CVPR #*****


# TinyHazardSynth: Industrial Grade Realistic Data Augmentation for Autonomous Driving Using 3D Modeling and Depth-Aware Occlusion Modeling

Anonymous CVPR submission

Paper ID *****

## Abstract

*We introduce **TinyHazardSynth**, a depth-aware* synthetic model-auditing *pipeline that can be used to stress-testing autonomous-driving detectors by* controllably *inserting photorealistic small road obstacles—situations rarely captured by public datasets—into real dash-cam videos. NeRF-reconstructed assets are rendered with exact camera intrinsics; precise occlusion handling arise from a two-stage fitting fusion of sparse LiDAR and Depth-Anything depth that converts relative estimates to metric scale. A MaskFormer semantic prior prevents ground and road clipping, and modular fog/shadow layers vary visibility to probe robustness. The fully-automated factory produces thousands of labelled frames and can wrap around any perception stack. On inserted tiny obstacles, the in-house detector achieved a recall of only 29.4%, indicating a high miss rate for rare, small-scale hazards. And, after targeted retraining on our clips, we lifted accuracy by 1.4 pp, demonstrating the pipeline's value for safety-critical model assessment. We leave as future work an investigation of how the same controllable-insertion pipeline adapts to other rare hazard types (e.g., deformable debris or transparent objects) and to public datasets such as nuScenes and KITTI.*

## 1. Introduction

Small static hazards—such as fist-sized rocks, dropped cargo, or road-surface debris—pose a disproportionate safety risk for autonomous vehicles, yet they appear only rarely in public driving datasets. Purely simulator-generated clips like Airsim and Carla [3, 6] help to enlarge coverage, but the gap in texture realism, motion blur and camera ego-motion still limits their usefulness once a model is deployed on real dash-cam streams, which is the data used for real world driving task.

We explore a complementary route: *in-video* insertion of realistic tiny obstacles directly into ordinary dash-cam footage. Our system, **TinyHazardSynth**, (i) captures obstacle assets with NeRF [5], (ii) renders them per frame using exact camera intrinsics, and (iii) achieves pixel-accurate occlusions by fusing sparse LiDAR with Depth-Anything predictions [8] to fit the relationship between relative depth and metric depth. A lightweight *MaskFormer* prior [2] preserves road and vehicle boundaries to help refine further, while optional fog and shadow layers vary visibility conditions. The pipeline produces more than thousands labelled frames and can feed either regular data augmentation *or* targeted robustness checks of existing detectors.

**Contributions**

1. A controllable dash-cam video insertion workflow focused on small road hazards, bridging the realism gap left by simulator-only data.
2. A two-stage fusion of LiDAR and monocular depth that converts relative estimates into metric scale, giving stable occlusions under fast camera motion.
3. An industrial-scale implementation that can synthesize thousands of frames; we intend to share a streamlined code snapshot after acceptance so that others can inspect our pipeline logic and replicate the core steps on their own inputs.

## 2. Literature Review

**Synthetic data for autonomous driving.** Large–scale simulators such as CARLA [3], AIRSIM [6] render traffic scenes with controllable lighting, weather and sensor rigs, enabling low–cost data generation and *in–sim* perturbations for robustness studies. Yet a non-trivial domain gap—texture realism, motion blur and long–tail occlusions— still limits transferable performance on real dash–cam footage. Consequently, there is a gap demand for production pipelines that augment real videos in real world dash camera captured scenarios.

**Object insertion in dash–cam videos.** A straightforward approach would be to composite 2-D cut-outs. However, this breaks down under strong parallax and, to the best of our knowledge, lacks any pipeline capable of handling complex viewpoint geometry or batch-level integration across video frames. NeRF-based approaches [5] allow high-fidelity asset capture. In practice, existing insertion tools assume either a fixed camera or pre-computed dense depth, conditions rarely met by long dash-cam sequences with sparse LiDAR. Few papers tackle *small* ground obstacles, whose limited pixel footprint amplifies any depth or mask error.

**Our position.** We bridge these gaps by (i) inserting NeRF-reconstructed tiny hazards into real videos using a LiDAR + Depth-Anything metric fusion, (ii) preserving road semantics via *MaskFormer* thereby further refining synthesis, and (iii) modulating visibility with controllable shadows—with additional diffusion-based weather effects left as future work—so that quantitative audits remain entirely within the real-image domain. The resulting clips both enrich training data *and* serve as a fine-grained, dash-cam-authentic testbed for obstacle-detection auditing.

# 3. Methodology

## 3.1. NeRF-based 3D Reconstruction for Obstacle Cutout Images Gathering

To generate accurate visual representations of small ground obstacles, we first employ Neural Radiance Fields (NeRF) [5] to reconstruct detailed 3D models in the `.ply` format from multiple views. Utilizing intrinsic and extrinsic camera parameters at various timestamps, we render obstacle images from precise virtual camera positions. This approach allows us to generate continuous and realistic obstacle representations for subsequent insertion into dash cam videos. In practice, obstacle cutout images can be cut out with other methods or even manually.

## 3.2. Accurate Depth Estimation and Occlusion Handling

A critical challenge in synthesizing realistic dash cam data arises when dynamic objects, such as animals, suddenly emerge from behind obstacles, creating complex occlusion relationships. Traditional depth estimation models, including Depth Anything V2 [8], output relative depth maps (0–255 values) rather than absolute distances, which is not compatible with LiDAR data. To bridge this gap, we incorporate LiDAR point clouds as ground truth data to assist depth conversion. However, projecting point clouds onto images typically results in "edge outlining effects," where the projected cloud is larger than the actual object in the image, producing numerous discrete, noisy values at object boundaries.

To address this, we introduce an innovative two-stage curve fitting approach using `Scipy` [7] optimization. Initially, we apply a preliminary fit to the LiDAR-based depth data and Depth Anything V2's relative depth output, defined by a function of the form: This first fitting pass is specifically utilized to filter outliers by removing points exceeding a defined margin threshold from the initial fit. Subsequently, we perform a second fit using the filtered dataset to accurately determine the final mapping parameters between relative and absolute depth, to reduce noise and enhancing occlusion realism.

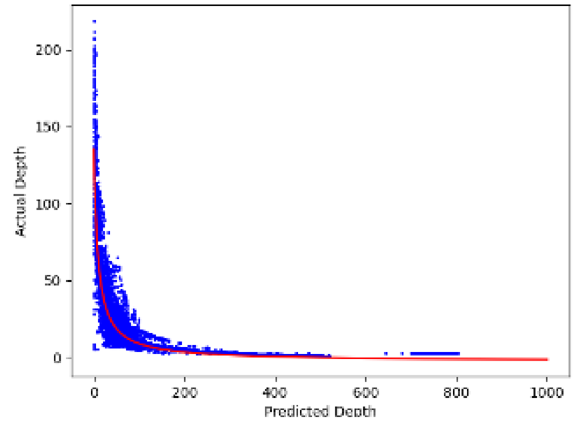


Figure 1. Fitting curve between predicted (relative) and actual depth. After outlier filtering, the final curve (in the form of y = a / (x + b) + c) aligns well with ground truth.

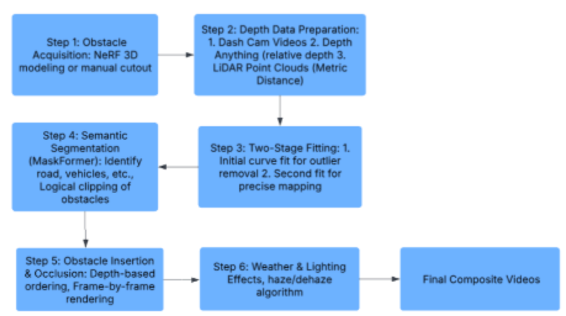


Figure 2. the whole flowchart of the pipeline

## 3.3. Semantic Segmentation for Clip Handling

Another practical challenge encountered is the "clipping" effect due to camera perspective, where objects appear partially embedded within road surfaces or vehicle structures, resulting in unrealistic visualization. This arises from perspective distortion where parallel lines converge in camera projection. To mitigate this issue, we employ the MaskFormer [2] semantic segmentation model (trained on the ADE20K dataset). MaskFormer enables the precise identification of road and vehicle hood areas, allowing logic-based controls to manage object visibility effectively. Specifically, we enforce rules to hide or show small objects visually positioned beneath vehicle hoods or incorrectly intersecting road surfaces, thus maintaining the semantic integrity and realism of synthesized videos.

| | Accuracy | Recall |
|---|---|---|
| Baseline (without synthetic clips) | 86.9 % | 29.4 % |
| + TinyHazardSynth clips | **88.3 %** | **36.6 %** |

Table 1. Internal obstacle-detection results. Synthetic tiny-hazard clips lift accuracy by 1.4pp and recall by 7–8pp (24.5 % relative).

## 4. Experimental Validation

The goal of this section is to examine whether the clips produced by our pipeline translate into measurable performance gains on real-world autonomous-driving tasks. While a full public benchmark is in preparation, we report the first numbers from an internal test bed provided by an industry partner.

The experiment uses proprietary dash-cam footage into which we insert *metal plates*—a canonical small, rigid hazard that is hard to detect because of its thin profile and low reflectivity. After one round of fine-tuning on the augmented set, the detector achieves the improvements listed in Table 1. Although the study covers a single class and a limited geography, the gain suggests that depth- and shadow-aware synthesis reveals failure cases otherwise under-represented in natural data and thus serves as a lightweight stress test of model robustness.

Figure 3. Frame pair before (up) and after (bottom) inserting a partially obscured fallen barricade at the hood–road junction.

Figure 4. Another example with additional synthetic shadowing to match scene illumination.

Future work may include extending the pipeline to additional object classes and investigating public release of a small validation subset.

## 5. Conclusion and Future Work

We have presented a depth-aware pipeline that inserts photorealistic tiny obstacles into dash-cam videos, combining NeRF asset capture with a LiDAR + Depth-Anything fitting scheme and a *MaskFormer* prior for clean semantic boundaries. On an internal test set the synthetic clips improved iron-plate detection by 1.4 pp in accuracy and about 25% in recall, indicating that such rare-event enrichment can benefit production models.

**Future directions**

1. **Wider sensor settings.**The modular design makes it feasible to plug in alternative depth sources—e.g. stereo or monocular SfM—but the impact of domain shift remains to be quantified.
2. **Richer environment effects.**Adding physics-based volumetric fog or lightweight generative post-processing could further close the realism gap under adverse weather.
3. **Broader evaluation.**Extending the study to deformable or transparent objects, and to public datasets such as nuScenes [1] and KITTI [4], would clarify how well the approach generalizes beyond metal plates.

All experiments in this paper rely on proprietary video due to corporate restrictions; public benchmarking is left to future collaborative work.

## References

[1] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 3

[2] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. arXiv:2107.06278. 1, 2

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, pages 1–16, 2017. arXiv:1711.03938. 1

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 3

[5] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421. Springer, Cham, 2020. 1, 2

[6] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635. Springer, Cham, 2018. 1

[7] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, ..., and SciPy 1.0 Contributors. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, 2020. 2

[8] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *NeurIPS*, 2024. arXiv:2406.09414. 1, 2