HazardNet: Road Debris Detection by Augmentation of Synthetic Models

Tae Eun Choe
tchoe@nvidia.com

Jane Wu
janehwu@stanford.edu

Xiaolin Lin
xiaolinl@nvidia.com

Karen Kwon
kkwon@nvidia.com

Minwoo Park
minwoop@nvidia.com

Abstract

We present an algorithm to detect unseen road debris using a small set of synthetic models. Early detection of road debris is critical for safe autonomous or assisted driving, yet the development of a robust road debris detection model has not been widely discussed. There are two main challenges to building a road debris detector: first, data collection of road debris is challenging since hazardous objects on the road are rare to encounter in real driving scenarios; second, the variability of road debris is broad, ranging from a very small brick to a large fallen tree. To overcome these challenges, we propose a novel approach to few-shot learning of road debris that uses semantic augmentation and domain randomization to augment real road images with synthetic models. We constrain the problem domain to uncommon objects on the road and allow the deep neural network, HazardNet, to learn the semantic meaning of road debris to eventually detect unseen road debris. Our results demonstrate that HazardNet is able to accurately detect real road debris when only trained on synthetic objects in augmented images.

1. Introduction

Object detection for autonomous vehicles mostly focuses on common objects and obstacles such as vehicles, bikes, pedestrians, traffic light/signs, and lane lines. However, equally important yet challenging objects to detect on the road are hazardous debris such as furniture, tires, fallen trees, animals, potholes, and more. When the height of road debris is larger than 15 centimeters, the treading vehicle may roll or detour from the path and potentially cause a fatal accident. Identification and localization of such hazardous objects is a critical task for both autonomous vehicles and regular road users. Based on the AAA traffic safety report[1], between 2011-2014, there were 200,000 crashes related to road debris, resulting in 39,000 injuries and 500 deaths. Two thirds of road debris are vehicle parts, unsecured cargo, and separated tow trailers. Since the most serious accidents occur on highways, where vehicles are traveling at high speeds, it is crucial for road debris detectors to identify objects early from a far distance.

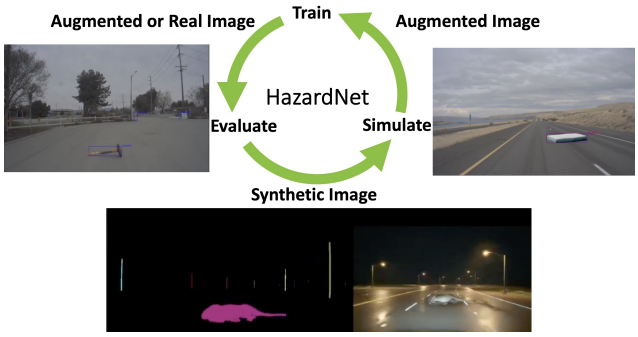

Figure 1. The cyclic workflow of HazardNet. Synthetic models of road debris are generated with domain randomization and rendered over real images using semantic augmentation. These augmented images are used to train HazardNet and evaluated on augmented and real images of road debris to assess few-shot learning performance. The evaluation results are used to tune the augmented data, and the cycle continues until performance converges.

There are two main challenges to road debris detection. The first is that data collection of road debris is extremely time-consuming, expensive, and dangerous. The conventional method requires physical vehicles to collect data. However, road debris are quite rare to encounter and thus requires long data collection times. In addition, when the data collecting vehicle approaches the road debris, it should either stop or detour, which are both dangerous maneuvers on the road. Though staging road debris on private roads is feasible, staging on various public roads is infeasible and limited by regulation. The second challenge is the broad variety of road debris, ranging from a small brick to a large detached trailer. The appearance and shapes of different hazardous road objects are difficult to categorize. For in-

[1] https://exchange.aaa.com/prevent-road-debris

stance, the appearance of a deceased animal and that of a ladder are quite dissimilar. An effective road debris detector should thus be able to identify an enormous number of distinct road objects, making it almost the same as a detector of all objects excluding common road entities.

We overcome the challenges of limited data and variability of road debris by using semantic augmentation with synthetic models and domain randomization. Contrary to conventional supervised learning methods on real labeled images, we propose to train a machine learning model, HazardNet, to detect hazardous road objects using few-shot learning. We generate training data for this objective by augmenting real images with several representative synthetic models in a semantically valid way. The main goal here is not only detecting road debris in training data but also learning the semantic context of road debris so that instances unseen during training can be detected at test time. This approach avoids dangerous and time-consuming data collection using a physical vehicle and speeds up the development process by using the synthetic data and recycling existing real data as well. See Figure 1.

To learn the general concept of road debris, we apply domain randomization by selecting several representative synthetic objects and placing them on the road in various locations and orientations and in images with different times of day and weather conditions. When a synthetic model is augmented on the image, it is augmented not in a random location but on the road as a hazardous obstacle using ground plane estimation. Since our main focus is to detect debris on the road, the synthetic model is placed on the path where the ego-vehicle moves forward. Our placement method also avoids overlap with common road elements such as vehicles, pedestrians, traffic signs, and vertical poles. Both domain randomization and semantic augmentation are essential for few-shot learning of visual and semantic road debris information. Even though there are an uncountable number of distinct road debris, the concept and meaning of road debris can be decoded using these two techniques.

In this paper, we mostly focus on road debris detection in the autonomous driving domain. However, the approach can be applied to other domains such as general detection of under-represented objects, medical imagery, speech recognition or any machine learning domain utilizing simulation data.

2. Previous Work

Synthetic Data: Recent work in deep learning has demonstrated the effectiveness of using synthetic data for object detection [13], semantic segmentation [47, 49], lane line detection [15], and optical flow [22, 32]. In the realm of autonomous vehicle perception, fully synthetic datasets such as CARLA [11], GTA5 [47], SYNTHIA [49], and Virtual KITTI [13] have been used to improve semantic segmentation of urban scenes [6, 8, 20, 26, 28, 39, 52, 57, 58, 64, 66, 67]. A number of these works use adversarial learning to facilitate domain adaptation from simulation to real data [6, 20, 26, 28, 58, 66, 67]. More similar to our approach, [56] performs domain randomization by rendering synthetic objects with random background images captured in the real world. However, the generated scenes do not follow any physical constraints, e.g. objects are placed at random 3D locations.

Augmentation: Rather than using completely synthetic data, several works augment real images by adding synthetic 3D models to the scene. In such cases, augmentation is typically applied to indoor scenes by placing objects in random locations [35, 53], and thus the resulting images need not bear resemblance to the physical world. Though road debris location has a strong association with the image ground plane, existing methods have no mechanism for realistically placing synthetic objects. More recently, [19] accomplishes image augmentation by generating a large dataset of 3D CAD models for various household objects and rendering each object over a cluttered background with added Gaussian noise. Another approach to augmentation is to add masked objects from real images to other images [12, 14, 27, 40].

Domain Randomization and Adaptation: Domain randomization is one of the most effective methods to reduce the sim-to-real domain gap. The concept was first introduced in [55] to expose deep neural networks (DNNs) to a wide range of different environments by randomizing the simulator when generating training data. This approach makes the assumption that if networks are trained on sufficiently varied synthetic data, models trained only in simulation can generalize to the real world without retraining. As in [55], domain randomization has most commonly been applied to robot manipulation and control [1, 23, 36, 50, 65]. Domain randomization for object detection is another application that has gained interest in the last few years [24, 39, 56, 64]. In a similar vein, the objective of domain adaptation is to align the source and target domains such that a model trained on the source domain can be applied to tasks in the target domain [3, 59]. Specifically for object detection, a number of recent works build upon Domain Adaptive Faster R-CNN [7], which adds domain adaptive components to Faster R-CNN [45] and trains the model in an adversarial manner. Subsequently, [18, 21, 42, 51, 70] have also applied adversarial learning to domain adaptation for object detection. [62] presents a categorical regularization framework on top of [7] that highlights the important image regions corresponding to categorical information.

Few-shot Learning: Few-shot, One-shot, or Zero-shot learning extrapolates beyond labeled data by inferring information about instances not seen during training [25, 48, 61, 63]. Recent work in zero-shot object detec-

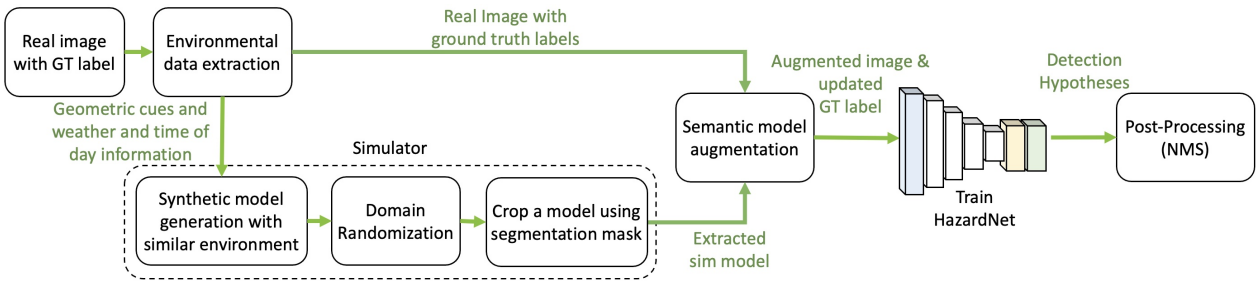

Figure 2. The data generation and training pipeline for HazardNet. Given a real image with ground truth labels, we use geometric cues and environmental information to randomly generate a synthetic model with texture corresponding to a similar environment. Once the model is selected, domain randomization is applied to render the synthetic model with random 3D pose, color tone, and visibility. Next, the rendered model is segmented and used to semantically augment the real image (e.g. taking into consideration overlapping objects and lane information.). We use the augmented images to train HazardNet and apply post-processing to the output predictions.

tion [2, 9, 29, 41, 46, 68, 69] focuses on detecting a set of unseen classes chosen to be excluded during training. In such cases, high-performance object detection networks, particularly YOLOv2 [44], have been used as backbones to provide a baseline for detection [9, 68, 69]. Most recently, [69] trained a conditional variational autoencoder to synthesize visual features for an input image, which was then used to re-train a confidence predictor to encourage detection of unseen objects.

Hazard Detection: Early approaches for image-based road hazard/obstacle detection use classical stereo reconstruction [33, 60] to detect small road obstacles at long distances. Following this work, geometric detection algorithms were used to detect obstacles from stereo images, including extensions of the Stixel algorithm [37], geometric clustering methods [4, 31], and the Fast Direct Planar Hypothesis Testing (FPHT) method [38, 43], which assumes that the road is planar. However, it is difficult to detect distant shallow objects with stereo vision. In addition, stereo vision and parallax-based approaches on the road are prone to fail and produce numerous false positive detections since road features are ambiguous and the 3D road profile breaks the planar assumption. More recently, advances in deep learning have shifted the focus to monocular obstacle detection [16, 34, 54]. For instance, [34] proposed an autoencoder framework that leverages semantic segmentation and anomaly calculation to detect obstacle regions in an input image. RGB-D images have also been used as input to DNN-based obstacle detection algorithms, including MergeNet [16] and RFNet [54]. [10] detects hazardous objects by segmentation -based anomaly detection.

3. Few-shot Learning of Road Debris

The lack of available data naturally makes few-shot learning an attractive approach to detecting real-world road debris. While we use synthetic road debris to train a DNN and detect those specific models, our ultimate objective is to detect unseen road debris in real-world images. To this end, we apply 1) semantic augmentation to meaningfully place synthetic objects in the scene and 2) domain randomization to increase the variety of road debris appearance. Semantic augmentation and domain randomization endow the DNN with semantic and visual understanding of road debris in the context of the ego vehicle and environmental conditions. The data generation and training pipeline is shown in Figure 2. Section 3.1 describes how 3D synthetic models are generated and saved as masked images given an input real-world image. Section 3.2 discusses domain randomization in the context of varying the appearance of the rendered synthetic models. Section 3.4 outlines the HazardNet architecture, and Section 3.5 considers important performance metrics for road debris detection.

3.1. Synthetic model generation

The most common and hazardous road debris are vehicle parts (tires, mufflers, hubcaps, bumpers), unsecured cargo (mattress, furniture, box, detached trailer), tree branches, and animals. Therefore, we collected 20 synthetic models of the most common road debris, including: cardboard boxes, small and big rocks, tires, wheels, wooden pallets, roadkill, wooden logs, traffic cones, barrels, mattresses, detached mufflers, trash cans, traffic sign bases, and detached trailers. When spawning objects on the road, their 3D information such as location in latitude/longitude, orientation (yaw/pitch/roll), lighting, weather condition, and time of day are saved as metadata. Since the appearance of the synthetic model should match the environmental conditions of the real image being augmented, we ensure that the weather conditions and times of day are consistent. The object is segmented using the instance segmentation mask generated by the simulator and saved as an image along with the mask and metadata.

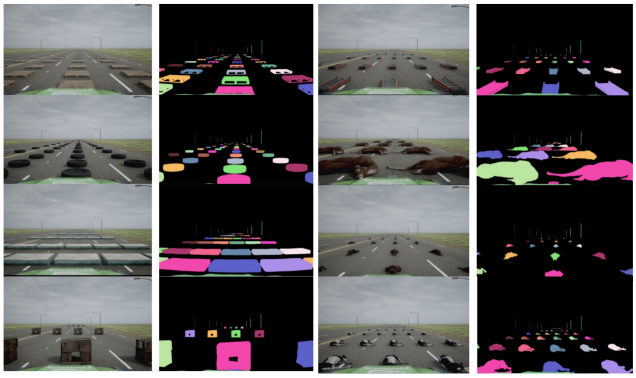

Figure 3. Road debris models spawned by a simulator and the corresponding segmentation masks.

3.2. Domain randomization on objects

The observable gap between synthetic models and real images includes appearances such as color, texture, and shadows. However, [56] showed that for DNNs, such artifacts from synthetic images are less important than the actual shape of objects. Therefore, we apply domain randomization to the generated models in Section 3.1 by randomly positioning the models in the simulator with different color and textures. Specifically, various road debris models in the simulator are generated by randomly sampling 3D position, 3D orientation, color tone, material, and visibility by fog or haze. Once these models are generated, one or more of these domain randomized instances are spawned in a simulated environment with environmental conditions that match that of the real image being augmented (using the metadata described in Section 3.1). Examples of domain randomization are shown in Figure 4. Most of the aforementioned randomization hyper-parameters were selected in a uniformly random manner. For model position, objects were placed up to 300 meters away from the camera or when the height of the 2D bounding box is at least 10 pixels. For model orientation, yaw, pitch, and roll angles were randomly selected. This orientation was further automatically corrected to follow the physics rules in the simulator, which considers model shape, road shape, and gravity.

3.3. Semantic model augmentation

In the last step of data generation, we augment the real images with the domain randomized synthetic models. During augmentation, we need to encode the semantic meaning of road debris, e.g. unusual objects on the road blocking a vehicle's path. By adding more semantic constraints for where road debris can be located, DNNs can more efficiently and accurately learn to distinguish road debris from other road elements. Therefore, synthetic models are placed on the planned path of the ego vehicle or its neighboring lanes (either left and right lanes or shoulders). Models

Figure 4. Examples of spawned ladder models using domain randomization with different times of day and weather conditions.

are not augmented in other unrelated lanes or in the sky to reduce the complexity of the road debris domain. Even though road debris can be on the sidewalk or in opposite lanes, debris out of the ego-vehicle's driving area are not considered for augmentation. In addition, we constrain the synthetic model to be on the ground and it cannot overlap with other objects identified by existing human-generated ground truth labels. Figure 5 shows examples of semantic augmentation with various synthetic models and environmental conditions. The green rectangles are human-labeled ground truth, the blue rectangles indicate acceptable augmentations, and the red rectangles are rejected for augmentation. We note that synthetic models in accepted augmentations are on the road, and overlap with other objects is accepted as long as the synthetic model is in front of existing objects (e.g. any occlusion introduced by augmentation must be physically plausible). In the case of the rejected augmentation shown in the bottom left image in Figure 5, the synthetic rock model is unnaturally above the truck. This case was filtered out by checking the y-coordinates of the bottom of overlapping rectangles. In the bottom right image, the synthetic wood pallet model was not an accepted augmentation since it was placed in the sky.

3.4. HazardNet architecture

ResNet [17] is used as the backbone architecture of HazardNet. However, DarkNet [44] or any DNN can be used since the main contribution of this paper is balanced data collection and sampling, domain randomization, and semantic augmentation. HazardNet learns features and outputs bounding box proposals, which are post-processed using non-maximum suppression (NMS) to output the final detection results. The architecture of HazardNet is shown in Figure 6. The final output layers ($120 \times 68 \times 5$) consist of one channel for detection confidence regression and four channels for bounding box positions, e.g. the normalized center position (x_c, y_c) and the width w and height h both divided by two. Binary cross entropy (BCE) in Equation (1) is used as the loss function of the confidence channel for each output pixel.

$$BCE(t,p) = t \cdot log(p) + (1-t) \cdot log(1-p-\epsilon) \quad (1)$$

where p is the predicted value of from the DNN, t is the corresponding ground truth value, and ϵ (we used 10^{-7}) is

Figure 5. Semantic model augmentation. Green rectangles: ground truth of dynamic objects labeled by a human. Blue rectangles: accepted semantic augmentation. Red rectangles: rejected augmentation.

to avoid numerical error when $p = 1$. Since real data was sampled balancing over micro operational design domains (μODD) as discussed in Section 4.1, the focal loss function [30] was not used. The loss function of the bounding box channels is the l_1 norm in Equation (2).

$$l_1(\boldsymbol{T}, \boldsymbol{P}) = ||\boldsymbol{T} - \boldsymbol{P}||_1 \tag{2}$$

where $\boldsymbol{T} = [x_c, y_c, w/2, h/2]$ is the ground truth bounding box and $\boldsymbol{P}$ is the prediction. The network is further optimized with an INT8 representation using TensorRT[2]. The inference time of an optimized HazardNet model, including post-processing times, is less than $5ms$ using an Nvidia RTX GPU. Therefore, more than 200 frames of $(960 \times 544 \times 3)$ resolution images can be processed in a second.

HazardNet is trained using the procedure in Figure 2. The developed pipeline is for production and all processes are automated without human intervention. Currently, synthetic models are semantically augmented into real images offline. However, we are working on completing a whole training pipeline for online semantic data augmentation in the training stage using a new real-time simulator. When training is complete, we evaluate the model using real road debris test data (see e.g. Section 4.3). We then extract all false positive (FP) and false negative (FN) cases and create a new set of synthetic data imitating the failure cases. Since

[2] https://developer.nvidia.com/tensorrt

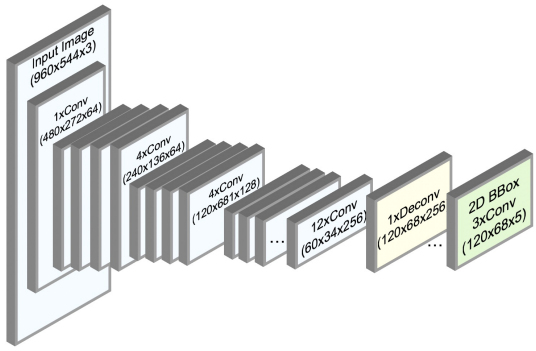

Figure 6. Architecture of HazardNet consisting of 263 layers and based on ResNet [17].

road debris in the real test data do not have matching synthetic models, we generate data based on the 3D size of road debris. 3D information is always available since all real-world objects are labeled using both cameras and lidar. We continue this cyclic workflow until the model performance (based on FP and FN) converges, as shown in Figure 1.

3.5. Performance metrics

Detectors trained to perform few-shot learning tend to output lots of false positive detections [69]. However, in road debris detection it is crucial to minimize false positive detection. One of the most important objectives of the road debris detector is thus the false positive detection rate (FPR): $FPR = FP/(FP + TN)$, where FP is the number of false positive detections and TN is the number of true negative detections. When road debris is detected, the vehicle needs to be stopped and sometimes may be required to brake suddenly. Harsh braking due to a false positive detection can cause severe accidents with neighboring vehicles on highways and thus must be avoided. Therefore, the FPR of HazardNet should be significantly low. Achieving both high recall and low FPR is challenging, so data augmentation using well-balanced, large sets of real data is key. Since the real data we use is collected from numerous operational design domains (ODD) and contains most road entities in a variety of scenarios, HazardNet is able to learn unusual road debris against common road entities. We examine the effects of training and evaluating on real data in the next section.

4. Experiments

4.1. Data collection and labeling

Among publicly available datasets, the nuScenes dataset [5] includes some images of traffic cones and barriers, which are categorized as hazards but common road elements with enough training data. [38] provides open road debris data for stereo vision but we were not able to use

it since their test data includes our synthetic models such as cardboard boxes, tires, logs, traffic cones, wood pallets. Therefore, we collected our own real road debris dataset for evaluation (i.e. the Real training and test datasets described in Sections 4.2 and 4.3, respectively). In addition, we used a well-balanced, in-house real dataset to apply semantic augmentation (i.e. for the Sim training and test datasets described in Sections 4.2 and 4.3).

First, real data without load debris was collected for Sim data. It was collected from various locations, lighting conditions, times of day, and weather conditions. There are four main axes of data collection buckets:

- Road type: highways, freeways, suburban roads, urban roads, rural roads, and dirt roads, indoor/outdoor parking lots.
- Time of day: day, night, dawn/dusk, sunset/sunrise
- Weather: clear, sun, moon, cloud, rain, snow, fog
- Objects: passenger car, emergency vehicles, heavy trucks, bicycle, motor bike, scooter, pedestrians with different level of traffic

Even data was collected considering above guidelines, the majority of data tends to be in clear day time on a straight road. When data is not balanced for each category, underrepresented categories tend to fail. For example, when uncommon construction trucks or snowing weather are quite rare, the DNN model trained on biased data fails on detecting such as trucks. With such difficulties, data should be collected considering the balance of data.

Secondly, the Real road debris were staged on the road in various locations and orientations. Staging was done because collecting real road debris data in-the-wild is extremely difficult, even with access to millions of customers' vehicle data. Collecting road debris data on highways is also dangerous and does not enable staging, though such settings are important to consider. As a result, the data was collected in a limited number of relatively static environments. We placed 30 different road debris on private roads and a vehicle equipped with all sensors (camera, lidar, radar) recorded data starting from about 200 meters away and drove towards the road debris. The staged real road debris are all unseen categories and unseen objects excluded from synthetic models. The unseen road debris includes trash bags, stuffed animals, wood branches, standing barriers, folded cardboard boxes, delineator posts, and more.

For all the real data used in this paper, the ego-vehicle was equipped with not only cameras but also lidar and radar sensors. When human labelers annotate objects and assign corresponding 3D distances from the vehicle, these radar and lidar sensors are fully utilized for more accurate 3D estimation of roads and objects. For every camera frame, human labelers annotated dynamic objects such as vehicles, bikes, and pedestrians, as well as static objects such as lane lines, road marks, road boundaries, traffic signs/lights, and vertical landmarks using all sensors.

Every image was also associated with other metadata such as weather, time/date, refined GPS/IMU signals (latitude, longitude, altitude, and orientation), and road condition (wet, snow, dirt, etc.). As mentioned in Section 3.1, this information is used when augmenting real images with synthetic models.

4.2. Training datasets

The experiments were conducted using three different training datasets, which we refer to as: Sim, Real, and Hybrid(Sim + Real). The Sim training dataset consists of about 5,000 sampled images balanced on various μODD as described in Section 4.1. The 3D synthetic road debris models were used to augment the images with domain randomization and semantic augmentation. The breakdown of each synthetic road debris subclass is shown in Figure 7. The Real training dataset consists of about 10,000 real images and has no overlap with the Sim dataset images. The road debris in each Real dataset image was labeled by humans as described in Section 4.1. The Hybrid training dataset is the combination of the Sim and Real training datasets, and thus has a total of about 15,000 images.

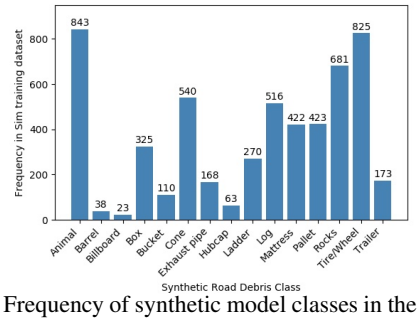

Figure 7. Frequency of synthetic model classes in the Sim training dataset.

4.3. Test datasets

We evaluated HazardNet on two test datasets (not used in training): Real and Sim, which were used for cross-validation of the model trained using few-shot learning and the model with fully supervised learning, respectively. The Real test dataset consists of about 3,000 images, and each image contains zero, one or multiple labeled road debris instances with various poses (see Section 4.1 for details). This real, unseen test dataset is used to evaluate the performance of HazardNet for few-shot learning. The Sim test dataset is generated in the same manner as the training dataset using domain adaptation and semantic augmentation of synthetic models on real images. The test data consists of about 5,000 images. This Sim test data is used to evaluate supervised

learning of HazardNet and represents an upper bound on performance.

4.4. Quantitative evaluation

To quantify the performance of HazardNet trained on the three training datasets, Sim, Real, and Hybrid (Sim+Real) described in Section 4.2, we compute the mean Average Precision(mAP), true positive rate (TPR), false positive rate (FPR), precision, and recall. For mAP, we divide each of the two test datasets into difficulty buckets: small (8-25 pixel height), medium (25-100 pixel height), large (100+ pixel height), and all. When evaluating all the test instances, we weight the mAP corresponding to each bucket proportional to object size. The corresponding weights for small, medium, and large objects are 0.5, 1, and 5. Table 1 shows cross-validation results for the Sim and Real training and tests: (1) sim2sim(97.79%) and real2real(76.36%) are the results of supervised learning with Sim and Real data, respectively. As expected, these supervised learning approaches result in high performance. (2) sim2real (trained on Sim and tested on Real data) is the proposed few-shot learning framework using HazardNet with **35.42%** mAP in the All category. The low performance in the large category (6.16%) is due to the limited number of augmented images with objects close to the ego-vehicle, since we mostly focus on detecting distant road debris. (3) The other cross-validation, real2sim (4.22%), provided the worst results. Since the Real dataset is staged in limited venues (as described in Section 4.1), the network overfit to those environments and was not able to extrapolate to the Sim dataset domain. (4) hybrid2real is a more interesting case. We combined Sim and Real data and tested on Real data, and the Hybrid model (hybrid2real, 80.53%) outperformed the supervised learning approach (real2real, 76.36%). These results imply that the few-shot learned sim2real and hybrid models are more generative than the real2real model with supervised learning.

Figure 8 plots precision-recall curves for all three models evaluated on the Real test dataset. All three cases have substantially high precision (more than 99%) even though the few-shot learned Sim model had lower recall. The confidence threshold value was determined on the validation dataset with the highest F-score, which was 0.3 for the Sim model. In addition, Figure 9 plots the receiver operating characteristic (ROC) curves corresponding to all three models evaluated on the Real test dataset. We observe that all three models have very low false positive detection rates (less than 0.06%), which is essential for road debris detection even though the few-shot learned Sim model was not as good as the Sim or Hybrid models.

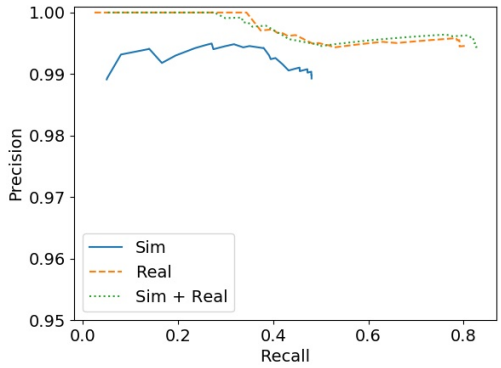

Figure 8. Precision-Recall curves for models evaluated on the Real test dataset.

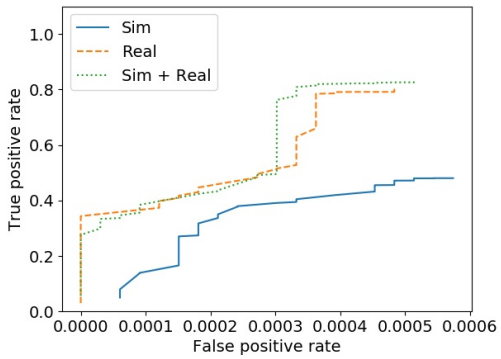

Figure 9. ROC curves for models evaluated on the Real test dataset.

4.5. Qualitative evaluation

To evaluate the efficacy of HazardNet in achieving few-shot learning, we also visualize detection results for real images containing unseen road debris (sim2real case). Figure 10 shows example detections for a variety of road debris classes and environmental conditions. While HazardNet was trained only on simulated road debris, the network is able to detect real debris from both small and large distances from the camera. In Figure 11, we also show examples of false negative detections.

5. Conclusion

We proposed a novel few-shot learning framework and deep learning model, HazardNet, which detects road debris for autonomous driving applications. We show that a small set of synthetic models can guide the DNN to learn to detect unseen real-world road debris. Our method can be extended to other applications where large-scale, real datasets are hard to acquire. During augmentation, the shadow of objects and the direction of lights were not considered and they were left for future research.

mAP(%)	Sim Test Dataset				Real Test Dataset			
Train Dataset	Small	Medium	Large	All	Small	Medium	Large	All
Sim[a]	96.05	99.56	100	97.79	35.11	41.44	6.16	**35.42**[b]
Real[c]	6.89	9.73	0.10	4.22	73.92	83.76	75.27	76.36[d]
Hybrid(Sim + Real)	95.03	99.13	100	97.13	74.79	91.78	80.41	80.53

[a] Sim is real image data with synthetic model augmentation, [b] mAP of few-shot learned HazardNet,
[c] Real is staged real road debris data, [d] mAP of the supervised learned network

Table 1. Mean average precision(mAP) scores using 0.5 IOU for HazardNet trained on Sim, Real, and Sim + Real training datasets. The models were evaluated on both the Sim and Real test datasets and the mAP scores are further divided into small, medium, and large road debris.

Figure 10. Detections of unseen road debris using HazardNet. Ground truth labels are drawn in blue, and HazardNet detections are drawn in red. The network is able to detect debris missed by human annotators (row 3, columns 3 and 4).

Figure 11. False negative detections of unseen road debris using HazardNet. Ground truth labels are drawn in blue, and HazardNet detections are drawn in red. Most of the false negative detections correspond to distant objects.

References

[1] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 2

[2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018. 3

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 2

[4] Alberto Broggi, Michele Buzzoni, Mirko Felisa, and Paolo Zani. Stereo obstacle detection in challenging environments: the viac experience. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1599–1604. IEEE, 2011. 3

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 5

[6] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1841–1850, 2019. 2

[7] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 2

[8] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018. 2

[9] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Zero-shot object detection by hybrid region embedding. *arXiv preprint arXiv:1805.06157*, 2018. 3

[10] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16918–16927, June 2021. 3

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 2

[12] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1301–1310, 2017. 2

[13] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 2

[14] Georgios Georgakis, Arsalan Mousavian, Alexander C Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836*, 2017. 2

[15] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXI*, volume 12366 of *Lecture Notes in Computer Science*, pages 666–681. Springer, 2020. 2

[16] Krishnam Gupta, Syed Ashar Javed, Vineet Gandhi, and K Madhava Krishna. Mergenet: A deep net architecture for small obstacle discovery. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5856–5862. IEEE, 2018. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[18] Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6668–6677, 2019. 2

[19] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[20] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 2

[21] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 749–757, 2020. 2

[22] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 2

[23] Stephen James, Andrew J Davison, and Edward Johns. Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. In *Conference on Robot Learning*, pages 334–343. PMLR, 2017. 2

[24] Rawal Khirodkar, Donghyun Yoo, and Kris Kitani. Domain randomization for scene-specific car detection and pose estimation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1932–1940. IEEE, 2019. 2

[25] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008. 2

[26] Kuan-Hui Lee, German Ros, Jie Li, and Adrien Gaidon. Spigan: Privileged adversarial learning from simulation. *arXiv preprint arXiv:1810.03756*, 2018. 2

[27] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4413–4421, 2019. 2

[28] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019. 2

[29] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8690–8697, 2019. 3

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[31] Roberto Manduchi, Andres Castano, Ashit Talukder, and Larry Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous robots*, 18(1):81–102, 2005. 3

[32] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 2

[33] Sergiu Nedevschi, Radu Danescu, Dan Frentiu, Tiberiu Marita, Florin Oniga, Ciprian Pocol, Thorsten Graf, and Rolf Schmidt. High accuracy stereovision approach for obstacle detection on non-planar roads. *Proc. IEEE INES*, pages 211–216, 2004. 3

[34] Toshiaki Ohgushi, Kenji Horiguchi, and Masao Yamanaka. Road obstacle detection method based on an autoencoder with semantic segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3

[35] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1278–1286, 2015. 2

[36] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018. 2

[37] David Pfeiffer and Uwe Franke. Towards a global optimal multi-layer stixel representation of dense 3d data. In *BMVC*, volume 11, pages 51–1, 2011. 3

[38] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1099–1106. IEEE, 2016. 3, 5

[39] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7249–7255. IEEE, 2019. 2

[40] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3836, 2017. 2

[41] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *Asian Conference on Computer Vision*, pages 547–563. Springer, 2018. 3

[42] Anant Raj, Vinay P Namboodiri, and Tinne Tuytelaars. Subspace alignment based domain adaptation for rcnn detector. *arXiv preprint arXiv:1507.05578*, 2015. 2

[43] Sebastian Ramos, Stefan Gehrig, Peter Pinggera, Uwe Franke, and Carsten Rother. Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1025–1032. IEEE, 2017. 3

[44] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 3, 4

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2

[46] Mahdi Rezaei and Mahsa Shahidi. Zero-shot learning and its applications from autonomous vehicles to covid-19 diagnosis: A review. *Intelligence-based medicine*, page 100005, 2020. 3

[47] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 2

[48] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning*, pages 2152–2161. PMLR, 2015. 2

[49] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 2

[50] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016. 2

[51] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 2

[52] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 84–100, 2018. 2

[53] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015. 2

[54] Lei Sun, Kailun Yang, Xinxin Hu, Weijian Hu, and Kaiwei Wang. Real-time fusion network for rgb-d semantic segmentation incorporating unexpected obstacle detection for road-driving images. *IEEE Robotics and Automation Letters*, 5(4):5558–5565, 2020. 3

[55] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization

for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 2

[56] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 969–977, 2018. 2, 4

[57] Apostolia Tsirikoglou, Joel Kronander, Magnus Wrenninge, and Jonas Unger. Procedural modeling and physically based rendering for synthetic data generation in automotive applications. *arXiv preprint arXiv:1710.06270*, 2017. 2

[58] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 2

[59] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[60] Todd Williamson and Charles Thorpe. Detection of small obstacles at long range using multibaseline stereo. In *Proceedings of the 1998 IEEE International Conference on Intelligent Vehicles*, pages 230–235. Citeseer, 1998. 3

[61] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017. 2

[62] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11724–11733, 2020. 2

[63] Xiaodong Yu and Yiannis Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *European conference on computer vision*, pages 127–140. Springer, 2010. 2

[64] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2100–2110, 2019. 2

[65] Fangyi Zhang, Jürgen Leitner, Zongyuan Ge, Michael Milford, and Peter Corke. Adversarial discriminative sim-to-real transfer of visuo-motor policies. *The International Journal of Robotics Research*, 38(10-11):1229–1245, 2019. 2

[66] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6810–6818, 2018. 2

[67] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Dong Liu, and Tao Mei. Transferring and regularizing prediction for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2020. 2

[68] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(4):998–1010, 2019. 3

[69] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11693–11702, 2020. 3, 5

[70] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 687–696, 2019. 2