Hybrid RESNET and Regional Convolution Neural Network for Accident Estimation

Abstract—**This paper explores accident estimation problem and introduces an intelligent deep learning framework by exploring outlier detection, vehicle detection, and accident estimation. The road states is first collected, where an intelligent filter based on SIFT extractor and Chinese restaurant process is used to remove noises. The extended region convolution neural network is then applied to identify the closer vehicles of the given driver. The residual network will benefits from the vehicle detection process to classify whether the current road state might cause accident or not. Finally, we propose a new hyper-parameters optimization model based on evolutionary computation that can be used to tune parameters of the suggested deep learning framework. The proposed solution was tested using the well known vehicle detection and accident estimation datasets. The results are very promising and show superiority over many current state of the art solutions in terms of runtime and accuracy performances.**

Index Terms—**Deep Learning, Vehicle Detection, Accident Estimation, Region Convolution Neural Network, Residual Network, Outlier Detection, Hyper-parameters Optimization.**

I. Introduction

The need of smart mechanisms in modern city became more than necessity. Accident prediction [1], [2] is one of the hot topic in smart city applications, where the goal is to propose intelligent systems in estimating accidents before it takes place. Deep learning is largely helped in this direction, however the existing solutions for accident prediction are not mature for production, and deployment due to the accuracy performance of such solutions [3]–[5]. This paper proposes an end-to-end hybrid deep learning framework for accident estimation in road safety environment.

A. Motivation

Hybrid deep learning [6]–[8] is a hot topic in intelligent transportation applications such as group anomaly detection, object detection, and accident estimation. Vehicle detection is the task of retrieving the cars in a given urban road scene [6], [9], [10]. Vehicle detection can be very useful for accident estimation, where the detected closer vehicles of the given car in the current road scene might be beneficial for predicting whether the current road scene caused accident or not. Motivated by the success of object detection, and accident estimation models in accurately capturing the different objects, and estimating the accident this paper presents an end-to-end framework for accident estimation based on the detected closer vehicles of the given car.

B. Contributions

We developed in this work HR2CNN (Hybrid RESNET and Convolution Neural Network for Accident Estimation), an intelligent hybrid framework for accident estimation. The framework uses different tasks including outlier detection, vehicle detection, and accident estimation. The road states is first collected, where an intelligent filter based on SIFT extractor and Chinese restaurant process is used to remove noises. The extended region convolution neural network is then applied to identify the closer vehicles of the given driver. The residual network will benefits from the vehicle detection process to classify whether the current road state might cause accident or not. Finally, we propose a new hyper-parameters optimization model based on evolutionary computation that can be used to tune parameters of the suggested deep learning framework. Holding these facts as notes, our key contributions in paper can be concluded as follows:

1) We develop a novel filtering algorithm based on SIFT extractor, and Chinese restaurant process which is used to remove the noises from the image database.
2) An extended vehicle detection algorithm based on region convolution neural network is developed, by employing hard negative exploration, and multi-scale training. In addition, a residual blocking model is used to estimate the road states whether they causes accidents or not.
3) We develop an intelligent evolutionary computation algorithm to accurately explore the hyper-parameters space for finding the optimal parameters of the HR2CNN framework.
4) We evaluate the HR2CNN by establishing massive experiments on well-known data for vehicle detection, and accident estimation. The results show that HR2CNN outperforms the baseline algorithms in both runtime and accuracy.

C. Outline

The remainder of the paper is organized as follows. Section 2 reviews the main existing works for the vehicle detection and accident prediction, followed by a detailed explanation of the framework proposed in this paper in Section 3. Section

4 presents the performance evaluation. Finally, Section 5 concludes the paper.

II. Related Work

Mostofa et al. [11] proposed an up-scaling generative adversarial network for identifying multi-scale vehicles by learning the hierarchical features. Wu et al. [12] suggested a transfer learning approach for multi-source vehicle detection, while incorporating fine-tuning unsupervised learning for creating the ground-truth intelligent transportation data. Arabi et al. [13] presented an intelligent system for detecting vehicles of construction. It adopted the pre-trained MobileNet for running the model on mobile and embedded devices. Tran et al. [14] used the convolution operators to extract the urban traffic features in a streaming way for vehicle detection. Chetouane et al. [15] studied the different of object detection architectures for identifying good and relevant bounding boxes for vehicles by exploring the Gaussian mixture analysis with Kalman filter and optical flow strategy. Chen et al. [16] developed an efficient deep convolution neural network for lightweight detection by reducing the memory access. Three new guidelines are also developed in order to find the optimal number of group convolution operators. Yang et al. [17] developed a solution based on Faster region convolution neural network to simultaneously detect both 2D and 3D vehicles from a single scene. It is multi-task solution which also integrates the orientation estimation and key point detection into generic deep convolution neural network. Hassaballah et al. [18] proposed a vehicle detection method for handling emergence restriction in camera functionalities such as the weather conditions. It is multi-scale based convolution Gaussian network which combines Gaussian mixture probability with the convolution neural network to detect vehicle in uncertain environment. Fan et al. [19] investigated the use of an approximate joint training strategy to learn the vehicle detection model using the faster region convolution neural network architecture [20]. According to the authors, the results reveal superiority of the proposed model while increasing with the anchors length from 64 to 512. Wang et al. [21] developed a weighted ensemble learning method which combines the region convolution neural network [22] and the "you only look once" [23] algorithms which are the state-of-the-art of the current object detection models. The developed ensemble method is decreased by the confidence level of the object detection, where it eliminates the detection outputs of errors instead of ignoring the output of the given model. Chen et al. [24] developed the cascade pyramid region proposal convolution neural network to learn from pseudo-images for vehicle detection. It also integrated the hybrid learning using the sparse points and residual network. Dinh et al. [25] transformed the vehicle bounding boxes to a binary map which is injected to the convolution neural network. The evolutionary algorithm is also integrated in the whole system to determine the correlation among the camera parameters of the different videos of vehicles. Kumar et al. [26] developed a deep learning network to identify vehicles captured in fisheye images. It explores different tasks such as depth estimation, visual odometry, semantic and motion segmentation, vehicle lens soiling detection. In the same context, Rashed et al. [27] explored the oriented bounding box, the ellipse, and the generic polygon for determining vehicles in fisheye by designing a curvature adaptive perimeter sampling strategy for deriving the polygon vertices. Li et al. [28] proposed a convolution neural network based on light enhancement in order to increase the ability of identifying cars at night environment. It also developed a generic system which is capable to convert daytime images to low-light images, and then use the resulted images in the training phase as ground-truth data. Chen et al. [29] developed an intelligent agent solution based on explainable reinforcement learning process for vehicle detection. The semantic birdeye mask is first deduced using the sequential latent model. The trained model is combined with reinforcement learning process to explain the learning outputs.

The existing solutions for vehicle detection suffer from several drawbacks. The first one is the scalability performance in terms of accuracy and computational cost in handling real setting scenarios. The second one is that these solutions are incomplete where only the detection model are learned, there is no end-to-end safely approach for autonomous driving scenario. Motivated by the success of the recent object detection models in accurately capturing the different objects, this paper explores an intelligent and end-to-end framework for the vehicle detection for autonomous driving settings.

III. HR2CNN: Hybrid RESNET and Region Convolution Neural Network for Accident Estimation

A. Principle

The detailed explanation of the HR2CNN (Hybrid RESNET and Region Convolution Neural Network for Accident Estimation) framework is given in this part. It is hybrid framework which evolves both RESNET classifier and the region convolution neural network to estimate the accident possibility in a given intelligent transportation environment. Figure 1 shows the graphical illustration of the HR2CNN framework. It is mainly performed in four stages: 1) Removing Noises: This stage aims to remove noises before the image processing. In this context, the outlier detection algorithm is used to identify images that deviate from the whole image input data. 2) Vehicle Detection: In this stage, an extension of the region convolution neural network architecture is made in order to accurately identify the closest vehicles of the given car. 3) Accident Estimation: After the vehicle detection process, the RESNET classifier is integrated in the whole system in order to estimate for each configuration, if it might cause an accident or not, and 4) Hyper-parameters Optimization: The proposed framework requires high number of parameters to be tuned, in this stage, we propose an intelligent mechanism to automatically identify the best hyper-parameters for all the previous stages of the HR2CNN framework.

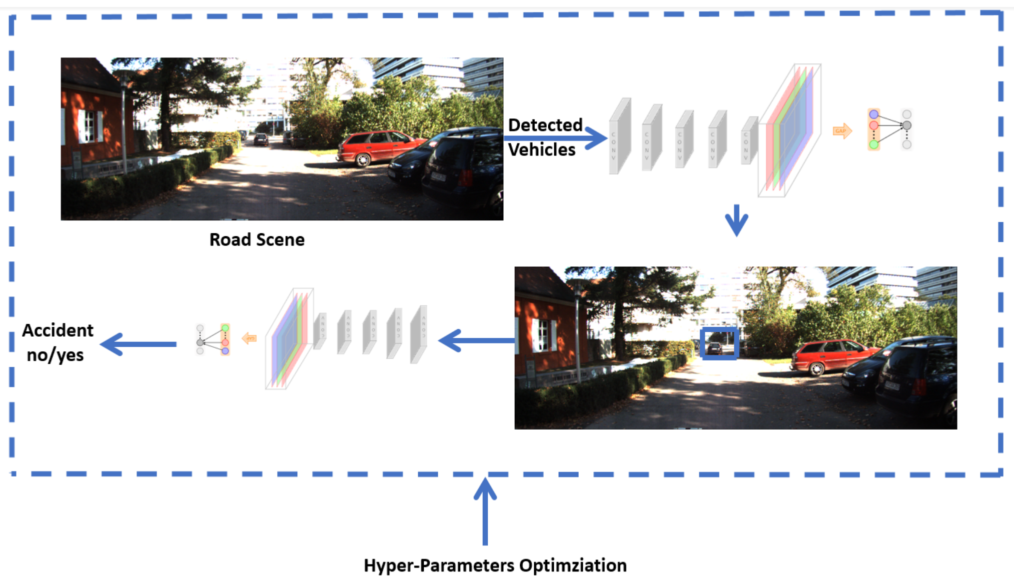

Fig. 1. HR2CNN Framework

B. Removing noises

With the appearance of emergent technology device, the image captured from sensors and cameras in the intelligent transportation environment are high resolution with immense number of pixels. From our experience, the number of pixel per image can be varied from $250,000$, and might exceed $4,000,000$ pixels. In addition, some image noises may be observed due to the error of the data acquisition. The need of data analysis before processing is crucial task, especially to mitigate generating high number of regions proposals, which can exceed billion of regions. This yield the vehicle detection process high time and memory consuming. This stage has the goal to remove outliers from the whole image database. We first consider the set of n images $I = \{I_1, I_2..I_n\}$, the outlier detection task aims to remove images from I, which deviates from the normal images. This strategy aims to prune the irrelevant images from the vehicle data images. The SIFT (Scale-Invariant Feature Transform) extractor [30] is first used to figure out the relevant features of the images . The scale space function, $S(I_i, \sigma)$, is first defined, which is based on the Gaussian kernel, K, as follows,

$$S(I_i, \sigma) = K(I_i, \sigma) * I_i, \tag{1}$$

We then identify the spatial information of each candidate keypoint based on the interpolation procedure. The spatial interpolated information is computed, which allows the stability of the extracted features. The Taylor function $Y(I_i, \sigma)$ is defined as interpolation function, given in the following,

$$Y(I_i, \sigma) = D + \frac{dY^T}{dI_i} I_i + 0.5 I^T \frac{d^2Y}{dI_i^2} I_i \tag{2}$$

The descriptor vector for the keypoints is then calculated by generating different orientation histograms of 4×4 pixel neighborhoods. After this procedure, the SIFT features for each image I_i is determined and noted by F_i.

In addition, a novel outlier detection algorithm for images is investigated in this research work. It is statistical analysis based algorithm. The detailed explanation of this algorithm is given in the following: We adopted the dirichlet process mixture model for identifying outliers from I. The set of all features

$$F = \{F_1, F_2...F_{|F|}\}$$

is transformed to k-dimensional space, where each point P_i is represented by

$$\{P_i, P_{i+1}...P_{i+k}\}$$

For instance, consider the features of ten different images.

$$F = \{F_1, F_2...F_{10}\}$$

with $k = 3$, various points are generated, each of which contains three different features. The first point contain the features $\{F_1, F_2, F_3\}$, the second one is $\{F_2, F_3, F_4\}$, the third one is $\{F_3, F_4, F_5\}$, and so on until the last point $\{F_8, F_9, F_{10}\}$. The feature are then reduced on two dimensions using the principle components analysis. The two derived dimension will be entered to the Dirichlet process for finding noises. Highly correlated groups are determined using the Chinese restaurant process. The image features are divided to the groups. Each image is assigned to the previously created group with a probability related to the number of images of such group. Otherwise, it is assigned to a new group. All images of the largest group are considered as normal, and be used in the next process, the others are considered are noises and removed from the image database.

C. Vehicle Detection

After the removing noises stage, the vehicle detection algorithm is executed to derive the closer vehicles from the given driver. We adopt the granular region convolution neural network [31] to efficiently identify the closer vehicles. It ensures uncertainty of the vehicles scenario, where vehicles can appear, and disappear at any time. This realized by incorporating probabilistic model represented by the derivative knowledge. The transfer learning is also integrated by using the pre-trained models on imagenet data [1]. The main operations of the vehicle detection stage is given in the following:

1) **Bounding Boxes Creation:** The goal of this operation is to generate the bounding box candidates. The convolution neural network is explored to accurately find the bounding box candidates. The refinement process is then executed with the regression model.
2) **Hard Negative Exploration:** The purpose of this operation is to reduce the model error. The hard negative bounding boxes are re-processed using the reinforcement learning. With this operation, the detection ratio is enhanced, and the false negative will be eliminated. In addition, unexpected behaviors of the drivers will be learned, which is common issue in intelligent transportation applications. Note that, the bounding box is considered as a hard negative if its intersection over union over the ground truth-region is less than 25%.
3) **Multi-Scale Training:** The aim of this operation is to generate bounding boxes with different sizes, this simulates the real scenario of the vehicles. Thus, vehicles and cars might be with different lengths, and widths. This might causes the regional convolution neural network based models inaccurate for vehicle detection process. To achieve this, various types of bounding boxes are generated, each type considers bounding of boxes of same height and same width.

D. Accident Estimation

After detecting the closer cars of the given driver, the next step is to estimate whether the given state of the road could cause an accident or not. The RESNET classifier is used to classifier the current state if it might be an accident or not. The input of the RESNET classifier is the current state of the road with the bounding box of each car, and the output is the class of this state, "0" for non accident, and "1" for accident. The classical deep neural architectures like AlexNet, and VGG perform the training in the entire layers, this reduces the performance of the model, in particular for deeper architecture such VGG19. RESNET is developed to address this issue by integrating new concept called "micro-architectures". The whole layers are organized in residual blocks, each of which processes different operators such as convolution, pooling, and batch normalization. Even though ResNet is much deeper than VGG16 and VGG19, this block partition based allows to effectively learn the different weights of such model. The residual blocks also allow to learn different patterns inside the data, which is missing in the state-of-the-art deep learning architecture.

[1] http://www.image-net.org/

E. Hyper-parameters Optimization

This section attempts to automatically retrieve the best parameters of the HR2CNN framework. We define the set of all parameters of the HR2CNN framework by $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 ... \mathcal{P}_{|P|}\}$. We also define the domain space of each parameter $\mathcal{P}_i$, noted $D(\mathcal{P}_i)$. It contains the possible values of $\mathcal{P}_i$. The configuration space $\mathcal{C}$ is represented by the possible configurations, where each configuration is the set of values of the all parameters in $\mathcal{P}$. Determining the optimal values of all parameters in $\mathcal{P}$ requires the exploration of all configurations in $\mathcal{C}$. This needs high computational and memory resources to proceed all configurations. In particular, for those parameters having continuous values such as the error rate. In addition, the number of all configurations is very immense, which depends to the number of all parameters, and the domain values of each parameters, It is set to $\prod_{i=1}^{|P|} |D(\mathcal{P}_i)|$. For example, if we only consider $1,000$ different values for epoch parameter (varied epoch from 1 to $1,000$, 100 different value for error rate (varied error rate from 0.01 to 1.00, and $2,000$ different values for the number of bounding boxes (from 1 to $2,000$), the number of all configurations in $\mathcal{C}$ will be 20 million configurations. Therefore, the traditional enumeration-based methods including branch and bound [32], and A* [33] will be bluntly blocked when processing the above use case. became inefficient for such high number of configuration. To solve this issue, we propose an efficient evolutionary computation based method to explore the configurations in $\mathcal{C}$. The elementary operations of the suggested approach are defined in the following:

1) **Population Initialization:** Any evolutionary computation algorithm is based on the initial population represented by the set of individuals. Each individual is defined by the possible values of each parameter in $\mathcal{P}$. For instance, if we consider the same example above, $(20, 0.75, 740)$ is a solution which represents the configuration: epochs set to 20, error rate set to 0.75, and number of bounding box set to 740. The population of our evolutionary computation algorithm should have fixed size, it means the same number of individuals. To better exploration of the configuration space, the individuals in one population should be heterogeneous. To ensure such diversity, the initial population is created by generating individuals with maximizing the distance among them. Thus, we define the distance between two solutions (individuals) S_1, and S_2, as follows,

$$D(S_1, S_2) = \sum_{i=1}^{|\mathcal{P}|} |S_1^i - S_2^i| \quad (3)$$

Note that S_1^i and S_2^i are the i^{th} value of the solution S_1, and S_2, respectively.
2) **Crossover:** Crossover operator allow the intensification process, which means it intensively explores one region in the configuration space. The following crossover operator is applied in each of two individuals of the current population: The crossover point is randomly

selected from 1 to $|\mathcal{P}|$ which allows to divide each individual into two parts, *left side*, and *right side*. The first child takes the left side of the first individual, and the right side of the second individual, where the second child takes the right side of the first individual, and the left side of the second individual. For instance, if we consider two individuals $(20, 0.75, 740)$, and $(10, 0.65, 820)$, and the crossover point is set to 2, two other individuals are generated, the first one is $(20, 0.75, 820)$, and the second one is $(10, 0.65, 740)$.

3) **Mutation:** Mutation operator allow the diversification process, which means it generates individuals far from the current region. A given parameter of each individual is randomly selected, and updated. For instance, consider the following individual generated by the crossover operator $(20, 0.75, 820)$, with the mutation operator, we can have the following individual $(15, 0.75, 125)$ by updating the first and the third elements and keeping the second element as it is.

First, the initial population is randomly generated, where each individual is created based on the population initialization. The crossover and mutation operators are then applied for exploring the configuration space. To maintain a consistent population size, each and every individual is evaluated making use of the object detection, and classification accuracy, where the best individual are kept, the others are removed. This process is repeated in multiple iterations until the max number of iterations is reached.

Algorithm 1 HR2CNN Algorithm

Input: $R = \{R_1, R_2, \dots, R_n\}$: the set of n road scenes in the training. $R_{new} = \{R^1_{new}, R^2_{new}, \dots, R^k_{new}\}$: the set of k new road scenes in the inference.
Output: $P(R_{new})$: prediction of the new road scenes whether they cause accident or not.
************Training*****************
$R \leftarrow OutlierDetection(R)$
$(M_{VO}, BB) \leftarrow VehicleDetection(R)$
$M_{AE} \leftarrow AccidentEstimation(BB)$
************Inference*****************
$P(R_{new} \leftarrow \emptyset$
for $R^i_{new} \in R_{new}$ **do**
    $BB^i_{new} \leftarrow M_{VO}(R^i_{new})$
    $P(R_{new}) \leftarrow P(R_{new}) \cup M_{AE}(BB^i_{new})$
end for
return $P(R_{new})$

Algorithm 1 presents the pseudo-code of HR2CNN algorithm. The input data is the set of n road scenes in the training, and the set of k new road scenes in the inference. The process starts by removing outliers and noises using the outlier detection step explained in Section III.B. The cleaned road scenes are trained in the object detection model as explained in Section III.C. The bounding boxes of the second step is trained to learn the accident prediction process as explained in Section III.D. As a results of the training phase, the weights of the two models, noted M_{VO}, and M_{AE} are adjusted. In the inference step, the propagation of weights of M_{VO}, and M_{AE} is performed for each new road scene, in order to predict whether it causes outlier or not. We remark that the training phase, performed only once independently from the number of road scenes in the inference, is the high time consuming task which includes several training models, and processing. However, the inference step contains only one loop, and needs simple propagation of the learned models in the training phase.

IV. Performance Evaluation

To validate the proposed HR2CNN framework, intensive experiments have been carried out for outlier detection, the vehicle detection, and the accident estimation. Several datasets are used in this research work, in the following, we give the detailed description of the datasets:

1) Kitti[2]: It is collected from camera of 15 drivers and 30 different pedestrians walking around the Karlsruhe city. It is used for vehicle detection, image retrieval, and place recognition tasks.
2) The BOXY vehicle data [34]: It is used for this purpose . It is a large vehicle detection dataset with almost two million annotated vehicles for training and evaluating object detection methods for self-driving cars on freeways.
3) Car object detection data [3]: It contains cars and vehicles in all views, and rotation. It is designed for object detection task.
4) Safe drive dataset [4]: It is related to the safe drive product which is used by the people who ride a bike or drive a car. It provides a warning to the user when an accident is possible to happen based on a particular scenario.
5) Accident dataset [5]: It contains the road status of 3,210 different cases which illustrates whether the road state might caused accident or not.
6) Self-Driving Cars [6]: This data contains different road status for given car in a self driving environment.

To evaluate the proposed framework, the mAP (mean Average Precision) is used as metric of comparison. mAP is largely used to test outlier detection, object detection, and accident estimation systems. It can be defined by,

$$mAP = \frac{\sum_{i=0}^{n} AvgP(i)}{n}, \tag{4}$$

where n is considered as the corrected objects handled among all objects, and $AvgP(i)$ is calculated as the precision results at i-rank. For example, the first i-ranked object is then taken into the consideration but ignored others. For outlier detection, the corrected objects handled are the corrected outliers. For vehicle detection, the corrected objects handled are the vehicle detected. For accident estimation, the corrected objects handled are the number of corrected accidents. The models are implemented on a machine fitted with Intel-Core

[2]http://www.cvlibs.net/datasets/kitti/
[3]https://www.kaggle.com/sshikamaru/car-object-detection
[4]https://www.kaggle.com/satishtilwani/safe-drive-dataset
[5]https://www.kaggle.com/jerrinbright/accident
[6]https://www.kaggle.com/alincijov/self-driving-cars

i7 processor and combined with NVIDIA GeForce GTX 1070 GPU. The HR2CNN is compared with the recent outlier detection, vehicle detection, and accident estimation solutions under varied number of images as input.

A. Outlier Detection Step

We compare the outlier detection step of the HR2CNN framework, with the following baseline outlier detection solutions:

1) LOF (Local Outlier Factor) [35]: LOF is a state-of-the-art outlier detection algorithm, it is based on distance reachability. It considers the set of images as point space, for each image the local outlier score is determined. This outlier score is based on neighborhood computation. If the local outlier score is close to 1, then the image is considered as normal otherwise it is outlier.
2) mRMRD (Minimum-Redundancy-Maximum-Relevance to-Density) [36]: This is a recent algorithm for outlier detection used for high dimensional data. This fits to the image data, where high number of pixels is considered for each image. It is unsupervised density based subspace selection method, which first groups the set of features on several sub-spaces. Instead of calculating the outlier score on each data point, it is determined for each subspace.

By varying the percentage of images used as input from 20% to 100%, Figure 2 shows the accuracy and the runtime of HR2CNN, the LOF, and mRMRD algorithms. The results reveal that the proposed solution outperforms the two baseline algorithms in terms of processing accuracy, and processing runtime. Thus, the mAP of the HR2CNN reached 73% for handling 100% of the Kitti data, whereas the mAP of the mRMRD does not exceed 71%, and the mAP of the LOF algorithm goes under 65% for dealing the same configuration. Regarding the runtime, the HR2CNN does not exceed 50 mile seconds for processing 100% of the Kitti data, whereas the runtime for the other algorithms reached 150 mile seconds for handling the same configuration as input. These results are obtained thanks to the efficient strategy used in the outlier detection process which is based on intelligent clustering process in find the outliers, whereas the other algorithms are based on traditional methods in determining the outliers.

B. Vehicle Detection Step

We compare the vehicle detection step of the HR2CNN framework, with the following baseline object detection solutions:

1) O-YOLO-v2 (Optimized You Only Look Once Version 2) [37]: It is based on YOLO algorithm which is considered the state-of-the-art object detection algorithms. The residual blocks are added on the YOLO-v2 for solving the gradient dispersion issue. In addition, convolution layers are added at different locations for better feature extraction and addressing the accuracy issue of the YOLO based methods.
2) Improved Faster RCNN (Improved Faster Region Convolution Neural Network) [19]: It is an extended version of Faster RCNN algorithm by proposing an approximate joint training strategy to learn the original images using the Faster RCNN architecture. An improvement of the size and the proportion of anchors is also investigated.

By varying the percentage of images used as input from 20% to 100%, Figure 3 shows the accuracy, and the runtime of HR2CNN, the O-YOLO-v2, and Improved Faster RCNN algorithms. The results reveal that the proposed solution outperforms the two baseline algorithms in terms of accuracy and it is very competitive with O-YOLO-v2 in terms of processing runtime. The mAP of the HR2CNN reached 84% for handling 100% of the Kitti data, whereas the mAP for the other algorithms goes under 75% for dealing the same configuration. Regarding the runtime, HR2CNN, and O-YOLO-v2 does not exceed 45 mile seconds for processing 100% of the Kitti data, whereas the runtime for the improved RCNN algorithm reached 85 mile seconds for handling the same configuration as input. These results are obtained thanks to the proposed operations during the vehicle detection process such as the bounding boxes creation, the hard negative exploration, and the multi-scale training. However, the other algorithms do not consider the size-varied of the objects which is common in intelligent transportation applications, in particular for vehicle detection.

C. Accident Estimation Step

We compare the accident estimation step of the HR2CNN framework, with the following baseline accident estimation solutions:

1) DSTGCN (Deep Spatio-Temporal Graph Convolutional Network) [1]: It is based on graph convolution neural network. It is performed on three steps. The first step is the learning the different correlation among the spatial information based on graph convolution operators. The second step employed the standard convolution to learn both the spatial and temporal dimensions. The third step is the interpretation of the semantic representation of contextual knowledge by adding an embedded layer in the whole architecture.
2) RFAP (Random Forest for Accident Prediction) [38]: It is based on random forest which is composed of the set of the decision trees, each of which gives a local decision of the current status of the road. An aggregation function is used to merge the local decisions of the trees and find the final prediction results whether the current status of the road causes accident or not.

By varying the percentage of images used as input from 20% to 100%, Figure 4 shows the accuracy and the runtime of HR2CNN, the DSTGCN, and RFAP algorithms. The results reveal that the proposed solution outperforms the two baseline algorithms in terms of accuracy and it is very competitive in terms of processing runtime. The mAP of the HR2CNN reached 91% for handling 100% of the Kitti data, whereas the mAP for the other algorithms goes under 86% for dealing the same configuration. Regarding the runtime processing, the gap between the HR2CNN, and the other algorithms does

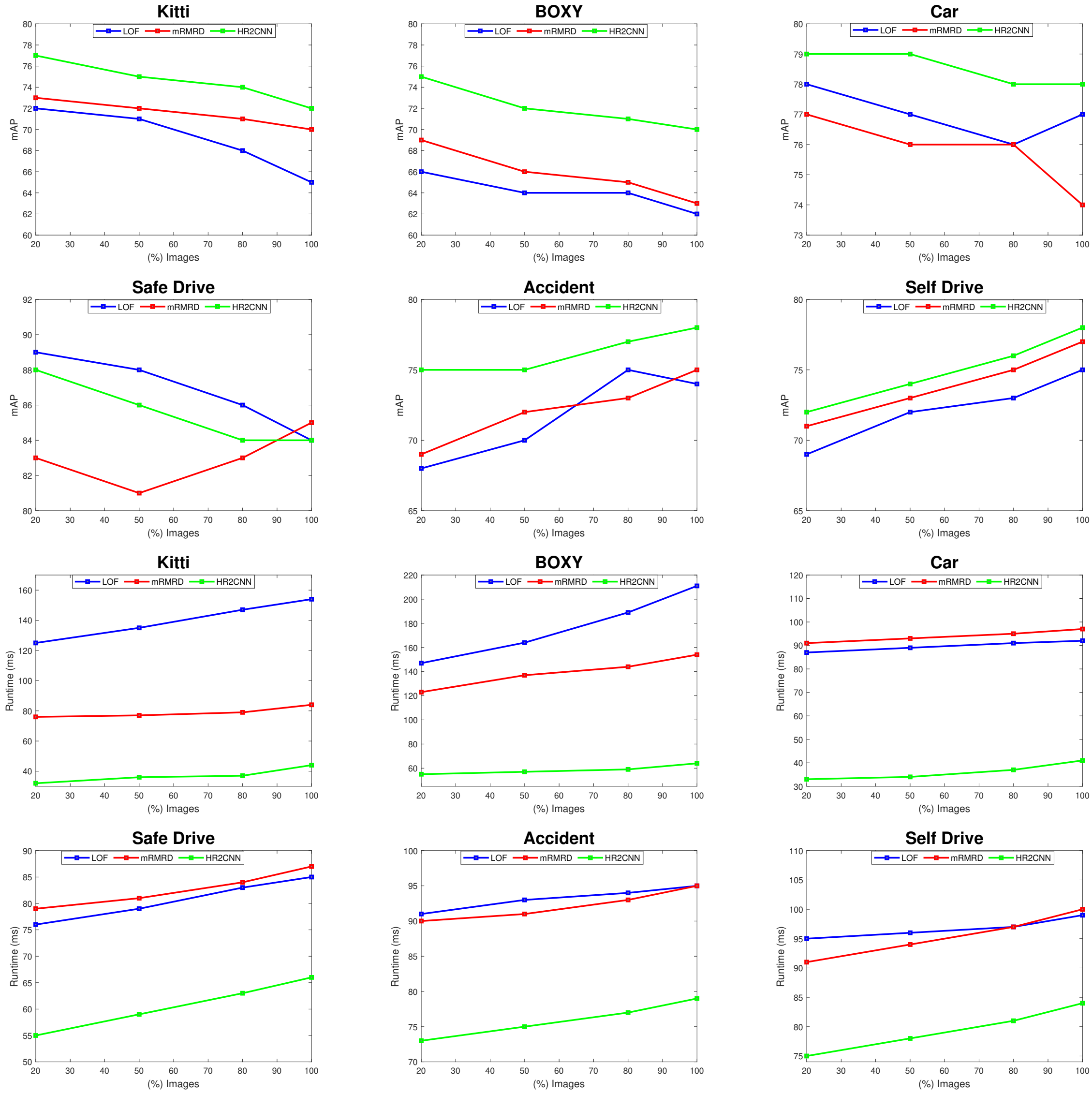

Fig. 2. Accuracy and Runtime of outlier detection step of the HR2CNN compared to the state-of-the-art outlier detection solutions.

not exceed 3 mile seconds for processing 100% of the Kitti data. The proposed framework needs more time compared to the two other algorithms because the HR2CNN is an end-to-end framework composed into three steps, outlier detection, vehicle detection and accident estimation processes. However, the two other algorithms are one pass based algorithms. This one pass reduces considerable the accuracy, where the HR2CNN removes the noises, find the vehicles closer to the given driven, and finally estimate the accidents from the detected cars.

V. Conclusion

This research work studied new deep learning framework for accident estimation by investigating the outlier detection, the object detection, and the classification in one pipeline. The outlier detection is first performed to remove noises from the set of original urban traffic data images. Hybrid SIFT extractor with Chinese restaurant process is developed to efficiently filter the noises. The extended region convolution neural network is then launched to detect the closer vehicles of the given driver. Different improvements have been suggested on the region convolution neural network model including, bounding box creation, hard negative exploration, and multi scale training. The binary classification benefits from the vehicle detection process, in order to distinguish between the road states that causes accident and the normal ones. To efficiently do such task, the RESNET algorithm is employed. Finally, and in order to better find the optimal parameter values of the proposed framework, an intelligent evolutionary computation algorithm is incorporated by developing diversification and intensification strategies for exploring the configuration space. Experimental evaluation is carried out to validate the applicability of the proposed framework using the well known vehicle detection, and accident estimation data. The results are very promising and show the proposed framework outperforms both the baseline

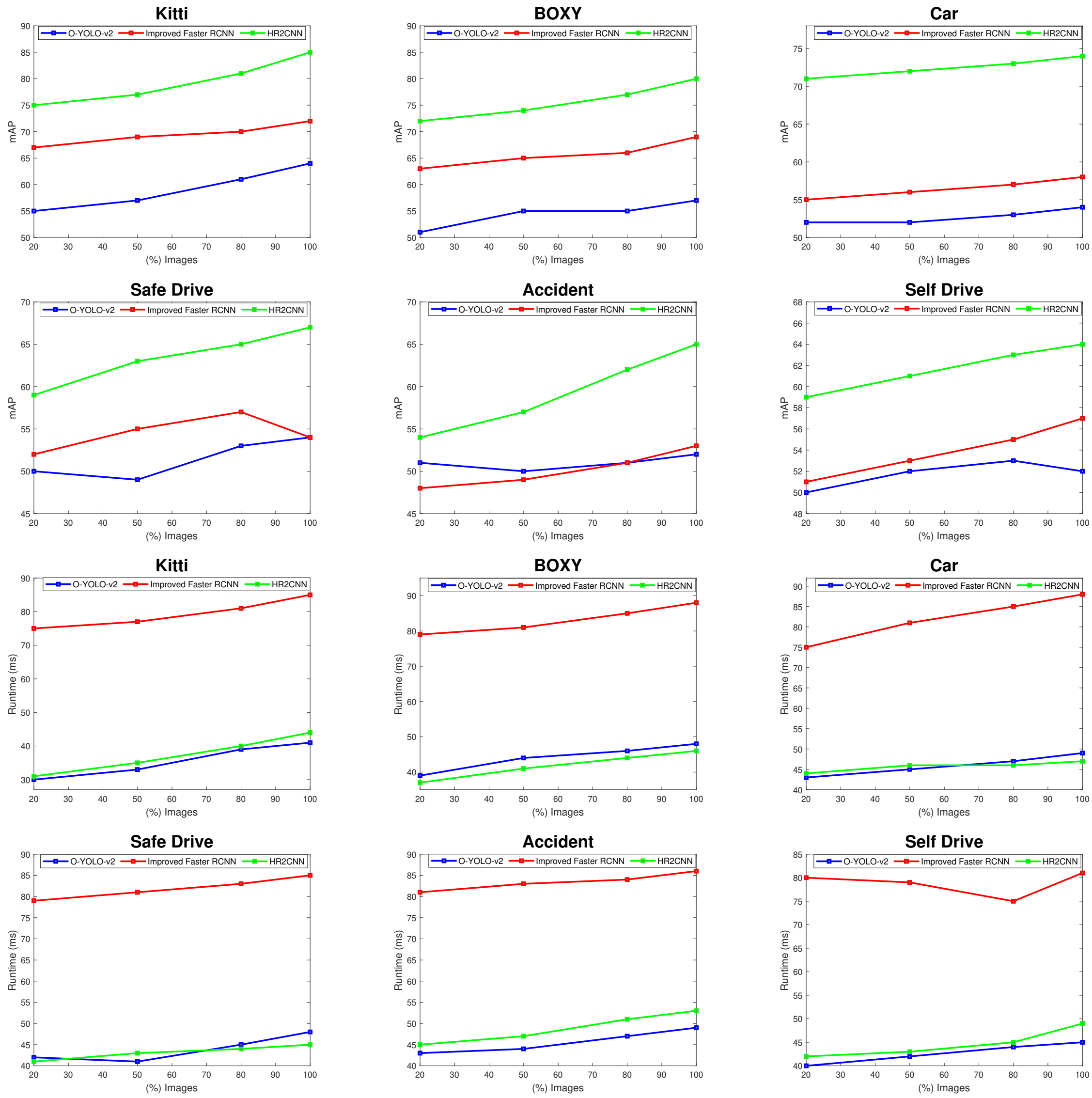

Fig. 3. Accuracy and Runtime of vehicle detection step of the HR2CNN compared to the state-of-the-art vehicle detection solutions.

vehicle detection, and the accident estimation solutions in terms of runtime, detection, and estimation accuracy. As future perspective, we plan to extend the proposed framework for dealing with large and big vehicle data in real time by exploring high performance computing tools. Handling 3D vehicle object data is also in our future agenda.

References

[1] L. Yu, B. Du, X. Hu, L. Sun, L. Han, and W. Lv, "Deep spatio-temporal graph convolutional network for traffic accident prediction," *Neurocomputing*, vol. 423, pp. 135–147, 2021.

[2] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Benchmark for evaluating pedestrian action prediction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1258–1268.

[3] B. Zhong, X. Pan, P. E. Love, J. Sun, and C. Tao, "Hazard analysis: A deep learning and text mining framework for accident prevention," *Advanced Engineering Informatics*, vol. 46, p. 101152, 2020.

[4] B. Zhong, X. Pan, P. E. Love, L. Ding, and W. Fang, "Deep learning and network analysis: Classifying and visualizing accident narratives in construction," *Automation in Construction*, vol. 113, p. 103089, 2020.

[5] J. Guerrero-Ibañez, J. Contreras-Castillo, and S. Zeadally, "Deep learning support for intelligent transportation systems," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 3, p. e4169, 2021.

[6] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2021.

[7] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, A. Cano, and J. C.-W. Lin, "A two-phase anomaly detection model for secure intelligent transportation ride-hailing trajectories," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[8] G. Srivastava, J. C.-W. Lin, A. Jolfaei, Y. Li, and Y. Djenouri, "Uncertain-driven analytics of sequence data in iocv environments," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[9] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Vehicle detection from uav imagery with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[10] J. Zhang, X. Jia, J. Hu, and K. Tan, "Moving vehicle detection for remote sensing video surveillance with nonstationary satellite platform," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 01, pp. 1–1, 2021.

[11] M. Mostofa, S. N. Ferdous, B. S. Riggan, and N. M. Nasrabadi, "Joint-srvdnet: Joint super resolution and vehicle detection network," *IEEE*

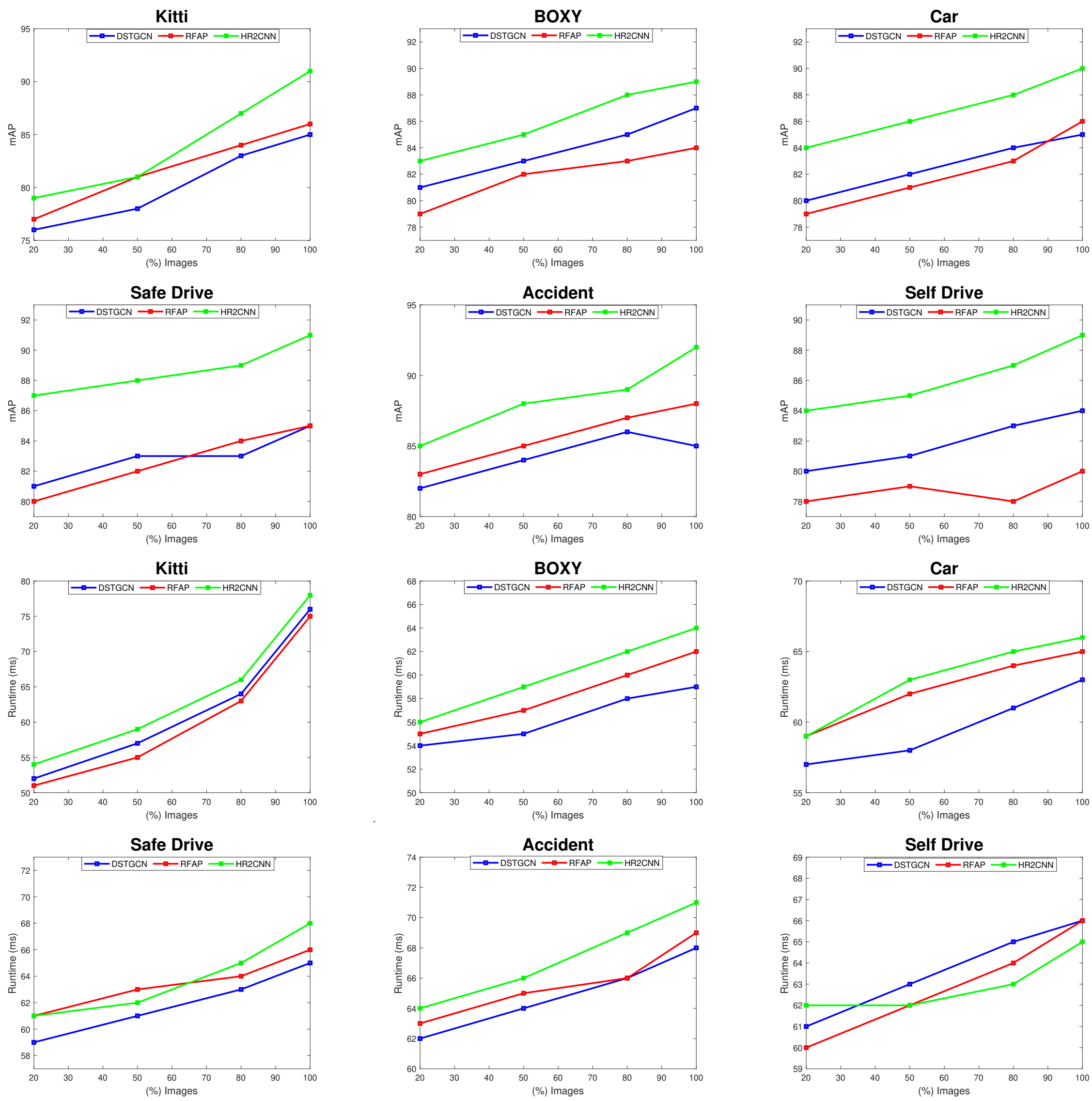

Fig. 4. Accuracy and Runtime of accident estimation step of the HR2CNN compared to the state-of-the-art accident estimation solutions.

Access, vol. 8, pp. 82 306–82 319, 2020.

[12] X. Wu, W. Li, D. Hong, J. Tian, R. Tao, and Q. Du, "Vehicle detection of multi-source remote sensing data using active fine-tuning network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 39–53, 2020.

[13] S. Arabi, A. Haghighat, and A. Sharma, "A deep-learning-based computer vision solution for construction vehicle detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 7, pp. 753–767, 2020.

[14] V.-T. Tran and W.-H. Tsai, "Acoustic-based emergency vehicle detection using convolutional neural networks," *IEEE Access*, vol. 8, pp. 75 702–75 713, 2020.

[15] A. Chetouane, S. Mabrouk, I. Jemili, and M. Mosbah, "Vision-based vehicle detection for road traffic congestion classification," *Concurrency and Computation: Practice and Experience*, p. e5983, 2020.

[16] L. Chen, Q. Ding, Q. Zou, Z. Chen, and L. Li, "Denselightnet: a light-weight vehicle detection network for autonomous driving," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10 600–10 609, 2020.

[17] W. Yang, Z. Li, C. Wang, and J. Li, "A multi-task faster r-cnn method for 3d vehicle detection based on a single image," *Applied Soft Computing*, vol. 95, p. 106533, 2020.

[18] M. Hassaballah, M. A. Kenk, K. Muhammad, and S. Minaee, "Vehicle detection and tracking in adverse weather using a deep learning framework," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[19] J. Fan, T. Huo, X. Li, T. Qu, B. Gao, and H. Chen, "Covered vehicle detection in autonomous driving based on faster rcnn," in *2020 39th Chinese Control Conference (CCC)*. IEEE, 2020, pp. 7020–7025.

[20] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[21] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Chen, and Y. Li, "Soft-weighted-average ensemble vehicle detection method based on single-stage and two-stage deep learning models," *IEEE Transactions on Intelligent Vehicles*, 2020.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[24] G. Chen, F. Wang, S. Qu, K. Chen, J. Yu, X. Liu, L. Xiong, and A. Knoll, "Pseudo-image and sparse points: Vehicle detection with 2d lidar revisited by deep learning-based methods," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[25] V. Q. Dinh, F. Munir, S. Azam, K.-C. Yow, and M. Jeon, "Transfer

learning for vehicle detection using two cameras with different focal lengths," *Information Sciences*, vol. 514, pp. 71–87, 2020.

[26] V. R. Kumar, S. Yogamani, H. Rashed, G. Sitsu, C. Witt, I. Leang, S. Milz, and P. Mäder, "Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2830–2837, 2021.

[27] H. Rashed, E. Mohamed, G. Sistu, V. R. Kumar, C. Eising, A. El-Sallab, and S. Yogamani, "Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2272–2280.

[28] G. Li, Y. Yang, X. Qu, D. Cao, and K. Li, "A deep learning based image enhancement approach for autonomous driving at night," *Knowledge-Based Systems*, vol. 213, p. 106617, 2021.

[29] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[30] S. Gupta, K. Thakur, and M. Kumar, "2d-human face recognition using sift and surf descriptors of face's feature regions," *The Visual Computer*, pp. 1–10, 2020.

[31] A. Pramanik, S. K. Pal, J. Maiti, and P. Mitra, "Granulated rcnn and multi-class deep sort for multi-object detection and tracking," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[32] S. Coniglio, F. Furini, and P. San Segundo, "A new combinatorial branch-and-bound algorithm for the knapsack problem with conflicts," *European Journal of Operational Research*, vol. 289, no. 2, pp. 435–455, 2021.

[33] Z. Bu and R. E. Korf, "A*+ bfhs: A hybrid heuristic search algorithm," *arXiv preprint arXiv:2103.12701*, 2021.

[34] K. Behrendt, "Boxy vehicle detection in large images," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[35] A. Tomlinson, J. Bryans, and S. A. Shaikh, "Using internal context to detect automotive controller area network attacks," *Computers & Electrical Engineering*, vol. 91, p. 107048, 2021.

[36] M. Riahi-Madvar, A. A. Azirani, B. Nasersharif, and B. Raahemi, "A new density-based subspace selection method using mutual information for high dimensional outlier detection," *Knowledge-Based Systems*, vol. 216, p. 106733, 2021.

[37] X. Han, J. Chang, and K. Wang, "Real-time object detection based on yolo-v2 for tiny vehicle object," *Procedia Computer Science*, vol. 183, pp. 61–72, 2021.

[38] X. Zhou, P. Lu, Z. Zheng, D. Tolliver, and A. Keramati, "Accident prediction accuracy assessment for highway-rail grade crossings using random forest algorithm compared with decision tree," *Reliability Engineering & System Safety*, vol. 200, p. 106931, 2020.