Signal, Image and Video Processing
https://doi.org/10.1007/s11760-023-02512-3

ORIGINAL PAPER

Cut-in maneuver detection with self-supervised contrastive video representation learning

Yagiz Nalcakan[1,2] · **Yalin Bastanlar**[1]

Received: 21 October 2022 / Revised: 28 December 2022 / Accepted: 24 January 2023

Abstract
The detection of the maneuvers of the surrounding vehicles is important for autonomous vehicles to act accordingly to avoid possible accidents. This study proposes a framework based on contrastive representation learning to detect potentially dangerous cut-in maneuvers that can happen in front of the ego vehicle. First, the encoder network is trained in a self-supervised fashion with contrastive loss where two augmented videos of the same video clip stay close to each other in the embedding space, while augmentations from different videos stay far apart. Since no maneuver labeling is required in this step, a relatively large dataset can be used. After this self-supervised training, the encoder is fine-tuned with our cut-in/lane-pass labeled datasets. Instead of using original video frames, we simplified the scene by highlighting surrounding vehicles and ego-lane. We have investigated the use of several classification heads, augmentation types, and scene simplification alternatives. The most successful model outperforms the best fully supervised model by ∼2% with an accuracy of 92.52%.

Keywords Contrastive representation learning · Vehicle maneuver classification · Driver assistance systems

1 Introduction

An important area of research regarding advanced driver assistance systems (ADAS) is the detection of intended maneuvers of nearby vehicles. It is also one of the tough challenges to develop fully autonomous vehicles. In this study, we propose a method to detect possible dangerous lane change maneuvers (cut-ins) performed by nearby vehicles, which are among the top three causes of fatal accidents, according to a report published by the U.S. Department of Transportation National Highway Traffic Safety Administration in 2018 [1]. Therefore, our study focuses on vehicles in front and only employs a single in-vehicle RGB camera, which brings simplicity to our approach compared to other studies in the literature that use multiple sensors including radar and LiDAR [2–5].

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 (BDD) dataset [6]. The dataset consists of videos collected via the vehicle's front camera on various highways. We have cut and labeled 875 video clips containing vehicle maneuvers belonging to cut-in or lane-pass classes (Fig. 1). Additionally, to evaluate our method's performance in a different dataset, we created a cut-in/lane-pass subset from the Prevention dataset [7], which consists of 500 videos (167 cut-in, 333 lane-pass).

Although the studies focused on cut-in maneuvers are few in number, we can discuss the studies on classifying vehicle maneuvers, which is a broader perspective. Some studies on that task try to predict the future trajectories of all the vehicles in sight using their previous positions. Then, those predicted trajectories are used to classify each vehicle's maneuver [2,8,9]. Some other studies try to classify maneuvers by using features like speed, acceleration, distance to the lane line, and distance between ego-vehicle and target vehicle, which are obtained from the image or image sequence data collected from ego-vehicle vision or surveillance cameras [3,10,11]. Recently, with the success of deep learning methods and higher computational power, vision data has been used directly as input to a deep neural net-

✉ Yagiz Nalcakan
yagiznalcakan@iyte.edu.tr

Yalin Bastanlar
yalinbastanlar@iyte.edu.tr

[1] Computer Engineering, İzmir Institute of Technology, 35430 Urla, İzmir, Türkiye

[2] TTTech Auto Turkey, 35260 İzmir, Türkiye

Published online: 01 March 2023

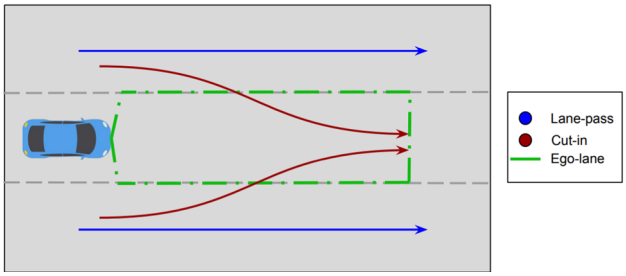

Fig. 1 Lane-pass and cut-in maneuvers (green area indicates considered safety region for ego vehicle)

work to extract various features and detect the target vehicle's maneuver [12,13].

In parallel to recent work, we develop a deep learning framework for cut-in maneuver detection. Unlike previous studies, we employ self-supervised contrastive learning following the video representation learning approach of Qian et al. [14]. Our framework comprises two phases: self-supervised training and fine-tuning for classification (Fig. 2). In the self-supervised training phase, the encoder network is pre-trained with unlabeled highway-recorded video clips with contrastive loss to learn vehicle maneuver representations and their interactions with the ego-lane. To enhance self-supervised learning, we convert recorded videos to high-level representations of the scene, which is done by segmenting vehicles and ego-lane and subtracting background. The same high-level representation extraction is applied to the prepared cut-in/lane-pass datasets and used to fine-tune the encoder while training a classification head to classify the scene whether a cut-in maneuver exists in front or not.

Our contributions can be summarized as:

1. We employed self-supervised learning for maneuver classification for the first time. Although our study only considers a certain maneuver type in a two-class classification setting, multi-class approaches, as well as other two-class detection studies (e.g. lane change detection) can benefit from contrastive representation learning to increase their performance.
2. We adopted self-supervised video representation learning to the simplified real-world video clips. We compared different simplification choices and various data augmentations for traffic scene representation.

2 Related work

Maneuver classification studies in the literature either use trajectories of surrounding vehicles or vision information directly as an input to detect their maneuvers. In trajectory-based approaches, former studies project each surrounding vehicle's trajectory on the ground plane using an on-vehicle camera, radar, or external sensors and classify the trajectory [2,8,9], whereas the latter extracts features from an image sequence and performs classification. While some studies included maneuvers like any lane-change or drift-into-ego-lane [2,8,9], some decided whether the maneuver is cut-in or not by predicting the future trajectory of the surrounding vehicles and then looking at whether this trajectory crosses into the lane of the ego-vehicle [3–5].

Most works in vision-based approaches use convolutional neural networks (CNNs). The usual practice is using a CNN to extract features from video frames and using an RNN or an LSTM as a classifier. In [11], features are extracted by a CNN on region-of-interest (ROI) and width, height, and center coordinate values are added to the feature vector. Then, the classification of the lane change maneuvers is performed by an LSTM. Their best model achieved 74.5% accuracy. Another approach in the same study [11] was converting movements of objects into contours in an RGB image and feeding CNNs with this motion history image. However, the performance was worse. Another study [13], that first crops ROIs from the original frames, exploited two modes of input video, which are high frame rate video itself and its optical flows. They compared two-stream CNNs and spatio-temporal multiplier networks. In a follow-up study [12], authors also included a slow-fast network (the one that uses videos of high and low frame rate) into the comparison which achieved 90.8% lane change detection accuracy and performed slightly better than other alternatives.

The studies discussed above used the Prevention dataset [7] which has left and right lane change and no lane change labeled videos. We see that the vision-based lane change classification results can reach ∼90% [12]. As will be detailed in Sect. 4.3, we also achieved a similar supervised learning accuracy for cut-in maneuver classification, but more importantly, we were able to improve the performance with self-supervised pre-training.

2.1 Self-supervised contrastive learning

Supervised learning usually requires a decent amount of labeled data, which is not easy to obtain for many applications. With self-supervised learning, we can use inexpensive unlabeled data to learn good enough representations and then fine-tune with supervised training with a smaller amount of labeled data. Self-supervised contrastive learning has recently gained popularity due to its achievements in computer vision [15,16]. Implemented with Siamese networks, these self-supervised approaches try to learn an embedding space in which similar samples stay close to each other while dissimilar ones are far apart. While MoCo [17] and SimCLR [18] use the negative (dissimilar) examples directly along with the positive (similar) ones, BYOL [19]

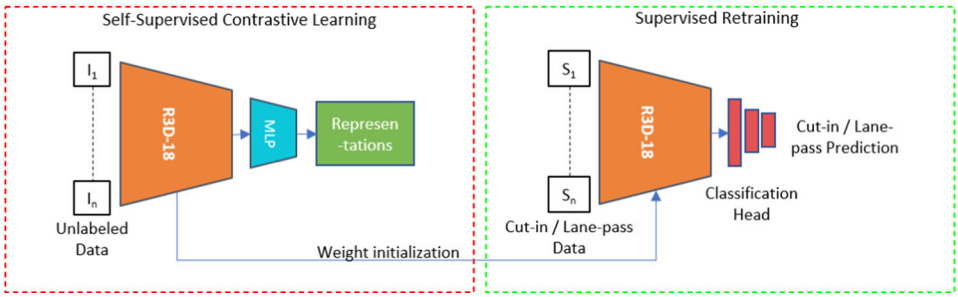

Fig. 2 Overview of the proposed framework. After self-supervised learning with contrastive loss, the MLP head is discarded, a new classification head is added to the 3D ResNet backbone, and supervised retraining is conducted

and SimSiam [20] achieved close performances just with the positive examples (different augmentations of the same sample). Qian et al. [14] introduced a self-supervised learning technique called Contrastive Video Representation Learning (CVRL) to obtain spatiotemporal visual features from unlabeled videos. The learned features are obtained through a contrastive loss function, which aims to bring together two augmented video clips from the same video in the embedding space while pushing apart the clips from different videos. In parallel, Tao et al. [21] proposed a method for learning video representations using an inter-intra contrastive framework. In addition to the negative samples coming from other videos, authors extracted negative samples from the same video (intra-negative samples) by applying undesired augmentations. Also additional inputs (e.g. optical flow) are used to get positive samples from the same video (intra-positive samples). Han et al. [22] introduced a self-supervised co-training approach for video representation learning, in which two different networks are trained concurrently on different inputs (RGB streams and optical flow) obtained from the same video. Each network provides 'hard' positives to each other. Lin et al. [23] combined contrastive learning with meta-learning in Meta Contrastive Network (MCN) to enhance video representation learning. Knights et al. [24] proposed a method to learn temporally coherent frame embeddings for video representation learning. They introduced a temporal coherence loss that encourages the embeddings of successive frames to be close.

The studies above concentrated on human action classification and worked on benchmark datasets of that task. Both action recognition and vehicle maneuver classification are tasks that involve video clips. However, there are some key differences between these two tasks. Actions may have specific features that could indicate what they are, like the tool employed (bike, golf club etc.) or the environment (in sky, on grass etc.) in which the action occurs. In addition, the order of the movements is not important most of the time (skiing, boxing etc.), which lets some studies in action recognition use some of the frames in video clips [24]. On the contrary, vehicle maneuver happens in the same environment (highways with similar background), there are no specific visual features to distinguish between different maneuvers and temporal information (order of movements) is the key to detect the maneuver of the vehicle. Our work is based on Qian et al.'s [14] self-supervised learning approach, but to overcome the challenges of vehicle maneuver classification, we employed simplified video clips and augmentations were chosen accordingly.

Fig. 3 Generation of scene representation with an example cut-in maneuver from BDD-100K Cut-in/Lane-pass Subset. Overlapping masks of vehicles and ego-lane are in different colors. The figure shows four frames of a single sequence, whereas the 3D network uses 20 of them for classification. The frame height is reduced from 600 to 400 pixels to remove ego-vehicle's hood and some sky

3 Methodology

3.1 Scene representation

Since the detection of maneuvers is directly related to vehicles and lane lines, we chose to feed the framework with a simpler representation of the scene, which includes the vehicles in front and ego-lane, rather than the original image sequence (Fig. 3). Vehicle representations were created by a state-of-the-art instance segmentation method [25] and the ego-lane mask was applied by vision-based methods. We also down-sampled the original video clips from 60 to 20 frames by taking one of every three frames.

3.2 Maneuver representation learning

Self-supervised training has been done by applying an InfoNCE contrastive loss [26] on feature tensors extracted from original and augmented video clips with a 3D-ResNet18 and MLP network. For a given video clip, contrastive loss regards augmented versions of the clip as positive pairs and other clips as negatives. Furthermore, it allows positive pairs to be close and lets others to be distant on feature space by

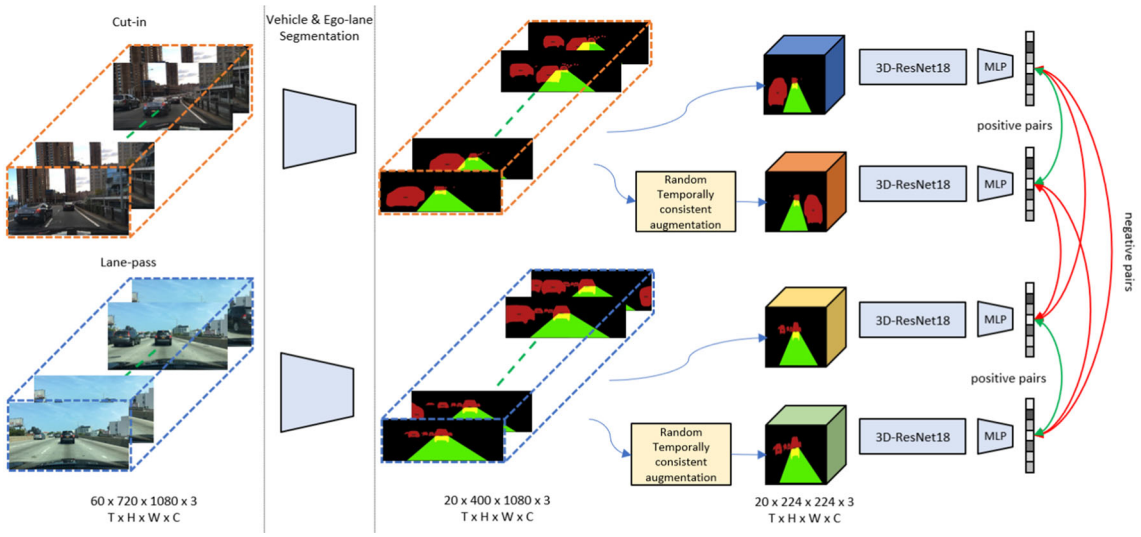

Fig. 4 Contrastive Maneuver Representation Learning. First, we create simplified video clips by extracting the vehicle and ego-lane masks from raw videos. Then, different temporally consistent augmentations (e.g. crop, flip, shear, rotation) are applied to the simplified video clips randomly before giving them as input to our video encoder (3D-ResNet18). For self-supervised learning, extracted feature tensors of each sample in mini-batch are compared with InfoNCE loss such that representations of positive pairs (green arrows) are brought together, and representations of negative pairs (red arrows) are put far apart

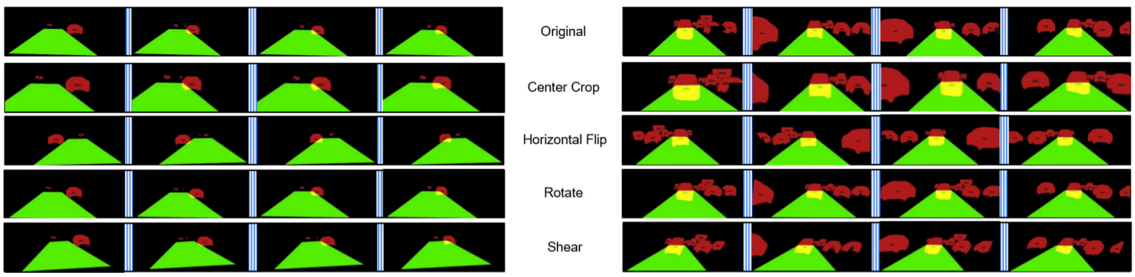

Fig. 5 Example outputs of applied augmentations on cut-in and lane-pass maneuvers. Each row shows a different augmentation of the original sequence (top row). Only *temporal elastic transformation* (TET) augmentation is not included in the figure. Since it stretches/shrinks the video sequence in time, showing the effect with a few frames is not possible

using the equation $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i$ and $\mathcal{L}_i$:

$$\mathcal{L}_i = -\log \frac{\exp\left(\text{sim}\left(z_i, z_i'\right)/\tau\right)}{\sum_{k=1}^{2N}\mathbf{1}_{[k\neq i]}\exp\left(\text{sim}\left(z_i, z_k\right)/\tau\right)} \tag{1}$$

where z_i, z_i' denotes the encoded representations of the two augmented clips of the ith video, N is the number of samples in the batch producing a total of $2N$ augmentations per batch, $\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v}/\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2$ is the inner product between two ℓ_2 normalized vectors, $\mathbf{1}_{[.]}$ is an indicator to exclude the self-similarity of video z_i, and $\tau > 0$ is a temperature parameter. Figure 4 shows details of the proposed maneuver representation learning phase.

3.3 Augmentations

To enable the self-supervised model to learn the spatial and temporal attributes of the scene, the augmentations we use should imitate different situations that may not be included in the labeled data set. At the same time, augmentations should not include cases that would not occur in real life. For that reason, we employed five different augmentations for video representation learning. Of these methods, *random rotation* and *random shear* were used to imitate the various differences in the road view of the in-vehicle camera, *horizontal flip* to simulate that the maneuver could take place on the opposite side, and *center crop* to simulate that the camera may have a narrower field of view. To ensure that representations are not affected by the random selection of augmentations, they are kept consistent temporally. In other words, the same

augmentation is applied to all frames of a video clip in the same way (Fig. 5).

In addition to the four augmentations mentioned above, considering that the speed of the maneuvering vehicle can change in time, we employed another augmentation which is called *temporal elastic transformation* (TET) [27]. The method works in one of the two different paths. In the first possibility, it stretches the beginning and the end of the video, shrinks the middle, and in the second, it does the opposite.

Algorithm 1 conveys the details of producing spatial and temporal augmentations in self-supervised contrastive learning.

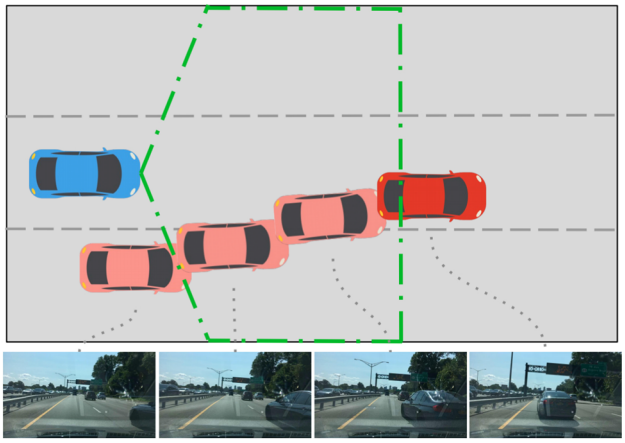

Fig. 6 Start and end points of maneuvers that are labeled as cut-in

4 Experiments

4.1 Dataset

We created unlabeled and labeled datasets from Berkeley Deep Drive Dataset [6]. Originally this dataset consisted of 100K driving videos labeled for ten 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 selected videos that happened on highways. 1220 video clips with unknown maneuver information were extracted to be used in the self-supervised learning phase. 875 video clips were cut for the fine-tuning phase, containing 405 cut-in and 470 lane-pass samples. Unlabeled versions of all 2095 video clips were used in self-supervised learning, while the labeled 875 video clips were used in the fine-tuning phase. All these clips cover two seconds of the action corresponding to 60 frames.

In addition to our main dataset, we created a Cut-in/Lane-pass subset from Prevention dataset [7] with the same labeling technique. As mentioned before, this subset consists of 500 videos (167 cut-in, 333 lane-pass). The unlabeled version of this subset was used in the self-supervised learning phase.

Figure 6 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 of whether a cut-in will occur or not. The lane change event occurs as the target vehicle enters the safety field (the polygon indicated with green lines in Fig. 6). The sequence is cut when the vehicle is entered the ego-lane with its full body (no need to be aligned in the center). For the lane-pass class, the vehicles those pass by the ego vehicle from the right-hand side or left-hand side are labeled during their stay in the safety field.

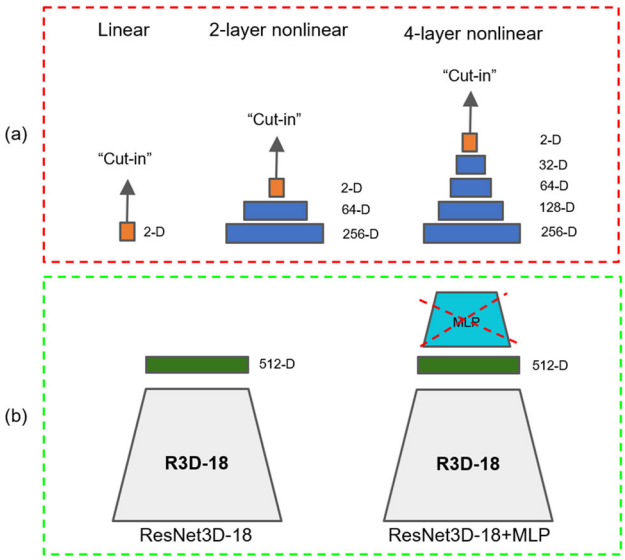

Fig. 7 Evaluated network architecture alternatives for classification head **a** and self-supervised training **b**

4.2 Video encoder

We processed video frames using ResNet3D-18 [28] with 3D convolution kernels instead of the 2D convolution kernels as in original ResNet architectures to encode spatio-temporal features of scenes. We trained two different backbone architectures, one is single ResNet3D-18, and the other is ResNet3D-18 with a multi-layer projection (MLP) head on top, as suggested in [14,18]. During supervised retraining, we dropped the MLP head and added the classification head onto ResNet3D-18 (Fig. 7b).

For the supervised retraining phase, we evaluated three different classification heads (one linear layer, two nonlinear layers, and four nonlinear layers) on top of backbones. Evaluated architectures are depicted in Fig. 7a.

Algorithm 1 Random Temporally Consistent Augmentation

Input: Video clip $V = \{I_1, I_2, \cdots, I_N\}$ with N frames `// N equal to 20 frames`
Crop: Randomly select a scale value **s** from [0.6, 1.0] Define $newwidth = width * \mathbf{s}$ and $newheight = height * \mathbf{s}$
Resize: Resize the cropped region to size of 224×224
Rotate: Randomly select a rotation degree **d** from $[-5°, 5°]$
Flip: Select a flag $\mathbf{F}_f$ from {0, 1} with 50% on 1
Shear: Randomly select a shear value **sh** for **x** and **y** coordinates from $[-0.2, 0.2]$
TET: Randomly select operation type from $\{-1, 1\}$
Select: Randomly select a number from [1, 5]
for $n \in \{1, 2, \cdots, N\}$ **do**
  $number = Select()$
  $I'_n = Flip(I_n)$ if $\mathbf{F}_f = 1$ and if $number = 1$
  $I'_n = Rotate(I_n)$ by d degree if $number = 2$
  $I'_n = Resize(Crop(I_n, newwidth, newhight))$ if $number = 3$
  $I'_n = Shear(I_n)$ transform **x** and **y** with **sh** shear value if $number = 4$
  $I'_n = TET(I_n)$ if $number = 5$
end
Output: Augmented video clip $V' = \{I'_1, I'_2, \cdots, I'_N\}$

Table 1 5-fold cross-validation results of both supervised baselines and self-supervised approaches on BDD-100K Cut-in/Lane-pass subset

Approach	Backbone	Classification head type	Best fold acc (%)	5-fold CV acc (%)	Best fold F1 score	5-fold CV F1 score
Baseline 1	MobileNetV2+LSTM	Linear	87.69	79.12	88.13	80.01
		2-layer nonlin	83.21	82.03	83.50	82.53
		4-layer nonlin	81.54	75.72	81.46	77.24
Baseline 2	R3D-18	Linear	87.79	85.19	87.91	85.19
		2-layer nonlin	93.89	90.69	93.94	90.59
		4-layer nonlin	91.60	88.70	91.84	88.70
Self-sup.1	R3D-18	Linear	94.66	91.15	94.58	91.18
		2-layer nonlin	**95.42**	92.37	**95.40**	92.30
		4-layer nonlin	94.66	**92.52**	94.58	**92.54**
Self-sup.2	R3D-18+MLP	Linear	93.13	90.99	93.23	90.98
		2-layer nonlin	93.89	92.21	93.85	92.18
		4-layer nonlin	94.66	92.21	94.61	92.16

Bold indicates the best value per metric

Self-supervised training of video encoder was performed with Adam optimizer [29], 0.1 as initial learning rate, 32 as batch size, and 0.1 for temperature τ. In the supervised re-training phase, the same optimizer was used, but the batch size was reduced to 8 and the learning rate to 0.001. Different numbers of epochs (between 200 and 500) were evaluated.

4.3 Experimental results

We compared the classification performances of three approaches by reporting the best fold and 5-fold cross-validation accuracies and F1 scores of each approach on the Cut-in/Lane-pass maneuver detection task. As supervised baselines, first we used a 2D-CNN(MobileNetV2 [30]) to extract features from each frame of the clip and give those as input to an LSTM (Baseline 1), secondly, the same 3D-CNN architecture of our CVRL approach which is a ResNet3D-18 was evaluated as Baseline 2. Baselines 1 and 2 achieved 5-fold CV accuracy of 82.03% and 90.69% with 2-layer nonlinear classification head on the BDD cut-in/lane-pass subset (Table 1). Classification accuracy increased by ∼2% and achieved 92.52% with self-supervised pre-trained ResNet3D-18 network and 4-layer nonlinear head.

When we performed the same experiment on the Prevention cut-in/lane-pass subset, Baselines 1 and 2 achieved 86.20% and 89.40% average accuracies with 4-layer nonlinear classification head respectively (Table 2). There, the proposed self-supervised trained ResNet3D-18+MLP network (with linear classification head) exceeded the performance of the best fully-supervised model by 2% and achieved 91.60%.

The most successful method among previous vision-based studies had reached a lane change classification accuracy of 90.8% with supervised learning (without cross-validation) on Prevention Dataset [12]. Although our task and the video samples in the experiments are not exactly the same (we created a cut-in/lane-pass subset), our 5-fold average accuracies of 92.52% and 91.60% are competitive. More importantly, experiments with self-supervised pretraining increased their fully supervised counterparts by 2–6% which indicates the

Table 2 5-fold cross-validation results of both supervised baselines and self-supervised approaches on the Prevention Cut-in/Lane-pass subset

Approach	Backbone	Classification head type	Best fold acc (%)	5-fold CV acc (%)	Best fold F1 score	5-fold CV F1 score
Baseline 1	MobileNetV2+LSTM	Linear	88.00	80.80	86.45	76.77
		2-layer nonlin	88.00	84.00	86.79	82.09
		4-layer nonlin	91.00	86.20	89.70	84.32
Baseline 2	R3D-18	Linear	93.00	88.60	92.20	87.09
		2-layer nonlin	91.00	88.80	90.14	87.44
		4-layer nonlin	92.00	89.40	90.94	88.00
Self-sup.1	R3D-18	Linear	93.00	90.20	92.00	88.77
		2-layer nonlin	92.00	88.60	90.85	87.01
		4-layer nonlin	89.00	87.00	87.36	85.03
Self-sup.2	R3D-18+MLP	Linear	94.00	**91.60**	93.33	**90.41**
		2-layer nonlin	93.00	90.20	92.00	88.79
		4-layer nonlin	**95.00**	90.00	**94.32**	88.58

Bold indicates the best value per metric.

Table 3 5-fold CV accuracies (%) with different data types to evaluate the effect of simplified scene representation. Video clips' original versions and simplified versions were evaluated with different highlighted information in the scene

Input type	Vehicle masks	Ego-lane mask	Linear layer	2-layer nonlinear	4-layer nonlinear
Original	x	x	57.29	58.86	59.00
Original	✓	x	57.57	60.14	60.14
Original	✓	✓	70.29	69.43	60.29
Simplified	✓	x	67.00	69.86	67.43
Simplified	✓	✓	**91.15**	**92.37**	**92.21**

Bold indicates the best value per metric.

Table 4 Experimental results of different augmentation types applied in the self-supervised learning phase. Self-sup.1 approach's (ResNet3D-18) results are given since it is the best performer in Table 1

Classification head type	Augmentations	Best fold acc (%)	5-fold CV acc (%)	Best fold F1 score	5-fold CV F1 score
Linear	Spatial	93.13	90.53	93.03	90.40
2-layer nonlin	Spatial	93.89	92.06	93.85	92.03
4-layer nonlin	Spatial	93.89	92.21	93.94	92.18
Linear	Temporal	88.55	87.63	88.50	87.58
2-layer nonlin	Temporal	85.50	81.22	85.33	81.02
4-layer nonlin	Temporal	83.21	80.31	82.95	80.01
Linear	Spatial&Temporal	94.66	91.15	94.58	91.18
2-layer nonlin	Spatial&Temporal	**95.42**	92.37	**95.40**	92.30
4-layer nonlin	Spatial&Temporal	94.66	**92.52**	94.58	**92.54**

Bold indicates the best value per metric.

potential of self-supervised learning for maneuver classification tasks.

4.4 Ablation study

Here, we present the ablation studies on augmentation types and data types. All experiments were performed on the BDD-100K Cut-in/Lane-pass subset.

First, we performed regards what would be the performance if we used original RGB frames with or without masks instead of simplified views. As results in Table 3 indicate, the simplified version outperformed the original RGB input type in either masked or unmasked form. We can also infer that the use of vehicle and ego-lane masks positively affects the classification performance in both input types.

In another ablation study, we investigate the contribution of individual augmentations to the performance of our best-

performing approach, which is Self-sup.1 in Table 1. We get the best results when all augmentations are on during the training (Table 4). We also see that the temporal augmentation itself is not sufficient, however, when added to spatial augmentations it increases the performance.

5 Conclusions

In this work, we investigated the effectiveness of self-supervised contrastive video representation learning to classify the scene whether it contains a cut-in maneuver or not. Additionally, traffic scenes were simplified to a representation of vehicles and ego-lane to enhance the self-supervised learning performance of the model. In the self-supervised learning phase, we learned from a large set of video clips without maneuver labels. Whereas a smaller labeled dataset was prepared and used to train the classifier. The proposed approach improved the fully supervised method's performance by ∼2% on 5-fold CV average accuracy. The results indicate that other supervised maneuver classification methods can benefit strongly from self-supervised learning.

Augmentations that we employed in our study are mostly extensions of the typical 2D augmentation types such as rotation, crop, flip, etc. In future, we plan to work on more creative versions where augmentations are applied to different vehicles and ego-lane separately.

Author Contributions Y.Nalcakan prepared all of the data sets and machine learning method codes and performed the experiments. Both authors wrote the manuscript, prepared the figures, and designed the detailed steps of the work. We confirm that the manuscript has been read and approved by both authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

Funding Y.Nalcakan is supported by the Scientific and Technological Research Council of Turkey (TUBITAK) 2244 Scholarship, Grant No: 2244-118C079. The numerical calculations reported in this paper were fully performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

Data availability The labeled and unlabeled simplified scene representation data is publicly available at Github.

Declarations

Conflict of interest We wish to confirm that there are no known conflicts of interest associated with this publication.

Ethical approval Not applicable.

References

1. Insurance Information Institute, Facts + Statistics: Highway safety. Accessed: 2022-06-28
2. Deo, N., Rangesh, A., Trivedi, M.M.: How would surround vehicles move? a unified framework for maneuver classification and motion prediction. IEEE Trans. Intell. Veh. **3**(2), 129–140 (2018)
3. Jeong, Y., Yi, K.: Bidirectional long short-term memory-based interactive motion prediction of cut-in vehicles in urban environments. IEEE Access **8**, 106183–106197 (2020)
4. Chen, Y., Hu, C., Wang, J.: Human-centered trajectory tracking control for autonomous vehicles with driver cut-in behavior prediction. IEEE Trans. Veh. Technol. **68**(9), 8461–8471 (2019)
5. Yoon, Y., Kim, C., Lee, J., Yi, K.: Interaction-aware probabilistic trajectory prediction of cut-in vehicles using gaussian process for proactive control of autonomous vehicles. IEEE Access **9**, 63440–63455 (2021)
6. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020)
7. Izquierdo, R., Quintanar, A., Parra, I., Fernández-Llorca, D., Sotelo, M.: The prevention dataset: A novel benchmark for prediction of vehicles intentions. In: ITSC (2019)
8. Altché, F., de La Fortelle, A.: An lstm network for highway trajectory prediction. In: ITSC (2017)
9. Scheel, O., Nagaraja, N.S., Schwarz, L., Navab, N., Tombari, F.: Attention-based lane change prediction. In: ICRA (2019)
10. Brosowsky, M., Orschau, P., Dünkel, O., Elspas, P., Slieter, D., Zöllner, M.: Joint vehicle trajectory and cut-in prediction on highways using output constrained neural networks. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI) (2021)
11. Izquierdo, R., Quintanar, A., Parra, I., Fernández-Llorca, D., Sotelo, M.: Experimental validation of lane-change intention prediction methodologies based on CNN and LSTM. In: ITSC (2019)
12. Biparva, M., Fernández-Llorca, D., Izquierdo-Gonzalo, R., Tsotsos, J.K.: Video action recognition for lane-change classification and prediction of surrounding vehicles. Preprint at arXiv:2101.05043 (2021)
13. Fernández-Llorca, D., Biparva, M., Izquierdo-Gonzalo, R., Tsotsos, J.K.: Two-stream networks for lane-change prediction of surrounding vehicles. In: ITSC (2020)
14. Qian, R., Meng, T., Gong, B., Yang, M.-H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: CVPR (2021)
15. Bastanlar, Y., Orhan, S.: Self-supervised contrastive representation learning in computer vision. In: Artificial Intelligence - Annual Volume 2022, IntechOpen. (2022)
16. Le-Khac, P.H., Healy, G., Smeaton, A.F.: Contrastive representation learning: a framework and review. Ieee Access **8**, 193907–193934 (2020)
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
18. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
19. Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: a new approach to self-supervised learning. Adv. Neural. Inf. Process. Syst. **33**, 21271–21284 (2020)
20. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR (2021)
21. Tao, L., Wang, X., Yamasaki, T.: Self-supervised video representation learning using inter-intra contrastive framework. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 2193–2201 (2020)
22. Han, T., Xie, W., Zisserman, A.: Self-supervised co-training for video representation learning. Adv. Neural. Inf. Process. Syst. **33**, 5679–5690 (2020)

23. Lin, Y., Guo, X., Lu, Y.: Self-supervised video representation learning with meta-contrastive network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8239–8249 (2021)
24. Knights, J., Harwood, B., Ward, D., Vanderkop, A., Mackenzie-Ross, O., Moghadam, P.: Temporally coherent embeddings for self-supervised video representation learning. In: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE. pp. 8914–8921 (2021)
25. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
26. Oord, A.v.d., Li, Y., Vinyals, O.: Representation Learning with Contrastive Predictive Coding. Preprint at arXiv:1807.03748 (2018)
27. Stamoulakatos, A., Cardona, J., Michie, C., Andonovic, I., Lazaridis, P., Bellekens, X., Atkinson, R., Hossain, M.M., Tachtatzis, C.: A comparison of the performance of 2d and 3d convolutional neural networks for subsea survey video classification. In: OCEANS 2021: San Diego–Porto, pp. 1–10 (2021)
28. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In: CVPR (2018)
29. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
30. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)