SynDRA: Synthetic Dataset for Railway Applications

Gianluca D'Amico, Federico Nesti, Giulio Rossolini, Mauro Marinoni, Salvatore Sabina, Giorgio Buttazzo
Scuola Superiore Sant'Anna, Italy
name.surname@santannapisa.it

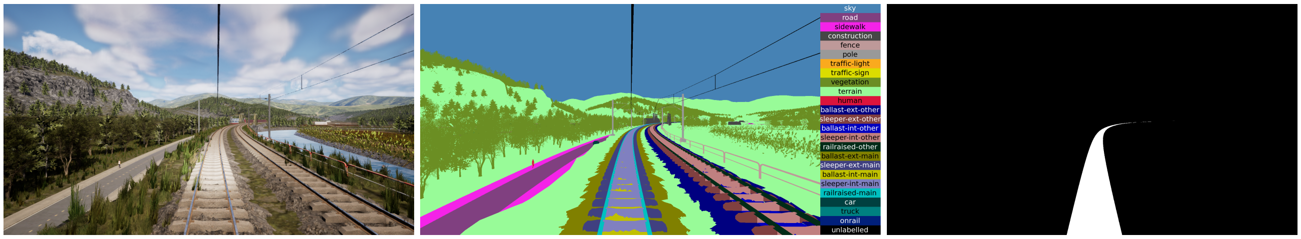


Figure 1. From left to right: sample RGB image from the SynDRA dataset, its corresponding semantic segmentation ground truth, and the extracted binary mask for ego track discrimination task.

Abstract

The use of deep learning techniques in railway environments faces significant obstacles, especially for computer vision tasks. Such obstacles are mainly due to the inherent safety concerns required for installing the proper equipment on a train and the substantial effort required to precisely annotate large datasets, especially for segmentation tasks. Public datasets of real-world images are quite scarce and suffer from severe limitations, such as coarse manual annotation or narrow range of scenarios. In addition, real-world datasets often do not contain scenes that represent critical situations. To address such limitations, this paper introduces SynDRA, a synthetic dataset of photo-realistic images generated using a railway simulator built on Unreal Engine 5. SynDRA offers precise pixel-level annotations across diverse scenarios, thereby facilitating more effective testing and training of deep learning models for semantic segmentation tasks in railway settings. The advantages of the proposed dataset are validated through a series of experiments that highlight the potential of SynDRA to enhance the performance of deep learning models in scenarios where real-world annotated data is scarce. The dataset is publicly available at the following link: https://syndra.retis.santannapisa.it

1. Introduction

In recent years, the railway sector has seen an increasing interest in computer vision tasks, driven by the need for automating a larger number of train operations [31], [2]. Despite such a growing interest, significant challenges remain to be solved to deploy deep learning models in railway environments. One of the primary obstacles is due to the safety concerns and permissions required to mount the proper equipment on the train. Another difficulty is the substantial effort required to annotate large datasets, especially for segmentation tasks.

The consequence of such difficulties is a scarcity of publicly available datasets in the railway domain. In fact, most of the datasets referenced in recent works [26] are not publicly accessible, which significantly slows down research and innovation. While the autonomous driving domain benefits from a variety of computer vision datasets [35], the railway domain is limited to only a few public datasets, such as RailSem19 [36], for coarse semantic segmentation, and OSDaR23 [30], for object and rail detection. These datasets have their own limitations, such as a coarse manual annotations or a limited scenario diversity.

Given these issues, simulators and synthetic datasets present a promising alternative. Although synthetic data is still far from perfect due to the distributional shift compared to real-world images, and challenges such as memory and computational issues when representing high-fidelity meshes, recent advances in graphic engines have significantly enhanced the quality of synthetic data. This progress makes them a viable solution for generating large amounts of data across diverse scenarios and operating conditions.

In particular, the use of dedicated simulators presents several advantages with respect to an ad-hoc data acquisition campaign: (i) high flexibility to replicate a variety of real-world scenarios with high fidelity; (ii) automatic and fine-grained annotation of each generated image for specific vision tasks; (iii) possibility of generating critical corner-case situations, which are difficult to replicate in a real environment, such as unauthorized pedestrian crossings or the presence of various obstacles on the track (e.g., animals, rocks, trees); and (iv) possibility of changing weather and lighting conditions, further enhancing the variety of the generated data. In addition, the data generated by graphical simulators can be used for proof-of-concept validation in conjunction with real-world data to enhance model performance, as demonstrated in previous studies [28], [33]. Despite these advantages, there are currently few simulators capable of generating synthetic data in railway environments [11]. To fill this gap, this paper introduces SynDRA, a public synthetic dataset specifically designed for computer vision tasks in railway applications.

Generated using a railway simulator developed with Unreal Engine 5 (UE5), SynDRA features photo-realistic images and corresponding ground-truth data for segmentation tasks within railway environments, as illustrated in Figure 1. The dataset is designed to be updated with new scenarios and operating conditions. Experimental results are also presented to assess how synthetic data can enhance the performance of vision models in these tasks. In summary, this paper provides the following contributions:

- It presents SynDRA, a novel synthetic dataset for visual segmentation tasks in railway applications. The dataset is designed to be easily extended with new environments or tasks, such as depth estimation and object detection, as well as additional data streams from different sensors, as LiDARs and infrared cameras.
- It presents a set of experiments that demonstrate the benefits of using synthetic images in combination with real-world samples for semantic segmentation and ego track discrimination tasks. The results indicate that incorporating synthetic data in the training phase enhances the model performance, especially when real-world annotated data are scarce.

The remainder of this paper is organized as follows: Section 2 analyzes the related literature; Section 3 presents the characteristics of the dataset and details about data generation; Section 4 shows the experimental results; and Section 5 states the conclusions and possible future directions.

2. Related Work

This section reviews the existing open datasets containing real-world or synthetic images of railway environments.

Real-world datasets. As mentioned in Section 1, only a few datasets for computer vision tasks in railway environments are publicly available. One of the most notable is RailSem19 [36], which includes 8,500 images from both railway and tram scenes taken from different countries, in all seasons, and under varying weather conditions. The dataset was constructed by accurately sampling 530 video sequences that were captured using a broad range of camera models and mounting positions. The major strength of this dataset is its great image variability, which is beneficial for training robust neural models. On the other hand, since images are filtered from real-world sequences, they lack temporal coherence. In addition, the semi-automated procedure used to annotate the segmentation masks of the images is not always precise and consistent.

Another key dataset is OSDaR23 [30], which includes 45 sequences, for a total number of 1,534 annotated multi frames, each composed of nine camera frames, one radar frame and six LiDAR frames. Annotations include diverse types of labels such as bounding boxes and rail polylines. Unfortunately, many of the subsequences are acquired while the train is stationary, leading to multiple frames with little or no variation. Additionally, the images in this dataset are not labeled for semantic segmentation tasks.

Other datasets provide valuable data for specific tasks, such as railway traffic signal detection and recognition [13, 16], place recognition [29], and pedestrian [32] or rail detection [19], but RailSem19 and OSDaR23 stand out for their breadth and depth, making them crucial for developing and testing visual perception tasks in railway environments.

Synthetic datasets. Given the limitations of current real-world datasets and the challenges outlined in Section 1 regarding extensive acquisition campaigns, the research community has turned to synthetic datasets to validate algorithms in fully controlled environments. These datasets can be divided into two categories: (i) those that modify real-world images using generative AI techniques and (ii) those generated by simulating virtual worlds.

Within the first category, Decker et al. [9] used Generative Adversarial Networks (GANs) to enhance RailSem19 by adding new images with different lighting and weather conditions. RailSet [38] is another dataset specifically designed for detecting anomalies in railway environments such as holes and rail discontinuities. To address the lack of publicly available anomaly samples, the authors used a deep learning algorithm known as StyleMapGAN [14] to synthetically generate images of abnormal scenes.

Similarly, Chen et al. [6] employed ChatGPT [22] and diffusion models [27] to generate synthetic data for detecting foreign objects on railroad transmission lines.

Finally, Li et al. [17] and Brucker et al. [5] used similar approaches to randomly combine images of various obstacles with traffic scenes taken from the driver's perspective to create a synthetic training dataset for obstacle detection.

The latter category requires the creation of virtual worlds and a simulator that generates sensory data, such as images or point clouds, which closely resemble real-world data.

Examples of simulation environments of this type are CARLA [10], widely used in the automotive domain, and Gazebo [15], widely used for robotics.

In the railway domain, a similar approach was followed by de Gordoa et al. [8], who developed a synthetic image dataset by extending CARLA (built on UE4), to generate railway imagery under diverse lighting and weather conditions and labels for tasks such as semantic segmentation, 3D bounding boxes, and odometry. D'Amico et al. [11] introduced TrainSim, a private simulation framework specifically developed for railway environments using UE4, capable of generating images and point clouds for tasks such as train localization and ego track discrimination. While TrainSim utilized an earlier version of Unreal Engine, our new framework leverages the latest version of UE5, unlocking a higher level of graphical fidelity, performance, and toolset capabilities. Notably, our framework leverages real-world databases, such as OpenStreetMap (OSM), to accurately reconstruct real-world environments with their inherent complexity. Additionally, it incorporates dynamic elements, including moving vehicles and pedestrians, to create realistic and highly detailed simulation scenarios that closely mirror real-world conditions.

Other synthetic datasets generated from simulated environments are RailEnV-PASMVS [4], which features images of track geometry including rail components like the rail profile, e-clip fastener, insulator pad, concrete sleeper, and ballast, and SARD [21], which focuses exclusively on point clouds of railway signals. Such datasets, however, fall outside the scope of this work as they target different tasks.

All these initiatives underline the growing importance of synthetic data in overcoming the various challenges associated with real-world data acquisition. By allowing a full control of the railway environment and the generated scenarios, synthetic datasets play a crucial role for advancing railway perception technologies. For a systematic review of public datasets for railway applications, please refer to the work by Pappaterra et al. [25].

3. Dataset generation and statistics

This section describes the procedure for generating the SynDRA dataset: Section 3.1 illustrates the simulation and dataset generation framework, Section 3.2 presents the virtual scenarios used, and Section 3.3 details the labeling policy adopted for semantic segmentation and introduces some dataset statistics. Some ethical considerations related to real-world assets gathered from different databases are discussed in the supplementary material.

3.1. Simulation Framework

The simulator used to generate SynDRA is built on UE5, leveraging its advanced rendering features to obtain photo-realistic images. The generation process is illustrated in Figure 2 and involves three key stages: Information Collection, Scenario Development, and Simulation Execution.

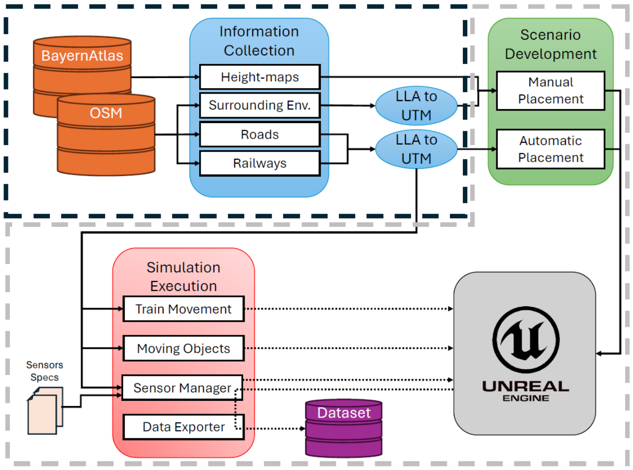


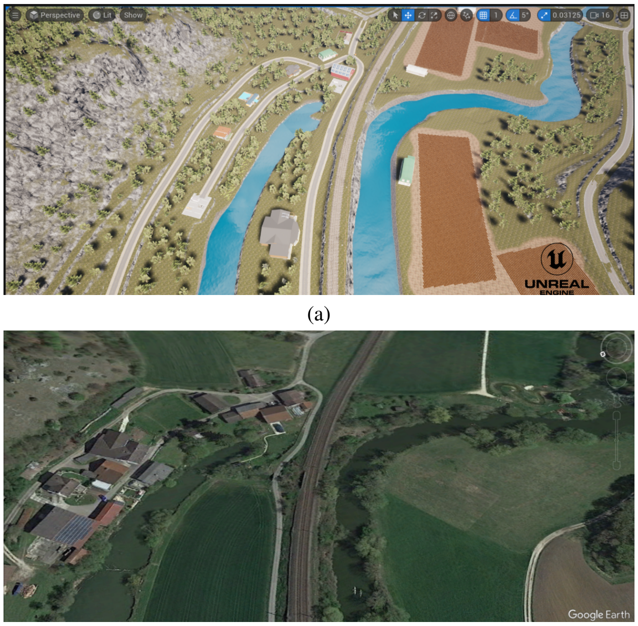
Figure 2. Block diagram of the dataset generation process.

Information Collection. Accurate emulation of real-world railway environments is achieved by collecting information on railway routes, road networks, surrounding landmarks, and height maps. Such data are used to generate the virtual scenarios that closely reflect the complexity and diversity of actual railway settings. While the simulator can accept data from any source, OpenStreetMap [23] (OSM), and BayernAtlas [1] are used. Additionally, highly realistic 3D meshes of railway infrastructure components are obtained through two UE5 plugins, TrainTemplate [37] and RailwaySystem [20], while other environmental assets are sourced from the Epic Games Megascan Marketplace [12] and various online marketplaces [1].

Scenario Development. The geo-referenced components gathered from OSM are placed in the virtual landscape automatically after a local coordinate conversion: OSM provides information on railway paths, road networks, surrounding landscapes and land use for specific areas (such as farmlands, buildings, forests, and more); the heightmaps from BayernAtlas are integrated into the virtual world to replicate the terrain's elevation accurately. Each scenario is designed to cover a 4km-by-4km area, consistent with the default landscape dimensions of UE5. Figure 3 compares the simulated landscape with its real-

[1] Primarily: https://www.turbosquid.com/ and https://sketchfab.com

world counterpart viewed from Google Earth [2].



(a)

(b)

Figure 3. Comparison between a SynDRA virtual scenario including roads, railways, and farmlands derived from OSM (a) and its real-world counterpart captured from Google Earth (b).

Simulation Execution. The framework simulates the movement of trains, vehicles, and pedestrians within the environment. Train movement is governed by a first-order dynamic system that incorporates maximum speed limits obtained from the OSM database. Virtual sensors are strategically positioned on the front of the train to capture data from the train's perspective. Specifically, the SynDRA dataset employs a stereo camera system, where each camera has a resolution of 1080×1920 pixels and a 90-degree horizontal field of view. The cameras are aligned with the ground and positioned 3.50 meters above the ground, with one camera offset 30 cm to the left and the other 30 cm to the right of the rail's center-line. The Sensor Handler manages the data acquisition process, while the Data Export module manages the storage and organization of the collected data for subsequent analysis through visual algorithms.

3.2. Scenarios

The SynDRA dataset currently includes four distinct 4km-by-4km scenarios, each carefully designed to represent different railway environments:

- Scenario 1: it features a field landscape with roads running parallel to the railway. The two-way railway includes multiple bridges crossing over roads and rivers, providing a varied terrain with both natural and man-made elements.
- Scenario 2: It is set along a two-way railway passing through a tunnel, surrounded primarily by farmlands and natural areas, with a small cluster of buildings near the tunnel's exit.
- Scenario 3: It includes a single rail track passing through a dense forest and an urban area. The rail track runs alongside a complex road network with multiple crossings, bordered by various buildings and parking areas on the right side.
- Scenario 4: It includes a two-way railroad starting from a urban area with a station expanding into three tracks before merging back into two. The route passes through forest and farmland areas.

Each scenario is enhanced with distinct textures and materials for the terrain, as well as various meshes for trees, rocks, and surrounding vegetation, providing a realistic and immersive environment.

The simulation framework can also generate various lighting conditions and weather effects. The dataset is primarily organized by visibility condition: (i) High-Visibility (HV) conditions, including scenarios with sunny weather during morning, afternoon, and evening light; and (ii) Low-Visibility (LV) conditions, includes scenarios under rain and fog, all captured with the same lighting condition.

Each scenario is traversed in both directions, capturing image sequences from different perspectives. For every scenario, the dataset provides 2 sequences (forward and reverse), repeated across 3 HV conditions and 2 LV conditions. These sequences are recorded from both left and right cameras, resulting in 12 HV sequences and 8 LV sequences per scenario, totaling 48 HV sequences and 32 LV sequences across all scenarios.

3.3. Labels Policy and Dataset Statistics

In SynDRA, class labels are listed in Figure 1 and include road, sidewalk, human, etc., as well as others railway-specific labels for a detailed and nuanced segmentation of the scene, as *ballast-ext-other*, *sleeper-ext-other*, *ballast-int-other*, *sleeper-int-other*, *railraised-other*, *ballast-ext-main*, *sleeper-ext-main*, *ballast-int-main*, *sleeper-int-main*, and *railraised-main*. The terms "main" and "other" are used to differentiate between the ego track (i.e., the railway on which the train with the camera is running) and other nearby or parallel tracks where other trains may be operating.

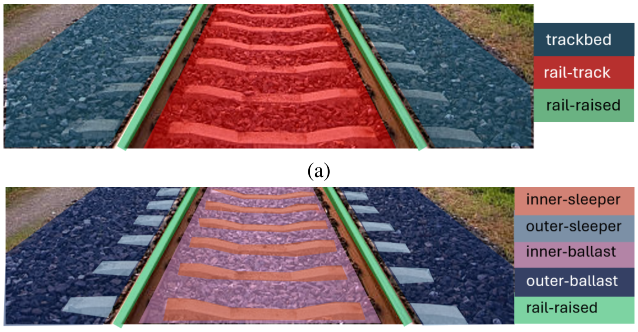
To ensure compatibility with existing datasets, class labels are aligned with those used in RailSem19 [36] whenever possible. As Figure 4 shows, while RailSem19 uses the *trackbed* and *rail-track* classes for both the ballast and the sleepers, in SynDRA these classes are divided into specific portions: the terms "int" and "ext" refer to the part between

[2] https://www.google.it/intl/it/earth/index.html

and outside the rails, respectively.

For the classes that are unique in SynDRA, a mapping for aggregating these classes into the broader categories defined in RailSem19 is straightforward to obtain. This approach allows SynDRA to be used alongside RailSem19, facilitating transfer learning and comparative studies.

This choice of class labels ensures both flexibility and domain-specific detail. In fact, by dividing the canonical RailSem19 railway-related classes into several sub-classes, a neural network may accurately discriminate specific portions of the track, and the role of the track itself in the current train operation. This distinction is useful for safety-critical applications, where precise localization of the ego-track is required, but it is highly adaptable to general-purpose computer vision, where more specific labels might not be crucial.



(a)

(b)

Figure 4. Correspondence between different classes in RailSem19 (a) and SynDRA (b). The mapping from SynDRA to RailSem19 combines *outer-sleeper* and *outer-ballast* into *trackbed*, *inner-sleeper* and *inner-ballast* into *rail-track*.

Semantic labels were generated using a specific feature of UE5, known as the custom stencil. In this process, each virtual object is assigned a unique 8-digit value, and a corresponding UE5 post-processing material is applied to capture this information during camera acquisition. UE5 ensures that labels are densely populated and automatically assigned, representing the visual elements within the scene.

A possible hurdle to the use of synthetic labels is the extremely fine granularity of the segmentation mask. In fact, overly fine-grained labeling, such as portions of sky through tree branches, or the difference between inner and outer ballast, may exceed the discriminating capacity of some models and confuse neural networks, leading to misclassification and reduction of segmentation accuracy. For similar reasons, differentiating between the ego-track and other tracks can introduce additional complexity, particularly in scenes with multiple tracks in close proximity.

Dataset Statistics. As discussed in Section 3.2, SynDRA encompasses four distinct scenarios, each tailored with specific objects during its design. Figure 5a illustrates the class distribution within each scenario, measured by the number of pixels. To facilitate a comparison with state-of-the-art real-world segmentation datasets, Figure 5b highlights the proportional differences between SynDRA and RailSem19. It is worth noting that SynDRA includes additional classes, requiring a specific mapping to the RailSem19 classes. Note that all RailSem19 statistics were calculated excluding images containing tram scenes, as they do not align with our railway environments or our railway-specific classes.

The statistics for the SynDRA dataset focus on a single sequence from each scenario. Specifically, we report the pixel counts from the left camera during a sunny, morning, forward-running sequence, as there is no difference between HV and LV segmentation masks, and distributions from the left and right cameras and forward and backward sequences are comparable.

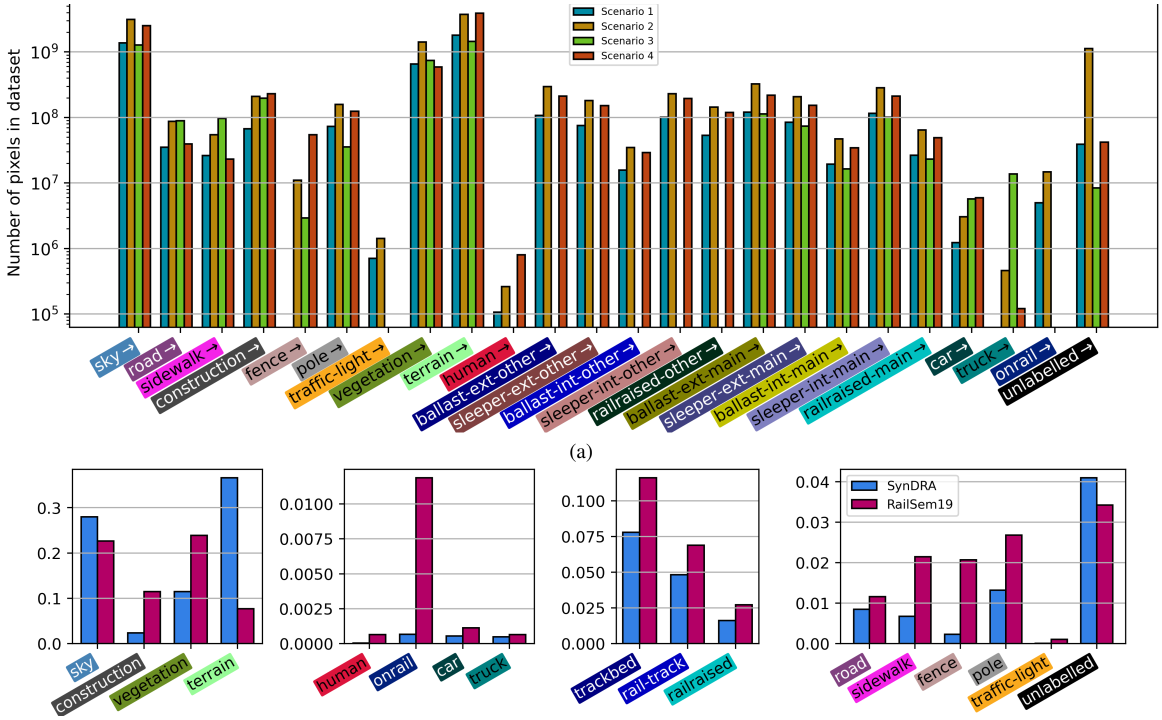
4. Experimental Results

This section presents the results of the experiments carried out on semantic segmentation (Section 4.1) and ego-track discrimination (Section 4.2). Such results showcase how a synthetic railway dataset might be used to increase the final performance, especially when real-world data are scarce. All the experiments have been performed on an NVidia Tesla A100 GPU with 40 GBs.

4.1. Semantic Segmentation

Semantic segmentation is one of the most widely applied tasks in scene understanding for driving applications [7]. Recent works have extended this task to railway scenarios, using RailSem19 [36], which is the largest railway dataset for semantic segmentation (see Section 2).

Implementation details. To evaluate the benefits of introducing synthetic samples of SynDRA in the training phase, we conducted several fine-tuning experiments on two neural network models, BisenetX39 [34] and DDRNet23Slim [24], pretrained on Cityscapes [7] to avoid learning the feature extractor from scratch. The fine-tuning was done using the RailSem19 dataset, filtering out all the images containing trams (which are not in the distribution of interest) and those where the ego-track was not recognizable due to the viewing angle, occlusions, or other factors. The filtered dataset includes 6572 images, where 5000 samples were used for testing, while a variable number of samples (i.e., 15, 25, up to 250 samples) was used for training to replicate situations where real-world training data is limited. Then, we trained the models with and without adding extra synthetic data from SynDRA. When adding SynDRA data, we used 50 samples from the morning sequence of



(a)

(b)

Figure 5. SynDRA dataset statistics. (a) Pixel count for each class in each scenario. The statistics for each scenario are taken by the left camera, morning, sunny, forward-running sequence; (b) Comparison between SynDRA and RailSem19 for the proportion of the number of the pixels of each class. Please note that due to the definition of additional classes in SynDRA, multiple classes have been mapped to a single class (*Railraised*, *Rail-track*, *Trackbed*); the scales are adapted to the proportion of each set of classes showed to allow a clearer visualization of the numerical values.

each scenario, for a total number of 200 synthetic samples. Note that a higher number of samples per scenario lead to many repetitive frames, thus increasing the risk of overfitting with SynDRA distribution (see suplementary material).

The fine-tuning process consisted of 1000 backward updates, where all the models parameters were updated. The Adam optimizer was used with a learning rate of 10^{-3} and a batch size of 16. Data augmentation was applied to both RailSem and SynDRA training images, using random crop (512,512), random horizontal flip, and jitter. The standard pixel-wise cross entropy was used as a loss function.

Regarding the addressed classes, potentially all those available in SynDRA could be used (see Figure 5b). However, to ensure consistent mapping between RailSem19 and SynDRA, we selected a subset of SynDRA classes, focusing on the most essential for railway applications, as those reported on the x-axis of the plot in Figure 6. Specifically, “Railway” includes the entire area within the tracks, comprising railraised and internal track objects (ballast and sleepers). “Trackbed” refers to the area outside the tracks, including the external part of ballast and sleepers. Since the models were originally pre-trained on Cityscapes [7], the last linear layers were modified to fit the addressed number of classes in railway scenarios.

Results. The results on test data after fine-tuning are presented in Figure 6 for both BisenetX39 (top) and DDRNet23Slim (bottom). As shown in the left and center bar plots, when limited labeled samples from RailSem are available (e.g., 15 and 25), the introduction of 200 samples from SynDRA yields significant benefits across all the addressed classes. Conversely, with a larger number of real-labeled samples (e.g., 50), the results indicate that additional synthetic samples have less impact, as the class-IoUs are approximately the same with and without SynDRA samples. These conclusions are consistent for both the tested models, highlighting how the use of SynDRA is beneficial when a small number of real training samples are available.

In the supplementary material, we provide additional results for different amounts of RailSem samples, and further experiments to understand the impact of using different numbers of samples from SynDRA.

For completeness, Figure 7 shows segmentation outputs obtained with DDRNet using 25 RailSem training samples with (d) and without (c) the 200 SynDRA samples. As high-

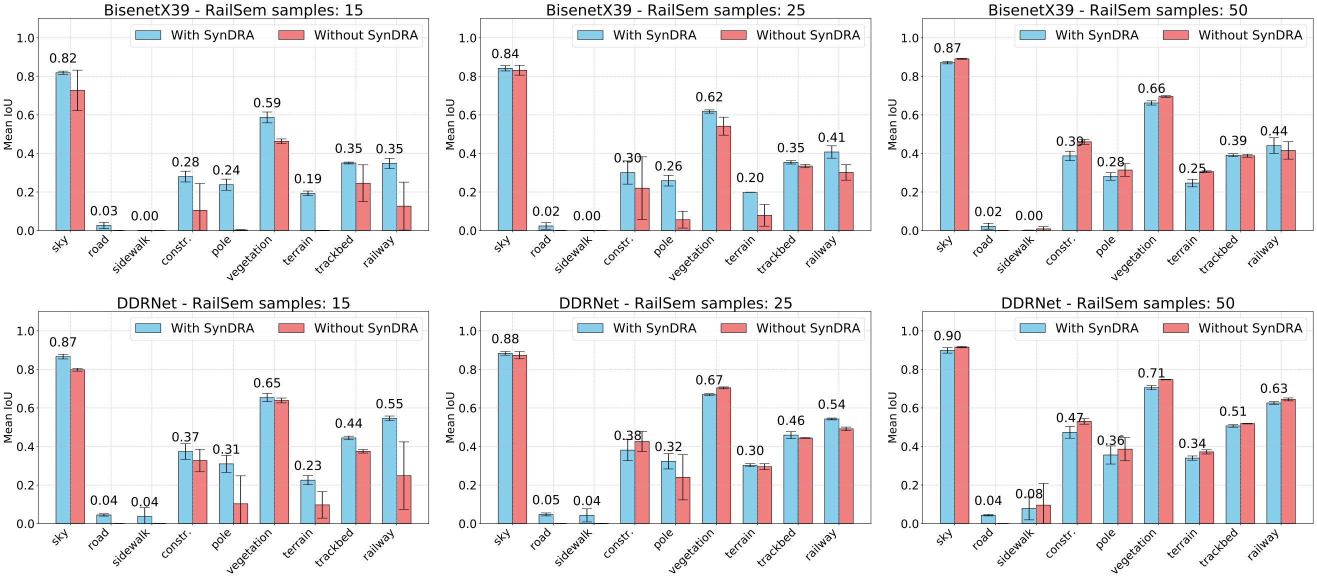


Figure 6. Finetuning results for Bisenet (top) and DDRNet (bottom) using different amounts of RailSem samples (15, 25, 50) with (blue) and without (red) SynDRA samples. The tests were conducted over 3 runs, with the variation shown by error bars. In these runs, the selection of samples from SynDRA and RailSem, as well as the seeds for applying data augmentation transformations, were randomized.

lighted in red, there is a notable improvement in the prediction of poles and the railway area, demonstrating the benefits of introducing synthetic data in the training phase when labeled real-world data is scarce. Note that, as shown in the ground truth annotations (b), the labels of RailSem are sometimes noisy and particularly coarse, with some railway objects being annotated quite ambiguously.

4.2. Ego-Track Discrimination

Ego-track discrimination is a segmentation-based task aiming at detecting the portion of the image between the two rails where the train is running. As a sub-problem of rail detection, it has been studied and researched in previous works [18], as it is an important step in the pipeline for the more complex task of obstacle detection on the track [26].

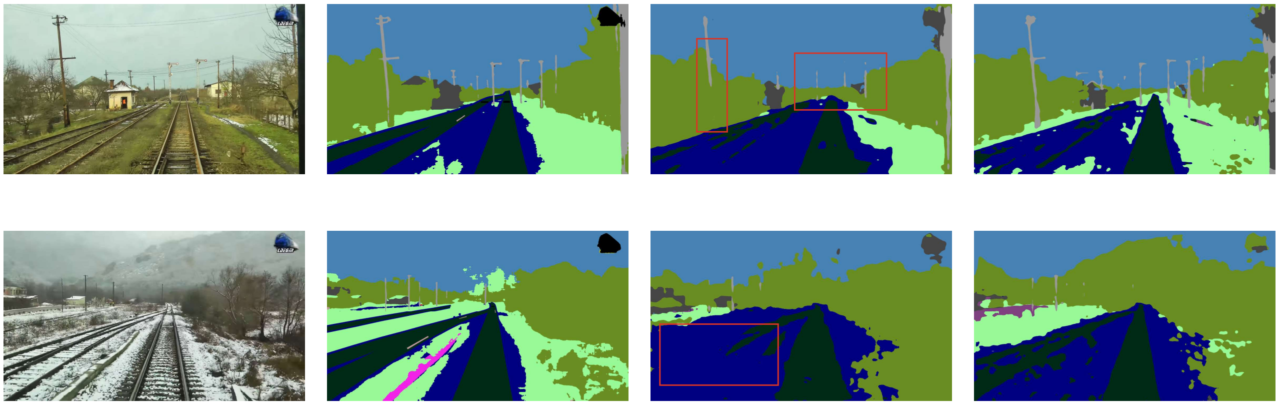
In this case too, the experiments were designed to show that using an additional synthetic dataset improves the performance of a model trained on real-world data, by first varying the number of RailSem19 samples while fixing the SynDRA portion, and then doing the opposite. The objective is to characterize how additional synthetic images affect the final (real-world) performance, especially when considering scarce availability of real-world images.

Implementation details. We used the DDRNet23Slim [24] architecture, pretrained on the Cityscapes dataset. The final layer was replaced with a linear layer with 2 output channels to suit the binary segmentation task. Similarly to the semantic segmentation experiments, we selected the RailSem19 dataset for training, filtering out not relevant images (see previous settings).

The final evaluations were performed on the OSDaR23 dataset, selecting all the images from the sequences where the train is moving, and only one from those where the train is still, for a total of 465 samples. As mentioned in Section 2, since OSDaR23 was built from short sequences, it is not really suitable for large-scale training (unlike RailSem19 which varies wildly in distribution); however, it closely resembles a practical industrial computer vision application in railway context, which is our specific interest. In both cases, the binary mask of the image region occupied by the track of the marching train was not readily available, but could be obtained with a simple algorithm starting from the annotated polyline of couples of rails. The mask was created by running a polygon filling algorithm available in OpenCV [3]. To discriminate the ego-track from the others, the most centered one was selected; however, manual checking was necessary on the RailSem19 dataset for the extreme variations of the viewing angle. Additional details on the labeling process are in the supplementary material.

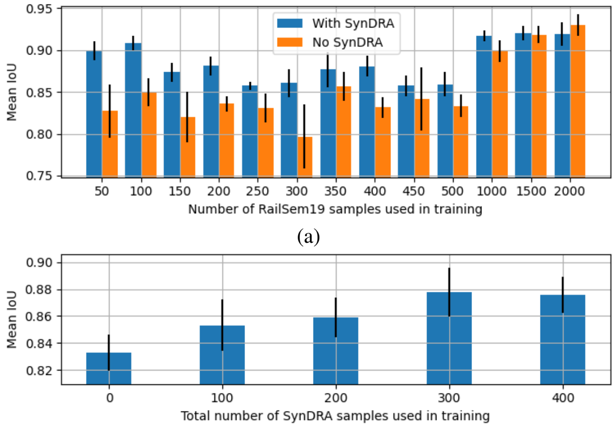
From each experiment we report the average maximum performance of 5 different 30-epochs runs, with starting learning rate 1e-3, optimized with Adam.

Results. Figure 8a summarizes the results, comparing the performance in terms of mIoU for the same architecture trained on increasing number of RailSem19 samples and a fixed number of SynDRA samples (50 for each sequence, 200 in total). Conversely, Figure 8b depicts the



(a) Railsem tested images (b) Ground Truth (c) Outputs w/o SynDRA samples (d) Outputs w/ SynDRA samples

Figure 7. Illustration from the tested samples of RailSem. The output predictions were obtained using a DDRNet trained with (d) and without (c) SynDRA samples. Areas of greatest interest are highlighted in red to show the improvements.



(a)

(b)

Figure 8. (a) Results for ego-track discrimination as a function of the number of RailSem19 samples, fixing the number of SynDRA samples to a total of 200. (b) Results for ego-track discrimination, fixing the number of RailSem19 samples to 500 and varying the total number of SynDRA used. The error bars depicts the standard deviation among the 5 randomized runs.

same metric when fixing the number of RailSem19 samples to 500, while varying the number of SynDRA samples used for training. Additional metrics are reported in the supplementary material. The performance on ego-track discrimination is non-negligibly improved by the introduction of the SynDRA samples, at least when less than 2000 samples from the real world are available. Notably, the use of SynDRA images not only improves performance but also helps stability, as the standard deviation decreases.

It is also important to note that increasing the number of RailSem19 samples is not always correlated with performance improvement, as there is a clear fluctuation in Figure 8a. This might be due to the fact that RailSem19 is a highly-varied dataset that includes all sorts of images. Hence, when a small subset is used, it is possible that the influence of an additional "bad" sample reflects in a drop in performance. Conversely, SynDRA images are much more similar to OSDaR23 and, up to a certain number of samples, help RailSem19 with the domain shift.

5. Conclusion

To address the limited availability of railway vision datasets and simulated environments, this work introduces SynDRA, a synthetic dataset generated using Unreal Engine 5. SynDRA provides photo-realistic images with precise pixel-level annotations. The design of these environments was carefully planned and supported by analysis, while the generation pipeline allows for easy extensions of new scenarios and integration of other annotations.

Experimental tests showed that SynDRA's annotated samples enhance the performance of vision models in semantic segmentation and ego discrimination tasks, specifically in situations where real-world annotated data is scarce.

Future work will build upon the versatility of the generation pipeline by extending the dataset with new scenarios, the integration of LiDAR sensors and object detection, and further analysis with low-visibility data. The current version of the dataset, which includes four distinct scenarios collected under varying conditions, is publicly available for download at https://syndra.retis.santannapisa.it.

References

[1] Bayerische Vermessungsverwaltung. Bayernatlas. https://geoportal.bayern.de/bayernatlas, 2024. 3

[2] Nikola Bešinović, Lorenzo De Donato, Francesco Flammini, Rob MP Goverde, Zhiyuan Lin, Ronghui Liu, Stefano Marrone, Roberto Nardone, Tianli Tang, and Valeria Vittorini. Artificial intelligence in railway transport: Taxonomy, regulations, and applications. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14011–14024, 2021. 1

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 7

[4] André Broekman and Petrus Johannes Gräbe. Railenvpasmvs: A perfectly accurate, synthetic, path-traced dataset featuring a virtual railway environment for multi-view stereopsis training and reconstruction applications. *Data in Brief*, 38:107411, 2021. 3

[5] Matthias Brucker, Andrei Cramariuc, Cornelius Von Einem, Roland Siegwart, and Cesar Cadena. Local and global information in obstacle detection on railway tracks. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9049–9056. IEEE, 2023. 2

[6] Zhichao Chen, Jie Yang, Zhicheng Feng, and Hao Zhu. Railfod23: A dataset for foreign object detection on railroad transmission lines. *Scientific Data*, 11(1):72, 2024. 2

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5, 6

[8] Jon Ander Iñiguez de Gordoa, Sara García, Itziar Urbieta, Nerea Aranjuelo, Marcos Nieto, Daniel Ochoa de Eribe, et al. Scenario-based validation of automated train systems using a 3d virtual railway environment. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 5072–5077. IEEE, 2023. 3

[9] Thomas Decker, Ananta R Bhattarai, and Michael Lebacher. Towards scenario-based safety validation for autonomous trains with deep generative models. In *International Conference on Computer Safety, Reliability, and Security*, pages 273–281. Springer, 2023. 2

[10] 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. 3

[11] Gianluca D'Amico, Mauro Marinoni, Federico Nesti, Giulio Rossolini, Giorgio Buttazzo, Salvatore Sabina, and Gianluigi Lauro. Trainsim: A railway simulation framework for lidar and camera dataset generation. *IEEE Transactions on Intelligent Transportation Systems*, 2023. 2, 3

[12] Epic Games, Inc. Quixel. Megascans. https://quixel.com/megascans, 2024. 3

[13] Jeanine Harb, Nicolas Rébéna, Raphaël Chosidow, Grégoire Roblin, Roman Potarusov, and Hatem Hajri. Frsign: A large-scale traffic light dataset for autonomous trains. *arXiv preprint arXiv:2002.05665*, 2020. 2

[14] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 852–861, 2021. 2

[15] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004. 3

[16] Philipp Leibner, Fabian Hampel, and Christian Schindler. Gerald: A novel dataset for the detection of german mainline railway signals. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 237(10):1332–1342, 2023. 2

[17] Dongtai Li and Jie Zhang. Intelligent road segmentation and obstacle detection for autonomous railway vehicle. *Advances in Mechanical Engineering*, 16(1):16878132231225312, 2024. 2

[18] Haoran Li, Qichao Zhang, Dongbin Zhao, and Yaran Chen. Railnet: An information aggregation network for rail track segmentation. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020. 7

[19] Xinpeng Li and Xiaojiang Peng. Rail detection: An efficient row-based network and a new benchmark. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6455–6463, 2022. 2

[20] Lyoshko. Railway system with ic train. https://www.unrealengine.com/marketplace/en-US/product/railway-system-with-intercity-train, 2021. 3

[21] Michael Neri and Federica Battisti. 3d object detection on synthetic point clouds for railway applications. In *2022 10th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6. IEEE, 2022. 3

[22] OpenAI. ChatGPT: Optimizing Language Models for Dialogue, 2023. 2

[23] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2024. 3

[24] Huihui Pan, Yuanduo Hong, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. *IEEE Transactions on Intelligent Transportation Systems*, 24(3):3448–3460, 2022. 5, 7

[25] Mauro José Pappaterra, Francesco Flammini, Valeria Vittorini, and Nikola Bešinović. A systematic review of artificial intelligence public datasets for railway applications. *Infrastructures*, 6(10):136, 2021. 3

[26] Danijela Ristić-Durrant, Marten Franke, and Kai Michels. A review of vision-based on-board obstacle detection and distance estimation in railways. *Sensors*, 21(10):3452, 2021. 1, 7

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[28] Alireza Shafaei, James J Little, and Mark Schmidt. Play and learn: Using video games to train computer vision models. *arXiv preprint arXiv:1608.01745*, 2016. 2

[29] Niko Sünderhauf, Peer Neubert, and Peter Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *Proc. of workshop on long-term autonomy, IEEE international conference on robotics and automation (ICRA)*, page 2013. Citeseer, 2013. 2

[30] Rustam Tagiew, Pavel Klasek, Roman Tilly, Martin Köppel, Patrick Denzler, Philipp Neumaier, Tobias Klockau, Martin Boekhoff, and Karsten Schwalbe. Osdar23: Open sensor data for rail 2023. In *2023 8th International Conference on Robotics and Automation Engineering (ICRAE)*, pages 270–276. IEEE, 2023. 1, 2

[31] Ruifan Tang, Lorenzo De Donato, Nikola Besinović, Francesco Flammini, Rob MP Goverde, Zhiyuan Lin, Ronghui Liu, Tianli Tang, Valeria Vittorini, and Ziyulong Wang. A literature review of artificial intelligence applications in railway systems. *Transportation Research Part C: Emerging Technologies*, 140:103679, 2022. 1

[32] Tugce Toprak, Burak Belenlioglu, Burak Aydın, Cuneyt Guzelis, and M Alper Selver. Conditional weighted ensemble of transferred models for camera based onboard pedestrian detection in railway driver support systems. *IEEE Transactions on Vehicular Technology*, 69(5):5041–5054, 2020. 2

[33] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 109–117, 2017. 2

[34] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 5

[35] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020. 1

[36] Oliver Zendel, Markus Murschitz, Marcel Zeilinger, Daniel Steininger, Sara Abbasi, and Csaba Beleznai. Railsem19: A dataset for semantic rail scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 4, 5

[37] ZerstorenGames. Train template. https://www.unrealengine.com/marketplace/en-US/product/train-template, 2019. 3

[38] Arij Zouaoui, Ankur Mahtani, Mohamed Amine Hadded, Sébastien Ambellouis, Jacques Boonaert, and Hazem Wannous. Railset: A unique dataset for railway anomaly detection. In *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*, pages 1–6. IEEE, 2022. 2