2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Joint Pedestrian Detection and Risk-level Prediction with Motion-Representation-by-Detection

Hirokatsu Kataoka[1], Teppei Suzuki[1], Kodai Nakashima[1], Yutaka Satoh[1] and Yoshimitsu Aoki[2]

***Abstract*— The paper presents a pedestrian near-miss detector with temporal analysis that provides both pedestrian detection and risk-level predictions which are demonstrated on a self-collected database. Our work makes three primary contributions: (i) The framework of pedestrian near-miss detection is proposed by providing both a pedestrian detection and risk-level assignment. Specifically, we have created a Pedestrian Near-Miss (PNM) dataset that categorizes traffic near-miss incidents based on their risk levels (high-, low-, and no-risk). Unlike existing databases, our dataset also includes manually localized pedestrian labels as well as a large number of incident-related videos. (ii) Single-Shot MultiBox Detector with Motion Representation (SSD-MR) is implemented to effectively extract motion-based features in a detected pedestrian. (iii) Using the self-collected PNM dataset and SSD-MR, our proposed method achieved +19.38% (on risk-level prediction) and +13.00% (on joint pedestrian detection and risk-level prediction) higher scores than that of the baseline SSD and LSTM. Additionally, the running time of our system is over 50 fps on a graphics processing unit (GPU).**

I. Introduction

To improve self-driving performance with a driving recorder, appropriate spatiotemporal understanding is required in addition to object detection. Herein, we focus on pedestrian detection and movement analysis for an advanced safety system.

Representative databases for traffic system and autonomous driving such as the KITTI [1] dataset and CityScapes [2], have increased in scale over the past decade. Unfortunately, they do not contain any traffic near-miss incidents (A traffic near-miss incident is defined as an event in which an accident is avoided through evasive driving action such as braking and steering [3]). The analysis of near-miss incident videos is still challenging because most existing pedestrian detection methods cannot determine whether or not a situation is dangerous. For instance, Figure 1 shows both normal (without any danger) and near-miss incident scenes, in which the vehicle-mounted drive recorders capture pedestrians. However, recent detection approaches often have difficulty identifying pedestrians in rapid motion because they have visual features, such as movement and posture, that are different from those of normal pedestrians. Moreover, recent object detectors cannot directly understand impending dangers in incident/accident scenes. Therefore, a pedestrian dataset that contains near-miss incidents and pedestrian locations would provide us with increased opportunities to recognize dangerous scenes in safety systems. However, the collection of pedestrian near-miss incidents is inevitably a very difficult task due to (i) the rarity in traffic scenes, and (ii) the region of a video capture involving a potential pedestrian incident being relatively small. The challenging problem must be conducted in order to avoid such a traffic near-miss incident.

Fig. 1. Normal (left) and near-miss incident (right) scenes on public roads. We believe that conventional pedestrian detection methods are unsuitable for detecting near-miss incidents because the two scenes include differences such as movement and pedestrian posture. The related work [3], [4] actually separates various driving scenes into near-misses and backgrounds to detect dangerous situations. Therefore, it was necessary to collect a database and construct an approach that would allow joint analysis of motion representation and pedestrian detection.

In this paper, we propose a novel framework called the Single-Shot MultiBox Detector with Motion Representation (SSD-MR) for joint pedestrian detection and risk-level prediction in traffic safety systems, including self-driving cars. Our goal is to predict pedestrian risk levels (high-/low-/no-risk) as well as perform localization with a vehicle-mounted driving recorder.

We summarize our contributions as follows:

Conceptual contribution: Since we believe that the analysis of various near-miss incidents is useful for avoiding risks, we have collected a new traffic database consisting of pedestrian near-miss incidents. This Pedestrian Near-Miss dataset (PNM dataset) is used to complete the joint task of pedestrian risk-level prediction (high-, low-, and no-risk) and pedestrian detection enables the simultaneous output of a pedestrian bounding box (bbox) and risk level.

Technical contribution: We also propose SSD [5] with motion representation (SSD-MR) as a concept of *motion-representation-by-detection*. SSD-MR processes the spatiotemporal vectors of each detected pedestrian with dilated convolution [6] and then uses that information to predict a risk level. Though the self-collected PNM dataset contains a lot of blurry images and cluttered backgrounds by vehicle-mounted driving recorders, SSD-MR overcomes the difficulties through the joint training with pedestrian detection and risk-level prediction.

Experimental contribution: Our results clearly show the

[1]National Institute of Advanced Industrial Science and Technology (AIST)
[2]Keio University

effectiveness of our approach when used in conjunction with SSD-MR and our self-collected PNM dataset. The data collection and sophisticated modeling of our proposed approach enable us to decisively detect the most at-risk pedestrian and the incident risk level.

II. Related work

Pedestrian detection. Since the deep neural networks (DNN) burst into public view in 2012 when AlexNet with ImageNet [7], [8], [9], various approaches have been proposed in the computer vision field and the region-based convolutional neural networks (R-CNN) [10] has become a critical algorithm in the object recognition field, while the search for a de-facto-standard object detection process has witnessed numerous advancements. The improvements made to sophisticated algorithms such as the Fast/Faster R-CNN [11], [12] have facilitated progress toward real-time object detection and researchers have proposed adjustments to make them more suitable for pedestrian detection (e.g.[13]. As a result, one-shot algorithms such as SSD [5] and you only look once (YOLO) [14], [15] now provide faster ways to detect many types of object.

The use of representative object detection algorithms has facilitated pedestrian detection [16], [17]. However, since pedestrian detection algorithms currently lack the motion representation required to understand more of the detailed information available in driving recorder videos, we have extended our detection algorithm to allow risk-level predictions using a space-time feature.

Temporal analysis. Dense trajectories (DT) and improved DT (IDT) models [18], [19] have been employed in video analysis. To represent a sophisticated motion vector, the DT model densely captures optical flows and combines HOG [20], HOF [21] and MBH [22] feature vectors. In the DNN era, several three-dimensional (3D) convolutional networks have been proposed by Ji *et al.* [23] and Tran *et al.* [24]. First, Ji *et al.* proposed a 3D-CNN model created from the input of multiple frames that could process spatiotemporal maps from various channels like gray, gradient, and optical flow. Meanwhile, Tran *et al.* presented the concept of 3D convolutional networks (C3D) as a means to directly capture a spatiotemporal representation in an image sequence. The C3D model employs a 3D convolutional kernel on xyt space obtained using only red-green-blue (RGB) sequences. Recently, a two-dimensional (2D)-based temporal model with a CNN known as a two-stream CNN [25] has proved to be a successful approach to action recognition. This model utilizes two CNN models, one of which is trained with stacked flow images while the other is trained with sequential RGB images. In the stacked flow images, dense optical flows are projected into a 2D image at each $x-$ and $y-$direction. Surprisingly, the two-stream CNNs outperforms the C3D model on representative human action datasets such as UCF101 [26] and HMDB51 [27] thereby indicating that 2D-kernel performance and two-stream models are suitable for action recognition on practical datasets. Although more recent action recognition studies [28], [29] show significant improvements in 3D-kernels, the data sought after in our study does not involve action recognition from a video sharing service. To robustly predict risk levels, we assign an SSD-MR that extracts temporal activations based on pedestrian detections. This allows our SSD-MR to effectively analyze sequential convolutional activations within the detected bbox without considering optical flows.

Fig. 2. Annotation example: We annotate risk-level {high-, low-, and no-risk} and bbox with {x, y, w, h} (height (h) and width (w)) at each frame. To improve annotation quality, the labels are checked by validators.

Traffic datasets. Efforts aimed at putting self-driving cars into practical use have included the application of the KITTI [1] and CityScapes datasets [2] as full-task benchmarks. Undoubtedly, KITTI is a well-organized benchmark and there are significant ongoing efforts to update the various algorithms such as stereo vision, traffic object detection, and visual odometry to a level sufficient to permit their use in self-driving cars. On the other hand, CityScapes provides well-defined semantic labels that can be used to train traffic scene parsing models. The most important issue in advanced driver assistance systems (ADASs) and self-driving cars must always be traffic accident avoidance. However, the above-mentioned vision-based databases are missing a perspective of accident/incident database collection. Along this line, the NTSEL [30], [31] and the Near-miss Incident DataBase (NIDB) [3], [4] have issued a highly motivated challenge aimed at achieving a more direct and active understanding of traffic accidents. Especially, the NIDB contains a large number of traffic videos captured in real-world situations. As a result, the database uses task-specific fine-tuning and semantic information to help us effectively understand incident scenes.

However, even using the NIDB, pedestrian locations in traffic incident scenes cannot be specified at a sufficient level of detail. In contrast, using the PNM dataset, we provide a large number of bbox annotations in addition to risk-level labels, which allows us to simultaneously train traffic risk levels (high-, low-, and no-risk) as well as pedestrian locations.

III. Pedestrian Near-Miss dataset (PNM dataset)

A. Dataset summary

In this section, we show the details of the PNM dataset, which is based on the NIDB [3] (Examples are shown in Figure 1). Although the collection of such data is very difficult due to the rarity of incidents in actual traffic scenes, we have managed to collect a total of 2,880 videos for the current PNM dataset. Each video consists of 10 - 15 seconds

of footage taken at 30 fps before/after a near-miss incident at 640×480 [pixel], and each video segment was annotated with a pedestrian's risk level {high-, low-, or no-risk} based on the low/high risk division outlined in III-B. Next, we applied 2,208 of the videos for training and used the other 672 videos for testing. The near-miss incident videos were obtained by vehicle-mounted driving recorders installed in more than 100 taxis.

B. Definition of traffic near-miss incident

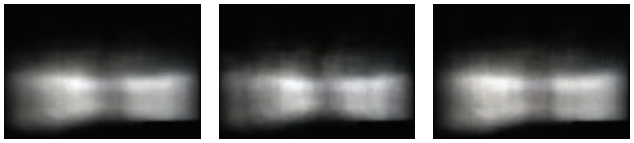
A traffic near-miss incident is an event in which an accident is avoided by driving operations such as braking and steering. Near-miss situations occur more frequently than collisions.

We evaluated risk levels as low or high based on the potential for a collision if drivers did not take appropriate actions such as emergency braking and/or evasive steering maneuvers. The high- and low-level risk categories correspond to the time-to-collision (TTC) [32]. In the case of a high-level risk, a driver must react in less than 0.5 s (TTC $<$ 0.5 s) to avoid a collision. For low-level risks, the TTC is more than 2.0 s (TTC $>$ 2.0 s). Videos containing mid-level risk (0.5s $\leq$ TTC $\leq$ 2.0s), which appear to show a mixture of high- and low-level risks, were excluded from the database. To train a deep CNN, a clear visual distinction should be made by the PNM dataset. This paper focuses on high- and low-level risks (without mid-level risk) in order to clearly divide the degree of risk. Especially in the risk-level prediction, we automatically divide risk into high or low. Therefore, human validator only checks the automatically annotated labels.

C. Collection, annotation, and cross-validation for the database

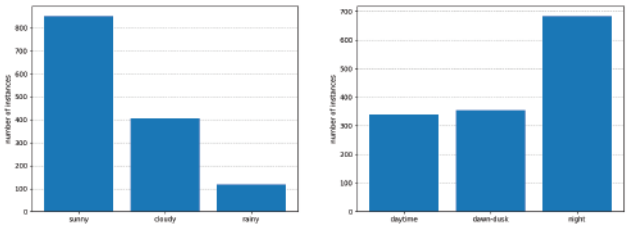
Although near-miss videos are difficult to collect, they are considered necessary for developing autonomous systems capable of driving safely in traffic. These video recording systems were triggered if there was sudden braking resulting in deceleration of more than 0.5 G. Each video was annotated according to its risk level {high-, low-, or no-risk}, and each frame was an annotated bbox with the {x, y, w, h} (height (h) and weight (w)) of the pedestrian in relation to the near-miss situation. See Figure 2 for examples. When no pedestrian is in the frame, the bbox does not appear. Of the training samples utilized, 1,030 videos were annotated as no-risk, 337 as high-risk, and 841 as low-risk. Test samples included 523 no-risk videos, 49 high-risk videos, and 100 low-risk videos. The number of validation set is following the near-miss dataset [3]. The other videos are used as training set.

To avoid ambiguity and prevent strong bias in our data annotations, three expert annotators (who have knowledge for computer vision) trimmed and categorized each video to 30 frames, after which they were assigned to a single category. Totally, PNM dataset contains 66,240 (2,208 [video] $\times$ 30 [frame]) video frames. The dataset was then cross-validated by the 2 annotators and 2 extra validators. The annotator and validators thoroughly checked the videos at least once.



(a) Overall frames (b) First 10 frames (c) Last 10 frames

Fig. 3. Averaged images with movement of bounding boxes. The pixel values are normalized for best view in color.



(a) Weather (b) Timezone

Fig. 4. Statistics in weather condition and timezone.

Note that the PNM dataset videos were collected from various vehicles, places on {*intersection*, *city areas*, and *major roads*}.

D. Dataset statistics

We list dataset statistics in Figure 3 and 4. Figure 3 illustrates the averaged images with movement of bounding boxes. Moreover, Figure 3(a), 3(b) and 3(c) show overall, first 10 frames and last 10 frames, respectively. Figure 4 denotes the statistics of weather and timezone. We separated the weather attribute into *sunny*, *cloudy* and *rainy* (see Figure 4(a)). Similarly, we divided the timezone into *daytime*, *dawn-dusk* and *night* (see Figure 4(b)).

IV. Pedestrian Risk Analysis

The process flow of our pedestrian risk analysis system is shown in Figure 5. Referring to the SSD-MR, we jointly execute *pedestrian detection* (section IV-A) and *risk-level prediction* (section IV-B). Our strategy aims at predicting a pedestrian risk-level by analyzing temporal activations from a detected pedestrian.

A. Pedestrian detection

To extract a temporal activation from a pedestrian area, we first must detect a pedestrian in a driving recorder video. Our baseline detector is designed based on the SSD [5], which is a de-facto-standard detection framework. The detailed SSD-based architecture is shown in Figure 6. The different point from the original SSD is to output a temporal activation x_T with 256-dim vector per frame.

Multi-scale voting with different layers. Multi-scale voting is processed with four different layers (conv4, 6-8) in order to output a region proposal at each layer and reliably detect a pedestrian in each driving recorder video. The multi-scale voting scores are obtained from Conv4, Conv6, Conv7, and Conv8 based on the SSD. As shown in Figure 7(a), voting scores are projected into the region around pedestrian as a distribution. The final detection result is decided using confidence scores from the anchor boxes (IoU $= 0.5$). The

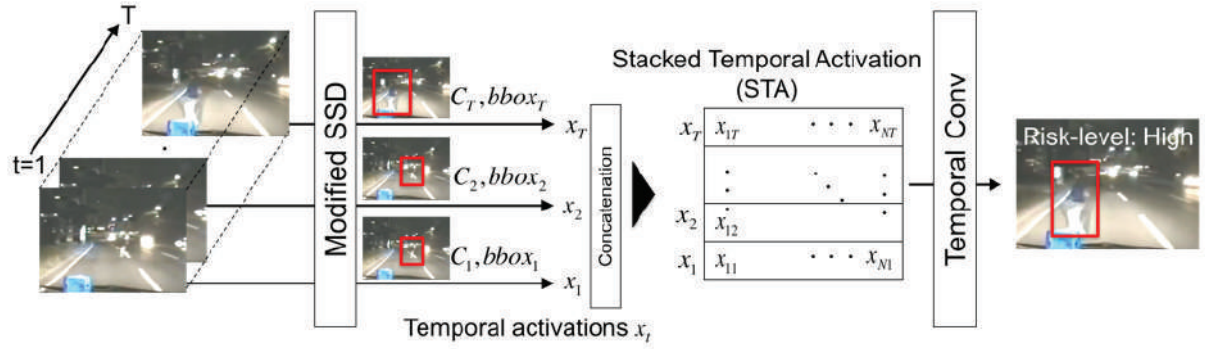


Fig. 5. Using our SSD-MR, we detect a pedestrian at each frame in order to extract a temporal activation N-dim for risk level prediction. After obtaining temporal activations from the T frames, stacked temporal activation (STA) were stacked with $T \times N$-dim to produce risk-level predictions. The detailed modified SSD architecture is shown in the next figure.

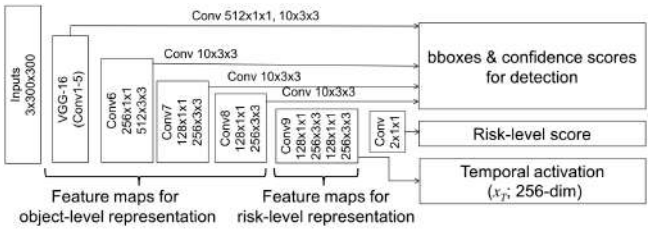


Fig. 6. Detailed modified SSD architecture. The kernel parameters used in the SSD are shown as *channels*×*width*×*height* like $256 \times 3 \times 3$. The first five convolutional layers follow the Visual Geometry Group (VGG)-16 [33] architecture. The detailed kernel parameters that result after the 6th layer (Conv6) are shown in the figure. In Conv9, a temporal activation x_T can be obtained for each frame.

maximum anchor box with the confidence score is the final detection result. The multi-scale vote detection process is shown in Figure 7(b).

Feature learning toward risk analysis. In order to obtain sophisticated temporal activation as a feature vector, we train our SSD-MR by assigning pedestrian risk-level labels and obtaining a temporal activation (x_T; the vector corresponds to the 256-dim vector in Figure 6) based on the output of Conv9.

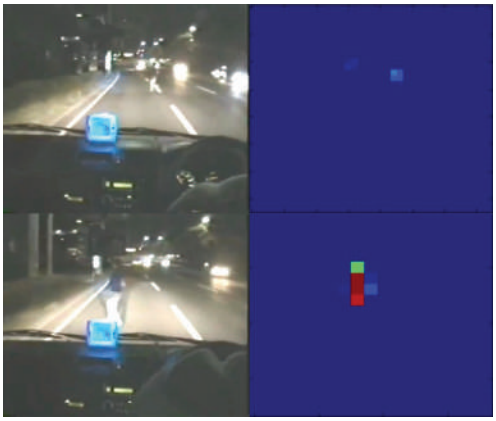
Since the SSD-MR is trained with risk-level labels and bboxes, we can evaluate risk-specified features from the detected pedestrian. The SSD-MR is trained with a multi-task loss function as shown below:

$$L(x, c, l, g, r) = \frac{1}{N}(L_{conf}(x, c) + L_{loc}(x, l, g)) + L_{risk}(r) \quad (1)$$

where the first and second terms $(L_{conf}(x, c) + L_{loc}(x, l, g))/N$ make up the loss function based on the SSD, $L_{risk}(r)$ is the softmax cross-entropy loss for risk prediction, and r is the set of risk-level annotations and the prediction.

B. Risk prediction

Pedestrian risk levels are predicted by using temporal activation in SSD-MR. We define y as the risk label $y \in \{high\text{-}, low\text{-}, or\ no\text{-}risk\}$ and $\mathbf{v_i} = \{v_{i1}, v_{i2}, ..., v_{it}\}(i \in 1, 2, ..., N)$ as temporal activations. To calculate the conditional probability $P(y|\mathbf{v})$, we can use the temporally stacked



(a) Visualization of multi-scale voting. The confidence scores obtained from the left input images with four different convolutional layers are shown in the right figures.

(b) The final detection results are decided based on confidence scores. (Left figure) Votes are cast for four different region proposals. (Right figure) The maximum confidence score is used as the final detection result.

Fig. 7. Pedestrian detection with modified SSD.

$t \times 256$-dim matrix called stacked temporal activation (STA), which consists of the stacked feature vectors given in the output of SSD-MR Conv9 (256-dim vector in Figure 6) in temporal order. We analyze the STA with a temporal convolution based on dilated convolution [6], which also allows easy expansion of the receptive field. Our temporal convolution is shown in Figure 8. If we were to use general convolution with a kernel size of two, 31 convolution modules would be required to obtain a receptive field sufficient to cover the T (=30) frames input. $L_{risk}(r)$ with temporal convolution is trained using softmax cross-entropy.

Next, we consider an evaluation between our SSD-MR and SSD with Long Short-Term Memory (LSTM) [34] (see Table III). In the pure comparison, the temporal convolution

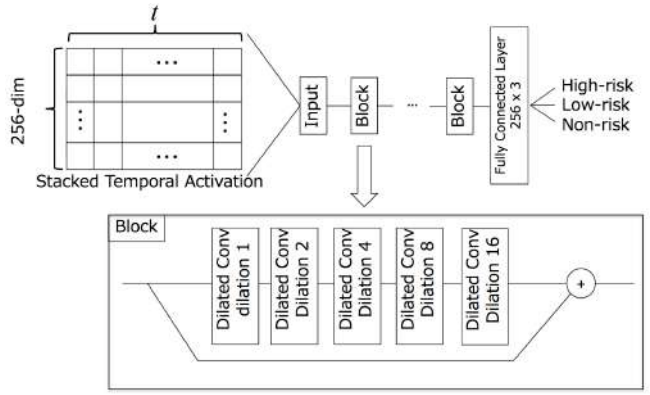


Fig. 8. Temporal convolution with stacked blocks. The kernel size of the dilated convolution is set to 2.

Fig. 9. Example of detection results. The blue rectangle is the detection result and the red rectangle is the ground truth on the detection failure frame.

has fewer parameters than the LSTM. When the LSTM is used, the number of parameters is $N \times N \times 8$ (input, input gate, output gate, and forget gate totally have weights for the current input and previous output). In contrast, our temporal convolution has fewer parameters with $2 \times 5 \times B$ (*#kernel size* $\times$ *#convolution* $\times$ *#block*). Further results and discussions are provided in the experimental section.

C. Implementation detail

In the detection part of SSD-MR, we use risk level labels if the annotations on the PNM dataset are *high* or *low*. In the test set, we have 149 videos with annotated risk levels.

In the temporal convolution in SSD-MR, we train with all training samples and evaluate all test samples.

The both parts were separately optimized, namely we initially trained the detection part with bbox ground truths then the temporal representation part (STA) is trained with the ground truths of risk-level score.

We set the learning rate to 0.001, the momentum to 0.9, and use a weight decay of 0.0005 for the detection part/0.001 for temporal convolution, and use a stochastic gradient descent (SGD) optimizer.

V. Experiment

In the section, we mainly show the recognition and detection performance for risk-level prediction (high-, low- and no-risk level) and joint task with pedestrian detection. However, at the beginning, we confirm how to construct our SSD model with multiple layers and voting mechanism because our dataset is different from conventional datasets like KITTI [1].

TABLE I

Pedestrian detection: Evaluation of pedestrian detection with precision and recall

	Approach	Precision	Recall	F-measure
3-layer SSD-MR	w/o vote	.9915	.6782	.8055
	w/ vote	**.9916**	.6830	.8089
4-layer SSD-MR	w/o vote	.9910	.6865	.8111
	w/ vote	.9911	**.7036**	**.8230**

TABLE II

Risk-level prediction: Comparison of block numbers

Approach	Ave. Recall (AR)	Ave. Precision (AP)	Ave. F-score
Block 1	.7417	.7419	.7305
Block 2	.8301	.6774	.6939
Block 3	**.7970**	.7212	**.7437**
Block 4	.7985	.6448	.6661
Block 5	.7422	**.7464**	.7381

A. Pedestrian detection

We evaluated the following properties in the pedestrian detection experiment:

- Which is better? A model with three convolutional layers (3-layer) or a model with four convolutional layers (4-layer)? (*3-layer* and *4-layer* in Table I. The 4-layer model is better.)

Next, we compared three convolutional layers (3-layer; Conv4, Conv6, and Conv7) with four convolutional layers (4-layer; Conv4, Conv6, Conv7, and Conv8) in SSD-MR. The 3- and 4-layer models were trained using 70,000 iterations on the PNM dataset. The pedestrian detection methods are validated with average precision (AP) and average recall (AR) in Table I. By using the plain setting (without multi-scale voting), the 4-layer model is +1.41% better than the 3-layer model with F-measure.

- With or without multi-scale voting in pedestrian detection? (*w/ multi-scale voting* and *w/o multi-scale voting* in Table I. With multi-scale voting is better.)

We compared pedestrian detection models both with and without multi-scale voting. From the results of our comparison, we can see that multi-scale voting is an effective method, especially in the recall scores (+0.48% for 3-layer and +1.71% for 4-layer). Both models perform better with multi-scale voting (+0.34%@3-layer and +1.19%@4-layer with F-measure).

Finally, we improved +1.75%@F-measure and +2.54%@recall with SSD-MR with four-layer, multi-scale voting. The visual results are shown in Figure 9. Here, it can be seen that our SSD-MR effectively detected various scales even though nighttime scenes are included in the test sample. Note that pedestrians cannot be detected when the outline is not clear because of halation and background darkness (the bottom-left in Figure 9). Pedestrian detection is obviously very challenging when halation is present. Although pedestrian detection does not work well at every

(a) IDT (b) SSD with LSTM (c) SSD-MR (ours)

Fig. 10. Confusion matrices for joint pedestrian detection and risk-level prediction. Our SSD-MR achieved averaged 79.33 score.

TABLE III
RISK-LEVEL PREDICTION: COMPARISON OF OUR APPROACH TO VARIOUS PARAMETERS AND RELATED WORKS. AR, AP AND AF INDICATE AVERAGE RECALL, AVERAGE PRECISION AND AVERAGE F-MEASURE.

Approach	AR	AP	AF
IDT[19]	.6167	.5531	.5828
Temporal Stream[25]	.3983	.3670	.3644
Spatial Stream[25]	.3515	.3436	.3178
Two Stream[25]	.3299	.2899	.2990
SSD with LSTM	.6647	.5667	.5499
SSD-MR (ours)	**.7970**	**.7212**	**.7437**

frame, our goal is to predict a risk-level (including no-risk labels) from some of the frames in which pedestrians are detected in temporal order.

B. Risk prediction

In the temporal convolution process, we tune the number of blocks as shown in Figure 8. Table II lists the relationship between the stacked block(s) and the performance with AR, AP, and F-score. The mean performance rate in the joint pedestrian detection and risk-level prediction task is evaluated in the table. The best model is three stacking blocks with the proposed model.

We compared our SSD-MR with representative models, specifically IDT, two-stream CNN, and SSD with LSTM. Basically, we employ the original tunings from IDT (HOG/HOF/MBH, codeword vector and support vector machine), two-stream CNN, and SSD with LSTM [34]. The two-stream CNN is trained on the PNM dataset in addition to the UCF101 pre-trained spatial- and temporal-stream. Additionally, the LSTM is assigned on behalf of the temporal convolution. Table III lists the results obtained on the PNM dataset.

Our approach achieves better results than the IDT and two-stream CNN (ours .7437 vs. IDT .5828 and two-stream CNN .2990 with F-score). Although a two-stream CNN is known as a representative motion model, it does not work well on the PNM dataset. We consider it likely that stacked flow images constructed by displacements of dense optical flow would include a large amount of noise. Moving vehicle-mounted cameras capture relatively large amounts of ego-motion compared with the movements of a pedestrian. Moreover, the videos included in the PNM dataset were recorded under a variety of adverse conditions, e.g. nighttime, rain, cluttered backgrounds, and viewpoint differences. The motion models including the IDT are disadvantageous to the conditions.

Interestingly, our temporal convolution with three stacked blocks outperformed the SSD with LSTM by +19.38% (.7437 vs. .5499 with F-measure). Against an LSTM, which has $256 \times 256 \times 4 \times 2$ parameters (256×256 fully connected layer, input and two gates have weights for the current input and previous output), our temporal convolution has fewer parameters: $2 \times 4 \times 3$ (kernel size, 4 convolutions and 3 blocks). Fewer parameters, faster training speeds, and a more accurate model are realized with dilated convolution.

Finally, we show the confusion matrices of our model, SSD with LSTM, and IDT in Figure 10. Our SSD-MR records 79.33% with averaged F-score which is +13.00% better than SSD with LSTM. Our SSD-MR model achieved the most balanced rates {80, 79, 79} at each {zero-, low-, high-}risk level in the joint pedestrian detection and risk prediction tasks. Other methods are slightly biased to zero-risk or high-risk from their prediction results. Please view the visual results in the supplementary video. Additionally, to measure the running time of our system, we ran SSD-MR on an NVIDIA GeForce Titan X with Pascal architecture. From the results obtained, we confirmed that our system runs at over 50 fps.

VI. CONCLUSION

We proposed a traffic near-miss detection architecture with temporal representations that jointly solves pedestrian detection and risk-level prediction. We also presented our Pedestrian Near-Miss dataset (PNM dataset), which provides pedestrian annotations of location and risk level. We demonstrated the effectiveness of our SSD with motion representation (SSD-MR). The proposal is superior to the other model including IDT and SSD with LSTM. We believe that the combination of pedestrian detector, stacked temporal activation (STA), and temporal convolution performs effectively in terms of fewer parameters, faster training speed, and increased accuracy based on the results of a pure comparison with SSD+LSTM. In future, we will continue to extend the dataset to improve overall performance.

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[2] 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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[3] H. Kataoka, T. Suzuki, S. Oikawa, Y. Matsui, and Y. Satoh, "Drive video analysis for the detection of traffic near-miss incidents." ICRA, 2018.

[4] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident DB." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in European Conference on Computer Vision. Springer, 2016, pp. 21–37.

[6] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," arXiv preprint arXiv:1511.07122, 2015.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 248–255.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision (IJCV), vol. 115, no. 3, 2015.

[10] R. Girshick, J. Ddonahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2015, pp. 580–587.

[11] R. Girshick, "Fast r-cnn," in International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448.

[12] 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, 2015, pp. 91–99.

[13] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2016, pp. 779–788.

[15] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2017, pp. 7263–7271.

[16] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?" in European Conference on Computer Vision (ECCV). Springer, 2016, pp. 443–457.

[17] X. Du, M. El-Khamy, J. Lee, and L. S. Davis, "Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection," in arXiv preprint arXiv:1610.03466, 2016.

[18] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 3169–3176.

[19] H. Wang and C. Schmid, "Action recognition with improved trajectories," in Proceedings of the IEEE International Conference on Comuter Vision, 2013, pp. 3551–3558.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. IEEE, 2005, pp. 886–893.

[21] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. IEEE, 2008, pp. 1–8.

[22] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in European Conference on Computer Vision (ECCV). Springer, 2006, pp. 428–441.

[23] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 35, no. 1, pp. 221–231, 2013.

[24] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in Proceedings of the IEEE International Conference on Comuter Vision (ICCV), 2015, pp. 4489–4497.

[25] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in Advances in neural information processing systems, 2014, pp. 568–576.

[26] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human action classes from videos in the wild." CRCV-TR-12-01, 2012.

[27] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition." International Conference on Computer Vision (ICCV), 2011.

[28] A. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6299–6308.

[29] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[30] H. Kataoka, Y. Aoki, Y. Satoh, S. Oikawa, and Y. Matsui, "Fine-grained walking activity recognition via driving recorder dataset." IEEE International Conference on Intelligent Transportation Systems (ITSC), 2015.

[31] H. Kataoka, Y. Aoki, Y. Satoh, S. Oikawa, and Y. Matsui, "Temporal and fine-grained pedestrian action recognition on driving recorder database," Sensors, 2018.

[32] Y. Matsui, M. Hitosugi, K. Takahashi, and T. Doi, "Situations of car-to-pedestrian contact," Traffic Injury Prevention, vol. 14, 2013.

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

[34] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," Neural computation, vol. 12, no. 10, pp. 2451–2471, 2000.