arXiv preprint arXiv:1903.00618*, 2019.

[8] R. Zhu, J. Fang, H. Xu, and J. Xue, "Progressive temporal-spatial-semantic analysis of driving anomaly detection and recounting," *Sensors*, vol. 19, no. 23, p. 5098, 2019.

[9] X. Peng, R. Liu, Y. L. Murphey, S. Stent, and Y. Li, "Driving maneuver detection via sequence learning from vehicle signals and video images," in *2018 International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 1265–1270.

[10] S. A. Zekany, R. G. Dreslinski, and T. F. Wenisch, "Classifying ego-vehicle road maneuvers from dashcam video," in *2019 IEEE Intelligent Transportation Systems Conference*. IEEE, 2019, pp. 1204–1210.

[11] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, 2018.

[12] J. M. Hankey, M. A. Perez, and J. A. McClafferty, "Description of the SHRP 2 naturalistic database and the crash, near-crash, and baseline data sets," Virginia Tech Transportation Institute, Tech. Rep., 2016.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] A. Palazzi, D. Abati, F. Solera, R. Cucchiara *et al.*, "Predicting the Driver's Focus of Attention: the DR (eye) VE Project," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1720–1733, 2018.

[15] Y. Xia, D. Zhang, J. Kim, K. Nakayama, K. Zipser, and D. Whitney, "Predicting driver attention in critical situations," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 658–674.

[16] J. Fang, D. Yan, J. Qiao, J. Xue, H. Wang, and S. Li, "DADA-2000: Can driving accident be predicted by driver attention? analyzed by a benchmark," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4303–4309.

[17] J. Fang, D. Yan, J. Qiao, and J. Xue, "DADA: A large-scale benchmark and model for driver attention prediction in accidental scenarios," *arXiv preprint arXiv:1912.12148*, 2019.

[18] M. Dozza and N. P.-e. Gonzalez, "Recognizing safety-critical events from naturalistic driving data," *Procedia - Social and Behavioral Sciences*, vol. 48, pp. 505–515, 2012.

[19] A. Breuer, J. Kirschner, S. Homoceanu, and T. Fingscheidt, "Towards tactical maneuver detection for autonomous driving based on vision only," in *Intelligent Vehicles Symposium*. IEEE, 2019, pp. 941–948.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[21] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[22] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, 2012.