A deep learning based autonomous distance estimation and tracking of multiple objects for improvement in safety and security in railways

Muhammad Abdul Haseeb[1]
haseeb@iat.uni-bremen.de

Guan Jianyu[1]
jianyu@uni-bremen.de

Danijela Ristić -Durrant[1]
risitc@iat.uni-bremen.de

Axel Gräser[1]
ag@iat.uni-bremen.de

[1]Institute of Automation
University of Bremen
Bremen, Germany

Abstract

In this paper, a novel approach for multiple object tracking and distance estimation from an on-board monocular camera, aiming at improvements in the safety and security of railways, is presented. The approach is based on deep learning architecture using a deep Convolutional Neural Network (CNN) object detection followed by a multi hidden-layer Gated Recurrent Neural Network (RNN) referred as DisNet-RNN Tracker, which consists of two sub-networks for distance estimation and bounding box prediction respectively. The DisNet-RNN Tracker learns and estimates the distance between the detected object and the camera sensor, and predicts the object bounding box based on sequential input from previous and current detection. The presented DisNet-RNN Tracker tracks multiple objects in case where object detection module fails to detect object. The presented method is evaluated on the real-world railway dataset recorded with the on-board Obstacle Detection System developed within a H2020 Shift2Rail project SMART - Smart Automation of Rail Transport. The presented work has potential to benefit other applications where reliable object detection, tracking and long-range distance estimation is needed such as autonomous cars, transportation and public security.

1 Introduction

In the last decade, the impact of digitalization on the railway sector has become evident. Digital technology is disrupting different components of railway operations including smart monitoring and surveillance systems that will change the way operators manage hazards and intrusions. A potential benefit of the digitalization is an improvement in safety and security thanks to obstacle and track intrusion detection as it will allow railways to address various types of risks in a smarter and more systematic way [1].

Following significant progress in development and implementation of reliable autonomous obstacle detection systems for autonomous cars, there is a tendency to use experience, knowledge and methods from the automotive sector for the development of autonomous on-board obstacle detection and track intrusion systems for future autonomous trains [2][3]. Although the main principals applied to obstacle detection for autonomous cars can be used, the autonomous detection and tracking in railways have to meet specific requirements. Some of these requirements are on-board long-range detection of obstacles on the rail tracks and detection of track intrusions (detection and tracking of objects in the track area [4]).

This paper is organized as follows. Section 2 introduces the related work in the field of vision-based object detection and tracking with emphasis on challenging conditions. Section 3 describes the methodology and explanation of proposed work for multiple object tracking and distance estimation. Section 4 presents the results and evaluation of the proposed approach. Finally, Section 5 presents the conclusions.

2 Related Work

Environmental perception system, which performs object detection and tracking task, is the core for autonomous vehicles. Object detection tremendously evolves along with the advancement in sensor technology, and with the development of classical and machine learning-based algorithms, while multiple object tracking seems less developed [5]. Object tracking is essential for many tasks of autonomous driving such as obstacle avoidance and intention prediction [6]. It is a critical task and it becomes more challenging for situations such as objects at far distance, low frame rate video sequence, frequent occlusion, camera vibration or movement and so on.

Mainly there are four object tracking methods categorized as region-based tracking [7], model-based tracking [8], contour-based tracking [9] and feature-based tracking [10]. All those methods rely on object detection. In general, tracking utilizes the detection information from previous frames to predict the detection in frames where detection is missing.

In [6] a traditional object tracking method based on mono camera for autonomous vehicles is present. Using the camera model to map pixel position into distance, the distance to the vehicles with respect to vehicle was measured. Further Kalman Filter (EKF) used to refine distance accuracy and track detected vehicles. The results show that method is capable to track 3D positions with sufficient accuracy.

Relatively modern machine learning-based methods [11] are introduced for object tracking based on Recurrent Neural Networks (RNNs) followed by long short-term memory (LSTM) and Gated recurrent units (GRU). The RNN based networks are often used for sequential data and thus also applicable for object tracking in video sequences.

3 DisNet-RNN Tracker: Robust object distance estimation and multiple objects tracking from a monocular camera

The workflow of DisNet-RNN Tracker based object detection, tracking and distance estimation from a single monocular camera is illustrated in Figure. 1. The frames captured by an RGB monocular camera are inputs to Object Detector Module. Different object detectors, which outputs bounding box and class of detected objects, can be integrated into the system. The resulted object bounding boxes from detection module further feed into Multiple Objects Mapping (MOM) module. The object mapping module matches previous objects detection results to the current detection results for the sake of objects tracking and assigning unique IDs for unmatched or newly detected objects. Further, the Features Calculation module extracts features of the objects bounding boxes and based on those features DisNet-RNN Tracker estimates object distance at current frame and predicts object bounding box in the following frame. In the system architecture in Figure 1, an example of the distance estimation and bounding box prediction based on prior detection information for a human walking along the rail tracks is shown.

In this paper, YOLO object classifier [12] trained with COCO dataset [13] is considered as an object detector module, whereas any other state-of-the-art object detector can be used. However, no matter which state-of-the-art object detection module is used, false detection or unprecise bounding boxes extraction cannot be avoided in cases such as object partially or fully occluded, object shadow, change in image quality due to illumination and the similarity of object texture or colour with the background. This problem is unfavourable for those applications where high reliability is demanded such as obstacle detection system for autonomous vehicles. DisNet-RNN Tracker, proposed in this paper, aims at reliable overall object distance estimation and object tracking in spite of failure of intermediate object detector module.

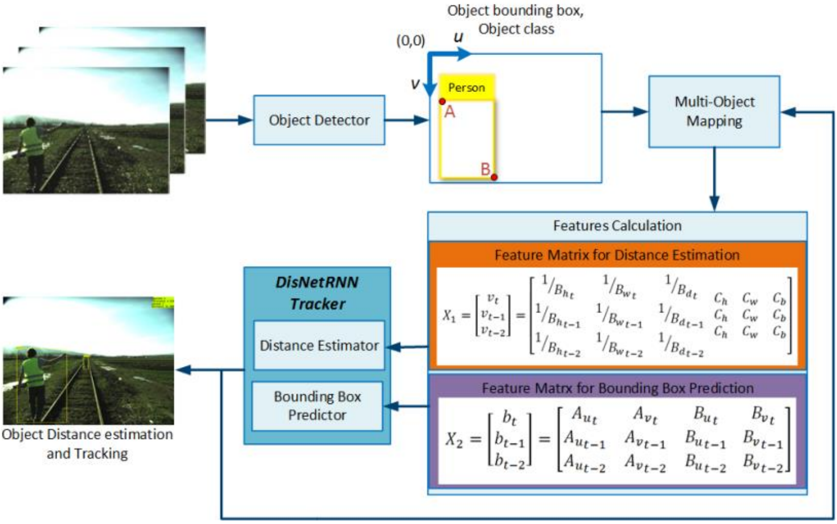
Object bounding box,
Object class
(0,0)
u
v
Person
A
B
Object Detector
Multi-Object Mapping
Features Calculation
Feature Matrix for Distance Estimation
$X_1 = \begin{bmatrix} v_t \\ v_{t-1} \\ v_{t-2} \end{bmatrix} = \begin{bmatrix} {}^1/_{B_{h_t}} & {}^1/_{B_{w_t}} & {}^1/_{B_{d_t}} & C_h & C_w & C_b \\ {}^1/_{B_{h_{t-1}}} & {}^1/_{B_{w_{t-1}}} & {}^1/_{B_{d_{t-1}}} & C_h & C_w & C_b \\ {}^1/_{B_{h_{t-2}}} & {}^1/_{B_{w_{t-2}}} & {}^1/_{B_{d_{t-2}}} & C_h & C_w & C_b \end{bmatrix}$
Feature Matrx for Bounding Box Prediction
$X_2 = \begin{bmatrix} b_t \\ b_{t-1} \\ b_{t-2} \end{bmatrix} = \begin{bmatrix} A_{u_t} & A_{v_t} & B_{u_t} & B_{v_t} \\ A_{u_{t-1}} & A_{v_{t-1}} & B_{u_{t-1}} & B_{v_{t-1}} \\ A_{u_{t-2}} & A_{v_{t-2}} & B_{u_{t-2}} & B_{v_{t-2}} \end{bmatrix}$
DisNetRNN Tracker
Distance Estimator
Bounding Box Predictor
Object Distance estimation and Tracking

Figure 1: DisNet-RNN Tracker based object distance estimation and tracking system from a monocular camera.

3.1 DisNet-RNN Tracker Architecture

The DisNet-RNN Tracker consists of two independent sub-networks based on Recurrent Neural Network (RNN) architecture. The reason to use RNN is that due to its unique characteristic of being suitable to work with sequential data and its memory cell able to preserve information from inputs provided in previous time moments [11]. In the scenario considered in this paper, two types of object detection failures are possible: the target objects are not detected in some frames, and the bounding boxes of some of the detected objects are not accurate. Using the RNN, which has a memory of previous inputs, can help to predict the object position and estimate distance more reliably. DisNet-RNN

Tracker uses the sequential data from previous two-time steps to improve the estimation of object distance at current time step and predicts the position and size of object bounding box in the next time step.

Figure 2 shows the DisNet-RNN Tracker architecture. The architecture consists of two subnetworks represented with two main blocks one under the other in Figure 2. The upper network is used to estimate the object distance named as *distance estimator* and the lower network named as *bounding box predictor* is used to predict the top left corner A, and the bottom right corner B, of the object bounding box. Prediction of these two bounding box corners' points relates to prediction of the bounding box position and size.

As shown in Figure 2, a three hidden layers network was adopted for the distance estimation. A deep recurrent neural network is stacked with Gated Recurrent Unit (GRU) layers and the output from the last GRU layer is connected to a fully connected output layer to perform final distance estimation. For the bounding box prediction, the network consists of a single hidden layer of GRUs with a fully connected layer as an output layer. A new loss function in training *distance estimator* network was defined and it is given in (1). This loss function calculates losses from all distance prediction results from three time-steps and in a similar way, the Mean Absolute Error (MAE) loss function for *bounding box predictor* network was defined given in (2).

$$loss\ function = \frac{1}{3n}\sum_{i=1}^{3}\sum_{j=1}^{n}\left|Y_{ij-true} - Y_{ij-estimated}\right| \tag{1}$$

$$\begin{aligned} loss\ function = \frac{1}{3n}\sum_{i=1}^{3}\sum_{j=1}^{n}&\left|Au_{ij-true} - Au_{ij-predicted}\right| + \left|Av_{ij-true} - Av_{ij-predicted}\right| \\ &+ \left|Bu_{ij-true} - Bu_{ij-predicted}\right| + \left|Bv_{ij-true} - Bv_{ij-predicted}\right| \end{aligned} \tag{2}$$

- n : training data numbers
- $Y_{ij-true}$: the ground truth distance of j_{th} training data at time step *i*.
- $Y_{ij-estimated}$- the estimated distance of j_{th} training data at time step *i*.
- $Au_{ij-true}$, $Av_{ij-true}$: the top-left corner coordinates of ground truth object bounding box of j_{th} training data at time step *i*.
- $Bu_{ij-true}$, $Bv_{ij-true}$: the bottom-right corner coordinates of ground truth object bounding box of j_{th} training data at time step *i*.
- $Au_{ij-predicted}$, $Av_{ij-predicted}$: the top-left corner coordinates of predicted object bounding box of j_{th} training data at time step *i*.
- $Bu_{ij-predicted}$, $Bv_{ij-predicted}$: the bottom-right corner coordinates of predicted truth object bounding box of j_{th} training data at time step *i*.

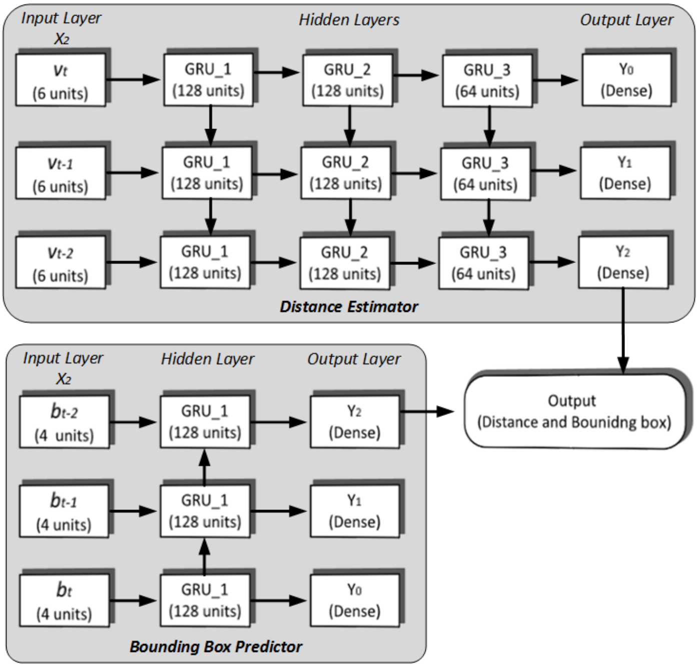

Figure 2: DisNet-RNN Tracker Architecture consists of two sub networks namely distance estimator on top and bounding box predictor in bottom

3.2 Dataset Preparation

Recurrent Neural Networks requires sequential dataset which needs to be prepared before training DisNet-RNN Tracker. As described earlier, DisNet-RNN Tracker consists of two sub independent RNN networks. In order to train and test these subnetworks, manually a dataset of size of 8000 sequential inputs was created using images recorded in real-world railways scenarios. Each sample of dataset represents extracted features from three subsequent frames. The annotation tool [14] provides the object bounding boxes coordinates, which were labelled together with ground truth distance. The ground truth for distance estimation network was recorded during the dataset generation using the GPS positioning system which later in offline phase allows calculating relative distance between train and objects. Whereas for bounding box prediction, the manually drawn bounding box on fourth frame considered as a ground truth as shown in Figure 3.

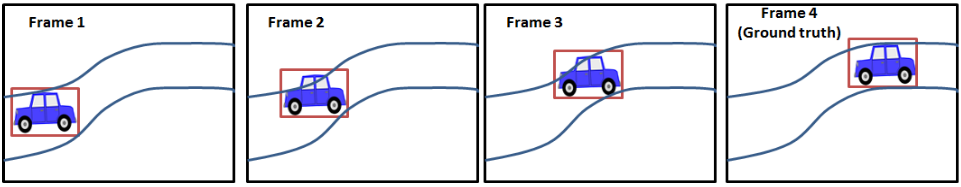

Figure 3: Dataset generation for bounding box prediction

In next section it is explained how the features were calculated and dataset was organized to train the DisNet-RNN Tracker, to learn distance estimation and bounding box prediction.

3.2.1 Features selection

As it is known from projective transformation, the object's size in the camera image is inversely proportional to the object's distance from the camera. The *distance estimator* network, trained to learn the relationship between the changes in sizes of the objects' bounding boxes in image with respect to change in the distances between the objects and the camera over time. The dataset for network *Distance Estimator*, organized in a form of a two-dimensional feature matrix X_1 for distance estimation is given in (6), where each matrix row represents the six features of the feature vector $\boldsymbol{v}$ of an object at time-steps *t*, *t*-1 and *t*-2 respectively. Vector $\boldsymbol{v}$ was calculated from manually annotated objects' bounding boxes for a class label as:

$$\boldsymbol{v} = [1/B_h \; 1/B_w \; 1/B_d \; C_h \; C_w \; C_b] \qquad (3)$$

- B_h: height of the object bounding box in pixels/image height in pixels
- B_w: Width of the object bounding box in pixels/image width in pixels
- B_d: diagonal of the object bounding box in pixels/image diagonal in pixels
- C_h, C_w, and C_b: average height, width and length of each object class in meters

C_h, C_w, and C_b are defined as uniqueness features which actually represent different classes. These features generalize the network to learn distance vs bounding box relation for multiple object classes. For example, the predefined features for object class person are the average height, width and breadth of the humans and similarly these features were predefined for other classes. These features do not have any meaning and contribution into distance learning but help to differentiate different object types.

Similarly feature matrix X_2 was calculated for network *Bounding Box Predictor*. Each row in X_2 matrix represents coordinates of the top left corner *A* and the bottom right corner *B* of an object bounding box at time-steps *t*, *t*-1 and *t*-2 respectively, where (*u*,*v*) are image point coordinates.

$$X_1 = \begin{bmatrix} \boldsymbol{v}_t \\ \boldsymbol{v}_{t-1} \\ \boldsymbol{v}_{t-2} \end{bmatrix} = \begin{bmatrix} {}^1/_{B_{h_t}} & {}^1/_{B_{w_t}} & {}^1/_{B_{d_t}} & C_h & C_w & C_b \\ {}^1/_{B_{h_{t-1}}} & {}^1/_{B_{w_{t-1}}} & {}^1/_{B_{d_{t-1}}} & C_h & C_w & C_b \\ {}^1/_{B_{h_{t-2}}} & {}^1/_{B_{w_{t-2}}} & {}^1/_{B_{d_{t-2}}} & C_h & C_w & C_b \end{bmatrix} \qquad (6)$$

$$X_2 = \begin{bmatrix} \boldsymbol{b}_t \\ \boldsymbol{b}_{t-1} \\ \boldsymbol{b}_{t-2} \end{bmatrix} = \begin{bmatrix} A_{u_t} & A_{v_t} & B_{u_t} & B_{v_t} \\ A_{u_{t-1}} & A_{v_{t-1}} & B_{u_{t-1}} & B_{v_{t-1}} \\ A_{u_{t-2}} & A_{v_{t-2}} & B_{u_{t-2}} & B_{v_{t-2}} \end{bmatrix} \qquad (7)$$

In order to make DisNet-RNN Tracker more robust as well as to make it work also in the situations where an object is detected in one or two subsequent frames and not only in three subsequent frames, the dataset shall be augmented and extended with modified feature matrices (7). Namely, as mentioned earlier, each row of matrix X_1relates to time-steps *t*, *t*-1 and *t-2*. By zero-padding of the first and the second row, the modified feature matrices are:

$$X_1 = \begin{bmatrix} & & 0 & & & \\ & & 0 & & & \\ {}^1/_{B_{h_t}} & {}^1/_{B_{w_t}} & {}^1/_{B_{d_t}} & C_h & C_w & C_b \end{bmatrix} \qquad X_1 = \begin{bmatrix} & & 0 & & & \\ {}^1/_{B_{h_t}} & {}^1/_{B_{w_t}} & {}^1/_{B_{d_t}} & C_h & C_w & C_b \\ {}^1/_{B_{h_{t-1}}} & {}^1/_{B_{w_{t-1}}} & {}^1/_{B_{d_{t-1}}} & C_h & C_w & C_b \end{bmatrix} \qquad (7)$$

In the same way, the extended dataset is generated for the feature matrix X_2. Using the extended datasets means that the network does not need to wait for an object to be detected in three continuous frames in time to predict the distance.

3.3 Training and testing phase

The dataset generated contain 8000 samples which were randomly split into training data 80%, validation data 10% and test data 10%. DisNet-RNN Tracker sub-networks were trained with Adam optimizer and with a learning rate of 1e-4. During the training, a mini-batch gradient descent algorithm with minibatch size of 100 and Early Stop technique with 20 tolerant epochs has been used. Finally, after 246 training epochs, the mini loss on test dataset according to equation 1 was 1.28.

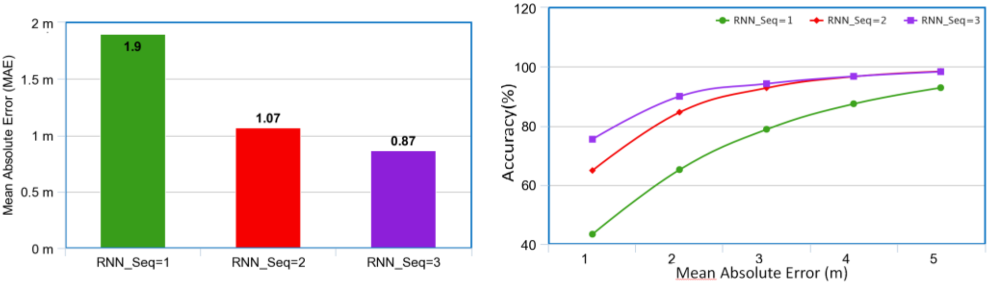

Figure 4: Left: Mean Absolute Error (loss) in distance estimation of DisNet-RNN Tracker for detection in 1, 2 and 3 subsequent frames. Right: Distance Accuracy vs Different mean absolute error

Figure 4 illustrates that with the use of more previous detection results, the mean square error reduces. This assures that the sequential input data really help in the estimation of distance and also shows the performance of trained model when the object is detected in three subsequent frames, two subsequent frames and only in one frame refer as RNN_Seq = 3, 2 and 1 respectively.

Besides mean absolute error (loss), another measurement parameter, distance accuracy Acc, is defined in (9). In the figure 4, the accuracy vs mean absolute error plots helps to understand the advantage of using sequential data. An estimation is considered accurate in $Acc_j$when estimation distance is in the range [y_{true-i} - j, y_{true-i}+ j]. Where j is Mean Absolute Error in the range of 1 to 5.

$$Acc_j = \frac{1}{n}\sum_{i=1}^{n}(|y_{true-i} - y_{estimated-i}| \leq j) \qquad (9)$$

	Acc1 (%)	Acc2 (%)	Acc3 (%)	Acc4 (%)	Acc5 (%)	MAE (m)
RNN_Seq =1	43.31	65.16	78.82	87.45	92.89	1.90
RNN_Seq =2	64.87	84.59	92.82	96.67	98.43	1.07
RNN_Seq =3	75.45	89.98	94.24	96.74	98.32	0.87

Table 1: Comparison result of DisNet-RNN Tracker Accuracy at different MAE

From the above two comparison results, it is clear that the performance of DisNet-RNN Tracker with more sequential data (previous detection results) is more accurate. The similar results for bounding box prediction network were obtained hence not included in this section.

3.4 Multiple Object Mapping (MOM)

Another problem needs to be solved before DisNet-RNN Tracker can be used in object tracking. The multiple detected objects in the current frame need to be associated with the multiple detected objects in previous frames to form sequential data (6)-(7), which is input to DisNet-RNN Tracker. Therefore, a Multiple Object Mapping (MOM) module is introduced to perform object association and generate sequential feature matrices for DisNet-RNN Tracker. The MOM module calculates the Intersection Over Union (IOU) of current detected and previously detected objects and based on high correlation associate the objects. The entire working of the MOM module is shown in Figure 5. If the objects do not correlate then MOM initialize the new tracker for newly detected objects and assign a unique ID.

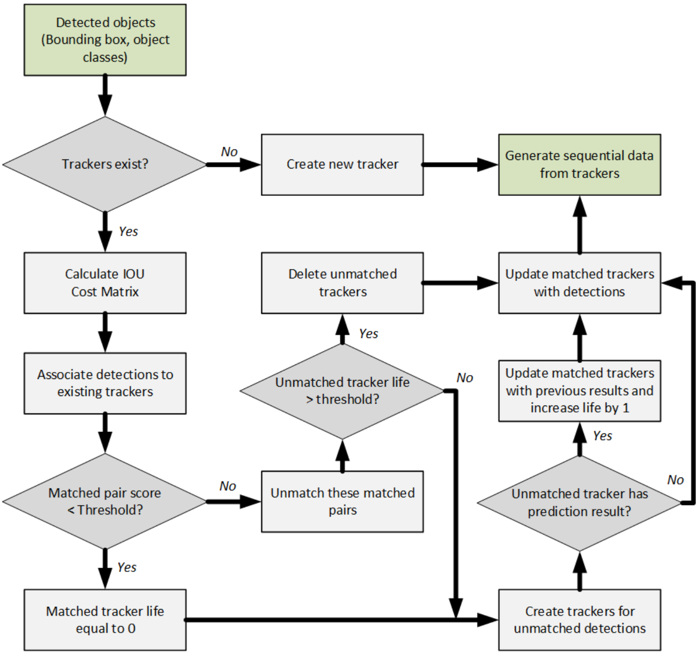
Detected objects (Bounding box, object classes)
Trackers exist?
No
Create new tracker
Generate sequential data from trackers
Yes
Calculate IOU Cost Matrix
Delete unmatched trackers
Update matched trackers with detections
Yes
Associate detections to existing trackers
Unmatched tracker life > threshold?
No
Update matched trackers with previous results and increase life by 1
Yes
Matched pair score < Threshold?
No
Unmatch these matched pairs
No
Unmatched tracker has prediction result?
Yes
Matched tracker life equal to 0
Create trackers for unmatched detections

Figure 5: Multiple Object Mapping based object association of current detected objects and previously detected objects

4 Evaluation

The DisNet-RNN Tracker based system for distance estimation and object tracking was evaluated on images recorded by a RGB monocular camera mounted on the frontal profile of the moving freight train during the field tests within H2020 Shift2Rail project SMART [15]. The RGB camera from The Imaging Source [16] provides images with the maximum resolution of 2592x1944 at 2 FPS.

Figure 6 shows the performance of DisNet-RNN Tracker on six subsequent frames from a video of a real-world dynamic railway scene with a truck parked in the rail track area (vehicle 3), which is potential intrusion, and a van (vehicle 4) crossing the unsecured crossing while the train is approaching the crossing. The estimated distance to the vehicle objects from the camera (i.e. from the locomotive), object class, and tracking ID are shown in the left upper corner of the each frame. The detection/prediction of the objects is marked by the object bounding boxes. The blue and brown coloured bounding boxes represent the prediction results, achieved by DisNet-RNN Tracker, whereas the red bounding boxes represent the object detection results of the YOLO object detection. As it is evident, the DisNet-RNN Tracker is able to track the object (vehicle 3), to predict its bounding box and to estimate the distance even in the case object is occluded by the other vehicle (frames 2 and 3) and YOLO object detector failed. Also, YOLO object detector failed even in the case of a fully visible object (vehicle 3) in frame 6, whereas the proposed DisNet-RNN Tracker achieved the object bounding box prediction and distance estimation.

During the train run, the ground truth was also measured using relative GPS positions of the objects (from the google maps) and the train (GPS on the train). According to the ground truth distance the van (vehicle 3) and truck (vehicle 4) were approximately 45 meters apart. The ground truth distances to the van were measured as 135m, 128 m, 123m, 120m, 117m and 108m respectively for six subsequent frames shown in Figure 4. Hence, the estimated distance from DisNet-RNN Tracker in comparison with the ground truth demonstrates also reliability of proposed method.

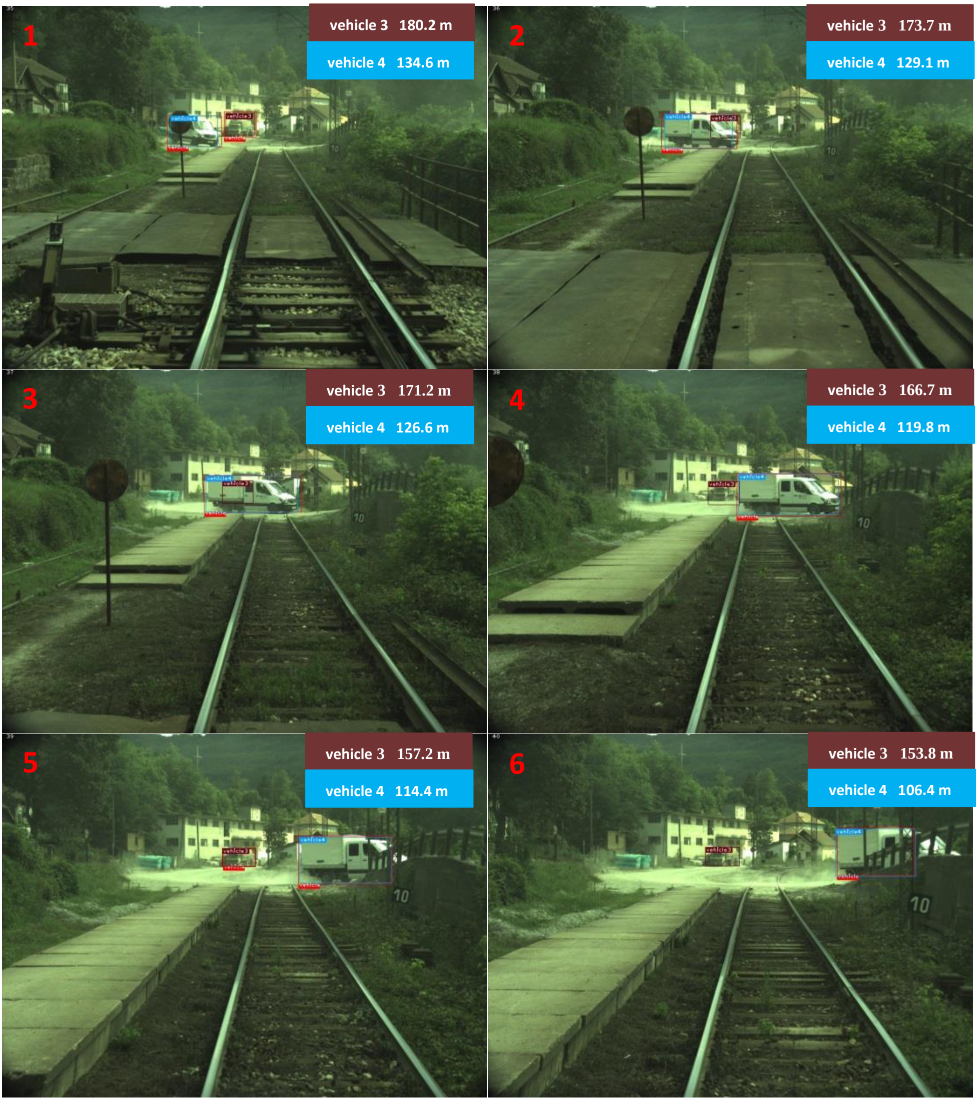

Figure 6: A real-world scene where truck is parked near rail track and the van crossing the rail track while train approaching.

5 Conclusion

Object tracking and distance estimation are crucial for safety critical applications such as obstacle and track intrusion detection in railways. The use of information from previous object detections improves the distance estimation and also enables the tracking of objects by prediction of the object's position in frames in which the object detector fails.

In this paper, DisNet-RNN Tracker consists of two sub independent network named as *distance estimator* and *bounding box predictor* is presented. Gated Recurrent Neural Network architecture is chosen for object distance estimation and object tracking. Using a monocular camera, the presented method can precisely track and estimate the distance to the target objects as shown by the evaluation results from a real-world railway scenario. However, the proposed object distance estimation system still has several limitations such as object occlusion for a longer period introduces an error which increases with time. So dependency on prediction results for a longer period is not recommended, as the error increases with time until the estimation is corrected by the detection.

Some efforts could be made in the following aspects in order to improve the performance and conquer the drawbacks of presented distance estimation and object tracking system. Moreover, for the improvement of distance estimation and object tracking, the dataset needs to be extended, for long.-range detection and tracking of multiple objects.

Acknowledgements

This research has received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 730836.

Special thanks to Serbian Railways Infrastructure, and Serbia Cargo for support for support in realization of the SMART Field tests.

References

[1] Shift2Rail Joint Undertaking, Multi-Annual Action Plan, Brussels, November 2015.

[2] Haseeb, M. A., Guan, J., Ristić-Durrant, D., Gräser, A. DisNet: A novel method for distance estimation from monocular camera, 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS, Spain, 2018.

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

[4] Wang Y, Zhu L, Yu Z, Guo B, An Adaptive Track Segmentation Algorithm for a Railway Intrusion Detection System. Sensors, Basel, 2019, DOI: 10.3390/s19112594

[5] Y. Zhang, Y. Huang and L. Wang, "Multi-task Deep Learning for Fast Online Multiple Object Tracking," 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), Nanjing, 2017, pp. 138-143.doi: 10.1109/ACPR.2017.58

[6] A. Kuramoto, M. A. Aldibaja, R. Yanase, J. Kameyama, K. Yoneda and N. Suganuma, "Mono-Camera based 3D Object Tracking Strategy for Autonomous Vehicles," 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, 2018, pp. 459-464. DOI: 10.1109/IVS.2018.8500482

[7] Yu Huang, T. S. Huang and H. Niemann, "A region-based method for model-free object tracking," Object recognition supported by user interaction for service robots, Quebec City, Quebec, Canada, 2002, pp. 592-595 vol.1. DOI: 10.1109/ICPR.2002.1044810

[8] Hyeong-Jun Cho, Sang-Hyeop Song, Jong-Hak Kim, Solima and Jun Dong Cho, "Simple object coordination tracking based on background modeling," 2015 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), Poznan, 2015, pp. 110-113. DOI: 10.1109/SPA.2015.7365143

[9] M. Yokoyama and T. Poggio, "A Contour-Based Moving Object Detection and Tracking," 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, 2005, pp. 271-276. DOI: 10.1109/VSPETS.2005.1570925

[10] L. Fan, "A Feature-Based Object Tracking Method Using Online Template Switching and Feature Adaptation," 2011 Sixth International Conference on Image and Graphics, Hefei, Anhui, 2011, pp. 707-713. DOI: 10.1109/ICIG.2011.102

[11] H. Hsu and J. Ding, "FasterMDNet: Learning model adaptation by RNN in tracking-by-detection based visual tracking," 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, 2017, pp. 657-660. DOI: 10.1109/APSIPA.2017.8282115

[12] Redmon, Joseph and Farhadi, Ali, YOLOv3: An Incremental Improvement, arXiv, 2018.

[13] COCO dataset, Available online: https://arxiv.org/pdf/1405.0312.pdf (accessed on 15.02.2019).

[14] Dutta, A. and Gupta, A. and Zissermann, A. “VGG Image Annotator (VIA)”, Available online: http://www.robots.ox.ac.uk/~vgg/software/via (accessed on 15.02.2019)

[15] Project SMART, http://smartrail-automation-project.net/

[16] The Imaging Source, GigE colour zoom camera. Available online: https://www.theimagingsource.com/ (accessed on 15.02.2019).