Deep Neural Networks for Railway Switch Detection and Classification Using Onboard Camera Images

Kanwal Jahan, Joshua Niemeijer, Nils Kornfeld and Michael Roth[1]

***Abstract*— Recent years have seen major advances in Artificial Intelligence (AI) methods for environment perception in intelligent transportation systems. Although most of them have been achieved in the automotive sector there is a similar demand in the railway domain. This paper investigates Deep Neural Network (DNN) based environment perception using vehicle-borne camera images from the rail domain. Specifically, railway switch detection and classification are addressed as a relevant example for a DNN application with potential use for landmark positioning, environment perception, and condition monitoring. The lack of large training data sets in the railway sector (in contrast to the automotive domain) is compensated by an appropriate DNN architecture, an anchor box ratio optimization scheme, and transfer learning. The presented experimental results advocate for the adopted approach.**

I. INTRODUCTION

The railway environment, as recorded by an onboard camera contains many objects of interest, including level crossings, traffic signs and signals, buffer stops, and railway switches. For selective regions, some of these assets are listed by OpenStreetMap[2]. In such areas, the correct detection of listed objects contributes to landmark recognition, localization, and navigation. Awareness about the objects like level crossings, buffer stops, traffic signals, and the presence of nearby railway vehicles ensures the safety and smooth operation of the railway system.

The recent advancement in Graphics Processing Unit (GPU) technology and the availability of big training data sets have enabled the researchers to use Deep Neural Networks (DNNs) on camera images for extracting useful features. Currently, there is a broad spectrum of applications ranging from simple object detection to providing artificial perception to a transportation system. Most of the achievements have been made in the automotive sector mainly due to the availability of a huge amount of training data. There is a similar demand in the railway domain too.

Attaining such advancement in the railway sector is rather straightforward due to the restricted motion of the vehicles on railway lines as there is no need to have a lane keeping, lane change, and intersection assistance system. The controlled motion of the vehicle on rail tracks limits the direction of motion except for the railway switches. Switches are special structures that enable railway vehicles to change the direction of motion. The blade position of a switch determines the driven path in the rail network [1].

[1]Kanwal Jahan, Joshua Niemeijer, Nils Kornfeld and Michael Roth are with the Institute of Transportation Systems, German Aerospace Center (DLR), 38108 Braunschweig, Germany `firstname.lastname@dlr.de` except M.Roth@dlr.de

[2] https://www.openstreetmap.org

In the context of condition monitoring, switches are more prone to failure and bear higher costs for maintenance [2]. However, they appear on the rail tracks infrequently so their detection can help to focus on the preventive measures of the selective railway tracks only. The task of switch detection has multi-fold benefits like providing support in environment perception leading to autonomous driving, landmark navigation, condition-based monitoring, and path prediction. As a DNN can solve the detection and classification of the blade position simultaneously, switches are the focus of our work. The main contributions of this paper are as follows.

- We reduce the dependency on labeled data by using very few images during the training phase.
- We investigate the methods of transfer learning [3] and perform anchor box aspect ratio optimization to compensate for the use of a small dataset.
- We have generated a small data set comprising publicly available camera videos to be able to perform the supervised learning.

Finally, our work should be seen as an effort to reduce the technology gap in the railway sector. Even though it appears much more conservative, there is great potential for novel technologies to create a competitive and environment-friendly rail sector.

The outline of the paper is as follows. Section II lists related literature and similar work. Section III describes the implemented approach. Experimental results are provided in Section IV.

II. RELATED WORK

The researchers usually divide the neural networks used for object detection into two categories i.e. one-stage detectors and two-stage detectors. Two-stage region-based convolutional neural networks are promising in terms of accuracy but introduce a trade-off in prediction speed [4]. As the name suggests, they perform the detection with the help of two networks. Firstly, a region proposal network is used to generate regions of interest. In the second step, feature extraction and object classification is performed on the proposed regions. One such widely used model is Faster R-CNN [5].

One-stage detectors like YOLO V3 [4] achieve high inference speed by considering the detection task as a regression problem. An image is divided into grid cells, each grid cell has a fixed number of anchor boxes. For each anchor box the network learns, (1) to predict the location offset of the anchor box, (2) if the predicted box contains any object (objectness score), and (3) the probability of the class the detected

object belongs to. All three tasks are normally taken care of by only one network which accelerates their processing. One-stage neural networks improve the speed of detection but, in general, suffer from the problem of class imbalance during the training phase and have lower detection accuracy. However, RetinaNet [6], is a one-stage detector that performs well on both the criteria of speed and accuracy by addressing the problem of class imbalance with the appropriate choice of the loss function.

There are different motivations for the detection of the switches, for instance, [7] and [8] analyze condition monitoring of switches, where they used classical image recognition approaches for distinguishing the switches from a regular straight rail. While we aim to use a deep neural network to detect and classify a switch into its states to support the task of autonomous driving mainly. So far, AI-based work in the railway domain is used for condition monitoring and fault detection in railway assets [9], [10] than the task of the switch detection itself.

Another important component is the availability of domain-specific data. In their survey paper, Hang Yin and Christian Berger [11] describe there are at least 27 data sets applicable to the automotive sector. Unlike the automotive sector, the railway sector lacks publicly available data [12], while some of the big training data sets for example CIFAR-100 [13], Pascal VOC 2012 [14], Microsoft COCO [15], and Open ImageNet [16] do contain limited labeled data representing the railway environment. For instance, in CIFAR-100 [13] only one out of the 100 classes is labeled as train with a total of 600 images. Similarly, PascalVOC-12 [14] has one class named train out of total 20 classes. The widely used COCO data set by Microsoft [15] has more than 200K images with 80 object categories. Only traffic sign and train subcategories are usable in the railway context. So far the most comprehensive dataset in the railway domain is provided as RailSem19 [12] containing labels on four topics out of which only one addresses switches. Out of 8500 total images, there are 1965 and 2083 instances of the left and right switches respectively.

An insufficient amount of training data in the railway sector has motivated us to investigate the critically important concept of transfer learning [3].

III. RAILWAY SWITCH DETECTION

This section describes our proposed approach to address the switch detection and classification task. Our application platform includes two cameras in total, placed at the back and front of the test vehicle to record video data from the cabin, capturing the perspective of a driver. The collected camera data is used to detect and classify the switches appearing on the path of the moving railway vehicle.

A. Definitions

Geometrically, a switch consists of two movable blades known as the switch rail [1] which distinguish a switch from the straight rail. A distinction of a switch from the straight rail is addressed as switch detection in this paper.

If the gap between the switch rail and stock rail is on the right side, the vehicle will take the right path as depicted in Fig. 1 and a left path if the gap is found at the left side. The only difference between the left and right configuration of a switch is made based on the gap present between a straight stock rail and switchblade. The task of identifying the state of a switch into a left and right configuration is termed a classification task. The further classification of a switch into its states helps to predict the driven path which is an important element to achieve autonomous driving.

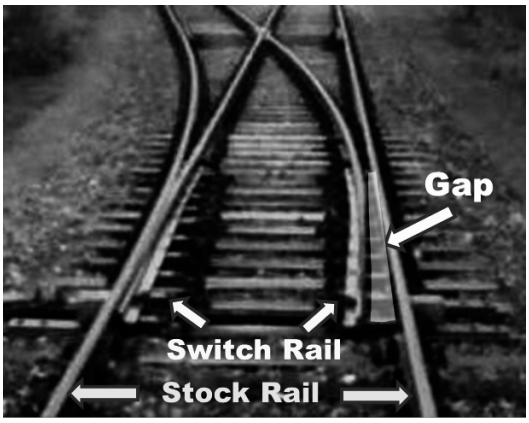


Fig. 1: A Representative Image of Right Switch.

B. DNN Architecture

There are many factors to consider for selecting a suitable network. For example, the confidence in the accuracy and speed of the prediction both are vital criteria for the system to work in real-time. Moreover, the switches appear on the straight rails infrequently which introduces an imbalance in the background (straight rail) and foreground (switches) classes. If the skewed class distribution is not addressed properly it leads to the network ignoring the non-dominating class completely and still achieving a good reduction of training loss. In such situations, the network ends up learning the background (dominating) class which has no practical use. RetinaNet [6] not only performs well on speed and accuracy criteria but also handles the class imbalance. The backbone framework consists of a residual network (ResNet-50) [17] and a Feature Pyramid Network (FPN) [18]. In the architecture of RetinaNet, two subnetworks running in parallel. One is termed as box regression head and the other as classification head.

In the box regression head, a small Fully Convolutional Network (FCN) is attached to each pyramid level. For every anchor box, on every pyramid level, the subnetwork outputs four offset values. These offset values are optimized during training to bring the difference between the ground truth object location and the anchor box as low as possible. Smooth $L1$ loss [19] is used to optimize the weights of the regression head. It is a combination of regular $L1$ and $L2$ loss. It behaves as $L2$ loss near the minima which is a quadratic loss and is sensitive to outliers. For the rest of the

graph, it acts like $L1$ (linear) loss with a steady gradient and with more tolerance to outliers.
The classification subnetwork estimates the probability for the presence of a particular class instance, for each anchor box and total object classes. Generally in one-stage detectors, the class imbalance is present between foreground (positive) and background (negative) classes. In RetinaNet the introduced [6] α-balanced focal loss not only balances negative and positive cases but also gives more importance to complex examples over the simple cases.

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{1}$$

The focal loss Eq. (1) [6], with hyper-parameter α_t (the weight assigned to the rare class) takes care of the class imbalance. It also down weights the loss contribution of easily classified examples and focuses on a sparse set of hard examples through the modulating factor γ.

C. Anchor Box Ratio Adaptation and Transfer Learning

During the regression task of RetinaNet [6], a small FCN draws several anchor boxes at each location. To make the learning efficient the width to the height aspect ratio of the drawn bounding box can be fed as a configuration parameter. K-means clustering is the simplest and most popular unsupervised learning algorithm used to divide a data set into a total of Kclusters. To compute the appropriate aspect ratios K-means clustering is used [20]. We estimate the best suited n number of centroids to represent the aspect ratios of the whole data set as much as possible. This helps to minimize the difference between the learned anchor box shapes and ground truth bounding boxes of the data set used.

Deciding on the number of anchor boxes is a crucial factor, as the number of boxes drawn at each location is a multiple of the number of ratios and used scales. Even though more aspect ratios represent the data set better but it comes with a trade-off in the training speed and computational overhead. For our generated data set, when the number of anchor boxes is 5, 7 and 9 they represent 77.70%, 80.30% and 85.42% of the total bounding boxes respectively. For RailSem19 [12] the same number of anchor boxes represent 59.80%, 61.10% and 64.32% of total bounding boxes. We use three fixed values of scales [18] and 9 aspect ratios which result in 27 boxes drawn at each location during the training.
DNNs have a strong dependence on big training data set but in our case, the major challenge is the lack of any such data. To address this, network-based transfer learning [3] is performed by initializing the weights of the network with the weights from a pre-trained network. In transfer learning knowledge gained with the help of a bigger data set can be transferred to a smaller but to some extent similar data set. Additionally, the lower layers of the backbone network which contain basic features (lines, edges, etc) can be frozen too. This helps the network to perform well on unseen data as during pre-training the model learns the generic features. It also introduces generalization and avoids the network to overfit on smaller data set.

D. DLR labelled Dataset

We have labeled a small data set at DLR which consists of 2500 instances of switches. The images are collected from a DLR-owned railroad vehicle and publicly accessible video platforms capturing the view from the driver's cabin. The data is not only collected from different vehicles, but also different rail tracks of various countries, with changing light (day, evening, and night time) and weather conditions (sunny, rainy, and snowy). Also, the cameras are installed at different heights on the vehicles with varying camera settings. These conditions introduce an enormous amount of diversity in the collected data. There are 1272 instances of switch left and 1218 instances of switch right. The reason to label only a few images is to keep the labeling cost and time minimal.

IV. EXPERIMENTS

We deploy our proposed approach on two datasets i.e. RailSem19 [12] and the DLR generated dataset respectively. Datasets are divided into 80 %, 10 %, and 10 % for train, validation, and test set correspondingly. We use the validation set to monitor the training performance and early stopping to avoid overfitting. The appropriate number of anchor box aspect ratios, as described in Sec. III-C, for each dataset is found to be 9. For performing the network-based transfer learning [3] and to initialize the weights of the training network, the weights of ResNet-50 previously trained on COCO data set are used. The images are resized to a lower dimension of 960x540 to speed up the training time. To further accelerate the training process and reduce the generalization error we use batch normalization [21]. The batch size of 8 is selected which is the maximum size supported by the used NVIDIA GPU- GeForce GTX 1080 Ti. The values used for hyper-parameters like γ and α (Eq. 1) are 2.0 and 0.25 [6]. Adam [22] with initial learning rate of $1e^{-5}$ and clipping norm .001 is used for the optimization.

A. Evaluation Protocol

To evaluate the detection performance of the network we use the Intersction over Union(IoU) metric. A total of 3 IoU thresholds are used i.e. 0 %, 50 %, and 75 %. An IoU threshold at 0 % interprets to, any detected object is considered for the classification irrespective if the predicted anchor box and ground truth bounding box have an overlap. Similarly, an IoU set at 100 % means the drawn anchor box and the ground truth bounding box have an exact match of coordinates.
To assess the classification performance we use the Precision-Recall (PR) curve which is more sensitive to misclassification of minority classes [23]. A PR curve closer to the upper right corner of the graph depicts the good classification performance by the network. As we already have the precision and recall values from the PR curve, we also calculate the $F1$ score which not only evaluates the network performance but also helps to determine the classification threshold. An $F1$ score of 1 is considered ideal and a 0 score means the network has not learned anything

rational. Similarly, precision and recall values closer to 1 indicate the high positive predictive value and sensitivity of the network.

B. Results on RailSem19

The first set of experiments is performed on RailSem19 [12]. It has labels on four topics, one of the topics is switches. Out of 8500 total images, there are 1965 and 2083 instances of the left and right switches respectively. The network is trained for 100 epochs and the training performance does not improve after the 30th epoch. The highest mAP achieved on the validation set is only 8 % at 30^{th} epoch.
Fig. 2 shows the PR curve of the selected model, on the test set (214 instances of switch-right and 201 of switch-left) of RailSem19 [12] dataset. The curves for all three IoU thresholds tend to remain on the lower-left corner illustrating the low values for precision and recall both. The best performing PR curve, with a larger area under the curve, is at 0% IoU.

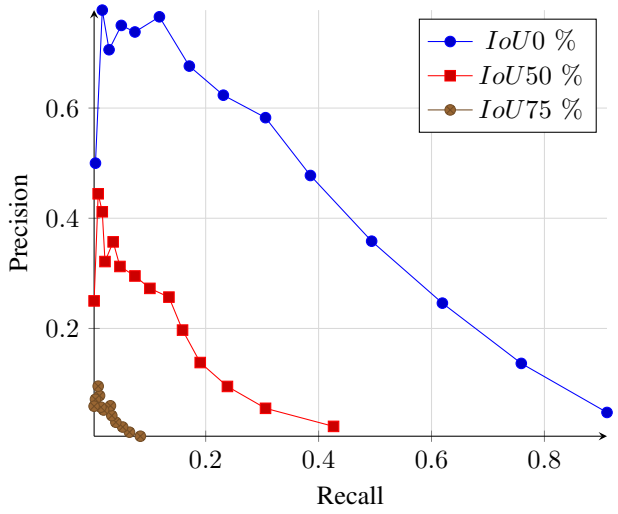


Fig. 2: Precision-Recall Curve of RailSem19.

TABLE I: $F1$ Score of RailSem19 Test Set

CT	F1	$F1_{50}$	$F1_{75}$
0.15	0.352	0.135	0.030
0.20	0.415	0.160	0.0344
0.25	0.426	0.176	0.0377
0.30	0.401	0.147	0.041
0.35	0.337	0.119	0.028

As shown in Tab. I too, the $F1$ score is highest at 0 % yet we select the threshold at 50 % giving equal importance to classification and detection tasks. $F1$ score at the classification threshold of 25 % shows the maximum score of 0.176. So, the selected values are 50 % and 25 % for IoU and classification thresholds, respectively. At the best settings, the precision and recall values are very low (ref: Fig. 2) i.e. 0.19 and 0.15 representing the number of false negatives and false positives both are high as compared to the true positives. The mAP among the classes for the test set is 5.13% and the average inference time is 40.1 ms i.e. the prediction speed of 24.95 fps.
The training results are not promising and do not improve much by using a larger network, introducing dropouts between dense layers, another optimizer, and varying batch size. We observed that the irregularity in the sizes of the bounding boxes is too high. Only 64% of the total bounding boxes are represented with 9 anchor box aspect ratios. Sizes of ground truths are sometimes too small to contain enough information about the geometrical details of the switch states. In rare cases, labels are incorrect too. In the original work [12], detection and classification of the left and right switches is stated as a challenging task with lower accuracy as compared to other classes.

C. Results on DLR Labelled Dataset

The training is performed for 100 epochs on labeled switched from the DLR-labelled dataset. The network reaches the highest training accuracy after the $30th$ epoch. We use early stopping and select $27th$ epoch which shows an mAP of 93.92% for the validation set.
Fig. 3 represents the PR curve of the selected model on the test set (125 instances of left switches and 130 instances of right switches). The gap between curves of 0% IoU and 50% IoU thresholds is minimal depicting good performance by classification network.

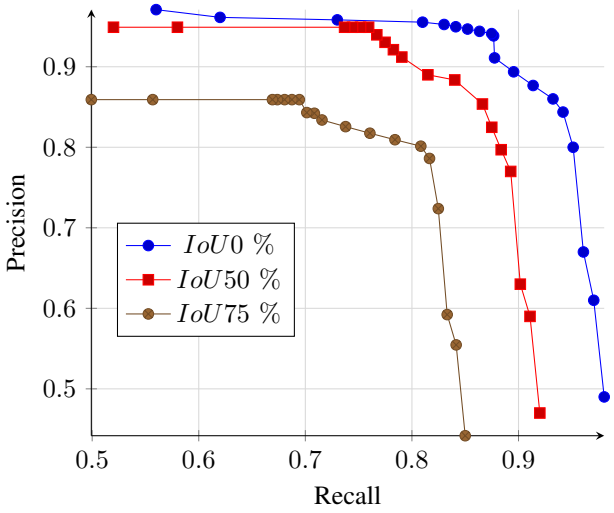
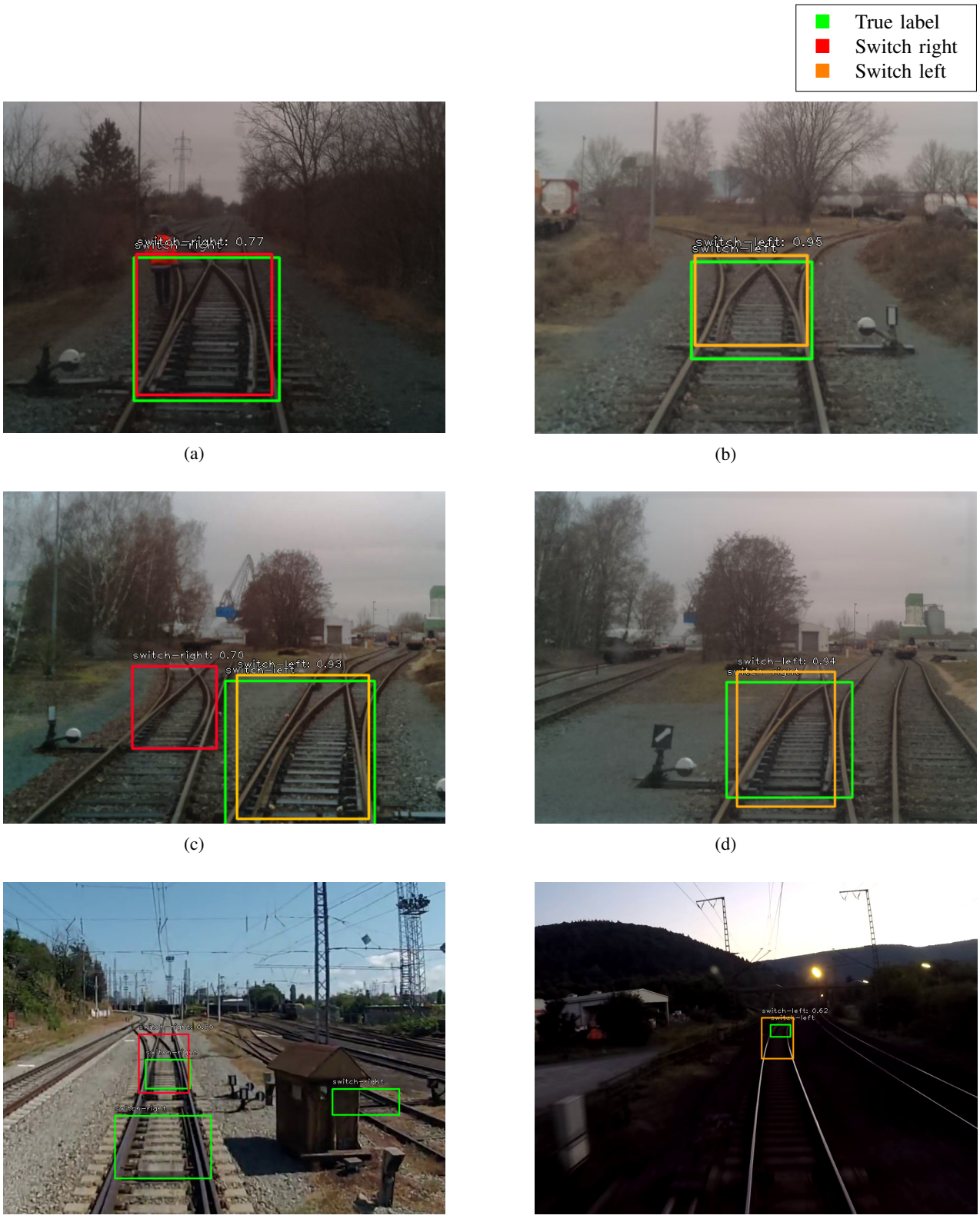


Fig. 3: Precision-Recall Curve of DLR Generated Data Set.

TABLE II: $F1$ Score of DLR Generated Test Set

CT	$F1_0$	$F1_{50}$	$F1_{75}$
0.35	0.911	0.892	0.778
0.40	0.918	0.893	0.770
0.45	0.921	0.898	0.768
0.50	0.890	0.886	0.759
0.55	0.876	0.871	0.741

Tab. II shows the $F1$ score of the test set with the respective IoU and classification thresholds.



(a) (b) (c) (d) (e) (f)

Fig. 4: Images predicted by network trained on self-generated data set at DLR. Fig. (a), (b), and (c) show correctly classified right, left and left switch with a confidence of 77%, 95% and 93%respectively. Fig. (d) shows wrongly classified switch-left. Fig. (e) and (f) are the predictions made on images from RailSem19 data set with a confidence of 90% and 62%.

The classification threshold which better represents the test set is 45 % for all three IoU thresholds. We select the IoU threshold to be at 50 %. At the selected IoU threshold the detection rate of a switch irrespective of its classification is 95.9%, representing the network can distinguish a switch from the straight rail with high accuracy. Fig. 5 shows the confusion matrix for the switch detection.

		Predictions	
		Switch	Not-switch
Ground Truth	Switch	237	13
	Not-switch	0	0

Fig. 5: Confusion matrix for switch detection.

While considering the classification task the precision and recall values at the selected thresholds are (ref: Fig. 3) 0.90 and 0.86, meaning the number of true positives is very high as compared to the false positives and false negatives. The $F1$ score of the test set with the 50 % IoU and 45 % classification thresholds is 0.898.
Fig. 6 shows the confusion matrix for the classification of the switch into left and right.

		Predictions	
		Switch-right	Switch-left
GT	Switch-right	122	8
	Switch-left	18	107

Fig. 6: Confusion matrix for switch classification.

Individually, the switch-left and switch-right classes have an average precision of 93% and 87.41%, respectively. Fig. 7 shows the precision, recall and $F1$ score for each class.

	Precision	Recall	F1
Switch-right	0.87	0.94	0.90
Switch-left	0.93	0.86	0.89

Fig. 7: Precision and recall values for switch classes.

While the mAP among classes is 90.21% with an average inference time of 49.5 ms i.e a prediction speed of 20.2 fps. As the cameras are installed on a shunter locomotive which move with a maximum speed of $20km/h$ the inference speed is good enough to deploy the approach real-time. The approach can be extended to high speed railway vehicles too with the help of advance GPU.
Fig. 4 shows the predictions made by the network on the test set are shown.Fig. 4(a) and 4(b) show correctly classified right and left switch respectively. Fig4(c) shows not only correctly classified switch left but also detects a switch right which was not labeled as we are interested in the driven rail track only. However, Fig 4(d) shows wrongly classified switch-left instead of switch-right as predicted anchor box fails to include the gap between the rails altogether. Fig 4(e) and 4(f) are the predictions made on RailSem19 data set. The network (trained on DLR data set) performs well when RailSem19 and our labeling policies match.

V. Conclusion

In this paper, we were able to detect and classify the railway switches, with accuracy, using transfer learning, anchor box optimization, and appropriate network selection to address existing challenges like the lack of suitable training data, and class imbalance while maintaining a good balance between prediction accuracy and inference time. This work resulted in the high accuracy of classification (90.21 %) and even higher accuracy (95.9 %) for the detection of a switch from the background by utilizing only 2000 images during the training phase. The high inference speed (20 fps enables the presented approach to be deployed for online applications.
In the future, we aim to compare the accuracy and inference speed using other state-of-art approaches like Faster R-CNN [5]. We also aim to improve the classification accuracy by benefiting and combining both the data sets while ignoring the smaller anchor boxes from RailSem19 [12] data set. We intend to deploy the gained insights for the detection of other railway objects too, to gain useful insights into the railway environment and profit from the deep learning methods in the railway domain.

Acknowledgment

This research is part of the project 5G Real-World Laboratory. The 5G Real-World Laboratory has received funding from the Federal Ministry of Transport and Digital Infrastructure.

References

[1] Jürgen Janicki, "DB Manual: Railway system knowledge – How the German rail system works," Tech. Rep., 2018.

[2] Robert Schuil Eric Baars Joern Christoffer Groos Daniela Narezo Guzman, Edin Hadzic, "Turning data driven condition now- and forecasting for railway switches into maintenance actions," *In: Transport Research Arena 2018 Conference Proceedings*, 2018.

[3] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu, "A Survey on Deep Transfer Learning," *The 27th International Conference on Artificial Neural Networks (ICANN 2018)*, 2018.

[4] Joseph Redmon and Ali Farhadi, "YOLOv3: An Incremental Improvement," *Computer Vision and Pattern Recognition (CVPR)*, p. 6, 2018.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *NeurIPS 2015*, Jan. 2016.

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal Loss for Dense Object Detection," *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] Mehmet Karakose, Erhan Akin, and Orhan Yaman, "Detection of Rail Switch Passages through Image Processing on Railway Line and use of Condition-Monitoring Approach," *International Conference on Advanced Technology and Sciences (ICAT)*, 2016.

[8] Canan Taştimur, Mehmet Karaköse, and Erhan Akın, "A Vision Based Condition Monitoring Approach for Rail Switch and Level Crossing using Hierarchical SVM in Railways," *International Journal of Applied Mathematics, Electronics and Computers*, pp. 319–319, 2016.

[9] Wasim Ahmad, *Artificial Intelligence-based Condition Monitoring of Rail Infrastructure*, Ph.D. thesis, University of Twente, 2019.

[10] Jinbeum Jang, Minwoo Shin, Sohee Lim, Jonggook Park, Joungyeon Kim, and Joonki Paik, "Intelligent Image-Based Railway Inspection System Using Deep Learning-Based Object Detection and Weber Contrast-Based Image Comparison," *Sensors (MDPI)*, vol. 19, no. 21, 2019.

[11] Hang Yin and Christian Berger, "When to Use What Data Set for Your Self-Driving Car Algorithm: An Overview of Publicly Available Driving Datasets," in *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, 2017, pp. 1–8.

[12] Oliver Zendel, Markus Murschitz, Marcel Zeilinger, Daniel Steininger, Sara Abbasi, and Csaba Beleznai, "RailSem19: A Dataset for Semantic Rail Scene Understanding," *Computer Vision and Pattern Recognition (CVPR)*, p. 9, 2019.

[13] Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Tech. Rep., 2009.

[14] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár, "Microsoft COCO: Common Objects in Context," *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, p. 8.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[18] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[19] Ross Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[20] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang, "Anchor Box Optimization for Object Detection," *Computer Vision and Pattern Recognition*, 2020.

[21] Andrew Ilyas Aleksander Madry Shibani Santurkar, Dimitris Tsipras, "How does batch normalization help optimization?," *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

[22] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," *ICLR 2015*, 2015.

[23] Jesse Davis and Mark Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, Pittsburgh, Pennsylvania, 2006, pp. 233–240.