Traffic Sign Recognition on Video Sequence Using Deep Neural Networks and Matching Algorithm

Ilya Belkin
Laboratory of Cognitive Dynamic Systems
Moscow Institute of Physics and Technology (National Research University)
Dolgoprudny, Russia
belkin.iv@phystech.edu

Sergey Tkachenko
Laboratory of Intelligent Transport
Moscow Institute of Physics and Technology (National Research University)
Dolgoprudny, Russia
tkachenko.sm@phystech.edu

Dmitry Yudin
Laboratory of Intelligent Transport
Moscow Institute of Physics and Technology (National Research University)
Dolgoprudny, Russia
yudin.da@mipt.ru

Abstract*—The paper analyzes data sets containing images with labeled traffic signs, as well as modern approaches for their detection and classification on images of urban scenes. Particular attention is paid to the recognition of Russian types of traffic signs. Various modern architectures of deep neural networks for the simultaneous object detection and classification were studied, including Faster R-CNN, Mask R-CNN, Cascade R-CNN, RetinaNet. To increase the efficiency of neural network recognition of objects in a video sequence, the Seq-BBox Matching algorithm is used. Training and testing of the proposed approach was carried out on Russian Traffic Sign Dataset and IceVision Dataset containing over 150 types of road signs and more than 65,000 marked images. For all the approaches considered, quality metrics are defined: mean average precision mAP, mean average recall mAR and processing time of one frame. The highest quality performance was demonstrated by the architecture of Faster R-CNN with Seq-BBox Matching, while the highest performance is provided by the architecture of RetinaNet. Implementation was carried out using the Python 3.7 programming language and PyTorch deep learning library using NVidia CUDA technology. Performance indicators were obtained on the workstation with the NVidia Tesla V-100 32GB video card. The obtained results demonstrate the possibility of applying the proposed approach both for the resource-intensive procedure for automated labeling of road scene images for new data sets preparation, and for traffic sign recognition in on-board computer vision systems of unmanned vehicles.

***Keywords*—image recognition, detection, traffic sign, deep learning, neural network, matching algorithm, software**

I. Introduction

Deep neural networks (DNN) have proven their worth in various fields of knowledge. This is especially true for computer vision and pattern recognition problems [1]. In recent years impressive results have been achieved in this area. That makes the proposed approaches [2, 3, 4] attractive for deploing in real applications. One such application is autonomous transport and machine vision systems which able to process and recognize road infrastructure. The main computer vision problems in this area are semantic segmentation [5] (e.g. roadway segmentation), object detection (traffic signs, traffic lights), image classification (traffic light recognition), instance segmentation (finding the outlines of pedestrians, cars etc).

Most of deep learning approaches are widely studied for object detection and instance segmentation on single image (still image detection). That is a sequence of frames is considered as a set of independent images and information between adjacent frames is not taken into account.

To leverage sequentual structure of the data deep neural networks of a special architecture were developed, which takes as an input several frames [6] or process them in recurrent manner [7]. Such approach not only increased detection quality on video but significantly improved processing speed.

The main drawback of such approaches is that they require annotated sequences of frames to train on. The standard approach to video annotation involves markup frames with some step, for instance one frame per second. At 30 frames per second data preparation for these models requires 30 times more work.

Another works are aimed at postprocessing of detection results on individual frames with taking into account the sequentual nature of the data. One such approache is Seq-NMS [8] which is adoptation of non-maximum suppression of detection results for whole sequences of detections. Another one is Seq-Bbox Matching [9], which will be discussed later, make it possible to merge separate detection of a single object in tubelet using not only spatial relationship between bounding boxes, but their semantic similarity too.

In our work we investigate modern architectures for object detection for the problem of traffic sign detection and recognition. We report quantitative results of five popular detection frameworks. All models were trained and validated on dataset of russian traffic signs RTSD [10, 11], which is the largest database of traffic signs publicly available. For testing we use new IceVision dataset [12], collected in Skolkovo using external cameras of autonomous car. We report such metrics as mAP [13], mAP as well as AP and AR for each class. We also implement and apply Seq-Bbox Mathing when testing on sparse annotated sequences and make our implementation publicly available.

II. Problem Definition

In this section we consider the definition of the problem of traffic signs detection and recognition on a video. We also list metrics and discuss approach used for algorithm validation is case when only some available images are annotated.

A. Problem Description

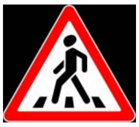
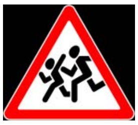
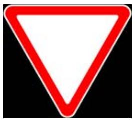
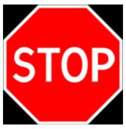
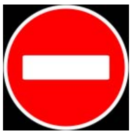
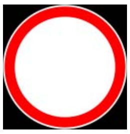
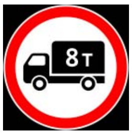
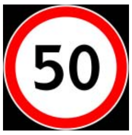
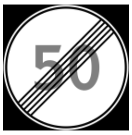
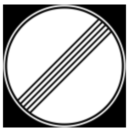
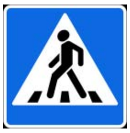
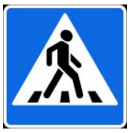
Formally, we have an ordered sequence of frames $F = (f_i)_1^N$, where $N$is a sequence length and $f_i$is an RGB image of

978-1-7281-4326-2/19/$31.00 ©2019 IEEE
DOI 10.1109/IC-AIAI48757.2019.00013

spatial resulution (w, h) obtained from RGB camera on the car. For each frame we need to obtain a set $b_i = \{o_i^j\}$, where each o_i^jhave structure of and specify a bounding box of width w, height $h$and with top-left corner at (x_{tl}, y_{tl})on the image. Each bounding box is a hypothese of that corresponded area on the image contains an object such as traffic sign in our case. Aslo it is required to determine traffic sign type. Among with $b_i$for each frame we find $d_i = (p_i^j)$, where each $p_i^j$is a vector of size $N + 1$ which contains probabilities that the corresponding object inside a bounding box belongs to a certain class, which is traffic sign type in out case.

There are about 200 traffic sign types. We choose only 12 most popular types and perform classification for them. Other types we refer as ‘other’. Traffic sign templates of chosen types are shown in the Table I. Note that we do not distinguish sign types 3.25 and 3.31 as well as 5.19.1 and 5.19.2 and in next sections refer them as 3.25+3.31 and 5.19 types respectively. Actually our model use next 11 classes: ‘1.22’, ‘1.23’, ‘2.4’, ‘2.5’, ‘3.1’, ‘3.2’, ‘3.4’, ‘3.24’, ‘3.25+3.31’, ‘5.19’, ‘other’.

TABLE I. TRAFFIC SIGN TYPES

 1.22	 1.23	 2.4	 2.5
 3.1	 3.2	 3.4	 3.24
 3.25	 3.31	 5.19.1	 5.19.2

B. *Metrics*

Most common metric for object detection is Mean Average Precision (mAP) [13]. There are different open-source implementations of this metric [14, 15]. It is used for quality estimation of multi-classe detectors.

It has different parameters, such as minimal object area, intersection over union threshold, maximum number of detections. Parameters we use for evaluation are different from ones in COCO challenge [16]. We do not consider objects smaller than 20x20 pixels for evaluation, since reliable annotation as well as detection of such objects is difficult and not essential in case of road infrastructure recognition. We use 50% IoU threshold because accurate localization of traffic signs is not needed. We also do not set up maximum number of detections because filtering of detections is a part of our pipeline.

Evaluation was performed on approximately every 30d frame, because only these images were annotated. However, we run detection on every frame to leverage Seq-Bbox Matching method.

Another metric we report is the speed of detector. We measure it in milliseconds. This number includes image preprocessing, which is simple resize and normalization, forward pass through network and postprocessing of detection results. All measurements were carried out on single Tesla V100 32GB and Intel Xeon Gold 6154.

III. DATASETS

In this section we give an overview of datasets with russian traffic signs which we used in our research: Russian Traffic Sign Dataset [10] and IceVision [12].

A. *RTSD*

RTSD is the largest dataset of traffic signs. It contains annotated images collected at different weather conditions, seasons, time of day. Particular characteristics of the dataset are shown in the Table II. Only about one third of all images actually contains traffic signs. Dataset contain images of two spatial resolutions. All images were captured on dashboard camera inside a car. Many images are blurred.

TABLE II. RUSSIN TRAFFIC SIGN DATASET (RTSD)

Characteristic	Value
Resolution	1280x720 + 1920x1080
Total number of frames	179138
Number of annotated frames	179138
Number of classes	156
Number of boxes	104358
Images with signs	59188

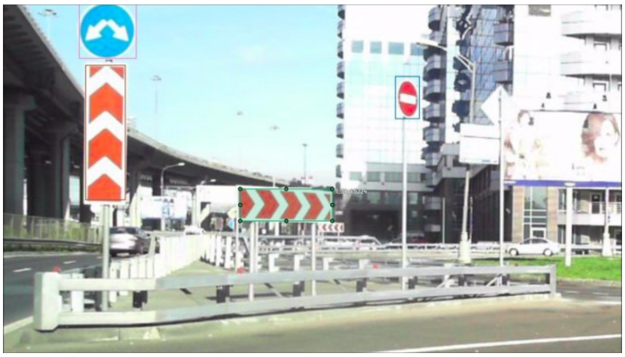


Fig. 1: RTSD image example. White arrows on the red are not presented in the original dataset, but were annotated by out team.

Traffic signs of types 3.11, 3.12, 3.13, 3.14, 3.16, 3.24, 3.25, 3.4, 6.2 have additional data: numeric value on the sign.

Original database does not contain 8.22 signs. We add this additional markup via opencv/cvat [17]. We make this additional markup also publicly available in MS COCO JSON format [18].

B. *IceVision*

IceVision dataset was presented during IceVision competition. It contains videos with annotated traffic signs captured in conditions of Russian winter and poor visibility. Some details are presented in Table III.

TABLE III. ICEVISION DATASET

Characteristic	Value
Resolution	2448x2048
Total number of frames	212965
Number of annotated frames	8563
Number of classes	178
Number of boxes	42758
Images with signs	6956

Some annotated traffic signs have associated additional information, such as text, presented on the sign. It relates to speed limit signs, information signs, etc. Aslo each sign have boolean lable ‘temporary’ and ‘occluded’. The first one show whether the sign is temporary, i.e. have yellow background, the second – whether the sign is partially hidden.

The data was collected in winter in Moscow region. There are day and night images captured at 30 frames per second on external camera on the car. Only about every 30d frame was annotated.

Markup contains almost all traffic signs presented on the image starting from 10x10 pixels. Such small objects are not recognizable by human, and even presence of such signs is not obvious.

IV. DEEP LEARNING APPROACH TO TRAFFIC SIGN RECOGNITION

In our work we tested five popular approaches to object detection: Mask R-CNN[3], Faster R-CNN[19], Cascade R-CNN[20], Cascade Mask R-CNN и Retina Net[21]. In this section we give a short overwiev of the key differences between these models (Fig. 2). All these models rely on backbone – convolutional neural network used for feature extraction from input image. We use ResNet50 as backbone in our experiments because it is lightweight and fast network which still able to achieve high results.

A. Faster R-CNN

Faster R-CNN is a two stage detector which consists of two neural networks. The first one is Region Proposal Network (RPN), which looks for proposals, which can contain objects. The second one is classifier of proposals. This detector is quite fast and have good recognition quality.

B. Mask R-CNN

Mask R-CNN is an extention of Faster R-CNN. Along with classifier it contains segmentation head. It is a bit slower, but provide us not only bounding boxes but object contours [22].

C. Cascade R-CNN

Cascade R-CNN is a multistep detector. The main difference from Faster R-CNN is that it applies consequently few detectors to increase final quality.

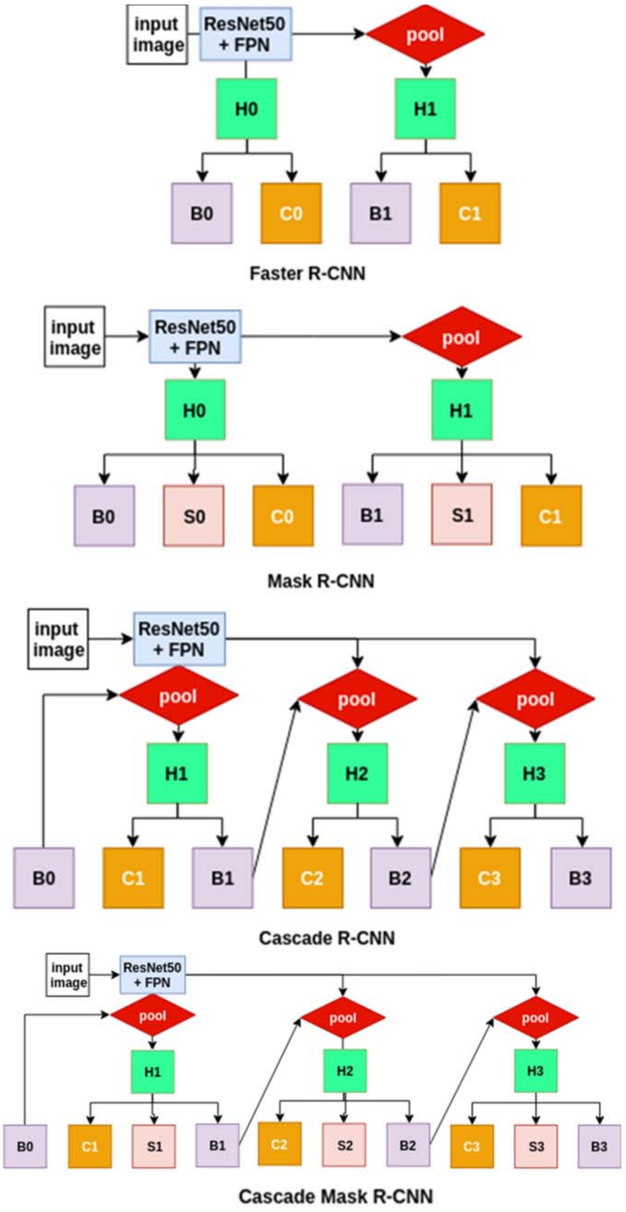


Fig. 2: Two stage detectors considered in this paper. Mask models have additional head which make pixel level classification. Cascade models perform refining of bounding boxes by feeding proposals from previous stage back to pooling layer.

D. Cascade Mask R-CNN

Cascade Mask R-CNN is an extention of Cascade R-CNN. Modification is similar to ordinary Mask R-CNN. This architecture typically provides a more accurate match of bounding boxes of objects [23].

E. RetinaNet

RetinaNet is a unified network with backbone and two subnetworks. Backbone is responsible for feature extraction from input image. The first subnetwork works over feature map and used for classification. The second one is used for bounding box regression.

V. Matching Algorithm

A typical problem when still image detection approach is applied to video is the lose of object for a few frames. Conversely it is possible when object not presented on previous and subsequent frames will be detected with high confidence. Consequent images have a lot in common for human, but neural networks may suffer from this difference. This problem lead to idea to reuse information from the past to calibrate our current predictions.

One possible approach is to use tracker and propogate bounding boxes on subsequent frames. Another approache is to smooth preditions on adjacent frames using geometric information as well as semantic features. Seq-Bbox Matching is able to increase detection quality with minimal computation overhead and with minimal modification of pipeline.

Seq-Bbox Matching allow us to merge predictions from the past with out current results. This method introduce a measure of similarity between two bounding boxes which is based on IoU, to leverage spatial relationship, and on semantic similarity, which is dot product between classification scores.

VI. Experimantal Results

In this section we describe training precedure and present validation curves for five architectures and test metrics on two sequences from IceVision, one is daily, another is nightly. All images were resized to fit into rectangle of size 1333x800.

A. *Training*

We use the same pipeline [24] for training all models. In detail we train them for 30 epochs with initial learning rate equals to 0.02 and reducing it by factor of 5 after 20 and 25 epochs. We use SGD with momentum equals to 0.9 and weight decay 0.0001. Also, to accumulate gradients early in training we use linear warmup for 500 iterations.

B. *Validation curves*

We use 20% of RTSD images for validation, the rest for training. We perform validataion every epoch and calculate mAP with IoU threshold 0.5. Validation curves are shown on the Fig.3.

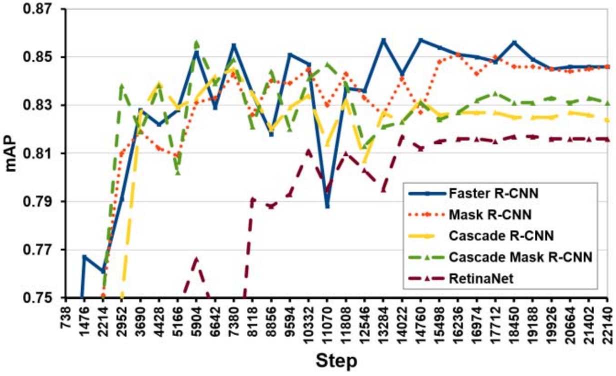


Fig. 3: Validataion curves. mAP correspond to IoU threshold 0.5. Some models show best results early in training, but worse after convergence.

Cascade models are prone to overfitting, since validation curve go down after 10 epochs. Faster R-CNN and Mask R-CNN show the best results on validation. RetinaNet is much worse than other architectures.

C. *IceVision results*

Testing results are shown in Table IV. RetinaNet is the fastest model and show best results on night images. Faster R-CNN is a bit slower but works well on daily images. Seq-Bbox Matching (SBM) lead to significant gain in mAP on night images for all models except RetinaNet, where it reduces quality dramatically. For daily images SBM makes results worse.

Tables V and VI containing more detailed results with per class Average Precision and Recall. Although detection quality may decrease in terms of mAP and mAR after applying SBM, it is possible to get positive gain for particular classes.

TABLE IV. Test results on IceVision dataset

Approach	Night		Day		Speed (ms)
	mAP	***mAR***	***mAP***	***mAR***	
Faster RCNN	43.7	51.1	**75.0**	80.1	42.8
Mask RCNN	36.3	46.5	73.7	78.1	46.4
Cascade RCNN	41.7	51.8	72.0	77.0	54.0
Cascade Mask RCNN	43.1	51.2	72.0	75.1	59.7
RetinaNet	**47.1**	**61.0**	65.6	74.8	**40.8**
Faster RCNN + SBM	45.6	52.1	69.6	**80.4**	43.0
Mask RCNN + SBM	42.0	48.9	71.4	78.2	46.6
Cascade RCNN + SBM	42.3	52.3	62.4	73.3	54.2
Cascade Mask RCNN + SBM	44.9	55.8	70.4	77.7	59.9
RetinaNet + SBM	37.4	56.8	53.1	75.1	**41.0**

TABLE V. Traffic sign detection results on day testing sample

Metrics	**Faster RCNN**	**Mask RCNN**	**Cascade RCNN**	**Cascade Mask RCNN**	**RetinaNet**	**Faster RCNN + SBM**	**Mask RCNN + SBM**	**Cascade RCNN + SBM**	**Cascade Mask RCNN + SBM**	**RetinaNet + SBM**
AP 1.22	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.83	**1.00**	0.83	0.83	0.83
AP 1.23	**0.97**	0.93	0.96	0.81	0.78	**0.97**	0.91	0.91	0.93	0.59
AP 2.40	**0.88**	**0.88**	**0.88**	**0.88**	0.72	**0.88**	**0.88**	**0.88**	**0.88**	0.63
AP 2.50	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.92	**1.00**	0.66	**1.00**	0.81
AP 3.10	0.03	0.06	0.06	0.00	0.00	**0.17**	0.06	0.02	0.00	0.00
AP 3.20	–	–	–	–	–	–	–	–	–	–
AP 3.24	0.91	0.94	0.90	0.93	0.82	0.94	**0.95**	0.91	**0.95**	0.54
AP 3.40	**0.95**	0.75	0.69	0.65	0.53	0.73	0.69	0.61	0.69	0.50
AP 5.19	0.26	0.34	0.27	**0.49**	0.40	0.14	0.23	0.16	0.35	0.34
AP 3.25+ 3.31	–	–	–	–	–	–	–	–	–	–
AP other	0.39	0.39	0.37	0.38	0.40	0.40	**0.41**	0.39	0.40	0.29
mAP	**0.75**	0.74	0.72	0.72	0.66	0.70	0.71	0.62	0.70	0.53

TABLE VI. TRAFFIC SIGN DETECTION RESULTS ON NIGHT TESTING SAMPLE

Metrics	**Faster RCNN**	**Mask RCNN**	**Cascade RCNN**	**Cascade Mask RCNN**	**RetinaNet**	**Faster RCNN + SBM**	**Mask RCNN + SBM**	**Cascade RCNN + SBM**	**Cascade Mask RCNN + SBM**	**RetinaNet + SBM**
AP 1.22	–	–	–	–	–	–	–	–	–	–
AP 1.23	0.57	0.38	0.39	0.35	0.72	**0.83**	**0.83**	0.53	0.51	0.74
AP 2.40	0.53	0.51	0.53	0.54	0.58	0.56	0.53	0.57	**0.61**	0.42
AP 2.50	0.39	0.30	0.15	0.23	0.32	**0.42**	0.41	0.20	0.38	0.33
AP 3.10	**0.55**	0.54	**0.55**	0.54	0.53	0.52	0.44	0.47	0.41	0.26
AP 3.20	–	–	–	–	–	–	–	–	–	–
AP 3.24	0.71	0.78	0.78	0.82	0.70	0.78	0.84	0.84	**0.86**	0.40
AP 3.40	0.42	0.06	0.50	**0.75**	0.52	0.42	0.00	0.50	0.65	0.50
AP 5.19	0.47	0.40	0.45	0.36	**0.47**	0.42	0.36	0.40	0.34	0.27
AP 3.25+ 3.31	0.29	0.29	0.41	0.29	0.41	0.15	0.37	0.29	0.28	**0.43**
AP other	0.52	0.53	0.56	0.54	0.52	0.56	0.55	**0.59**	0.56	0.35
mAP	0.44	0.36	0.42	0.43	**0.47**	0.46	0.42	0.42	0.45	0.37

VII. CONCLUSION

In this work we study different methods of object detection for the problem of traffic sign recognition on video sequence. We study still image detection approach and found that it may be improved with Seq-Bbox Matching. However, if the quality of detection is high, applying SBM may give worse results. We tested Faster-RCNN, Mask-RCNN, Cascade R-CNN, Cascade Mask R-CNN and RetinaNet on IceVision and RTSD datasets and found that simple RetinaNet gave the best results. We think it is due to overfitting of larger models. For further improvement data augmentation and other regularization techniques as well as hyperparameter optimization should be applied.

ACKNOWLEDGMENTS

This study was carried out under the contract with the Scientific-Design Bureau of Computing Systems (SDB CS) and supported by the Government of the Russian Federation (Agreement No 075-02-2019-967).

REFERENCES

[1] C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” In NIPS, 2013.

[2] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, “Hybrid task cascade for instance segmentation,” Technical report, 2018.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” arXiv:1703.06870, 2017.

[4] Z. Cai, and N.Vasconcelos, “Cascade r-cnn: Delving into high quality object detection,” arXiv:1712.00726, 2017.

[5] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” CVPR, 2015.

[6] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Detect to track and track to detect,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3038–3046, 2017.

[7] X. Zhu, J. Dai, L. Yuan, and Y. Wei, “Towards high performance video object detection,” CVPR, 2018.

[8] W. Han et al., “Seq-NMS for video object detection,” arXiv: 1602.08465, 2016.

[9] H. Belhassen, H. Zhang, V. Fresse, and E.-B. Bourennane, “Improving Video Object Detection by Seq-Bbox Matching,” 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2019.

[10] Laboratory of computer graphics and multimedia, http://graphics.cs.msu.ru/en/node/1266.

[11] V. Shakhuro, and A. Konushin, “Russian traffic sign images dataset,” Computer Optics, vol. 40, no. 2, 2016, pp. 294–300.

[12] A. L. Pavlov, P. A. Karpyshev, G. V. Ovchinnikov, I. V. Oseledets, and D. Tsetserukou, “Icevisionset: lossless video dataset collected on russian winter roads with traffic sign annotations,” in 2019 International Conference on Robotics and Automation (ICRA), IEEE, pp. 9597–9602, 2019.

[13] M. Everingham, L. Van Gool, C.K.I. Williams, et al. Int J Comput Vis, 88: 303, 2010, https://doi.org/10.1007/s11263-009-0275-4.

[14] Metrics for object detection, https://github.com/rafaelpadilla/Object-Detection-Metrics.

[15] COCO Dataset API, https://github.com/cocodataset/cocoapi.

[16] COCO – Common Objects in Context, http://cocodataset.org.

[17] Computer Vision Annotation Tool, https://github.com/opencv/cvat.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft COCO: Common objects in ´ context,” In ECCV, 2014.

[19] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” In NIPS, 2015.

[20] Z. Cai, and N.Vasconcelos, “Cascade r-cnn: Delving into high quality object detection,” arXiv:1712.00726, 2017.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” arXiv:1708.02002, 2017.

[22] D. Yudin, A. Ivanov, and M. Shchendrygin, “Detection of a Human Head on a Low-Quality Image and its Software Implementation,” International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 42, 2/W12, 2019.

[23] D. A. Yudin, A. Skrynnik, A. Krishtopik, I. Belkin, and A. I. Panov, “Object Detection with Deep Neural Networks for Reinforcement Learning in the Task of Autonomous Vehicles Path Planning at the Intersection,” Optical Memory & Neural Networks (Information Optics), Vol. 28 № 4, 2019.

[24] Open MMlab Detection Toolbox and Benchmark, https://github.com/open-mmlab/mmdetection