Deep Learning Based Classification of Pedestrian Vulnerability Trained on Synthetic Datasets

Jens Schleusner, Lothar Neu, Nicolai Behmann, Holger Blume
Institute of Microelectronic Systems
Leibniz Universität Hannover
Appelstr. 4, 30167 Hannover, Germany
{jens.schleusner, lothar.neu, nicolai.behmann, blume}@ims.uni-hannover.de

***Abstract*—The reliable detection of vulnerable road users and the assessment of the actual vulnerability is an important task for the collision warning algorithms of driver assistance systems. Current systems make assumptions about the road geometry which can lead to misclassification. We propose a deep learning-based approach to reliably detect pedestrians and classify their vulnerability based on the traffic area they are walking in. Since there are no pre-labeled datasets available for this task, we developed a method to train a network first on custom synthetic data and then use the network to augment a customer-provided training dataset for a neural network working on real world images. The evaluation shows that our network is able to accurately classify the vulnerability of pedestrians in complex real world scenarios without making assumptions on road geometry.**

***Index Terms*—neural networks, advanced driver assistance, pedestrian detection, synthetic dataset**

I. Introduction

The increasing complexity of traffic and an increasing number of traffic areas shared by motorised and non-motorised road users make driver attention a scarce resource that must be guided towards relevant features in the vehicle environment. In the automotive sector, different technologies are available to highlight pedestrians, cyclists and other vulnerable road users (VRUs) in order to draw the driver's attention towards them and therefore improve the driver's steering and braking performance [1]. Head-up displays, matrix headlights and other augmented reality devices can be used to perform this task. Modern vehicles are equipped with sensors for Advanced Driver Assistance Systems (ADAS), which can provide various 2D and 3D sensor data streams. Extracting information about other road users requires a classification filter that detects and selectively highlights VRUs who share the traffic area with the driver.

II. Related Work

LIDAR and vision-based systems can be used to detect pedestrians and other vulnerable road users in the vehicle's path. The authors of [2] show the fusion of visual detection based on Fast R-CNN with LIDAR data. [3] focusses on the detection of cyclists with LIDAR. Their detection is based on Faster R-CNN and was trained on synthetic depth images. LIDAR provides high-quality 3D images, but has a limited range and is more expensive than camera-based solutions. Therefore, LIDAR is typically used for research, prototyping and high-end vehicles.

Optical systems can be tailored to massproduced vehicles as they provide good results even when using consumer-grade cameras. Vision-based collision warning and avoidance systems typically consist of two algorithmic steps.

The first step is the detection of all vulnerable road users within the camera image. Algorithms such as HOG [4] are used in automotive systems, but are replaced by neural networks that perform better under difficult conditions. In [5], a SegNet is used to detect pedestrians and cyclists in optical images.

The second step following the pedestrian detection is classification into different classes of vulnerability to assess the risk of endangerment. The vulnerability can be calculated based on the size and position of the detected objects within the frame. In [6], VRU detection using a 2D pose estimation network and fixed image regions for classification is shown. A straight road geometry with sidewalks on the left and right side is assumed. Objects in the top, left and right parts of the frame are therefore considered less vulnerable than objects in the front center part. On straight roads, this assumption works well, but fails in more complex road environments. When the vehicle turns, VRUs visible in front of the car can safely walk on the opposite sidewalk, but VRUs on the side towards the turn are at risk. Similar problems arise for narrowing and widening of the road, T-intersections and at the end of declines.

III. Proposed Work

The VRU classification task requires a semantic understanding of the scene. Ideally, all aspects of a situation can be assessed from a single image without relying on temporal information or tracking. One possible solution is the use of a CNN for semantic scene labelling like PSPNet [7]. Semantic information about the relative position of the road and VRUs can then be generated by analyzing the neighboring pixels at the base point of objects. This works well for an open, unobstructed view of the scene, but requires near-perfect segmentation. This analysis fails if the base points of the objects are occluded, which is usually the case in an urban environment (Fig. 1). Furthermore, a full semantic segmentation requires a lot of computational power. To highlight VRUs, a detection system based on bounding boxes is sufficient

 DOI: https://doi.org/10.1109/ICCE-Berlin47944.2019.8966161

Fig. 1. Camera view generated using our virtual environment. Pedestrian bounding boxes show ground truth vulnerability (red/green).

Fig. 2. Synthetic semantic segmentation showing the vulnerability (red/green) of occluded pedestrians on the sidewalk (pink) and on the road (purple)

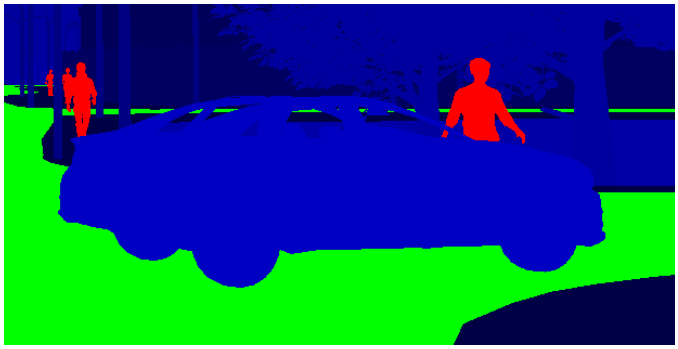

Fig. 3. Preprocessed training image with modified colors to emphasize the relation of pedestrians and road objects.

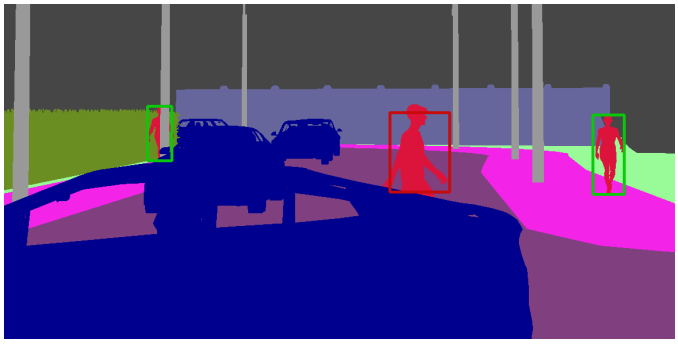

Fig. 4. Bounding boxes generated by the intermediate CNN_1. The three pedestrians are detected as 68% safe, 88% vulnerable and 100% safe.

and allows execution in real-time. We therefore propose an approach in which the pose of VRUs is directly assessed in relation to traffic areas with a YOLOv3 network [8] trained on synthetic and real world training data.

IV. Data Acquisition

There are no pre-labeled datasets available to classify the location of road users in relation to the traffic areas in which they move. Existing datasets such as Cityscapes [9] contain semantic labels for pedestrians, road areas and sidewalks, but it is a complex task to estimate the relative 3D position from these data alone. Moreover, these data cannot easily be hand-labeled from regular image datasets because it is very difficult to judge relative locations in occluded or ambiguous situations. We therefore use a virtual environment to generate our own custom synthetic training data.

A. Virtual Environment

Our simulation uses Unreal Engine [10] and the Microsoft AirSim plugin [11] with custom Python code to generate the synthetic training datasets. We modelled an area around an urban intersection based on satellite data and populated the scenario with vehicles and pedestrians on the sidewalk and crossing the road (Fig. 1). An actor vehicle with virtual cameras can be placed anywhere in the environment. We used the camera extrinsics provided by the Cityscapes dataset for our vehicle camera to match the field of view of the generated semantic segmentation. At each simulation tick, a downward facing 3D linetrace is executed for the pedestrians within the 3D game environment to detect if they are on the road object in order to provide 3D-referenced ground truth classification information (Fig. 2).

B. Rendering Configuration

Besides regular rendering of color images, AirSim's special render modes can be used to create Cityscapes-compatible semantic segmentation based on the virtual scenario. We have replaced the predefined pedestrian class with the two classes "vulnerable road user" and "safe road user". In order to generate bounding boxes for individual pedestrians even if they overlap, the classification is further extended to support instance based segmentation.

C. Dataset Recording

The setup of the recorded pedestrians, vehicles and objects in the synthetic data must be plausible to be used for transfer training. Therefore, random locations along a predefined plausible spline within our road layout are chosen for the actor vehicle. After placement of the actor, collisions are resolved before recording of our images to get images of regular driving scenes. Each recorded dataset contains 20,000 images with Cityscapes semantic segmentation and ground truth classification data. The classification is available for each instance as a bounding box and at pixel level (Fig. 2).

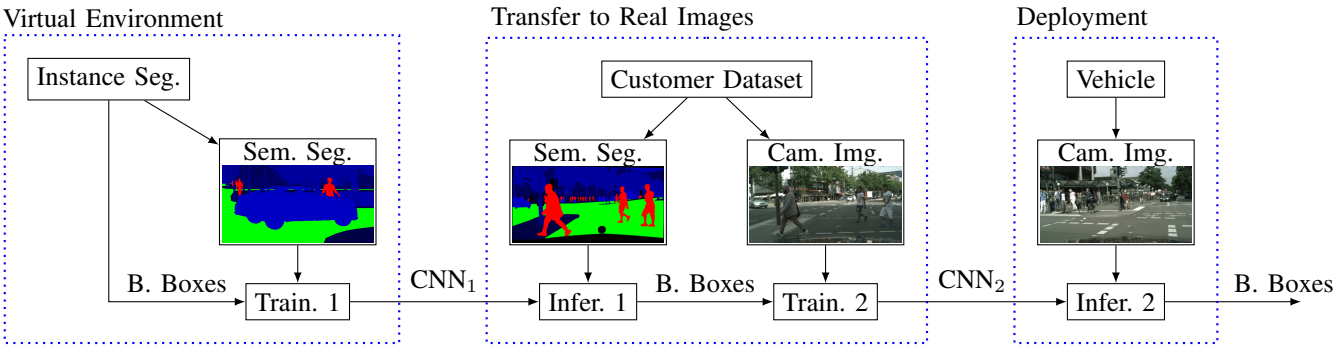

Fig. 5. The CNN-training consists of two steps. The virtual environment provides instance segmented images that are split into semantic segmentation and classification bounding boxes for training of CNN_1. The customer provides a reference dataset (Cityscapes). CNN_1 is used to infer bounding boxes from semantic segmentation for the second training step. Bounding boxes and camera images are used for the training of CNN_2. This can then be deployed to the vehicle to infer vulnerability on live camera images.

V. Neural Network Training

In order to classify the VRUs, we use a YOLOv3 Convolutional Neural Network (CNN) architecture and train it on our generated synthetic dataset using Darknet [12] on a cluster of three Nvidia Tesla V100.

A. Network Architecture

The original YOLOv3 network [8] is configured for general object detection and classifies its results with 80 labels. We used the published network as a starting point and reduced the number of classes from 80 to 2. This initial network detected objects covering a large part of the image. Our objects are usually farther away from the vehicle and therefore often visually small. The YOLOv3 detection anchors [8] have been adapted in our network to improve the detection of visually small VRUs. YOLOv3 is a multilabel classifier that supports classification using labels that are not mutually exclusive. Since our classes are inherently disjunctive, a softmax layer has been added. This layer selects the class with the highest probability to distinguish between vulnerable and non-vulnerable road users.

B. Two Step Training

A major problem with using synthetic data for CNN training is overfitting. In a virtual environment, only a limited number of textures, objects, and other assets are available. Therefore, training images are not realistic and diverse enough compared to real world images. Networks trained on synthetic data cannot be used directly for inference based on real camera images. Our solution to this problem is a two-step training process (Fig. 5). Instead of using the rendered images directly as input, we use the virtual environment to render images with a semantic segmentation compatible to the Cityscapes dataset and corresponding images containing our classification data. Then we train an intermediate CNN_1 with the segmented data as input and the generated classification as ground truth. This first network enables the translation of regular semantic segmentation into classified segmentation. In an inference step (Infer. 1), this network can then be applied to the ground truth semantic segmentation provided by the Cityscapes training dataset to generate new end-to-end training data with classification bounding boxes. This data is then used in a second training step (Train. 2) as ground truth to train the final CNN_2 on the real world Cityscapes images.

C. Data Preprocessing

Bounding boxes around the pixels for the instantiated classes are generated and saved as ground truth data. Each frame is further processed to generate training data. All pedestrian classes and instances are merged into a single pedestrian class that corresponds to the pedestrian class of Cityscapes. The training results are bad when using semantic segmentation directly as input, because the pseudo colors of the classes are chosen arbitrarily and therefore similar colors do not correspond to similar objects. Our evaluation showed that parked bicycles (dark red) are treated as pedestrians (light red) because they have a very similar red color. We therefore modified the classification colorscheme to emphasize the relation between pedestrians, the road surface, and all other objects. The three segmentation classes are mapped to the three independent dimensions of the RGB colorspace. Pedestrians are marked in red, the road surface green and the other classes are evenly distributed within the blue channel (Fig. 3).

D. Deployment

A major advantage of the two-stage training approach is that the classification task can be learned independently of the image dataset, since the transfer to real images takes place in a second training (Train. 2) step (Fig. 5). A customer can provide images from his vehicles camera system and corresponding semantic segmentation. Our intermediate CNN_1 then converts the segmentation into ground truth bounding boxes for training on the real images. The resulting CNN_2 can then be deployed to the vehicle and used with live video data to infer bounding boxes showing VRU vulnerability. These bounding boxes can then be used by augmented reality devices to highlight the VRUs at risk.

VI. Performance Evaluation

Our synthetic dataset consists of training, validation and test images. Training images are used to train the CNN. The unbiased validation dataset is used to evaluate parameter sets during training. We use the synthetic test dataset (Fig. 4) for the final evaluation of the intermediate CNN.

Our application of pedestrian highlighting depends on correct classification and precise localization of the objects. As a quality measure, we have chosen the standard mean average precision (mAP) metric [13].

For 11 detection thresholds [0,0.1,...,1], the bounding boxes of the detections are extracted. For each bounding box x of the two classes, we calculate the intersection over union (IOU) with the reference bounding boxes r of the same class.

$$\text{IOU} = \frac{x \cap r}{x \cup r} \qquad \text{Recall} = \frac{x \cap r}{x} \qquad \text{Precision} = \frac{x \cap r}{r}$$

If the IOU is greater than 50%, the bounding box is considered a valid detection. For these, recall and precision curves are generated. The mean average precision can then be calculated according to [13]. Table I shows that our detector, which achieves a mAP of 67.5%, lies between the original YOLOv3, which is trained for 80 classes and a LIDAR-enhanced network, that uses images as well as additional depth data.

TABLE I
Average Precision for our Network compared to a LIDAR-enhanced approach and the original YOLOv3

Method	mAP (50% IOU)
original YOLOv3 [8]	57.9%
proposed YOLOv3	67.5%
VGG + LIDAR [2]	75.7%

We applied our final CNN_2 to the Cityscapes test images (Fig. 6) to show the performance in real world applications. Pedestrians of different sizes are detected and correctly classified. Our approach shows good results even under difficult geometric conditions with vulnerable pedestrians in the foreground center and non-vulnerable pedestrians in the background.

VII. Conclusion

We showed a two stage training process for deep learning based classification of vulnerable road users. In the first stage, we train the specific classification task on synthetic semantic segmentation data generated with an application-specific virtual environment. In the second stage, the learned classification is transferred to real world data by applying the first network to a semantic segmentation dataset. The generated information is then used to train a network on real camera images. This approach enables the creation of scene-aware vulnerability maps for complex driving scenarios and enables autonomous driving applications on consumer-level hardware.

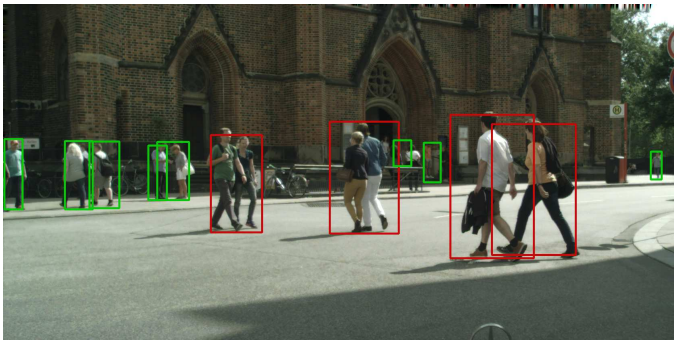

Fig. 6. Classification of our end-to-end CNN_2 applied to a Cityscapes image. Pedestrians on the sidewalk are classified as >65% safe (green). Pedestrians on the road are classified as >97% vulnerable (red)

References

[1] H. Kim, A. Miranda Anon, T. Misu, N. Li, A. Tawari, and K. Fujimura, "Look at me: Augmented reality pedestrian warning system using an in-vehicle volumetric head up display," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 2016, pp. 294–298.

[2] T. Kim and J. Ghosh, "Robust detection of non-motorized road users using deep learning on optical and lidar data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 271–276.

[3] K. Saleh, M. Hossny, A. Hossny, and S. Nahavandi, "Cyclist detection in lidar scans using faster r-cnn and synthetic depth images," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *international Conference on computer vision & Pattern Recognition (CVPR'05)*, vol. 1. IEEE Computer Society, 2005, pp. 886–893.

[5] A. Nurhadiyatna and S. Lončarić, "Semantic image segmentation for pedestrian detection," in *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*. IEEE, 2017, pp. 153–158.

[6] S. K. Maurya and A. Choudhary, "Deep learning based vulnerable road user detection and collision avoidance," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2018, pp. 1–6.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[9] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] "Unreal engine 4," https://www.unrealengine.com, accessed: 2019-04-01.

[11] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[12] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013–2016.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun 2010. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4