2020 IEEE Intelligent Vehicles Symposium (IV)
October 20-23, 2020. Las Vegas, USA

Attention R-CNN for Accident Detection

Trung-Nghia Le [1], Shintaro Ono[2], Akihiro Sugimoto[1], and Hiroshi Kawasaki[3]

[1]National Institute of Informatics, Japan
[2]University of Tokyo, Japan
[3]Kyushu University, Japan

***Abstract*— This paper addresses accident detection where we not only detect objects with classes, but also recognize their characteristic properties. More specifically, we aim at simultaneously detecting object class bounding boxes on roads and recognizing their status such as safe, dangerous, or crashed. To achieve this goal, we construct a new dataset and propose a baseline method for benchmarking the task of accident detection. We design an accident detection network, called Attention R-CNN, which consists of two streams: one is for object detection with classes and one for characteristic property computation. As an attention mechanism capturing contextual information in the scene, we integrate global contexts exploited from the scene into the stream for object detection. This introduced attention mechanism enables us to recognize object characteristic properties. Extensive experiments on the newly constructed dataset demonstrate the effectiveness of our proposed network. The dataset and source code are publicly available on our project page.** [1]

I. Introduction

Development and production for automated driving system (ADS) have been recently growing at a rapid pace. Indeed, significant advances for ADS have been achieved during the last decade [6]. Despite such great advances, how to drive a car safely still remains a challenge as a high number of collisions caused by autonomous cars is reported [31]. This suggests the requirement of developing ADS capable of appropriately reacting to any situations in huge diversity that happen on real-world roadways. To meet the requirement, traffic scene understanding such as pedestrian/road object detection [24], [26], semantic segmentation [7], [14], or scene flow estimation [15], [27] has been developed as the key in ADS to achieve safe driving and to reduce road accidents.

Object detection aims to locate a single or multiple object(s) in an image or a video by drawing bounding boxes with their classes. With the era of deep learning, convolutional neural network (CNN) based object detectors [18], [19] have achieved remarkable progress in recent years. Existing work has applied deep learning-based object detectors to detect road objects in videos captured using surveillance cameras [24], [37] or car-mounted cameras [14], [26]. Existing work is, however, limited to detect only appearance of objects with classes such as pedestrian, car, or bus; the characteristic properties of objects such as old/new or safe/damaged are not recognized. In order to enrich the ability of reaction of ADS against various situations, such characteristic properties of objects should be also addressed. In this sense, analyzing road objects still has an important missing step.

This paper advances road object analysis further by targeting not only object detection with classes but also recognition of each object characteristic property. We call this task *accident detection*. We propose a two-stream network, called Attention R-CNN, for accident detection. Leveraging advantages of cutting-edge deep model [18], the proposed Attention R-CNN consists of the stream for detecting object appearances with classes (appearance stream) and the stream for computing object characteristic properties (characteristic stream). In the characteristic stream, we aim to exploit both global and local contexts of objects. More precisely, we extract the global context of an object using features computed from entire of the scene, and the local context of an object using features computed from inside the bounding box corresponding to the object. We introduce an attention mechanism that integrates global contexts of the scene into local features of objects. This attention mechanism allows us to effectively recognize the object characteristic properties. We also provide a new dataset for accident detection to promote this new task. Extensive experiments on the newly constructed dataset demonstrate the effectiveness of our proposed Attention R-CNN.

In summary, the main contributions of this work are highlighted as follows:

- We address the new task of accident detection, in which not only object appearance bounding boxes with classes are detected, but their characteristic properties are also recognized.
- We present Attention R-CNN for accident detection that consists of the appearance stream for detecting object bounding boxes with classes and the characteristic stream for recognizing object characteristic property. The characteristic stream possesses the attention mechanism that exploits local and global contexts in the scene.
- We provide a 100-video dataset for accident detection (each video lasts 3.3 seconds). Our newly constructed dataset has the ground-truth manually annotated to every frame in a video. Each frame consists of the ground-truth of object appearance bounding boxes with classes and object characteristic properties.

Corresponding author. Email: ltnghia@nii.ac.jp
This work was done when Trung-Nghia Le was with University of Tokyo

[1] https://sites.google.com/view/ltnghia/research/accident-detection

978-1-7281-6673-5/20/$31.00 ©2020 IEEE

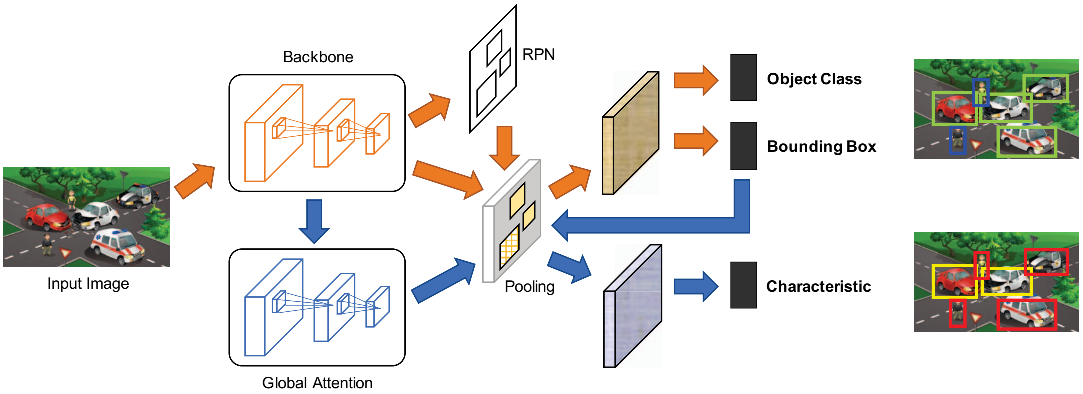


Fig. 1: The overview of our proposed network. Attention R-CNN consists of two streams, namely, the appearance stream (orange flow) for object class detection and the characteristic stream (blue flow) for object characteristic computation.

II. Related Work

A. Accident Categorization and Anticipation

Accident categorization, which is similar to image/video classification, is to compute the probability that a collision could happen in a(n) image/video where the collision is divided into various categories such as low risk/safe and high risk/accident. Kim et al. [13] trained a classifier from virtual and real video scenes to identify dangerous vehicles from a regional image. Taccari et al. [33] fed deep features to Random Forest to classify crash and near-crash events from car-mounted videos and telematics data. Wang et al. [35] categorized accident videos into three levels of collision risk using two-stream CNN. Corcoran et al. [2] classified each frame in a given input video sequence into four categories from low to critical risks, using a two-stream recurrent CNN.

Accident anticipation, on the other hand, is to predict a collision could happen in future video frames. Chan et al. [1] proposed a dynamic attention network based on Recurrent Neural Network (RNN) for anticipating accidents in car-mounted videos before they occur. To anticipate traffic accidents, Suzuki et al. [32] built a new loss function for RNN to gradually learn earlier anticipation. Yao et al. [38] used an unsupervised deep learning framework to predict traffic participant trajectories and locations for anomaly anticipation in egocentric videos.

There are usually many vehicles and persons on the road. However, existing work outputs only the accident event of the whole image/video without identifying which road objects collided. Meanwhile, our method can point out the location of road objects involved in an accident event.

B. Object Detection

Generally, there are two types of detectors depending on whether or not a region proposal network (RPN) is used: one-stage (proposal-free) and two-stage (proposal-based). One-stage detectors (*i.e.* YOLO [28], [29], SSD [21], FCOS [34], RetinaNet [19]) are, in general, faster in training and inference, but with lower detection accuracy, compared to flexible two-stage detectors (*i.e.* R-CNN [10], Fast R-CNN [9], Faster R-CNN [30], R-FCN [5]). We employ two-stage detectors in this paper accordingly.

Our proposed network is built on top of Faster R-CNN [30], a state-of-the-art two-stage detector, by adding a characteristic stream having an attention mechanism to support for accident detection.

III. Proposed Method

A. Attention R-CNN Overview

Figure 1 depicts the overview of our proposed Attention R-CNN. Attention R-CNN consists of two streams for two different tasks: the appearance stream and the characteristic stream. The appearance stream detects object appearance bounding boxes with their classes. Meanwhile, the characteristic stream utilizes the results from the appearance stream to recognize characteristic properties for all detected objects.

B. Appearance Stream

Faster R-CNN [30] is a state-of-the-art method presenting for two-stage detectors. Recent detectors [18], [25] usually follow the architecture of Faster R-CNN. However, the original Faster R-CNN is not effective in detecting road objects due to imbalance category and high density. Therefore, to efficiently detect objects on roads, we employ the appearance stream by adopting Faster R-CNN with modification. Particularly, we improve multi-scale features extracted from the backbone and RoI features exploited from the head network. We also solve imbalance of object categories in the training process through balance loss functions.

Leveraging advantages of cutting-edge deep models, we use ResNet-50 [11] followed by Feature Pyramid Network (FPN) [18] as the backbone for the entire network. Following [25], we balance multi-level features by combining all features in the FPN with the average feature: $\widetilde{F_i} = F_i + F_{ave}$, where $F_{ave} = \frac{1}{N}\sum_{i=1}^{N} F_i$ denotes the average feature. To integrate multi-level features and preserve their semantic hierarchy at the same time, we first resize the multi-level features to an intermediate size. The obtained features are then rescaled using the same but reverse procedure to strengthen the original features.

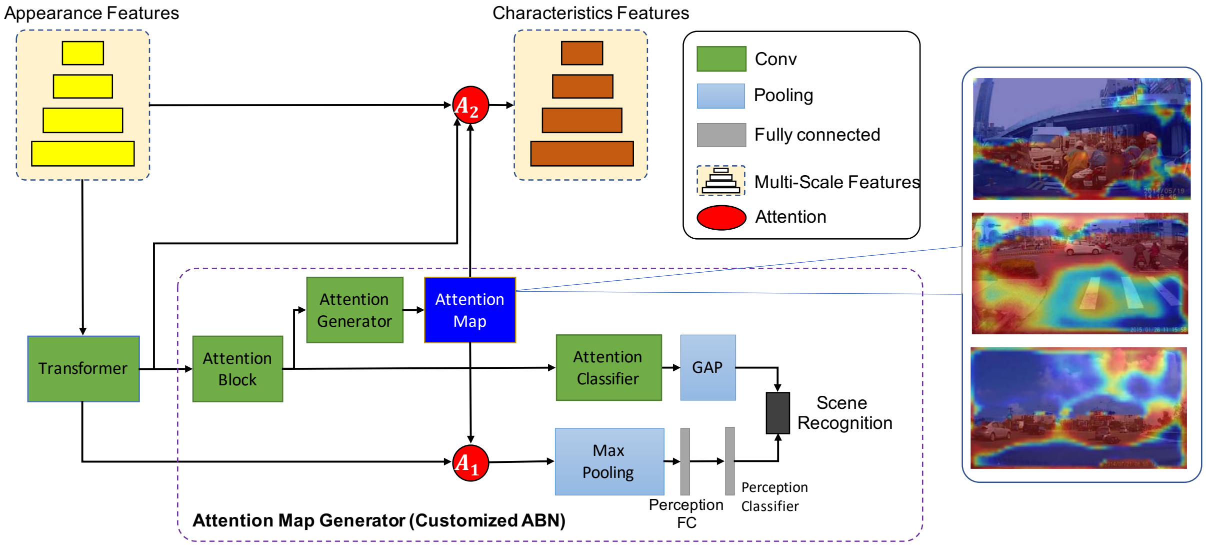


Fig. 2: Architecture of Global Attention and examples of generated attention maps.

Following the standard design in two-stage detectors, we first detect the possible positions containing objects via Region Proposal Network (RPN) [30]. RPN shares weights with the main backbone and outputs bounding boxes (RoI/object proposal) at various sizes. For each RoI, a fixed-size feature map (i.e., 7 × 7) is pooled from the image feature map using the RoIPool layer [30]. We here replace the original RoIPool [30] layer with a Precise RoI Pooling layer (PrRoI) [12] because of its outstanding effect. The RoIPool works by dividing the RoI into a regular grid and then max-pooling the feature map values in each grid cell. This quantization, however, causes misalignment between the RoI and the extracted features due to the harsh rounding operations when mapping the RoI coordinates from the input image space to the image feature map space and when dividing the RoI into grid cells. On the other hand, PrRoI [12] does not have the problem of misalignment. Indeed, PrRoI uses average pooling instead of max pooling for each bin and has a continuous gradient on bounding box coordinates. That is, one can take the derivatives of some loss function with respect to the coordinates of each RoI and optimize the RoI coordinates.

After that, we extract features inside these proposals to compute object positions and object classes. Inspired by the head network architecture proposed by Lin et al. [18], RoI features are fed into a stack of four 3 × 3 × 256 convolution (conv) layer. Each conv layer is followed by a Group Normalization (GN) layer [36] and a ReLU layer .

Road object categories are obviously heavy imbalance. To address this problem, we employ balance loss functions. The balance loss of the appearance stream $\mathcal{L}_{\text{baApp}}$ is computed as follows:

$$\mathcal{L}_{\text{baApp}} = \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{objLoc}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}}, \tag{1}$$

where $\mathcal{L}_{\text{obj}}$ and $\mathcal{L}_{\text{objLoc}}$ are the output of the RPN, $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{loc}}$ are defined on the output of the head network.

The objectness loss is the Focal Loss [19] computed as follows:

$$\mathcal{L}_{\text{obj}}(p, u) = -\alpha(1 - p_u)^{\gamma} \log p_u, \tag{2}$$

where p_u is the probability output for the true class u. Parameters $\alpha = 0.25, \gamma = 2$ are computed similarly to [19]. The object classification loss is the Class-Balance Softmax Cross-Entropy Loss [4] computed as follows:

$$\mathcal{L}_{\text{cls}}(p, u) = -\frac{1 - \beta}{1 - \beta^{n_u}} \log p_u, \tag{3}$$

where p_u is the softmax output for the true class u, and n_u is the number of samples in the ground-truth class u. Parameter $\beta = 0.999$ is computed similarly to [4]. The bounding box regression losses $\mathcal{L}_{\text{objLoc}}(t^u, v)$ and $\mathcal{L}_{\text{loc}}(t^u, v)$ is computed as the Balance L1 Loss [25] between the regressed box offset t^u (corresponding to the ground-truth object class u) and the ground-truth box offset v:

$$\mathcal{L}_{\text{BL1}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} Balance_{\text{L}_1}(t_i^u - v_i), \tag{4}$$

where

$$Balance_{\text{L}_1}(x) = \begin{cases} \frac{\alpha}{b}(b|x| + 1)\ln(b|x| + 1) & \text{if } |x| < 1 \\ \gamma|x| + C & \text{otherwise,} \end{cases} \tag{5}$$

where $\alpha \ln(b + 1) = \gamma$. Parameters $\alpha = 0.5, \gamma = 1.5$ are computed similarly to [25].

C. Characteristic Stream

The characteristic stream consists of two modules: Global Attention and head network as shown in Fig. 1. The Global Attention converts features extracted from the backbone of the appearance stream to characteristic features, which is then useful for object characteristic property computation. The head network outputs the probability of characteristic categories for all the objects detected by the appearance stream.

1) Global Attention: Features extracted from the appearance stream are not effective in object characteristic property computation. This is because appearance features do not necessarily reflect implicit properties of objects (even though these features can recognize car/motor, they may not be able to distinguish safety/damage). To address this problem, we build on top of the backbone the Global Attention module to convert appearance features to effective characteristic features. The Global Attention involves three components: transformer, attention map generator, and attention mechanism.

Transformer: It converts appearance features to characteristic features. We first extract an average feature from multi-level features of the backbone. We note that multi-level features are resized to the same size before the extraction. The average feature is then fed into a stack of four $3 \times 3 \times 256$ conv layers. Each conv layer is followed by a GN layer [36] and a ReLU .

Attention map generator: This is developed by employing the Attention Branch Network (ABN) [8] with modifications. The customized ABN contains an attention branch and a perception branch. The attention branch consists of a block of five conv layers. The first four conv layers has $3 \times 3 \times 256$ kernel. Each conv layer is followed by a GN layer [36] and a ReLU . The last conv layer is $1 \times 1 \times K$, where K is the number of classes. After that, we use a $1 \times 1 \times K$ conv layer for pixel-wise classification, followed by a Global Average Pooling (GAP) layer [17] to generate a probability vector of classes. In order to aggregate K feature maps, a $1 \times 1 \times 1$ conv layer is plugged-in right after the conv block as the attention generator. Fig. 2 illustrates examples of the generated attention map. On the other hand, the perception branch outputs the probability of each class, following the standard design of classification models. Similarly to [8], the generated attention map is applied to the transformed feature by an attention mechanism to enhance feature maps: $G = (1 + M) \cdot F_{tf}$, where M is the attention map, F_{tf} is the transformed feature, and G is the new feature. A Max Pooling layer is used to obtain a fixed-size 64×64 feature. The feature is then fed into two fully connected layers, yielding feature vectors with 1024 and K channels, respectively.

The loss of Global Attention is also the loss of ABN, computed as follows:

$$\mathcal{L}_{\text{gbAtt}} = \mathcal{L}_{\text{att}} + \mathcal{L}_{\text{per}}, \tag{6}$$

where $\mathcal{L}_{\text{att}}$ denotes the training loss at the attention branch, and $\mathcal{L}_{\text{per}}$ denotes the training loss at the perception branch. All the two losses are the Class-Balance Softmax Cross-Entropy Loss [4] computed as follows:

$$\mathcal{L}_{\text{CBCE}}(p, u) = -\frac{1-\beta}{1-\beta^{n_u}} \log p_u, \tag{7}$$

where p_u is the softmax output for the true class u, and n_u is the number of samples in the ground-truth class u. Parameter $\beta = 0.999$ is computed similarly to [4].

Attention mechanism: It aims to output characteristic features from multi-level appearance features F_i by combining them with the transformed feature F_{tf} and the attention map M (see block Attention A2 in Fig. 2) as follows:

$$\tilde{F_i} = (1 + M) \cdot (F_{tf} + F_i), \tag{8}$$

where $\tilde{F_i}$ is multi-level characteristic features, corresponding to appearance features F_i. In practice, we directly apply the formula below to utilize the pre-computed feature g:

$$\tilde{F_i} = (1 + M) \cdot F_i + G. \tag{9}$$

2) Head Network: To compute the characteristic properties of detected objects, for each target, we employ the head network to exploit RoI features. For each target RoI, RoI feature is exploited directly from the region inside the target RoI through a PrRoI layer [12]. Similarly to the appearance stream, the RoI features $\tilde{F}$ are first fed into a stack of four $3 \times 3 \times 256$ conv layers, in which each of them is followed by a GN layer [36] and a ReLU . They are then fed into two fully connected layers, yielding feature vectors with 1024 and K channels, respectively.

The loss of characteristic classifier is the Class-Balance Softmax Cross-Entropy Loss [4] computed as follows:

$$\mathcal{L}_{\text{chaCls}}(p, u) = -\frac{1-\beta}{1-\beta^{n_u}} \log p_u, \tag{10}$$

where p_u is the softmax output for the true class u, and n_u is the number of samples in the ground-truth class u. Parameter $\beta = 0.999$ is computed similarly to [4].

The balance loss of the characteristic stream $\mathcal{L}_{\text{baCha}}$ is computed as follows:

$$\mathcal{L}_{\text{baCha}} = \mathcal{L}_{\text{gbAtt}} + \mathcal{L}_{\text{chaCls}}. \tag{11}$$

IV. Accident Detection Video Dataset

To promote researches on accident detection, a publicly available dataset is mandatory. We thus construct an **A**ccident **D**etection **V**ideo (ADV) dataset. We emphasize that no other dataset is publicly available for accident detection. To build the new dataset, we selected 100 raw videos recorded by Chan *et al.* [1], each of which involves at least a dangerous/crashed event, and annotated ground-truth for accident detection. We initially annotated a bounding box with its object label for all road objects at every video frame where we used the semi-supervised annotation process [16]. We used 7 popular semantic labels to annotate road object classes. They are *car, bus, truck, motorbike, bicycle, pedestrian*, and *rider*. After that, we manually labeled three accident categories for all objects: *safe, dangerous,* and *crashed*.

Our newly constructed ADV dataset is the first dataset designed explicitly for the task of accident detection. The dataset consists of 100 videos, each of which has 100 frames, with total 88,800 object bounding box ground-truth. Some examples are shown in Fig. 3 with the corresponding ground-truth label annotations.

The main difference of our ADV dataset from existing road object datasets [3], [23], [39] having a dominant number

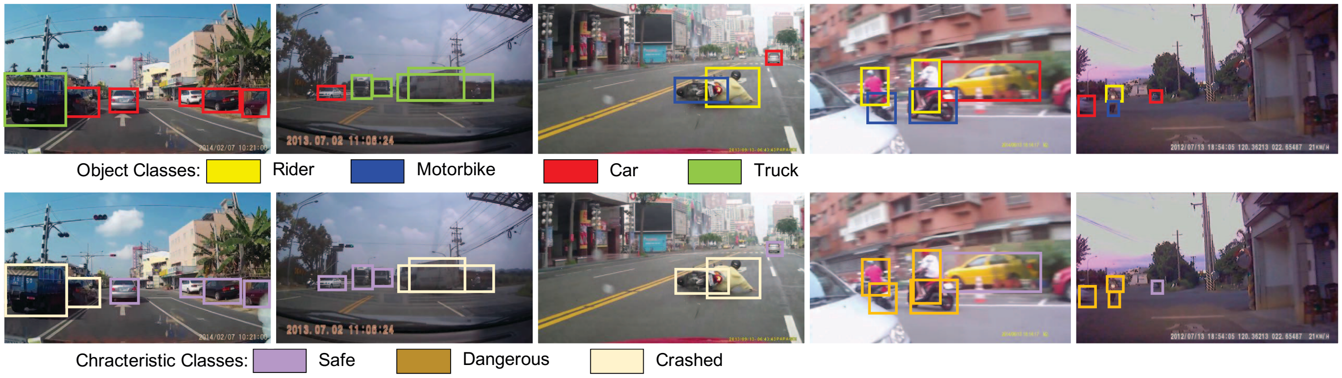
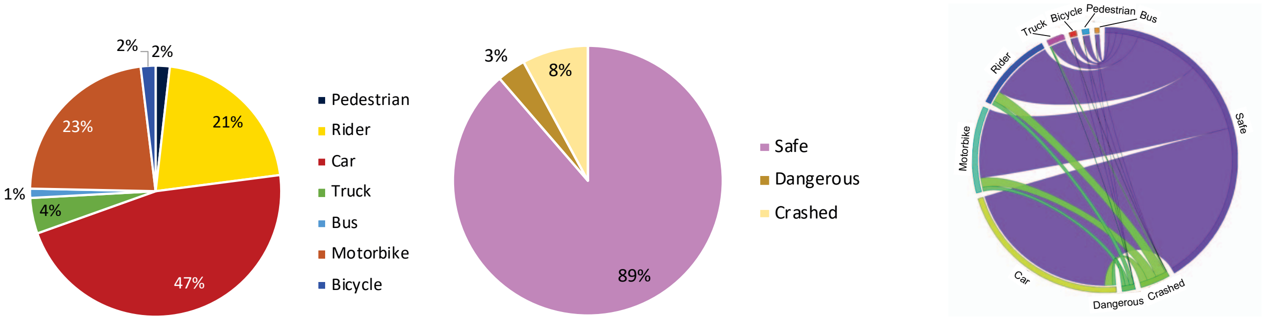


Fig. 3: Examples of Accident Detection Video (ADV) dataset. The first row is object class ground-truth and the second row is accident ground-truth. We do not highlight tiny objects and unimportant objects (blurred, heavily occluded).



(a) Object categories.

(b) Accident categories.

(c) Object-Accident distribution.

Fig. 4: Categories distribution over the ADV dataset.

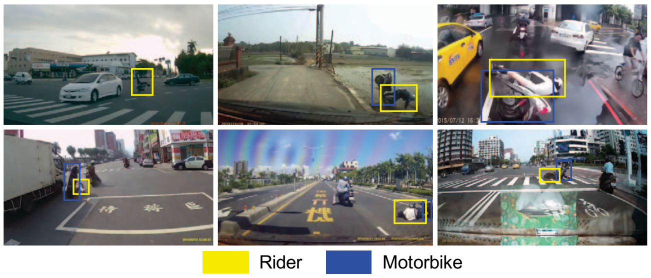


Fig. 5: Strange poses of persons and vehicles in the ADV dataset when accidents happen. We highlight only objects with strange poses.

TABLE I: Number of objects at each image over different datasets (%). Risky objects indicate dangerous and crashed vehicles and persons.

Dataset / Class	Persons	Two-Wheel Vehicles	Four-Wheel Vehicles	Risky Objects
Cityscapes [3]	6.82	1.65	9.43	0
MVD [23]	3.45	0.70	8.35	0
BDD [39]	1.38	0.15	10.88	0
ADV	2.08	2.23	4.75	1.03

of four-wheel vehicles is that our ADV dataset consists of a large number of persons and two-wheel vehicles. This reflects the specialty of many Asian countries (cf. Table I). The ratios of each category are shown in Fig. 4. We remark that we count only images having object bounding box ground-truth because existing datasets do not have ground-truth for all images. In the ADV dataset, it is noteworthy that when accident events happen, strange poses of persons and vehicles appear (cf. Fig. 5), which never existed in other datasets [3], [23], [39]. This makes the detection more challenging than existing datasets. The ADV dataset also contains videos in different weather (sunny, rainy, snowy) and timeline (daytime and night) to increase the difficulty of the detection.

V. Experiments

A. Benchmark Dataset and Evaluation Criteria

Our constructed ADV dataset is randomly divided in to the *trainval* set and the *test* set with ratios 80% and 20%.

All reported results follow standard COCO-style Average Precision (AP) metrics that include AP (averaged over IoU thresholds), AP50 (AP for IoU threshold 50%), AP75 (AP for IoU threshold 75%) [20]. In addition, mean Average Precision (mAP) denotes the AP over all the categories.

B. Implementation Details

For fair comparisons, all experiments are implemented on PyTorch, and based on the published code of Mask R-CNN

TABLE II: Experimental results on ADV dataset. The best results are shown in **blue**. Methods are evaluated by mAP, which can justify both bounding box and recognition of each detection.

Method	Improved	Global	Object Class			Object Characteristic			Total		
	Architecture	Attention	mAP	mAP50	mAP75	mAP	mAP50	mAP75	mAP	mAP50	mAP75
Faster R-CNN*			16.4	35.9	13.1	10.0	19.7	9.0	13.2	27.8	11.1
Appearance Stream*	✓		34.2	57.7	34.8	17.3	28.3	17.4	25.8	43.0	26.1
Attention R-CNN	✓	✓	**34.2**	**57.7**	**34.8**	**18.9**	**31.1**	**18.9**	**26.6**	**44.4**	**26.9**

TABLE III: Detection results of Attention R-CNN in each category (mAP50).

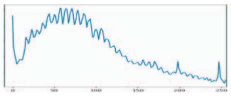
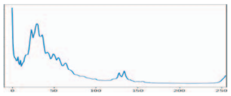
Detection Results of Object Categories

Car	Bus	Truck	Motorbike	Bicycle	Pedestrian	Rider
82.1	1.9	70.1	80.4	56.4	34.2	78.5

Detection Results of Characteristic Categories

Safe	Dangerous	Crashed
72.8	2.5	18.1

TABLE IV: Detection results of Attention R-CNN at different conditions.

Environment	Day	Night
Color Intensity Distribution		
mAP50	45.0	38.6

Benchmark [22] .

We trained detectors with two GTX 1080Ti GPUs (4 images per GPU) for 20 epochs. We used the Stochastic Gradient Descent (SGD) optimization with a moment $\beta = 0.9$ and a weight decay of 0.0001. The training process was conducted by initially fine-tuning the available pre-trained model from the MS-COCO dataset [20] on the BDD dataset [39] to transfer domain from general objects to road objects. After that, we fine-tuned models on the newly constructed ADV dataset.

It is not easy to directly train the whole network in an end-to-end manner; different learning rates are employed for different streams. Here, we employ a two-stage optimization approach to solve this problem. We first train the appearance stream with an initial learning rate of 0.02, using the loss $\mathcal{L}_{\text{baApp}}$. We then train the characteristic stream with an initial learning rate of 0.002, using the loss $\mathcal{L}_{\text{baCha}}$. The learning rate decrease by 0.1 after 8, 11, and 16 epochs, respectively.

C. Experimental Results

We emphasize that Attention R-CNN is the first work for the task of accident detection, meaning that no state-of-the-art method is available for comparison. In this section, we thus compare our proposed Attention R-CNN with baselines. We investigate the impact of different components in our proposed Attention R-CNN, such as Global Attention and improved network architecture of the appearance stream, including balance losses. Experimental results are shown in Table II.

Figure 6 shows the visual comparison of different methods. As illustrated in the figure, our Attention R-CNN yield better results than Faster R-CNN. Our results are close to the ground truth and focus on risky objects (e.g., dangerous and crashed vehicles and persons). As shown in Table II, our proposed Attention R-CNN significantly outperforms all baselines.

Faster R-CNN*: Faster R-CNN [18] is designed only for object class and bounding box detection. To additionally compute object characteristics, we attach more two fully connected layers, followed by a classifier after the RoI pooling layer for object characteristic recognition, denoted by *Faster R-CNN**. Table II shows that our Attention R-CNN significantly outperforms Faster R-CNN*.

Global Attention: We investigate the performance of the Global Attention by comparing our completed Attention R-CNN with the one without Global Attention, denoted by *Appearance Stream**. This baseline is implemented similarly to our appearance stream with an additional head network for object characteristic detection. As shown in Table II, our completed Attention R-CNN surpasses Appearance Stream*, highlighting the contribution of our Global Attention.

Improved network architecture: We suspect that one of the major factors affecting the accident detection was the network architecture (including the backbone, the head network, and loss functions). As also seen in Table II, all networks with the improved network architecture (i.e., Appearance Stream* and Attention R-CNN) significantly outperforms Faster R-CNN* based on the old designed architecture. This clearly shows the importance of the new network architecture and balance losses in the proposed method.

D. Analysis and Discussion

We show results of Attention R-CNN in different conditional environments in Fig. IV. Our proposed method can achieve good performance in both daytime and nighttime, which have differences in light intensity distribution.

Results of Attention R-CNN in each category are shown in Table III. mAP scores of *Bus* and *Dangerous* are too low, compared with other categories. Due to the imbalance of the ADV dataset (cf. Fig. 4), bus takes only 1% of objects in the ADV dataset, leading to miss-detect to car or truck.

In dangerous events, almost vehicles change only their behaviors but not appearance. Meanwhile, the proposed Attention R-CNN is frame-by-frame processing, which lacks temporal information to detect behavior changes. In the future, we will investigate effectiveness of temporal information to improve the performance of the system.

VI. Conclusion

We addressed a new task of accident detection, and provided a dataset for this task. We gave the baseline for the task on the provided dataset. We believe that our dataset will promote new advancements in accident detection.

Our proposed Attention R-CNN network for accident detection consists of two streams: object detection stream and object characteristic stream. The object characteristic stream employs the attention mechanism that exploits local and global contextual-levels of a detected object using not only its corresponding region but also entire of the scene to recognize the object characteristic property. This leads to significant improvement in both object class detection and object characteristic detection, establishing a baseline on our provided dataset.

Besides extending the quantity of the dataset, developing a way to exploit temporal information from videos is left for future work.

Acknowledgements

This research was, in part, supported by Committee on Advanced Road Technology (CART) 2019 of Ministry of Land, Infrastructure, Transport and Tourism (MLIT) of Japan and JSPS/KAKENHI 18H04119, 18K19824 of Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan.

References

[1] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun. Anticipating accidents in dashcam videos. In *ACCV*, pages 136–153, 2016.

[2] G.-P. Corcoran and J. Clark. Traffic risk assessment: A two-stream approach using dynamic-attention. In *Conference on Computer and Robot Vision*, pages 166–173, 2019.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[4] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

[5] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.

[6] A. Davies. The wired guide to self-driving cars. https://www.wired.com/story/guide-self-driving-cars/, 2018.

[7] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *CVPR*, June 2019.

[8] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *CVPR*, pages 10705–10714, 2019.

[9] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, June 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, June 2016.

[12] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, pages 784–799, 2018.

[13] H. Kim, K. Lee, G. Hwang, and C. Suh. Crash to not crash: Learn to identify dangerous vehicles using a simulator. In *Association for the Advancement of Artificial Intelligence*, 2019.

[14] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar. Panoptic segmentation. In *CVPR*, June 2019.

[15] H.-Y. Lai, Y.-H. Tsai, and W.-C. Chiu. Bridging stereo matching and optical flow via spatiotemporal correspondence. In *CVPR*, June 2019.

[16] T.-N. Le, A. Sugimoto, S. Ono, and H. Kawasaki. Toward interactive self-annotation for video object bounding box: Recurrent self-learning and hierarchical annotation based framework. In *IEEE Winter Conference on Applications of Computer Vision*, 2020.

[17] M. Lin, Q. Chen, and S. Yan. Network in network. *International Conference on Learning Representations*, 2014.

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[22] F. Massa and R. Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018.

[23] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.

[24] K.-T. Nguyen, T.-H. Hoang, M.-T. Tran, T.-N. Le, N.-M. Bui, T.-L. Do, V.-K. Vo-Ho, Q.-A. Luong, M.-K. Tran, T.-A. Nguyen, T.-D. Truong, V.-T. Nguyen, and M. N. Do. Vehicle re-identification with learned representation and spatial verification and abnormality detection with multi-adaptive vehicle detectors for traffic video analysis. In *CVPR Workshops*, June 2019.

[25] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019.

[26] Y. Pang, J. Xie, M. H. Khan, R. M. Anwer, F. S. Khan, and L. Shao. Mask-guided attention network for occluded pedestrian detection. In *ICCV*, October 2019.

[27] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, June 2019.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[29] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017.

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[31] J. Stewart. Global autonomous driving market outlook. https://www.wired.com/story/self-driving-car-crashes-rear-endings-why-charts-statistics/, 2018.

[32] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident db. In *CVPR*, pages 3521–3529, 2018.

[33] 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 *International Conference on Intelligent Transportation Systems*, pages 2460–2465, Nov 2018.

[34] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, October 2019.

[35] Y. Wang and J. Kato. Collision risk rating of traffic scene from dashboard cameras. In *International Conference on Digital Image Computing: Techniques and Applications*, pages 1–6, Nov 2017.

[36] Y. Wu and K. He. Group normalization. In *ECCV*, pages 3–19, 2018.

[37] D. Xu, W. Xie, and A. Zisserman. Geometry-aware video object detection for static cameras. In *British Machine Vision Conference*, 2019.

[38] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins. Unsupervised traffic accident detection in first-person videos. In *International Conference on Intelligent Robots and Systems*, 2019.

[39] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018.

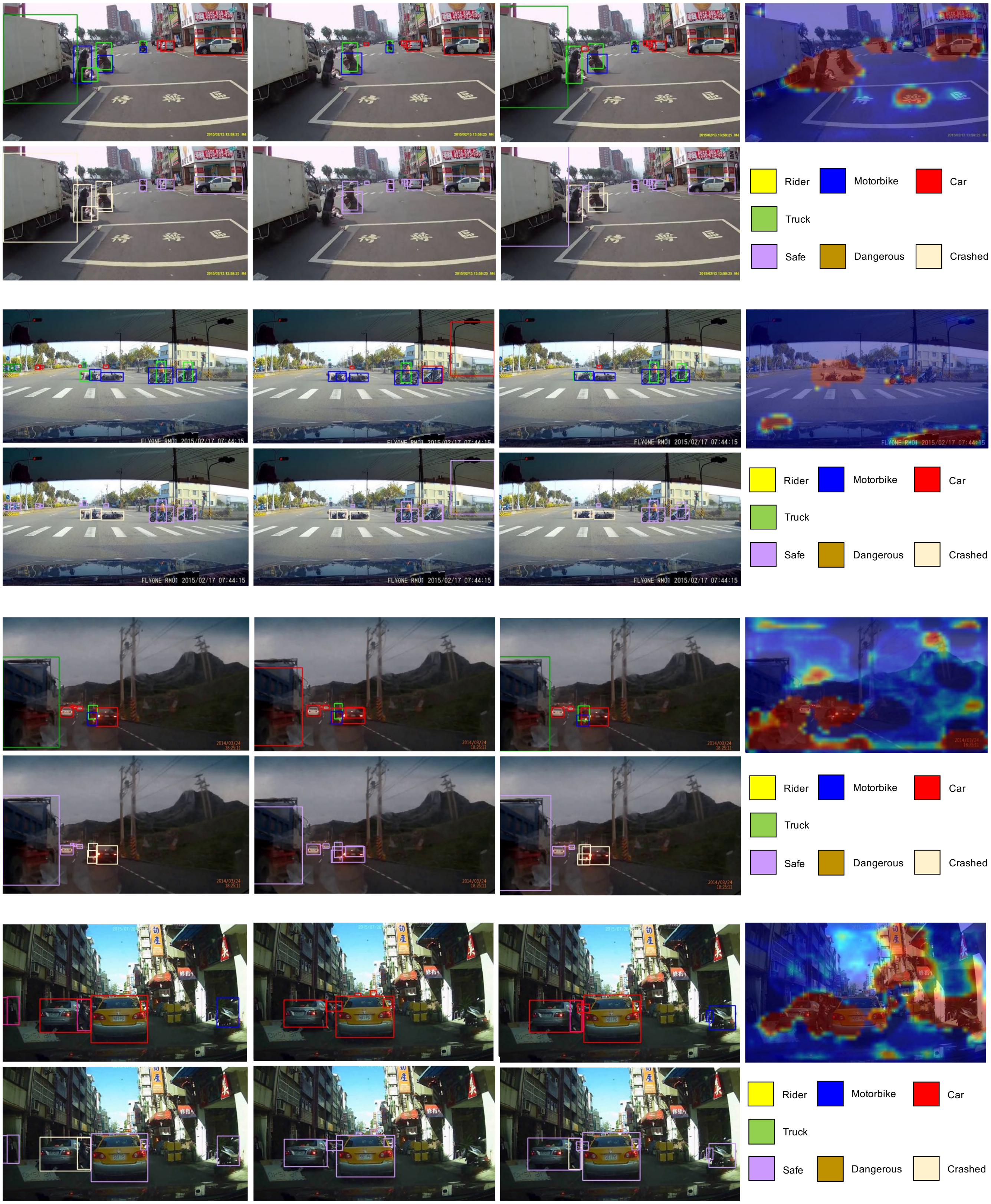


Fig. 6: Visualization of some results the ADV dataset. From left to right, original video frames with ground-truth are followed by results of Faster R-CNN*, our proposed Attention R-CNN, and attention maps generated from our method, respectively. For each image, the first row is object class and second row object characteristic detection results.