# RailSem19: A Dataset for Semantic Rail Scene Understanding

Oliver Zendel Markus Murschitz Marcel Zeilinger Daniel Steininger Sara Abbasi
Csaba Beleznai
AIT, Austrian Institute of Technology, Giefinggasse 2, 1210, Vienna, Austria
{oliver.zendel;markus.murschitz;marcel.zeilinger.fl;daniel.steininger;
sara.abbasi;csaba.beleznai}@ait.ac.at

## Abstract

*Solving tasks for autonomous road vehicles using computer vision is a dynamic and active research field. However, one aspect of autonomous transportation has received little contributions: the rail domain. In this paper, we introduce the first public dataset for semantic scene understanding for trains and trams: RailSem19. This dataset consists of 8500 annotated short sequences from the ego-perspective of trains, including over 1000 examples with railway crossings and 1200 tram scenes. Since it is the first image dataset targeting the rail domain, a novel label policy has been designed from scratch. It focuses on rail-specific labels not covered by any other datasets. In addition to manual annotations in the form of geometric shapes, we also supply dense pixel-wise semantic labeling. The dense labeling is a semantic-aware combination of (a) the geometric shapes and (b) weakly supervised annotations generated by existing semantic segmentation networks from the road domain. Finally, multiple experiments give a first impression on how the new dataset can be used to improve semantic scene understanding in the rail environment. We present prototypes for the image-based classification of trains, switches, switch states, platforms, buffer stops, rail traffic signs and rail traffic lights. Applying transfer learning, we present an early prototype for pixel-wise semantic segmentation on rail scenes. The resulting predictions show that this new data also significantly improves scene understanding in situations where cars and trains interact.*

## 1. Introduction

Autonomous driving for road vehicles is currently a rapidly developing topic promising improvements for convenience and road safety. Camera-based sensors are crucial components needed to solve many open tasks for autonomous driving. Image/video datasets are at the center of the current push in data-driven machine learning allowing much progress in this field. Although the number of public datasets for road scenes has increased over the last years, rail applications have been left out. There is a noticeable lack of research for autonomous trains and trams. This also affects the safety of autonomous cars at rail intersections and city environments where cars and trams share the same roads. Such scenes are underrepresented in existing road datasets (e.g. [5], [16], [2]).

In this work, we present the first public dataset for semantic scene understanding of image sequences covering the rail domain. Full pixel-wise annotations are labor-expensive and contain many areas already covered by road annotation datasets. Instead, we define a new geometry-based label policy capturing relevant semantic content. This closes crucial gaps for both road and rail vehicles with little redundancies. Our main contributions are:

- An analysis of existing datasets to identify existing rail-relevant data and the data gap (Section 2).
- A novel label policy for the rail domain (Section 3).
- A new dataset called RailSem19 (rail semantics dataset 2019) which is the result of an extensive data collection campaign, frame selection, and annotation work (Section 4; freely available at www.wilddash.cc).
- A semantic-aware automated process to join the manually created geometric shape annotations and pre-generated pixel-wise label maps (weakly supervised) to gain annotations suitable for pixel-wise semantic segmentation (Section 5). Direct use in road scenarios is supported by our Cityscapes-compatible labels.

The experimental result Section 6 showcases tasks that can be tackled using the new dataset. It gives strong empirical evidence that our dataset closes the previously identified data gap. Finally, Section 7 summarizes our contributions.

## 2. State-of-the-Art

To the best of our knowledge, RailSem19 is the first public dataset for solving tasks based on images from the ego-vehicle view for the rail domain. This section summarizes

Figure 1. Examples from RailSem19: front views from train and tram rides in varying seasons, lightings, and environments

other, more generic datasets which contain some rail categories. It is divided by three computer vision (CV) tasks connected to semantic understanding: image classification, object detection and dense semantic segmentation.

### 2.1. Classification

Bernardi et al. [3] provide a general summary of public datasets for image classification and automatic image description. Some of the datasets also contain rail-relevant labels:

- *CIFAR 100* dataset [11] has one sub-class for trains within the *vehicles 1* superclass. There are 600 images present in the dataset for this class. The very small size of 32x32 pixels per image makes this dataset unfit for robust real-world data.
- In the *PASCAL VOC2012* challenge [6], the class *trains* is represented by a total of 544 example images.
- The *Microsoft COCO* dataset [14] contains 3745 images annotated with the class *trains*. The vast majority of images from the classes *traffic lights* (4330) and *stop signs* (1803) are taken from regular road scenes without any rail context.
- The popular 1000 *ImageNet* [21] ILSVRC2017 object categories from the Synset dataset contain labels for *freight_car* (1311 images), *passenger_car* (986 images), *streetcar* (1309 images), *electric_locomotive* (1217 images), *steam_locomotive* (1187 images), and *bullet_train* (1205 images) with a total of 7215 images annotated using any of these labels. The extended *Synset* dataset contains a few more leaf nodes (e.g. *diesel_locomotive*, *tender*).
- The *Open Images Dataset V4* [13] is a collection of over 9.2 million images with bounding box annotations. Overall, 9284 images contain a total of 10506 annotations for the label for *train*. These images are mainly taken from passenger's views (e.g. from the platform towards the approaching train). They also include pictures of toy and model trains. There are only two images with both a traffic light and a train annotation. No other road/rail-relevant label is present with an overlap of the *train* label.
- There are some annotated datasets (e.g. YFCC-100M [23], SBU1M [17]) which automatically link title and descriptions to images from *flickr*, but these unmoderated results contain huge label noise.

### 2.2. Detection

Multiple classification datasets provide bounding box annotations for the mentioned instance classes: both *MS COCO* and *Open Images Dataset* support all mentioned classes. *ImageNet* contains a reduced set of label classes and offers bounding boxes for the rail-relevant classes.

### 2.3. Semantic Segmentation

Some of the dense (i.e. pixel-wise) annotated datasets focusing on road scenes contain also rail elements (as seen from the car's perspective):

- Among the 35 label classes of the *Cityscapes* dataset [5], there are two specifically relevant for rail scenarios: *rail track* and *train*. In contrast to Railsem19, the *rail track* label encloses both the rails and the area between them. *Cityscapes* contains 117 images with 131 annotated rail tracks and 167 images with 194 annotated trains. The average area relative to the total image area is 6.55% for *rail track* and 4.07% for *train*.
- The *Mapillary Vistas* dataset [16] has a far more extended label policy. The public dataset has 66 labels while the commercial dataset contains 152 classes. For the rail domain, the public label classes[1] only include the same two classes as *Cityscapes*: *construction–flat–rail-track* and *object–vehicle–on-rails*. In the combined training and validation sets there are 710 images with

[1] No public information or sample data indicates additional rail-relevant labels in the commercial edition.

a *construction–flat–rail-track* annotation and 272 annotations of *object–vehicle–on-rails*. The average area of *construction–flat–rail-track* annotations is 3.79% relative to the total image area, and 2.1% for *object–vehicle–on-rails*.

- The *COCO-Stuff dataset* [4] offers dense pixel-wise annotations for 182 labels from the *COCO* dataset. This includes these rail-relevant labels: *platform*, *railroad*, and *train*. Since platforms can be any surface other objects stand on, only those platforms with a railroad or train in the image are counted. In total, the dataset contains 4761 trains with an average relative area of 19.1%, 2839 railroads with 11.6% and 1015 such platforms with 11.7%.
- The *KITTI* dataset [2] supplies dense pixel-wise annotations for *Cityscapes* labels. It contains 47 *rail track* and 18 *train* annotations, with 2.26% and 2.39% average size, respectively.

Images from the rail domain in present datasets are a mix of indoor views, road views, and images taken by spectators. Sequences and examples taken from the ego-perspective of a train are nearly nonexistent.

One CV task for the rail domain has been covered by research: rail track maintenance. Specialized equipment captures images of the rail track from the bird's-eye view which are used to identify cracks and imperfections in the rails (see [8]).

Some datasets for the rail domain have been introduced to help solve non-CV machine learning tasks. They focus on track network topology (e.g.[12]), train configurations (e.g. [15]), and crash prediction modelling (e.g.[1]).

## 3. Label Policy for Rail Applications

A new label policy is presented to fix the identified gaps of existing semantic datasets for the rail domain. The following description also helps to understand the rail-specific terminology in the rest of the paper.

The new annotations focus on four topics:

- rails: The basis for rail transportation.
- switches: Dynamic elements allowing splitting and merging of tracks.
- traffic signs and traffic signals: Static and dynamic infrastructure to direct rail traffic.
- crossings, trains, and platforms: Other traffic participants and places where road/passenger interactions occur.

Other aspects like background and road traffic participants are already covered by various road semantic segmentation datasets mentioned in Section 2.2. One of the most unique components of the new dataset is rail annotations. Two types of rails are distinguished:

- *rail*: Rails designed to be in contact with trams and train wheels.
- *guardrail*: Metal rails which have no contact to train wheels; used to increase the structural integrity of rail tracks

Both types of rails are typically elongated and their centerline can be approximated using a curve. One Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)[7] spline annotation is used for each rail. Polygon annotations of typical rail occluders are defined to capture the visual impact of objects occluding the rails: *person*, *car*, *truck*, *pole*, and *fence*. All other occluders of rails are annotated using the generic *rail-occluder* label.

Switches are places where a rail track can split or two tracks join to continue as one track. For the detection of railway track switches, there are five relevant bounding box annotation labels: *switch-left*, *switch-right*, *switch-unknown*, *switch-indicator*, and *switch-static*. The two movable rails that form part of a switch are called blades. The switches' state can be detected by identifying which of the two blades is not in contact with the adjacent static rail: if the left blade has no contact, a train traveling over the switch would be diverted to the left side (and vice-versa for the right side). Figure 2 illustrates this relation. The switches' state and thus the projected path of a rail vehicle can be detected by identifying the gaps next to switch blades. For switches where this visual state identification is not possible, the label *switch-unknown* is used. In many cases, the state of a switch is also indicated by a nearby signal or rotating display. These elements are labeled as *switch-indicator* whereas typical motor/power boxes next to switches are labeled as *switch-static*. Static elements have no way of communicating the switches' current state (left/right) but serve as a valuable clue that a switch is nearby on the track.

The traffic signals and signs are also annotated using bounding box annotation. The front and backside are differentiated. Although the signal's backside can sometimes reveal its state (e.g. for raised arms), typically the backside of traffic signs is not descriptive and therefore not annotated. Figure 3 showcases example sections from RailSem19 for each annotation class.

The choice of using bounding box annotations for some labels is based on a cost/benefit analysis to minimize total annotation effort. Both traffic signs and traffic lights can be reasonably annotated with axis-aligned bounding boxes without too much negative space. They must be oriented so that train drivers can clearly see them. Railway crossings are annotated using bounding boxes: the surrounding rail annotation and the clear edges between road and track bed make more detailed polygons redundant. In addition, all rail vehicles visible and all station platforms get full polygon annotations.

Figure 2. Switch explanation using image crops from RailSem19: the three panels on the left show an example for *switch-left*, the three on the right a *switch-right*. The first panel shows the original image, the second panel the annotated rails in cyan and the area of the gap between switch blade and outer rail in red. The last panel illustrates where the rail vehicle is driving when moving forward: *switch-left* directs the train to the left; *switch-right* to the right.


Figure 3. Examples from RailSem19 for each class in our rail annotation label policy; f.l.t.r: rails with guardrails in the middle; *switch-indicator*; *switch-static*; *traffic-signal-front*; *traffic-signal-back*; *traffic-sign-front*; *crossing*; *train*; *platform*; *buffer-stop*.

## 4. Creating a new Dataset

A large collection of suitable video sequences showing rail sequences from the ego-vehicle's point of view forms the foundation of the RailSem19 dataset. A lot of data is available thanks to a large online community of rail enthusiasts who upload driver's cabin videos. Over 1000 suitable sequences have been identified for RailSem19 through contacts via railway community portals and search engines. Their authors have either released videos under a permissive license, or they have given explicit consent to allow the inclusion of short samples to the dataset. In the end, agreements have been made for 530 sequences covering over 350 hours of rail and tram traffic from 38 different countries, in all four seasons, and under varying weather conditions. The huge variety of camera models, mounting positions, and different lighting conditions also increase dataset variability. Figure 1 illustrates the variations of scenes present in RailSem19.

Selecting a diverse set of center frames from the large set of sequences poses an additional challenge. The dataset should contain little redundancy but have many examples of the objects mentioned in the label policy. For this task, a new fast image similarity measure has been created specifically tuned for ego-motion sequences: *egodist*. The proposed metric works by calculating normalized cross-correlation between adjacent frames and central crops of adjacent frames. The central crops approximate image areas that are visible if the ego-vehicle moves forward by an assumed frame motion. The position and sizes of the crops have been tuned manually to match the egomotion created by the sequence's framerate. By resizing these patches to a small image size of $64 \times 36$ pixels, the following computations are sped up while small variations in image noise can be ignored. Three image similarities are computed: *egodist* of adjacency frames for the lower third of the image nearest to the train, the *egodist* of adjacent frames for the full height, and the maximum *egodist* with a random subsample of 84 frames from the whole sequence. An interactive interface is used to choose appropriate thresholds for each of the similarities to remove data automatically. This approach works well to focus on frames where crossings, platforms, and rail switches are largest in the image. Predictably, it has some serious problems with fast curves. During the first phase, approx. 60000 frames have been automatically selected from roughly half a million initial frames grabbed from all image sequences using constant framerates. All results are inspected visually and again about 80% of frames have been removed. In total, this process allowed to create a diverse subset of about 10100 frames with little redundancies in a manageable time frame. The resulting frames are annotated and multiple rounds of quality assurance are performed to achieve a consistently high level of quality. A small subset of frames is held back for the creation of upcoming challenges resulting in the final number of 8500 frames with over 110000 annotations for RailSem19 (see Table 1 for details).

## 5. Generating dense labels

For many background labels and some object labels, current state-of-the-art semantic segmentation solutions for road scenes can infer labels with a mean intersection-over-union (IoU) of well over 90%. A hand-selected subset of inference results is used as a backdrop for the rasterizing of geometric rail labels to generate dense annotations for pixel-wise semantic segmentation training and testing.

Rasterizing polygon annotations is straightforward. Bounding box annotations are processed in a two-step approach: if there is at least 50% overlap between bounding box area and the inferred reference result, then the intersection between the two annotations is used in the resulting

Table 1. Number of annotations per class for Railsem19; cursive labels are rail occluders.

| **label** | **buffer-stop** | **crossing** | **guard-rail** | **train-car** | **platform** | **rail** | **switch-ind.** | **switch-left** | **switch-right** | **switch-unknown** | **switch-static** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 218 | 2885 | 4873 | 1951 | 2482 | 58322 | 2007 | 1965 | 2083 | 2491 | 2519 |
| **label** | **track-sign-front** | **track-signal-back** | **track-signal-front** | | *person-group* | *car* | *fence* | *person* | *pole* | *rail-occluder* | *truck* |
| **count** | 7404 | 3448 | 5768 | | 62 | 172 | 291 | 234 | 10875 | 3920 | 11 |

image. Otherwise, the full bounding box area is filled with the respective label. However, rasterizing the spline rail annotations poses three challenges:

- Rail thickness is unknown (splines have no width).
- Information for pairing rails into rail tracks is missing.
- Rail trackbed extent surrounding the rails is missing.

Figure 4 illustrates steps used to solve these open points:
**Tapered rail regions:** Equidistant points $P_1 ... P_N$ ($N = 100$) along each PCHIP spline annotation are generated and their normals are computed. Local thickness $w$ is computed for each point by assuming perspective linear scaling between $w_{init}$ at the bottom image border (= closest-to-camera) and zero at the semi-automatically approximated horizon line. Along given normal vectors two sample points are generated on both sides of the polygon at a distance of $w/2$ to create tapered polygonal region representation for each rail (left in Figure 4).
**Rail pairing:** The pairing of rails is crucial for the identification of drivable regions. This is safety-relevant to prevent potential collisions. Each resampled rail polygon is matched against the other rail polygons to obtain rail pairs automatically. For each point of polygon $P_a$, the distance to the nearest point of polygon $P_b$ is computed. A pair is kept for further comparison if no distance exceeds a spatially varying distance threshold (governed by the same linear scaling as before).

The next comparison step examines the shape similarity between two hypothetical matches. Each resampled polygon is split into $K$ ($K = 2$) segments and between segment pairs, a cosine-distance-based correlation measure is computed. Only candidates with consistent shape similarity along their length are kept. The computed shape similarity measures are inserted into an affinity matrix and correspondences are resolved by a greedy association scheme. A one-to-one pairing constraint is enforced. The automated pairing scheme yields correct associations for about 80% of all cases. Some parallel rails with equidistant spacing create ambiguities, thus requiring manual linking or unlinking of erroneous associations.
**Spatial extension around paired rails:** Based on the established rail pairings, the extended trackbed is generated outwards in a manner similar to the generation of tapered rail regions (see Figure 4). No dense labels are added for switches as they are well solved as part of object detection tasks. The layout and area of most parts that form a switch are represented with *rail-raised*, *rail-track*, and *trackbed* labels. Platforms are represented as part of the *sidewalk* label and buffer stops use the *construction* label.

Five new rail-specific labels are introduced: a) *rail-track*, *rail-raised*, and *trackbed* for rail-exclusive areas and b) *tram-track*, *rail-embedded* for mixed areas where road and rail vehicles can drive (see Table 2). The classes *road* or *sidewalk* are used for outward areas next to *rail-embedded* areas (analogues to *trackbed*). Masks are automatically generated to differentiate between the two modes a) and b). These masks are created using the *crossing* geometric annotations and the meta information for each source sequence classifying it as either a tram or a train sequence.

The above rasterizations produce a foreground which is filled by background labels from a pre-trained model of WideResNet38 + DeepLab3 using in-place activated batch normalization (InPlace-ABN) [20]. A public version of the model [19] pretrained on *Mapillary Vistas* [16] from the authors of InPlace-ABN is used for RailSem19 background labels. Dense labeling for RailSem19 uses 19 labels in total to provide an easy transition for users applying transfer learning from existing semantic segmentation networks trained on the 19 Cityscapes training labels (see Table 2). The amount of retraining is thus kept to a minimum, making the dense labels directly applicable to road scenes. All automatically generated labels are inspected manually and adaptations are made to correct detected faults.

Table 2 shows the resulting distribution of dense-label classes for all 8500 RailSem19 frames. The raw geometric data concentrates on thin objects (e.g. rails, guard rails, and switch states) and instances while the dense data focuses on usability for interpreting the whole scene and identifying car-drivable sections for mixed rail/road scenarios. Both versions are available for free which allows researchers to quickly start using RailSem19 for multiple tasks out of the box.

## 6. Experiments

This section presents multiple experiments run to illustrate the variety of tasks that can be tackled in the rail domain thanks to RailSem19 data. We do not claim to solve

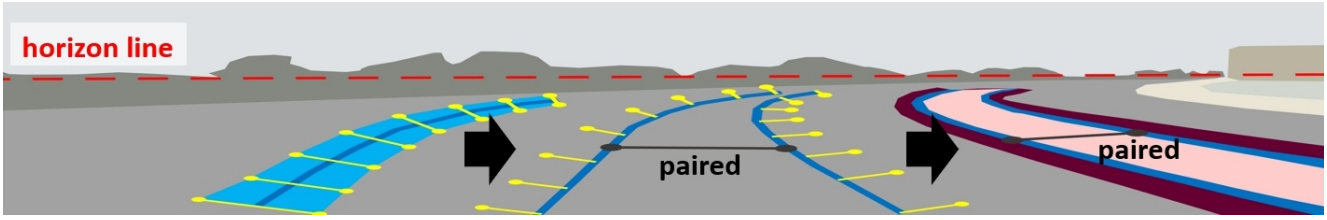


Figure 4. Illustration showing the steps of computing spatially tapered elongated polygon structures around rails (left) and around paired rails (middle). The pairing and the computed spatial extension allows creating the *rail track* and *trackbed* labels.

Table 2. Mapping of Cityscapes (CS) training labels to RailSem19 dense labels including color legend. Pixel-wise statistics: **Mean/Median** = percentage over all Railsem19 pixels; **In Frames** = percentage of frames containing pixels with the resp. label. **mIoU RS19** = results of semantic segmentation experiment from Section 6.2. *: Cityscapes class *rail-track* (not part of default training labels) is used for both the rail and the space between the rails while RailSem19 uses two distinct labels; bi/motorcy.: bicycle+motorcycle.

| **Color RS19** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Label RS19** | **road** | **sidewalk** | **con-struction** | **tram-track** | **fence** | **pole** | **traffic-light** | **traffic-sign** | **vegetation** | **terrain** |
| **Label CS** | **road** | **sidewalk** | **building + wall** | **-** | **fence** | **pole** | **traffic-light** | **traffic-sign** | **vegetation** | **terrain** |
| **Mean** | 5.2 % | 3.1 % | 12.2 % | 3.1 % | 2.5 % | 2.8 % | 0.3 % | 0.2 % | 23.3 % | 6.6 % |
| **Median** | 2.3 % | 1.0 % | 5.8 % | 2.3 % | 0.8 % | 1.8 % | 0.1 % | 0.1 % | 19.8 % | 3.6 % |
| **In Frames** | 48.1 % | 57.2 % | 72.1 % | 51.7 % | 48.9 % | 60.0 % | 37.1 % | 32.3 % | 83.3 % | 61.2 % |
| **mIoU RS19** | 50.5% | 54.0% | 72.4% | 40.1% | 49.8% | 60.2% | 36.8% | 32.5% | 84.1% | 59.5% |
| **Color RS19** | | | | | | | | | | |
| **Label RS19** | **sky** | **human** | **rail-track** | **car** | **truck** | **trackbed** | **on-rails** | **rail-raised** | **rail-embedded** | **void** |
| **Label CS** | **sky** | **person + rider** | * | **car** | **truck + bus** | **-** | **on-rails** | * | **-** | **bi/motorcy. + void** |
| **Mean** | 22.3 % | 0.5 % | 5.9 % | 0.8 % | 0.7 % | 10.3 % | 3.3 % | 3.4 % | 1.5 % | 5.4 % |
| **Median** | 22.9 % | 0.1 % | 5.4 % | 0.1 % | 0.1 % | 9.0 % | 0.2 % | 2.7 % | 1.1 % | 1.3 % |
| **In Frames** | 94.5 % | 6.0 % | 86.2 % | 13.8 % | 4.6 % | 87.6 % | 15.4 % | 87.2 % | 14.6 % | 57.2 % |
| **mIoU RS19** | 94.7% | 46.7% | 81.9% | 50.6% | 20.7% | 67.8% | 62.8% | 71.5% | 45.0% | — |

the problems fully and robustly. These experiments deliver first impressions for the respective tasks showcasing the value and utility of our new dataset.

### 6.1. Image Classification

The first range of experiments focuses on image classification tasks: classify images into rail-relevant classes. Interesting classes are based on the available annotations of the new dataset: all bounding box annotations, as well as polygon annotations which are not occluders, are used to create class-wise crops from all RailSem19 images. Negative crops are generated by randomly picking areas that do not contain any class label from all RailSem19 frames. For each label, a single-class classifier is trained. The positive class consists of the label's samples whereas remaining samples are used as the negative class. The *fastai* framework [9] is used to train a densenet161 [10] (pretrained on ImageNet) for this classification task on both RailSem19 and all identified rail-relevant classes from Open Images dataset (see Section 2.1). In general, the networks converge successfully, achieving accuracies of over 90%. Classifiers trained on RailSem19 perform well on both test datasets while classifiers trained on the Open Images dataset see a significant reduction in accuracy when tested on RailSem19. The differences between viewpoints in RailSem19 vs. Open Images (ego-vehicle view vs. platform view) have significant impact on network performance in real-world rail scenarios. After the single-class classifiers, a multi-class classifier is trained on RailSem19 to cover all classes from RailSem19. Again, the same densenet161 architecture is used until training and validation losses flatten (five epochs). The last two rows in Table 6.1 show relevant results. Figure 5 plots the confusion matrices from both the validation and test datasets. In general, the multi-class classifier approaches the performance of the single-class classifier for most classes. The exceptions are *buffer-stop* and the *switch-left*/*switch-right* labels. For *buffer-stop*, the number of examples in the dataset is quite small (218 instances in total). The classes of *switch-left* vs. *switch-right* are hard to distinguish. Only the changes of the switch blades determine the difference, otherwise they look the same when compared to the other classes. Another clas-

Table 3. Results of classification experiments. Accuracy for each dataset and class; RS19 = RailSem19; Open Images = OImg; d1 / d2 means training was done on d1 and evaluation on d2's separate test dataset; The first four rows are results for the single-class experiments. The last two rows show the results from the multi-class experiment. Here evaluation on both validation (RS19 - val) and test (RS19 - test) dataset are shown.

| **Dataset** | **train** | **traffic-light** | **traffic-sign** | **switch-static** | **buffer-stop** | **switch-ind.** | **switch-left** | **switch-right** | **platform** | **crossings** |
|---|---|---|---|---|---|---|---|---|---|---|
| RS19 / RS19 | 93.6% | 94.0% | 92.8% | 91.8% | 86.9% | 84.3% | 90.3% | 89.6% | 96.7% | 92.4% |
| RS19 / OImg | 83.8% | 86.5% | 74.5% | — | — | — | — | — | — | — |
| OImg / RS19 | 76.9% | 67.1% | 57.8% | — | — | — | — | — | — | — |
| OImg / OImg | 98.4% | 98.5% | 98.6% | — | — | — | — | — | — | — |
| RS19 - val | 90.5% | 90.8% | 93.0% | 86.0% | 41.9% | 63.3% | 44.3% | 61.3% | 91.5% | 89.6% |
| RS19 - test | 89.0% | 91.9% | 90.8% | 88.6% | 54.5% | 66.1% | 53.7% | 62.9% | 91.1% | 90.2% |

sification experiment is run, differentiating only *switch-left* vs. *switch-right* instances. Again a densenet161 pretrained on ImageNet is fine-tuned to distinguish between the two classes. After initial strong confusion of both classes in the multi-classification experiment, the image crops have been expanded by 30% on the x and 125% on the y-axis to capture more context. Labels which do not fully overlap with the image or with dimensions $< 28$ pixels are ignored leading to a reduced dataset size with $1460/1539$ images for switch-left/switch-right. Multiple data augmentation transforms are used to create a larger training set (warping and small rotations). Accuracy after 20 epochs of training is still only 67%, proving this to be a challenging classification task.

### 6.2. Semantic Segmentation

A semantic segmentation network is now trained and evaluated using the dense approximations which are supplied as an extension to the geometric labels of the RailSem19 dataset (see Section 5).

Cityscapes is used as a basis for this experiment to prevent any bias introduced by using weak supervision for the dense labels pretrained on Mapillary Vistas dataset [16]. The baseline for this experiment is an FRRNB model [18] pretrained on Cityscapes which is fine-tuned on RailSem19. The model is part of the pytorch-semseg framework [22] allowing training and retraining of various semantic segmentation networks. A set of $4000$ images is randomly selected from RailSem19 for this experiment. The $4000$ images are split into subsets of $3000$, $500$, and $500$ images for training, validation, and testing. This replicates the size and split of the Cityscapes dataset. The FRRNB model is parameterized for an input size of 512x512 pixel and a batch size of two (allows training on a single RTX2080Ti GPU). The FRRNB model is first trained on the Cityscapes dataset from scratch for 60 epochs (approx. 24h) until the validation loss flattens. The evaluation on the separate default test set result in a mean IoU of $62.7\%$ for the $19$ Cityscapes training classes.

In the second phase, the RailSeg19 training and validation set is used for another $60$ epochs. As discussed in Section 5, many of the default $19$ Cityscapes labels have been reused for RailSem19, some are unions of existing Cityscapes labels. This allowed for an easy transfer learning approach: switching datasets and fine-tuning the whole network on the new data (no layers were frozen). The resulting network has a mean IoU of $57.6\%$ when evaluated on the separate test set. Table 2 lists the individual results per category. Figures 6 and 7 illustrate example output predictions taken from the test and validation subset. In general, these early results show that a network is able to learn the new rail labels. Although not perfect, clear differentiation between *rail-track* and *tram-track* areas are visible.

## 7. Conclusion

We present a new dataset for semantic scene understanding for autonomous trains and trams: RailSem19. It consists of a diverse set of 8500 rail scenes from 38 countries in varying weather, lighting, and seasons. A semi-automated workflow is presented which creates dense semantic labels from geometric annotations. In multiple experiments, typical computer vision scene understanding tasks are tackled: classification and dense image segmentation. The early results from these experiments show a clear benefit of using the new dataset for rail-relevant topics. Raw annotation data and the generated dense maps of RailSem19 are available for free at `www.wilddash.cc`.

## 8. Acknowledgement

The research was supported by ECSEL JU under the H2020 project grant agreement No. 737469 AutoDrive - Advancing fail-aware, fail-safe, and fail-operational electronic components, systems, and architectures for fully automated driving to make future mobility safer, affordable, and end-user acceptable. Special thanks go to all authors who allowed us to use their video material, Gustavo Fernandez Dominguez, Thomas Kadiofsky, Bernhard Rainer, and Christian Zinner for valuable inputs during discussions.

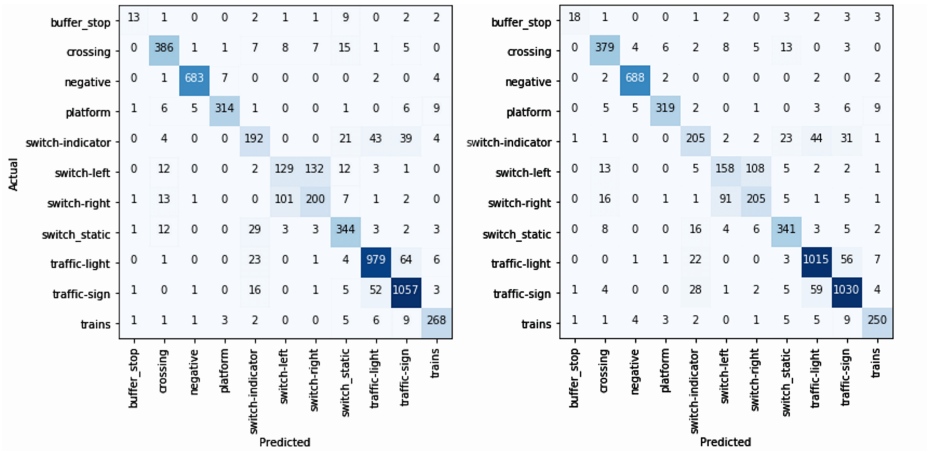


Figure 5. Confusion matrices for the all-classes classifier experiment on RailSem19 data; left: validation dataset, right: test dataset.

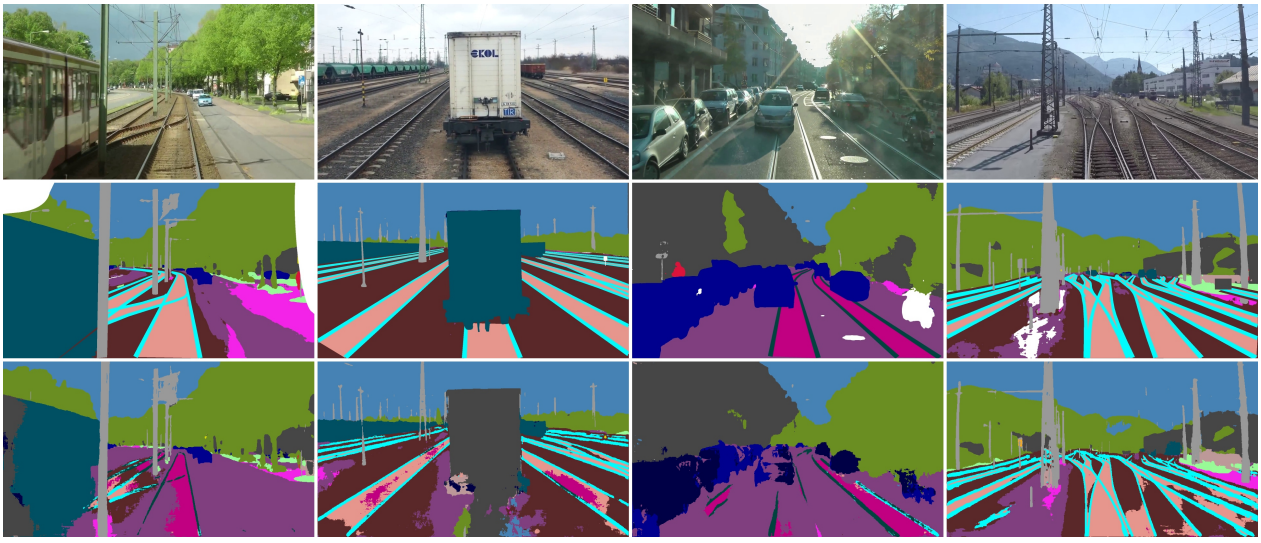

Figure 6. Example results from FRRNB experiment 6.2 for frames taken from RailSem19's validation subset (input images, dense ground truth, and prediction from FRRNB model). For the color legend, see Table 2.

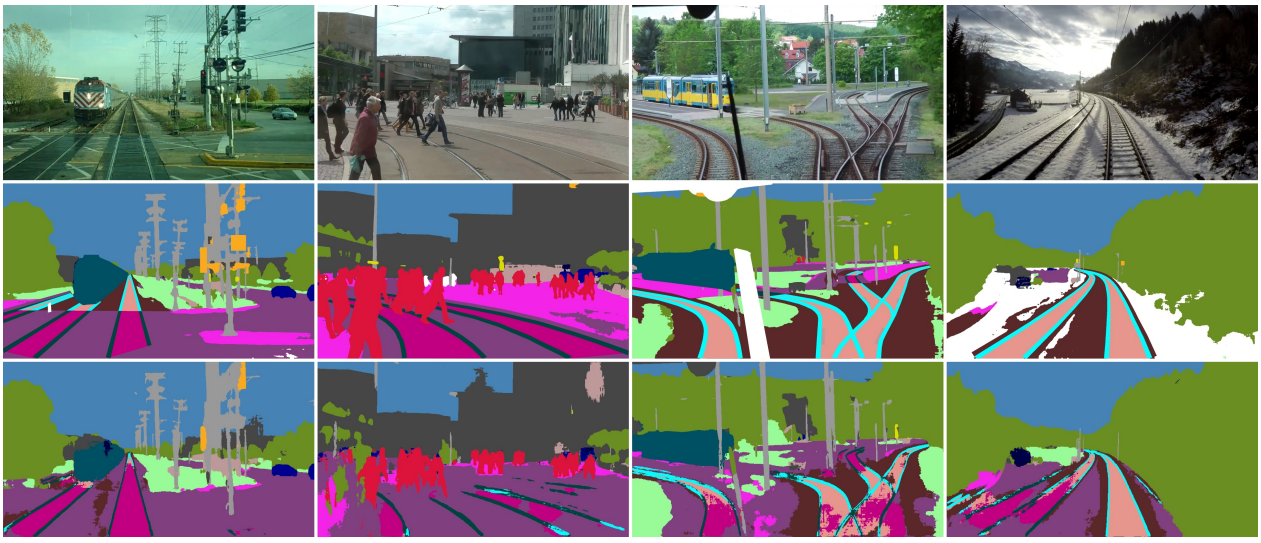

Figure 7. Further example results from FRRNB experiment 6.2 for frames taken from RailSem19's test subset.

## References


[1] "Federal Railroad Administration - Office of Safety Analysis. https://safetydata.fra.dot.gov/OfficeofSafety/default.aspx. Accessed: 2019-03-14. 3

[2] H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018. 1, 3

[3] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 55:409–442, 2016. 2

[4] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018. 3

[5] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, volume 2, 2015. 1, 2

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2

[7] F. N. Fritsch and R. E. Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, 1980. 3

[8] X. Gibert, V. M. Patel, and R. Chellappa. Deep multitask learning for railway track inspection. *IEEE Transactions on Intelligent transportation systems*, 18(1):153–164, 2017. 3

[9] J. Howard et al. fastai. https://github.com/fastai/fastai, 2018. Accessed: 2019-03-14. 6

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6

[11] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 2

[12] R. Kujala, C. Weckström, R. K. Darst, M. N. Mladenović, and J. Saramäki. A collection of public transport network data sets for 25 cities. *Scientific data*, 5:180089, 2018. 3

[13] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 2

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[15] D. Michie, S. Muggleton, D. Page, and A. Srinivasan. To the international computing community: A new east-west challenge. *Distributed email document available from http://www. doc. ic. ac. uk/˜ shm/Papers/ml-chall. pdf*, 1994. 3

[16] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4990–4999, 2017. 1, 2, 5, 7

[17] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *Advances in neural information processing systems*, pages 1143–1151, 2011. 2

[18] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4151–4160, 2017. 7

[19] S. Rota Bulò, L. Porzi, and P. Kontschieder. Wideresnet38 + deeplab3 segmentation model pre-trained on mapillary vistas. 5

[20] S. Rota Bulò, L. Porzi, and P. Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 2

[22] M. P. Shah. Semantic segmentation architectures implemented in pytorch. *https://github.com/meetshah1995/pytorch-semseg*, 2017. 7

[23] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015. 2