Bounding Box Dataset Augmentation for Long-range Object Distance Estimation

Marten Franke, Vaishnavi Gopinath, Chaitra Reddy, Danijela Ristić-Durrant, Kai Michels
Institute of Automation, University of Bremen
Otto-Hahn-Allee 1, 28359 Bremen, Germany
franke@iat.uni-bremen.de

Abstract

Autonomous long-range obstacle detection and distance estimation plays an important role in numerous applications such as railway applications when it comes to locomotive drivers support or developments towards driverless trains. To overcome the problem of small training datasets, this paper presents two data augmentation methods for training the ANN DisNet to perform reliable long-range distance estimation.

1. Introduction

Vision-based *obstacle detection* (OD) followed by distance estimation is crucial for numerous safety critical applications involving moving elements, ranging from robotic manipulators to driven and driverless vehicles. Object distance estimation in driving applications can provide vital information for a vehicle to avoid collisions and adjust its speed for safe driving. The distance range is determined by specific challenges of an application. In railway, the only way for a train to avoid a collision is to come to a complete stop before making contact with an obstacle, because it cannot change its path of travel. So collision avoidance is only possible if the detection distance exceeds the train's stopping distance. The exact train stopping distance depends on different factors such as, among others, the mass distribution of the train, the speed of the train when the brakes are applied and the deceleration rate available with maximum brake application [1]. For example, according to national regulations in most EU countries, the stopping distance of a freight train pulling the 2000 t cargo for the speed of 80 km/h is approximately 700 m. This long-range stopping distance represents a specific challenge of trains when compared to road vehicles.

As a result of developments of Artificial Intelligence (AI), in recent years, there has been an expansion in research and development of machine learning-based methods for OD for rail transport [2][3]. However, the majority papers published so far on OD in railways are concerned with OD only and they do not explicitly discuss estimation of distances between individual detected obstacles and on-board cameras, although this is recognized as an important function. For example, in the work presented in [4], an on-board thermal camera was used which had a distance range of up to 1500 m. The paper presents results of OD within the camera visibility range, on the rail tracks' portions visible in the camera image. However, no details are given on the estimation of distances to individual detected objects. The only known published work explicitly describing obstacle distance estimation with an on-board vision system is presented in [5]. The main part of this on-board OD system is the *artificial neural network* (ANN)-based distance estimator named *DisNet* that estimates the distance between each detected object in the camera images and the on-board camera, using the features of the object's *Bounding Box* (BB) extracted by the *deep learning* (DL)-based object detector. The initial evaluation tests performed in an operational railway environment demonstrated that this integrated on-board vision-based extended the state-of-the-art by providing long-range OD and identification of obstacles in the mid-range (from 80 m up to beyond 200 m) and in the long-range (up to 1000 m) [6]. However, these initial tests indicated also a need for further improvement of reliability of OD and accuracy of distance estimation. In this paper, the improvement of both parts of *DisNet* system, OD and distance estimation, is presented. The OD improvement is achieved by transfer learning of applied DL-based object detector using custom long-range railway dataset. The distance estimation improvement is achieved by re-training of the *DisNet* network with an augmented BB dataset.

2. Related work

Traditionally, object distance estimation in computer vision is most commonly done using stereo-vision, in which depths are estimated by triangulation using the two stereo images and solving stereo correspondence problems [7]. For short-range distance estimation applications, stereo vision provides satisfactory results in spite of several usual shortcomings such as unreliable stereo correspondence so-

lution in textureless image regions. For example, in [8] an error of 6 cm on a target 30 m from the current vision system with a stereo-vision system was achieved in the automated aerial refuelling application. Achieved error is an acceptable error in considered application where the object detection distance range, distance from the cameras on a tanker to the refueling contact point, is a short one, about 30 m.

However, stereo vision-based distance estimation is characterized with inaccuracy in estimation of larger distances [9]. In order to overcome problems of stereo vision-based distance estimation, a number of authors have proposed solutions based on monocular cameras. Monocular depth estimation has been considered by many *convolutional neural network* (CNN) methods as a method presented in [10]. The authors proposed a fully convolutional architecture, encompassing residual learning, to model the ambiguous mapping between monocular images and depth maps. However, in this as well as in other related work the analyzed depth range was up to 70 m – 80 m.

In the railway domain, characterized with a need for long-range distance estimation, the only research found using a monocular camera is *DisNet* [5]. As explained above *DisNet* is a distance regressor that studies the geometric relation that maps an object BB with a certain width and height, which is extracted by a CNN-based object detector, to a distance value. *DisNet*-based distance estimation method was developed by the authors after their experimental confirmation that the distance estimation error was larger for the stereo camera system with a longer baseline, as the calibration error of the system with the longer baseline was bigger than the calibration error for the system with a shorter baseline. This was proved to be particularly problematic for the applications where long-range OD is needed, such as railway applications, as a stereo system with a longer baseline is needed for long-range distance estimation [11]. This paper further demonstrates the usefulness of *DisNet* in long-range OD and presents a novel method for improvement of *DisNet* performance by re-training the initial distance regressor with an augmented long-range BB dataset.

3. DisNet: object bounding box-based distance estimation

A machine learning-based method named *DisNet* was developed in the project SMART (Smart Automation of Rail Transport) [12] to support autonomous long-range obstacle detection by providing direct estimation of the distances between the imaged objects and the monocular onboard cameras mounted on the front of the train. The *DisNet*-based object distance estimation system consists of two parts as illustrated in Figure 1. The first part is deep learning-based OD and the second part is ANN-based distance estimation named *DisNet*. The details of the development of both parts are given in [5]; in the following an overview of the complete system is given.

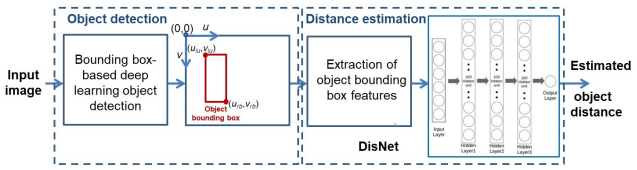


Figure 1: *DisNet*-based object distance estimation system

The main concept in *DisNet* is the ANN-based learning of the relationship between the size of the object in the camera image and the distance of the object from the camera. The size of an object in the image is expressed by the size of the so-called object BB, which is the smallest rectangular box containing the object area in the image. Bearing this in mind, the object detector in the *DisNet* system (illustrated in Figure 1) can be any BB-based DL method which extracts the BB of an object detected in the input image as well as the object class, such as different variants of YOLO [13] and CenterNet [14]. In this paper, the *DisNet*-based system that includes YOLO release YOLOv3 [15] as the object detector is considered. The main advantage of YOLO is its speed, making it appropriate for real-time applications such as OD in railways, which was the main reason for its selection for the SMART OD system and for the work presented in this paper.

The distance estimation part in the *DisNet* system (illustrated in Figure 1) is a feed-forward ANN named *DisNet*. It consists of three hidden layers, each containing 100 hidden units. The *DisNet* input layer consists of six neurons corresponding to six features, which are the parameters of the object BB extracted by the implemented YOLO-based object detector. The output layer has one neuron. The output of this neuron is the estimated distance between the camera and the object in the image which was detected by the object detector and which was bounded with the object BB. For the training of *DisNet*, a supervised learning technique was used. This method required a dataset including both inputs and outputs (outputs being the ground truth distances). The initial training of *DisNet* was done by using the parameters of manually extracted BBs of 2000 objects (of the classes *person* and *car*), which were in the distance range 0 m – 60 m from the static camera (camera mounted on a test-stand), as the inputs. The training outputs, that is the ground truth distances, were real distances between the camera and the objects in dataset measured by a laser scanner. For the purpose of this paper, the initially trained *DisNet* is named *Initial DisNet* as opposed to the *DisNet* system, which was re-trained with an augmented long-range BB-dataset as explained in Section 4.

The YOLO model, used for OD in presented *DisNet*

system, was originally trained with the Microsoft COCO dataset [16] of images of everyday scenes containing common objects in their natural context, consisting of 328,000 images of 80 easily recognizable objects classes. In total, 2.5 million objects are labeled in the images of the dataset, and about 3500 images of railway scenes are labeled with the object class *train*. However, COCO dataset does not contain images of explicit railway scenes with objects on the rail tracks and, moreover, it does not contain images of distant objects. In order to enable the YOLO model to detect objects in railway scenes, with particular focus on distant objects, the following was done:

- A custom *long-range dataset* was generated specific to the railway environment consisting of objects in the long-range (0 m - 1000 m) from the cameras;
- Using the generated custom long-range dataset COCO-trained YOLO model was re-trained using transfer learning. In total, 998 images captured with SMART RGB cameras were used, with 2238 labeled objects of class *person*, multiple classes of vehicles and multiple classes of animals. These images were recorded in the dataset generation field tests as described in the following.

3.1. Custom long-range dataset generation

(a) (b)

Figure 2: SMART field tests for dataset generation: (a) Test-stand with the SMART sensors viewing the rail tracks and an object (of class *person*) on the rail track; (b) SMART vision sensors for obstacle detection integrated into sensors' housing mounted on the frontal profile of a locomotive below the headlights.

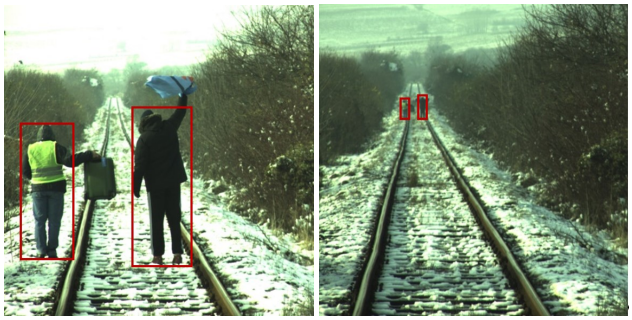
In order to collect relevant high-quality and high-volume training data to re-train the COCO-trained YOLO model for obstacle detection in railways, during the lifetime of the project SMART two types of field tests were performed on the Serbian railways' test sites, static and dynamic [12]. The static field tests were conducted on the location of the straight rail tracks in length of about 1100 m in different times of the day and night and in different weather conditions in November 2017, March and November 2018 and March 2019. During the static field tests, the cameras were mounted on a static test stand and the members of the SMART Consortium imitated potential static and dynamic (movable) obstacles (pedestrians) on the rail tracks located on different distances from the SMART test-stand, as illustrated in Figure 2a. In dynamic field test performed in July 2018, the cameras were integrated into sensors' housing mounted on the front profile of an operational locomotive (Figure 2b). The test was performed with an in-service train of the operator Serbia Cargo, pulling 21 wagons with total mass of 1194 t and total train length of 458 m. The test length was 120 km on the Serbian part of the pan European corridor X.

Table 1: Dataset structure

Image frame No.	Object class	Top left corner	Bottom right corner	BB width	BB height	BB diagonal

Figure 3: Example images from custom long-range railway RGB dataset recorded in dynamic field tests. Different object classes on/ near the rail tracks (humans, different vehicles, animals).

The SMART dataset comprises approximately eight hours of video recorded by the train mounted SMART OD system in the dynamic field tests and six hours of video recorded by test stand mounted SMART cameras in the static field tests. The series of recorded videos were converted into sequential frames of images. In the dataset images both static and moving obstacles are present, including humans, vehicles, bicycles and animals. Some examples of dataset images recorded in dynamic field tests are shown in Figure 3 and some examples of dataset images recorded in static field tests are shown in Figure 4. As given in Table 1, each object in the dataset is described by a number of parameters, including the information about the class of the object and the information about object BB. (x_{ul}, y_{ul}) and (x_{br}, y_{br}) are respectively image coordinates of the left up-



(a) (b)

Figure 4: Example images from custom long-range railway RGB dataset recorded in static field tests. Objects from the class *person* on 60 m (left) and on 600 m (right) from the cameras mounted on a test-stand.

per corner of the object BB and image coordinates of the right bottom corner of the object bounding box. The obstacle distance range covered by recordings in dynamic field tests was determined by the real-world operational environment. It is important to note that during the dynamic field tests, because of the rail-tracks configuration, there were no segments longer than 600 m viewed with on-board cameras were the obstacles could be recorded, so that distance range covered by SMART real-world railway dataset covers distances up to 600 m. Considering the images from the static field test, described above, the total distance range covered by generated custom long-range dataset is 0 m – 1000 m.

3.2. Transfer Learning of object detector

For the purpose of re-training of the COCO-trained YOLO model using the transfer learning and generated long-range dataset, the annotated images have been segregated into training and testing image sets using randomization in the ratio 4:1.

Transfer learning works on the principle of using the knowledge gained by a model to solve problem *X*, to be used to solve another problem *Y*. In the presented work, the initial YOLO model trained with COCO dataset for the purpose of general OD was re-trained to detect objects in railway environments with long distance range of up to 1000 m.The performed procedure of transfer learning consisted of four steps as follows:

1. The COCO-trained YOLOv3 model was loaded, along with it's weights, which was efficient in detecting general objects of 80 classes in COCO dataset. The existing YOLO model's lower layers, called the *body*, focused on detecting basic and important features of objects.

2. The last three layers (three layers out of total number of layers 252), which are called the *head* and which are specific to the custom long-range dataset are cut. These layers are replaced by random layers.

3. The added random layers are fine-tuned by training the network on the custom long-range railway dataset while the weights in the *body* (initial lower layers) are frozen.

4. After fine-tuning of the *head*, the entire network is unfrozen and the model is trained again, to allow small weight adjustments throughout the network to obtain a re-trained YOLO model.

The fine tuning of the YOLO model for the purpose of model optimization to obtain better OD results was done by tuning the hyperparameters such as, *batch size*, *number of epochs* and *learning rate*. Hyperparameter optimization finds a set of hyperparameters that yields an optimal model which minimizes a pre-defined loss function on the given test data. This process consisted of following steps:

1. With each iteration one hyperparameter value is varied and the model is trained by keeping the other hyperparameter values constant.

2. The effect of this change is analyzed by measuring the performance of the model using the metric - *Mean Average Precision* (mAP). First, the *Intersection over Union* (IoU) is calculated for each detection and this calculated IoU is used to arrive at the corresponding precision and recall values. The Average Precision (area under precision v/s recall curve) is calculated for all object detections belonging to each object class. The mean of the Average Precisions over all object classes gives the mAP of a model.

3. If there is an improvement in the mAP value, the hyperparameter value is further increased or decreased in the same direction until local maximum is reached.

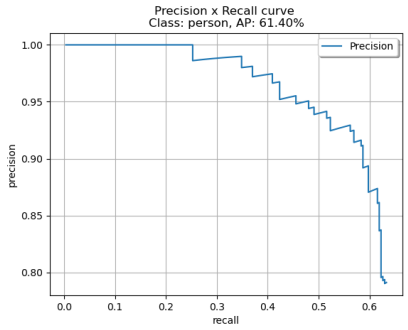
4. The same process is implemented for other hyperparameter values until an optimum set of hyperparameter values producing maximum mAP is obtained.

After performing the above procedure for all three considered hyperparameters, finally the following set of optimal hyperparameters was achieved: *batch size* $= 32$; *number of epochs* $= 200$; *learning rate* $= 1e-3$.

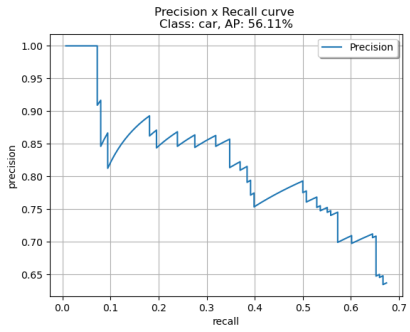
With the achieved optimum set of hyperparameters, the optimized YOLO OD model had mAP values shown in Figure 5 for the object classes *person* and *car*.

4. Data augmentation for re-training of DisNet

Besides the improvement of OD in railway domain through the transfer learning procedure described above, in



(a)



(b)

Figure 5: Precision v/s recall curves for mAP evaluation for object class *person* (a) and *car* (b)

order to improve the reliability of BB-based object distance estimation, re-training of *Initial DisNet* using an augmented BB long-range dataset was performed as explained in following.

As explained in Section 1, the training dataset used for training of *Initial DisNet* consisted of parameters of manually extracted BBs of 2000 objects (of the classes *person* and *car*), which were in the distance range 0 m – 60 m from the static camera (camera mounted on a test-stand), as the inputs. More precisely the BB dataset consists of six-dimensional feature vectors **v** assigned to training samples. The vector **v** assigned to an individual object BB is:

$$\mathbf{v} = [\frac{1}{B_h}, \frac{1}{B_w}, \frac{1}{B_d}, C_h, C_w, C_d]^T \tag{1}$$

In (1), B_h, B_w and B_d are respectively height, width and diagonal of an object BB, which are calculated as:

$$B_h = \frac{v_{ul} - v_{br}}{\text{Image height in px}} \tag{2}$$

$$B_w = \frac{u_{br} - u_{ul}}{\text{Image width in px}} \tag{3}$$

$$B_d = \frac{\sqrt{(v_{ul} - v_{br})^2 + (u_{br} - u_{ul})^2}}{\sqrt{\text{Image height in px}^2 + \text{Image width in px}^2}} \tag{4}$$

where (u_{ul}, v_{ul}) and (u_{br}, v_{br}) are respectively the image coordinates of the left upper corner and the right bottom corner of the object BB, as illustrated in Figure 1. The above BB features are invariant to camera's image resolution, hence *DisNet* can be used with a variety of cameras independently of image resolution. Features C_h, C_w and C_d in (1) are the values of average height, width and depth of an object of a particular class. For example, for the class *human* C_h, C_w and C_d are respectively 175 cm, 55 cm and 30 cm, while for the class *car* these parameters are 160 cm, 180 cm and 400 cm respectively. The features C_h, C_w and C_d represent three-dimensional object features that complement information on object bounding boxes extracted from two-dimensional images and so they give more information to distinguish different objects.

As the initial training dataset covered only distance range 0 m – 60 m accuracy of *DisNet*-based object distance estimation for long-range was not sufficiently reliable. In order to improve performances of *DisNet*-based distance estimation a BB-dataset augmentation was performed to obtain a large training dataset of sufficient diversity in sense of covering broader ranges of distances including long-range distances up to 1000 m. For the creation of synthetic BB-data, two different methods were used:

1. Image transformation-based data augmentation;
2. Projective transformation-based data augmentation.

4.1. Image transformation-based BB data augmentation

The main idea behind the BB-dataset augmentation using image transformations was to obtain the BBs of objects of one class (so-called transformed object class) by transforming the BBs of objects of another class for which the BBs sizes and corresponding ground truth distance were a priory known (so-called reference object class). To illustrate this procedure, here, the object classes *person* and *car* are considered as the reference object class and transformed object class respectively. The reference object class dataset consisted of 264 images recorded in SMART long-range static field test conducted in November 2018; two examples of these images are shown in Figure 4. During this field test, for the purpose of long-range dataset generation, two persons imitated potential obstacles on the straight line track. Starting from the test-stand with mounted RGB cameras, they walked 1000 m along the rail track and back. Every 5 m, while walking in each direction, they gestured distinctively, so that the camera images recorded at the moments of the gestures could be used for the ground truth distance annotations in dataset generation. These camera images were extracted from the whole recorded video and manually drawn BBs of the objects (persons walking along the rail tracks) were labeled with ground truth distances.

Such created BBs were transformed by employing image transformations variations onto the existing images. The performed transformation operations are: rotation, scaling, and translation. Object BBs are extracted from the transformed images and these BBs represent the objects of different geometries and classes at the same ground truth distances as the objects from the reference image. The extent of transformation are applied to the image sequentially in the following order:

1. Rotation factor: The rotation in degrees of an image about the image centre where the anti-clock wise rotation is positive by convention. This determines the orientation of an object in an image.

2. Scaling factor: The extent to which the image is scaled in x and y directions. This is expressed in terms of a vector which determines the size of an object in an image.

3. Translation factor: The translation vector is again a tuple with image translation factors along x and y axes.

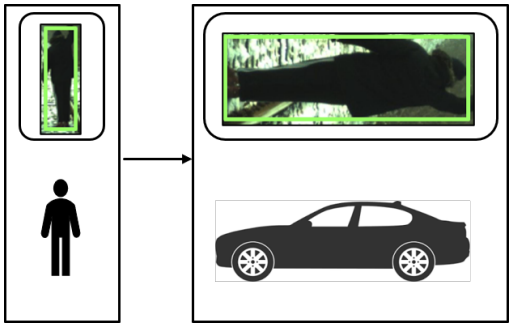
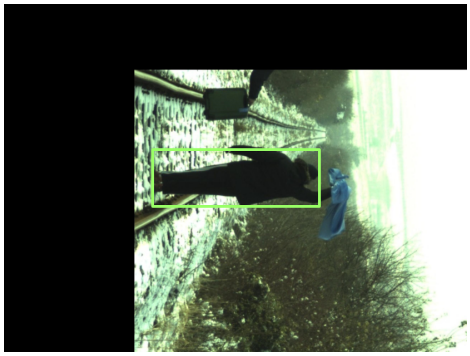


Figure 6: Transformation of BB of an object of class *person* to resemble the BB of the object of class *car*. The shape of the bounding box of a horizontal human figure is scaled along both x and y directions to achieve a bounding box shape equivalent to that of a car. The extent of scaling along the individual axes is determined by the ratio of the objects' real world dimensions.

An example of BB dataset augmentation is given in Figures 6 and 7. The reference image shown in Figure 7a, containing BB of an object from the class *person* at a certain distance d from the camera when the image was captured, was transformed to the image shown in Figure 7b. By applying the transformation metrics: rotation factor = 90°, scaling factors upon rotation = $(2.5, 1)$, translation factors upon rotation and scaling = calculated to translate the bounding box to the image centre. The object BB in transformed image resembles the BB of an object from the class *car* at the same distance d from the camera as illustrated in Figure 6.

(a) Original image



(b) Augmented image

Figure 7: Transformation of reference image (a) with object from class *person* to synthetic transformed image (b)

4.2. Projective transformation-based BB data augmentation

As above explained, BB data augmentation using image transformation enables augmentation of reference (original) BB dataset of one object class by BBs of objects from different classes, which are at the same distances as the original objects. In order to augment BB dataset so to generate synthetic object BBs corresponding to different distances, BB augmentation-based on projective geometry (Figure 8) was performed.

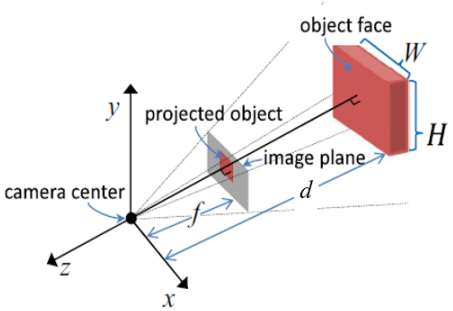


Figure 8: Projective geometry: d - object distance, f - camera focal length, $W\&H$ - real-world dimensions (width height) of the object. Image adopted from [17].

According to projective geometry illustration in Figure 8, the following relationship between the real-world size of the object and the size of the object BB in the image holds:

$$h = f(\frac{H}{d}) \quad \text{and} \quad w = f(\frac{W}{d}) \tag{5}$$

Starting from the priory known sizes of BB of objects from the class *person* of real-world height (1.6 m and 2 m) from the original images from above described static field test and corresponding distances, the focal length f was calculated. Using calculated f and formula (5) the parameters of synthetic BBs of objects from class *person* of real-world heights (1.5 m, 1.6 m, 1.7 m, 1.8 m, 1.9 m, 2 m) were calculated for distances d in the range from 0 m to 1000 m, which were not covered by original dataset.

As the result of BB data augmentation by both of the above described procedures, image transformation and projective transformation, finally an augmented BB dataset of about 10000 BBs with corresponding distances from the range 0 m – 1000 m was obtained. This augmented dataset was used for re-training of *Initial DisNet*. Resulting re-trained *DisNet* is referred to as *DisNet* in the following evaluation section.

5. Evaluation

The evaluation of *DisNet* re-trained with the augmented BB dataset was done on testing images recorded in SMART dynamic field tests (testing images were different from those used for transfer learning of YOLO object detector). In total, 741 BBs extracted by the re-trained YOLO object detector were used for the evaluation of *DisNet*. Out of these 741 BBs, 654 were BBs of objects from class *person* and 87 were BBs of objects from the class *car*. For all 741 BBs the corresponding ground truth distances were known as calculated in dynamic field tests using the GPS coordinates of the moving train and Google maps GPS coordinates of the objects (obstacles) locations (e.g. at crossings and at known locations near railway infrastructure).

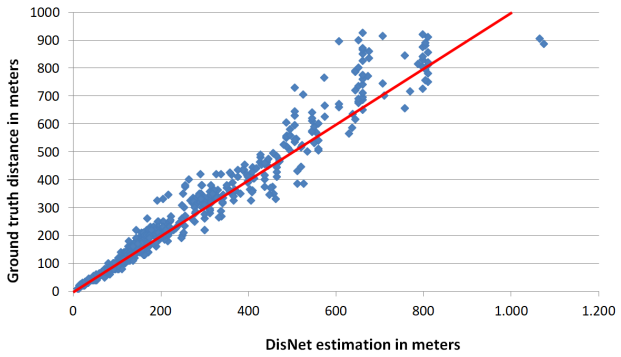


Figure 9: *DisNet* estimation v/s Ground truth distances

The diagram shown in Figure 9 illustrates the extent to which the *DisNet* distance estimations (blue points) differ from the ground truth data (red line). Obviously, the average absolute estimation error of *DisNet* increases with increasing object distance. However, it was calculated that the average relative error of *DisNet* increases slightly with increasing object distance, but is steady about 10%. An error of 10% on long-range distances can be considered as acceptable for obstacle detection in railways, as it would mean that an object at real distance of 800 m would be detected as being at 720 m being still above train braking distance. This underestimation is assumed as it can be seen from Figure 9, in general, *DisNet* tends to underestimate distance rather than overestimate it. In total, in 434 evaluation cases (out of 741) the distance between camera and object was estimated to be lower than the ground truth distance.

In order to evaluate further performance of *DisNet*, the following comparison to *Initial DisNet* performance was performed. A relative error of 10% and 20% were considered as threshold values and the number of distance estimations below 10%, between 10% and 20% and above 20% for different distance ranges were calculated for *DisNet* and *Initial DisNet*. The evaluation results are shown respectively in Figure 10 and Figure 11.

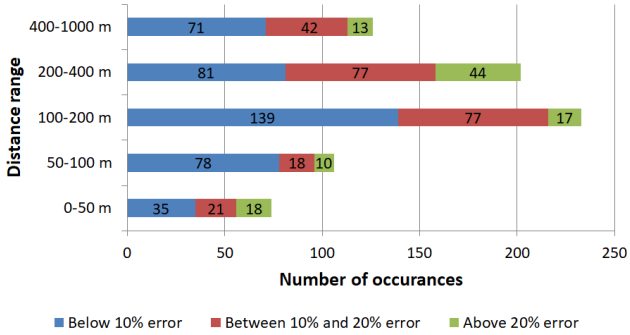


Figure 10: *DisNet* distance estimations below 10%, between 10% and 20%, and above 20% error

Overall, 639 of the total 741 *DisNet* estimations were determined to have a relative error of less than 20%, while for *Initial DisNet* 72.2% of the estimations were above the threshold of 20%. With respect to the 10% threshold, 54.5% were below it for DisNet, but only 14.8% were below it for Initial DisNet. As evident from Figures 10 and 11, *DisNet* outperforms *Initial DisNet* particularly in long-range distance estimations. This is expected result as *Initial DisNet* was trained with BB dataset covering only short range 0 m to 60 m; while *DisNet* was obtained by re-training the *Initial DisNet* with augmented dataset covering range 0 m to 1000 m.

Further evaluation of *DisNet* was performed by comparison of *DisNet* distance estimation results before and after re-training with the augmented dataset based on *RMSE*

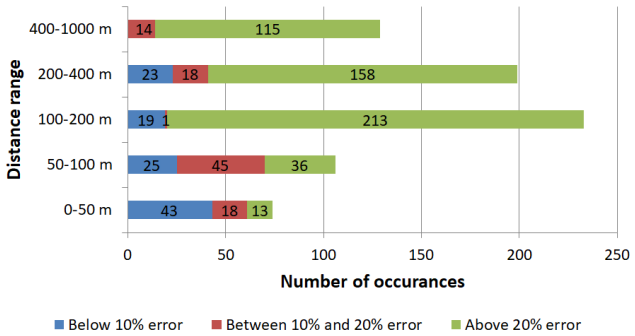


Figure 11: *Initial DisNet* distance estimations below 10%, between 10% and 20%, and above 20% error

(Root mean squared error) of the predicted distances from the ground truth distances as the evaluation metric. The RMSE indicates by how much on average the estimation differs from the ground truth.

$$RMSE = \sqrt{\frac{\sum (D_{est} - D_{GT})^2}{N}} \tag{6}$$

where D_{est} is estimated object distance, D_{GT} is ground truth distance and N is number of identified objects in an image.

Table 2 shows that *DisNet*'s performance for both object classes improved significantly after re-training, reducing the RMSE from 38.2% to 10.9%.

Table 2: RMSE comparison between *Initial DisNet* and *DisNet*

RMSE	Car	Person	**Total**
Initial DisNet	38,20%	38,21%	38,20%
DisNet	13,72%	10,52%	10,90%

An example of long-range OD and distance estimation with the improved *DisNet* system presented in this paper is shown in Figure 12. As obvious, the car in the railway scene that was on real distance of 524 m from the train, was correctly detected and its distance was estimated as of 495,9 m giving a relative error of 5,36%.

6. Conclusion and Outlook

Based on the shown evaluation results it can be concluded that BB dataset augmentation to cover long-range distances contributed significantly improves long-range distance estimation. However, there is still place for improvement so that average relative error is decreased. It can be expected that the re-training of *DisNet* with even larger BB dataset that would be augmented so to uniformly cover

Figure 12: Detected car with ground truth distance of 524 m and *DisNet* estimation of 495.9 m

different distance ranges with larger numbers of samples (BBs) would lead to further improvement. In this context, it would certainly be useful to perform the data augmentation for the selected object classes from different object perspectives (resulting in different ratios with respect to object width and height), so that *DisNet* continues to learn the objects at different distances from different perspectives.

Figure 13: Poor distance estimation of *DisNet* based on occluded part of the object

However, as shown in Figure 13, the *DisNet* approach may also result in poor distance estimation if the detected object is partially occluded and the image coordinates of the object cannot be made out by OD network in its entirety. In the example shown, *DisNet* estimates the occluded car at a distance of 614.3 m, while the ground truth distance is 385 m. If the "correct" BB size of the car were given to *DisNet*, a good distance estimation of 345.9 m (9.45% error) would result. A future approach could be to use traditional image processing to check if the object is completely visible in the found BB and if necessary to estimate the real BB coordinates including the occluded object's part.

Acknowledgement

This research received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation program under Grant No. 881784.

References

[1] D. Barney, D. Haley and G. Nikandros. Calculating Train Braking Distance. In Proceedings of the *Safety Critical Systems and Software (SCS01), 6th Australian Workshop on Safety-Related Programmable Systems*, St Lucia, Queensland, Australia, 6 July 2001; pp. 23-30.

[2] M. Yu, P. Yang, S. Wei. Railway obstacle detection algorithm using neural network. In Proceedings of the *6th International Conference on Computer-Aided Design, Manufacturing, Modeling and Simulation* (CDMMS 2018), Busan, South Korea, 14-15 April 2018; pp. 0400171–0400176, doi: 10.1063/1.5039091.

[3] T. Ye, X. Zhang, Y. Zhang, J. Liu. Railway Traffic Object Detection Using Differential Feature Fusion Convolution Neural Network. In *IEEE Transactions on Intelligent Transportation Systems* 2020, 22(3); 1375-1387, doi: 10.1109/TITS.2020.2969993.

[4] R. Kapoor, R. Goel, A. Sharma. Deep Learning Based Object and Railway Track Recognition Using Train Mounted Thermal Imaging System. In *Journal of Computational and Theoretical Nanoscience* 2018, 17(11); pp. 5062–5071, doi: 10.1166/jctn.2020.9342.

[5] M. A. Haseeb, J. Guan, D. Ristić-Durrant and A. Gräser. A Novel Method for Distance Estimation from Monocular Camera. In Proceedings of the *10th Planning, Perception and Navigation for Intelligent Vehicles* (PPNIV18), IROS, Madrid, Spain, 1 October 2018; Volume 10.

[6] D. Ristić-Durrant, M. A. Haseeb, M. Franke, M. Banić, M. Simonović and D. Stamenković. Artificial Intelligence for Obstacle Detection in Railways: Project SMART and Beyond. In *Dependable Computing – EDCC 2020 Workshops*; Bernardi, S., Vittorini, V., Flammini, F., Nardone, R., Marrone, S., Adler, R., Schneider, D., Schleiss, P., Nostro, N., Løvenstein Olsen, R., Di Salle, A., Masci, P., Eds.; Springer: Cham, Switzerland, 2020; pp. 44-55, doi: 10.1007/978-3-030-58462-7_4.

[7] A. Leu, D. Aiteanu, A. Gräser. High Speed Stereo Vision Based Automotive Collision Warning System. In *Applied Computational Intelligence in Engineering and Information Technology*; Precup, R.-E., Kovács S., Preitl S., Petriu E., Eds.; Springer: Berlin, 2012; pp. 187-199, doi: 10.1007/978-3-642-28305-5_15

[8] A. Lee, W. Dallmann, S. Nykl, C. Taylor and B. Borghetti. Long-Range Pose Estimation for Aerial Refueling Approaches Using Deep Neural Networks. In *Journal of Aerospace Information Systems* 2020, 17(11); pp. 634-646, doi: 10.2514/1.I010842.

[9] P. Pinggera, D. Pfeiffer, U. Franke, R. Mester. Know Your Limits: Accuracy of Long Range Stereoscopic Object Measurements in Practice. In *Computer Vision — ECCV 2014, Part II*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 96–111, doi: 10.1007/978-3-319-10605-2.

[10] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In Proceedings of the *2016 Fourth International Conference on 3D Vision* (3DV), Stanford, CA, USA, 25-28 October 2016; pp. 239-248, doi: 10.1109/3DV.2016.32.

[11] D. Ristić-Durrant, M. A. Haseeb, D. Emami, A. Gräser, V. Nikolić, I. Ćirić, M. Banić, B. Brindić, D. Nikolić, D. Radovanović, F. Eßer, C. Schindler. SMART concept of an integrated multi-sensory onboard system for obstacle recognition. In Proceedings of the *7th Transport Research Arena* (TRA) 2018, Vienna, Austria, 16-19 April 2018.

[12] SMART project. Available online: `http://www.smartrail-automation-project.net/` (accessed on 29 May 2020).

[13] J. Redmon, S. Divvala, R. Girshick and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR2016), Las Vegas, NV, USA, 27-30 June 2016; pp. 779-788, doi: 10.1109/CVPR.2016.91.

[14] X. Zhou, D. Wang and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. Available online: `https://arxiv.org/abs/1904.07850v2` (accessed on 29 July 2021).

[15] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv: 1804.02767*, 2018. Available online: `https://arxiv.org/pdf/1804.02767.pdf` (accessed on 29 July 2021).

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision — ECCV 2014, Part V*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[17] H. Hachiya, Y. Saito, K. Iteya, M. Nomura and T. Nakamura. Distance estimation with 2.5D anchors and its application to robot navigation. In *ROBOMECH* 2018, 5(1); pp. 1-13, doi: 10.1186/s40648-018-0119-5.