УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
май–июнь 2020 Том 20 № 3 ISSN 2226-1494 http://ntv.itmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
May–June 2020 Vol. 20 No 3 ISSN 2226-1494 http://ntv.itmo.ru/en/
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК УНИВЕРСИТЕТ ИТМО ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ май–июнь 2020 Том 20 Номер 3

doi: 10.17586/2226-1494-2020-20-3-418-424

REAL TIME DETECTION AND CLASSIFICATION OF TRAFFIC SIGNS BASED ON YOLO VERSION 3 ALGORITHM

V.N. Sichkar, S.A. Kolyubin

ITMO University, Saint Petersburg, 197101, Russian Federation
Corresponding author: vsichkar@itmo.ru

Article info
Received 29.04.20, accepted 26.05.20
Article in English

For citation: Sichkar V.N., Kolyubin S.A. Real time detection and classification of traffic signs based on YOLO version 3 algorithm. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2020, vol. 20, no. 3, pp. 418–424 (in English). doi: 10.17586/2226-1494-2020-20-3-418-424

Abstract
The issue of effective detection and classification of various traffic signs is studied. The two-stage method is proposed for creation of holistic model with end-to-end solution. The first stage includes implementation of effective localization of traffic signs by YOLO version 3 algorithm (You Only Look Once). At the first stage the traffic signs are grouped into four categories according to their shapes. At the second stage, an accurate classification of the located traffic signs is performed into one of the forty-three predefined categories. The second stage is based on another model with one convolutional neural layer. The model for detection of traffic signs was trained on German Traffic Sign Detection Benchmark (GTSDB) with 630 and 111 RGB images for training and validation, respectively. Classification model was trained on German Traffic Sign Recognition Benchmark (GTSRB) with 66000 RGB images on pure "numpy" library with 19 × 19 dimension of convolutional layer filters and reached 0.868 accuracy on testing dataset. The experimental results illustrated that the training of the first model deep network with only four categories for location of traffic signs produced high mAP (mean Average Precision) accuracy reaching 97.22 %. Additional convolutional layer of the second model applied for final classification creates efficient entire system. Experiments on processing video files demonstrated frames per second (FMS) between thirty-six and sixty-one that makes the system feasible for real time applications. The frames per second depended on the number of traffic signs to be detected and classified in every single frame in the range from six to one.

Keywords
traffic signs detection, deep convolutional neural network, YOLO v3, traffic signs classification, detection accuracy

УДК 004.855.5 doi: 10.17586/2226-1494-2020-20-3-418-424

ДЕТЕКТИРОВАНИЕ И КЛАССИФИКАЦИЯ ДОРОЖНЫХ ЗНАКОВ В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ АЛГОРИТМА YOLO ВЕРСИИ 3

В.Н. Сичкар, С.А. Колюбин

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
Адрес для переписки: vsichkar@itmo.ru

Информация о статье
Поступила в редакцию 29.04.20, принята к печати 26.05.20
Язык статьи — английский

Ссылка для цитирования: Сичкар В.Н., Колюбин С.А. Детектирование и классификация дорожных знаков в реальном времени на основе алгоритма YOLO версии 3 // Научно-технический вестник информационных технологий, механики и оптики. 2020. Т. 20. № 3. С. 418–424. doi: 10.17586/2226-1494-2020-20-3-418-424

Аннотация
Исследован эффективный метод обнаружения и классификации различных категорий дорожных знаков. Для построения целостной модели с комплексным решением был предложен метод с двумя этапами. На первом этапе метод включает выполнение эффективной локализации дорожных знаков на основе алгоритма YOLO версии 3 (You Only Look Once). Для первого этапа дорожные знаки группируются в четыре категории в соответствии с их формой. На втором этапе выполняется точная классификация обнаруженных дорожных знаков в соответствие с одной из заранее определенных 43 категорий. Второй этап построен на модели с одним сверточным нейронным слоем. Модель обнаружения дорожных знаков обучается на GTSDB (German Traffic Sign Detection Benchmark) с 630 и 111 RGB-изображениями для обучения и валидации соответственно. Модель классификации обучается

на GTSRB (German Traffic Sign Recognition Benchmark) с 66000 RGB-изображениями, с помощью библиотеки «numpy», фильтрами сверточного слоя размерностью 19 × 19, и достигла точности 0,868 на наборе данных для тестирования. Результаты экспериментов показали, что обучение глубокой нейронной сети первой модели только с 4 категориями для определения координат дорожных знаков выдает высокую точность mAP (mean Average Precision), достигающую 97,22 %. Дополнительный сверточный слой второй модели, добавленный для окончательной классификации, создает эффективную целостную систему. Эксперименты по обработке видеофайлов показали FPS (frames per second) в диапазоне 36 и 61, что делает систему пригодной для использования в реальном времени. FPS зависел от количества дорожных знаков, которые должны быть локализованы и классифицированы в каждом отдельном кадре, и находились в диапазоне от 6 до 1.

Ключевые слова
детектирование дорожных знаков, глубокая сверточная нейронная сеть, YOLO v3, классификация дорожных знаков, точность детектирования

Introduction

One of the important feature of the modern cars and future fully autonomous vehicles is their vision that makes them capable to sense the environment, assist the driver and take control in dangerous situations. Efficient and accurate traffic signs detection is still a challenging issue due to a number of real life factors that influence image quality, including various natural backgrounds, lightning and blurriness.

Traffic signs recognition challenges were addressed by researchers implementing deep convolutional neural networks [1–4], region-based convolution neural networks [5, 6], histogram of oriented gradients feature with support vector machine [7–9]. However, most of the algorithms were developed for detection of a small number of categories. Certain algorithms are focused only on classification problem, leaving predicting issue of traffic signs location on the image without attention. Multiple categories of traffic signs to be detected and classified remains an open issue.

This research presents YOLO (You Only Look Once) algorithm of version 3 [10–12] aimed at architecture of deep CNN (Convolutional Neural Network) creation for traffic sign recognition. Recently, deep CNNs were not considered for real time applications due to highly complex mathematical computations. However, modern GPUs (graphics processing units) were especially developed to implement high performance.

The proposed method incorporates two stages for accurate localization of traffic signs on images and for further classification of the cut fragments. The objective is to develop a system that can effectively detect and classify traffic signs with performance that makes it able to be applied in real time applications.

To train developed architecture based on YOLO version 3 model, GTSDB (German Traffic Sign Detection Benchmark) [13] and GTSRB (German Traffic Sign Recognition Benchmark) [14] datasets were used. The first dataset was applied to train localization stage that predicts coordinates of traffic sings and returns them as output. Classification stage was trained on the second dataset and takes an output from the first stage as an input.

Architecture of proposed method

The proposed method consists of two stacked together models. The first model (model-1) is trained to locate traffic signs separated into 4 categories [12]. The second model (model-2) is trained to classify located fragments of the image into one of the 43 classes [15]. Therefore, model-1 and model-2 solve detection and classification issues accordingly. Architecture of the system is shown in Fig. 1.

After traffic sign is detected by model-1, cut fragment is fed to the model-2. Before feeding cut fragment to the model-2 it was resized to the resolution 32 × 32 and preprocessed in the same way as it was done for training (normalization and subtraction of mean image). The result is a class of traffic signs (one of 43 classes) that is returned to the model-1 and used as a label for the appropriate bounding box to be drawn on the input image.

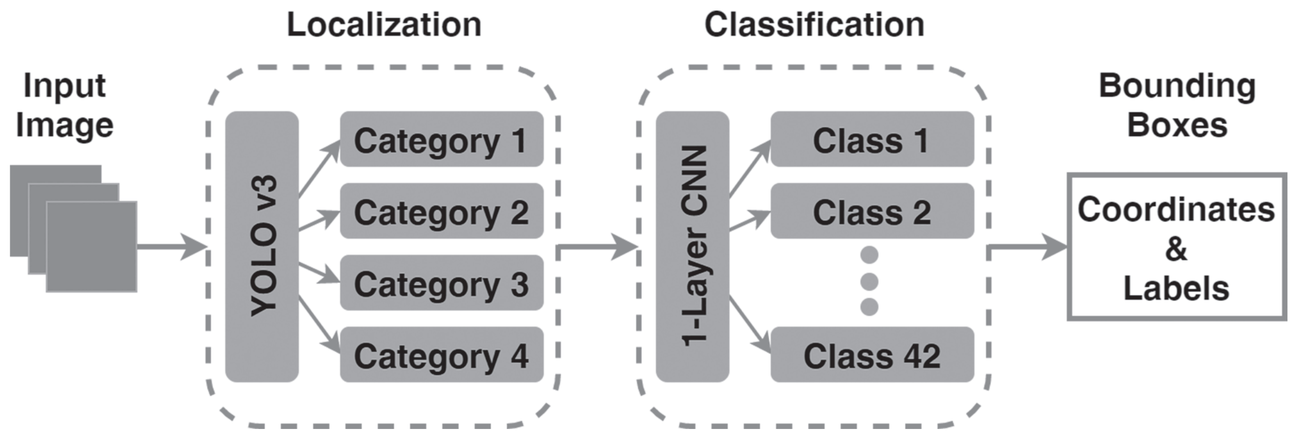

Fig. 1. General flowchart of the entire system

Table 1. Types of traffic signs separated into categories

Prohibitory	Danger	Mandatory	Other
speed limit; no overtaking; no traffic both ways; no trucks	priority at the next intersection; danger; bend; uneven road; slippery road; road narrows; road crossing; construction; traffic signal; snow; animals	go right; go left; go straight; go right or straight; go left or straight; keep right; keep left; roundabout	restriction ends; priority road; give way; stop; no entry

Traffic signs for the model-1 are grouped into the following categories: prohibitory, danger, mandatory and other as shown in Table 1.

The first category, prohibitory, consists of circular traffic signs that have white background and red border line. The second category, danger, consists of triangular traffic signs that have white background and red border line. The third category, mandatory, consists of circular traffic signs that have blue background. The last category, other, consists of traffic signs that do not belong to previous categories.

Model-1 was trained on GTSDB that has 900 RGB images. The dataset includes images with no traffic signs to be used for training. However, it was decided to exclude such images. Resulted dataset was divided into sub-datasets for training and validation in proportion 85 % and 15 %, respectively. The total amount of images for training and validation is 630 and 111. The number of excluded images without traffic signs is 159.

Annotations of bounding boxes in GTSDB are in a single txt file for all images. Originally, coordinates of bounding boxes are defined as top left corner and bottom right corner. Consequently, coordinates were converted into YOLO format as following: centre of bounding box in x, centre of bounding box in y, object width and object height. Calculated coordinates were normalized by real image width and real image height in order to be in the range between 0 and 1. Calculations were made by the following equations:

$$centerX = \frac{(X_{\max} + X_{\min})}{2} \cdot \frac{1}{w},$$

$$centerY = \frac{(Y_{\max} + Y_{\min})}{2} \cdot \frac{1}{h},$$

$$width = (X_{\max} - X_{\min}) \cdot \frac{1}{w},$$

$$heidth = (Y_{\max} - Y_{\min}) \cdot \frac{1}{h},$$

where $X_{\min}$, $Y_{\min}$, $X_{\max}$ and $Y_{\max}$ are original coordinates; w and h are real image width and real image height respectively.

After conversion, annotations were written into text files next to every image with the same names as image files have. As a result, every image has an annotation file where class number and bounding boxes coordinates are recorded. Every single line describes one bounding box. Images themselves were converted from PPM (Portable PixMap) to JPG (Joint Photographic Experts Group) format in order to make possible training of the model-1 in Darknet framework.

The CNN architecture of the model-2 is described in the paper [15]. It has one convolutional layer with 32 filters, ReLU (Rectified Linear Unit) activation function, one downsampling layer with 2 × 2 maximum factor, and hidden affine layer with 500 neurons that is followed by the output layer with 43 neurons as number of classes.

Model-2 was trained on GTSRB with 66000 as total amount of RGB images. The dataset was divided into sub-datasets for training, validation and testing as following: 50000, 12000 and 4000 images, respectively. Before training sub-datasets were normalized by dividing pixels values on 255 and further preprocessed by subtracting mean image that was calculated from images for training.

The dimension of convolutional layer filters for model-2 was chosen equal to 19 × 19 as it has the highest training accuracy [15].

Model-1 uses mAP (mean Average Precision) metric to evaluate accuracy every 1000 iterations during training. To calculate mAP for the entire model-1, firstly, the average precision (AP) is calculated for every class among 4: prohibitory, danger, mandatory and others. Then, the mean of these calculated APs across all classes produces mAP.

AP, in turn, is calculated by considering an area under interpolated Precision (axis y) and Recall (axis x) curve [16, 17]. The curve represents performance of the trained model-1 by plotting a zig-zag graph of Precisions values against Recalls values. Firstly, 11 points are located on Recall curve as following: (0; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1). Then, the average of maximum Precision values is computed for these 11 Recall points.

Precision illustrates how accurate predicted bounding boxes are and demonstrates an ability of the model-1 to detect relevant objects. Recall illustrates all correct predictions of bounding boxes among all relevant ground truth bounding boxes and demonstrates an ability of the model-1 to detect all ground truth bounding boxes.

To plot zig-zag graph, detected bounding boxes are collected and organized in descending order according to their confidences. Next, Precision and Recall are calculated for every detected bounding box by the following equations:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{alldetections},$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{allgroundtruth},$$

where TP (True Positive) represents the number of bounding boxes with correct predictions; FP (False Positive) represents the number of bounding boxes with wrong predictions; FN (False Negative) represents the number of ground truth bounding boxes that were not detected.

To identify that prediction is correct or wrong (True or False) IoU (Intersection Over Union) is used. If IoU is more or equal than a threshold, then predicted bounding box is considered as TP. If IoU is in the range (0; threshold), then predicted bounding box is considered as FP. For this study, the threshold is set to 50 %, indicating that predicted bounding box is correct (TP) if IoU is equal to or is more than 0.5. Consequently, mAP for the entire model-1 can be reported as following: mAP@0.5.

Experimental results

Model-1 used Darknet framework to be trained in. Parameters used for the training are described in Table 2. As can be seen from Table 2, the network size (input) has dimension 608 × 608. Before feeding to the network, the input images were resized to this spatial dimension by the framework without keeping aspect ratio. The 608 × 608 dimension was chosen as the next level up of standard, default 416 × 416 dimension [10–12] and was aimed to increase the accuracy of detecting.

Images were also collected in the batches with 64 items each. Sixteen was set as a number of subdivisions. Batch parameter represents the number of images that were processed during one iteration. Subdivision parameter represents the number of mini batches in one batch that GPU has processed at once. Weights were updated after each such iteration.

To predict bounding boxes, anchors (priors) were used at each scale. The anchors were calculated by k-means clustering for COCO dataset. The width and height of anchors is used to calculate predicted bounding boxes spatial dimensions. The total number of predicted bounding boxes is 10647 (507 for scale 1, 2028 for scale 2 and 8112 for scale 3) that were further filtered with non-maximum suppression technique.

The chosen framework also gave the possibility to augment data on fly during training. The last three parameters in Table 2 randomly changed saturation, exposure and hue during training.

Fig. 2 shows loss and mAP during training with 8000 as total number of iterations.

Table 3 shows mAP results calculated on validation images every 1000 iterations. It also shows the highest found mAP during training on the particular iteration point that is 5700. The 111 images for validation have unique number of ground truth bounding boxes that is 176. To calculate mAP, IoU threshold and confidence threshold were set to 0.5 and 0.25, respectively.

As can be seen from Table 3, the total number of detections at 5700 iterations is 271. After filtering by thresholds (IoU and confidence), the total number of bounding boxes at this iteration point for all four classes is as following: TP = 167, FP = 7, FN = 9. It means, that there are 3 and 4 bounding boxes with wrong predictions (FP) for the classes, mandatory and other, respectively. Nine ground truth bounding boxes among 176 were not detected (FN).

Model-2 was trained on pure “numpy” library during 9000 iterations and reached 0.868 accuracy on testing dataset, 0.867 accuracy on validation dataset and 0.963 accuracy on training dataset for the 19×19 dimension of convolutional layer filters [15].

Test experiments for entire system were performed by GPU Tesla V100 with 16 Gb of RAM (Random Access Memory). Fig. 3 shows testing results.

Average FPS results on processing video files were in the range between 36 and 61 and depended on the number of traffic signs in every frame of the video that, in turn, was in the range from 6 to 1 respectively.

In order to utilize the trained model for inference in real time, the following embeddable GPUs can be applied instead of Tesla V100: Jetson Nano, Jetson TX2, Jetson Xavier NX, Jetson AGX Xavier. These platforms have RAM in the range of 4–32 Gb and can process minimum 4 and maximum 36 video streams in parallel. The specifications make it possible to apply them in real time for the proposed method after training.

Conclusion

The study presents the recognition problem for a variety of traffic sign classes. Due to a number of categories and small amount of images in the dataset for training, it was proposed to separate processes of detection and classification into different models. As for detection, deep convolutional YOLO version 3 model was trained on GTSDB to predict locations of traffic signs among

Table 2. Parameters for the model-1 to be trained with

Parameter	Value
network size (input width, height)	608 × 608
batch	64
subdivisions	16
learning rate	0.001
learning rate decay	0.0005
anchors, scale 1 (large object)	(116, 90), (156, 198), (373, 326)
anchors, scale 2 (medium objects)	(30, 61), (62, 45), (59, 119)
anchors, scale 3 (small objects)	(10, 13), (16, 30), (33, 23)
saturation	1.5
exposure	1.5
hue	0.1

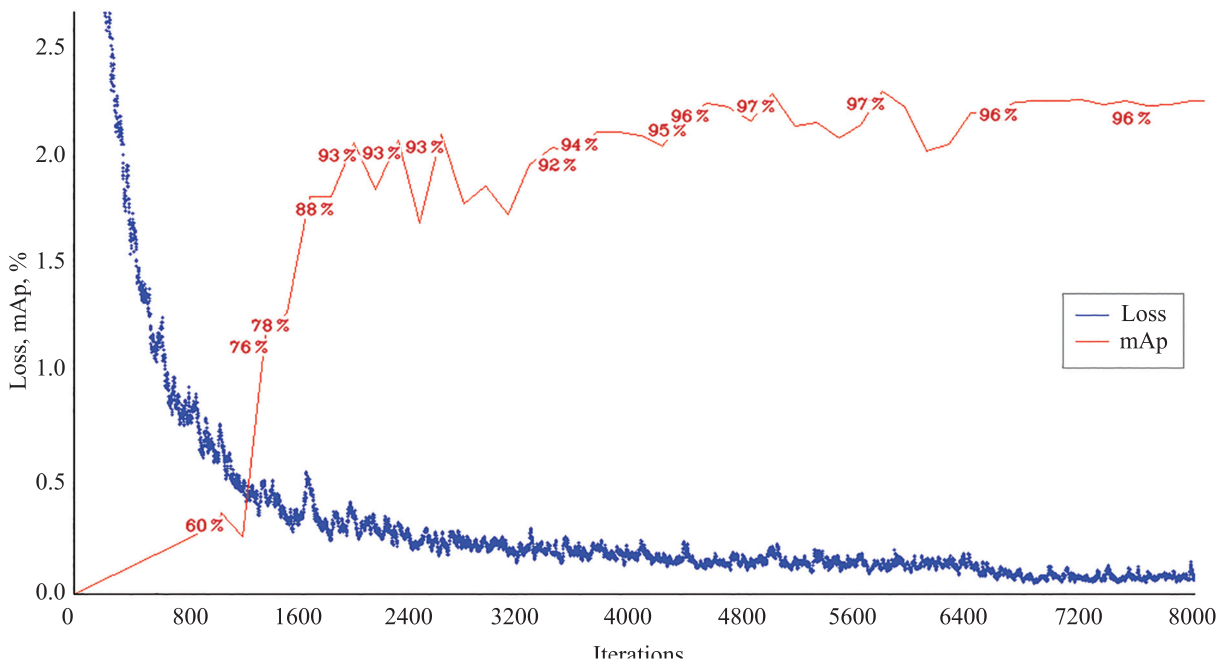

Fig. 2. Loss and mAP graph during training of model-1

4 categories. Further, one-layer convolutional model, trained on GTSRB, was stacked to utilize final classification among one of the 43 classes.

Experiments showed that training of deep network with only 4 categories to detect traffic signs gives high mAP@0.5 accuracy reaching 97.22 % and that is more than in other approaches with considerable number of categories. One more convolutional layer stacked in order to implement classification, creates efficient and fast system that can be used in real time applications.

The proposed method can be compared with other implementations by CNNs. In [1], the authors used 6 categories of Swedish traffic signs dataset (STSD) reaching the average Precision accuracy equal to 97.69 % and the average 92.9 % Recall accuracy. In [2], the authors used 10 categories of STSD reaching mAP@0.5 accuracy equal to 95.2 %. The authors in [2] also used DFG dataset (Slovenian company DFG Consulting d.o.o.) with 200 categories reaching mAP@0.5 accuracy equal to 95.5%.

Table 3. mAP results during training

Iterations	Detections	Average precision				mAp
		Prohibitory	Danger	Mandatory	Other	
1000	5472	57.46 % (TP = 59, FP = 29)	76.32 % (TP = 16, FP = 4)	41.80 % (TP = 12, FP = 9)	63.17 % (TP = 25, FP = 9)	59.69 %
2000	804	95.89 % (TP = 76, FP = 4)	84.47 % (TP = 20, FP = 8)	82.65 % (TP = 21, FP = 8)	84.37 % (TP = 37, FP = 2)	86.85 %
3000	1551	86.47 % (TP = 74, FP = 20)	99.67 % (TP = 24, FP = 6)	89.90 % (TP = 25, FP = 31)	87.91 % (TP = 39, FP = 21)	90.99 %
4000	336	95.89 % (TP = 76, FP = 2)	99.28 % (TP = 23, FP = 0)	78.71 % (TP = 20, FP = 2)	92.66 % (TP = 40, FP = 4)	91.64 %
5000	431	97.00 % (TP = 76, FP = 7)	100.00 % (TP = 24, FP = 0)	95.88 % (TP = 24, FP = 1)	93.43 % (TP = 41, FP = 3)	96.58 %
5700	271	96.25 % (TP = 78, FP = 0)	100.00 % (TP = 24, FP = 0)	98.19 % (TP = 25, FP = 3)	94.44 % (TP = 40, FP = 4)	97.22 %
6000	317	96.25 % (TP = 78, FP = 1)	99.83 % (TP = 24, FP = 1)	86.79 % (TP = 22, FP = 4)	97.09 % (TP = 43, FP = 6)	94.99 %
7000	222	96.25 % (TP = 78, FP = 0)	100.00 % (TP = 24, FP = 0)	92.20 % (TP = 23, FP = 0)	96.95 % (TP = 42, FP = 4)	96.35 %
8000	271	96.25 % (TP = 78, FP = 1)	100.00 % (TP = 24, FP = 0)	92.32 % (TP = 24, FP = 1)	96.93 % (TP = 42, FP = 2)	96.37 %

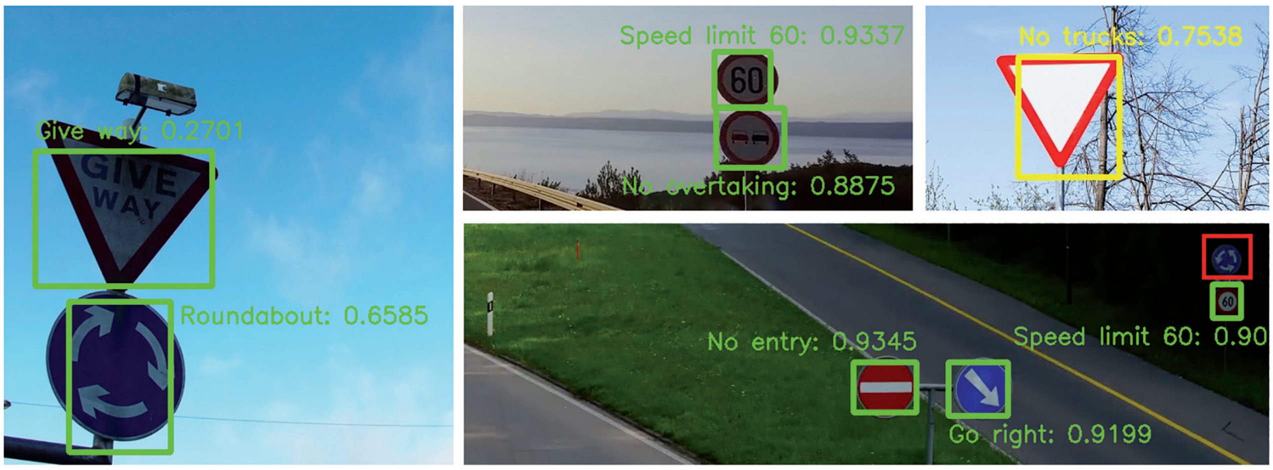

Fig. 3. Detected and classified traffic signs. True detections are shown in green, misclassified in yellow and missing in magenta

Future research is aimed to improve the current model by integrating unsupervised networks. Deep autoencoders are unsupervised neural networks that are trained to differentiate input. This feature of autoencoders is planned to be used in order to detect only traffic signs leaving any other objects. The autoencoder can be trained on GTSRB with only traffic signs on images without a background. Then, images with background (GTSDB) can be passed through trained autoencoder and an output can be used as an input to the deep convolutional YOLO model.

References

1. Zhu Y., Zhang C., Zhou D., Wang X., Bai X., Liu W. Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing*, 2016, vol. 214, pp. 758–766. doi: 10.1016/j.neucom.2016.07.009
2. Tabernik D., Skocaj D. Deep learning for large-scale traffic-sign detection and recognition. *IEEE Transactions on Intelligent Transportation Systems*, 2020, vol. 21, no. 4, pp. 1427–1440. doi: 10.1109/TITS.2019.2913588
3. Chung J.H., Kim D.W., Kang T.K., Lim M.T. Traffic sign recognition in harsh environment using attention based convolutional pooling neural network. *Neural Processing Letters*, 2020, in press. doi: 10.1007/s11063-020-10211-0
4. Mehta S., Paunwala C., Vaidya B. CNN based traffic sign classification using adam optimizer. *Proc. of the International Conference on Intelligent Computing and Control Systems (ICCS 2019)*, 2019, pp. 1293–1298. doi: 10.1109/ICCS45141.2019.9065537
5. Ren S., He K., Girshick R.B., Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, vol. 39, no. 6, pp. 1137–1149. doi: 10.1109/TPAMI.2016.2577031
6. Shrivastava A., Gupta A., Girshick R.B. Training region-based object detectors with online hard example mining. *Proc. 29th IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2016), 2016, pp. 761–769. doi: 10.1109/CVPR.2016.89
7. Zaklouta F., Stanciulescu B. Real-time traffic-sign recognition using tree classifiers. *IEEE Transactions on Intelligent Transportation Systems*, 2012, vol. 13, no. 4, pp. 1507–1514. doi: 10.1109/TITS.2012.2225618
8. Ellahyani A., Ansari M.E., Jaafari I.E., Charfi S. Traffic sign detection and recognition using features combination and random forests. *International Journal of Advanced Computer Science and Applications*, 2016, vol. 7, no. 1, pp. 6861–6931. doi: 10.14569/IJACSA.2016.070193
9. Zaklouta F., Stanciulescu B. Real-time traffic sign recognition in three stages. *Robotics Autonomous Systems*, 2014, vol. 62, no. 1, pp. 16–24. doi: 10.1016/j.robot.2012.07.019
10. Redmon J., Divvala S.K., Girshick R.B., Farhadi A. You only look once: Unified, real-time object detection. *Proc. 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91

Литература

1. Zhu Y., Zhang C., Zhou D., Wang X., Bai X., Liu W. Traffic sign detection and recognition using fully convolutional network guided proposals // Neurocomputing. 2016. V. 214. P. 758–766. doi: 10.1016/j.neucom.2016.07.009
2. Tabernik D., Skocaj D. Deep learning for large-scale traffic-sign detection and recognition // IEEE Transactions on Intelligent Transportation Systems. 2020. V. 21. N 4. P. 1427–1440. doi: 10.1109/TITS.2019.2913588
3. Chung J.H., Kim D.W., Kang T.K., Lim M.T. Traffic sign recognition in harsh environment using attention based convolutional pooling neural network // Neural Processing Letters. 2020. in press. doi: 10.1007/s11063-020-10211-0
4. Mehta S., Paunwala C., Vaidya B. CNN based traffic sign classification using adam optimizer // Proc. of the International Conference on Intelligent Computing and Control Systems (ICCS 2019). 2019. P. 1293–1298. doi: 10.1109/ICCS45141.2019.9065537
5. Ren S., He K., Girshick R.B., Sun J. Faster R-CNN: towards real-time object detection with region proposal networks // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. V. 39. N 6. P. 1137–1149. doi: 10.1109/TPAMI.2016.2577031
6. Shrivastava A., Gupta A., Girshick R.B. Training region-based object detectors with online hard example mining // Proc. 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016). 2016. P. 761–769. doi: 10.1109/CVPR.2016.89
7. Zaklouta F., Stanciulescu B. Real-time traffic-sign recognition using tree classifiers // IEEE Transactions on Intelligent Transportation Systems. 2012. V. 13. N 4. P. 1507–1514. doi: 10.1109/TITS.2012.2225618
8. Ellahyani A., Ansari M.E., Jaafari I.E., Charfi S. Traffic sign detection and recognition using features combination and random forests // International Journal of Advanced Computer Science and Applications. 2016. V. 7. N 1. P. 6861–6931. doi: 10.14569/IJACSA.2016.070193
9. Zaklouta F., Stanciulescu B. Real-time traffic sign recognition in three stages // Robotics Autonomous Systems. 2014. V. 62. N 1. P. 16–24. doi: 10.1016/j.robot.2012.07.019
10. Redmon J., Divvala S.K., Girshick R.B., Farhadi A. You only look once: Unified, real-time object detection // Proc. 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016). 2016. P. 779–788. doi: 10.1109/CVPR.2016.91

11. Redmon J., Farhadi A. YOLO9000: Better, faster, stronger. *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017, pp. 6517–6525. doi: 10.1109/CVPR.2017.690
12. Redmon J., Farhadi A. YOLOv3: An incremental improvement. *arXiv*, 2018, abs/1804.02767.
13. Houben S., Stallkamp J., Salmen J., Schlipsing M., Igel C. Detection of traffic signs in real-world images: The German traffic sign detection benchmark. *Proc. of the International Joint Conference on Neural Networks (IJCNN 2013)*, 2013, pp. 6706807. doi: 10.1109/IJCNN.2013.6706807
14. Stallkamp J., Schlipsing M., Salmen J., Igel C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 2012, vol. 32, pp. 323–332. doi: 10.1016/j.neunet.2012.02.016
15. Sichkar V.N., Kolyubin S.A. Effect of various dimension convolutional layer filters on traffic sign classification accuracy. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2019, vol. 19, no. 3, pp. 546–552. doi: 10.17586/2226-1494-2019-19-3-546-552
16. Davis J., Goadrich M. The relationship between precision-recall and ROC curves. *ACM International Conference Proceeding Series*, 2006, vol. 148, pp. 233–240. doi: 10.1145/1143844.1143874
17. Everingham M., Van Gool L., Williams C.K., Winn J.M., Zisserman A. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 2010, vol. 88, no. 2, pp. 303–338. doi: 10.1007/s11263-009-0275-4

11. Redmon J., Farhadi A. YOLO9000: Better, faster, stronger // Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). 2017. P. 6517–6525. doi: 10.1109/CVPR.2017.690
12. Redmon J., Farhadi A. YOLOv3: An incremental improvement // arXiv. 2018. abs/1804.02767.
13. Houben S., Stallkamp J., Salmen J., Schlipsing M., Igel C. Detection of traffic signs in real-world images: The German traffic sign detection benchmark // Proc. of the International Joint Conference on Neural Networks (IJCNN 2013). 2013. P. 6706807. doi: 10.1109/IJCNN.2013.6706807
14. Stallkamp J., Schlipsing M., Salmen J., Igel C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition // Neural Networks. 2012. V. 32. P. 323–332. doi: 10.1016/j.neunet.2012.02.016
15. Sichkar V.N., Kolyubin S.A. Effect of various dimension convolutional layer filters on traffic sign classification accuracy // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 3. С. 546–552. doi: 10.17586/2226-1494-2019-19-3-546-552
16. Davis J., Goadrich M. The relationship between precision-recall and ROC curves // ACM International Conference Proceeding Series. 2006. V. 148. P. 233–240. doi: 10.1145/1143844.1143874
17. Everingham M., Van Gool L., Williams C.K., Winn J.M., Zisserman A. The pascal visual object classes (VOC) challenge // International Journal of Computer Vision. 2010. V. 88. N 2. P. 303–338. doi: 10.1007/s11263-009-0275-4

Authors

Valentyn N. Sichkar — Postgraduate, Software Engineer, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 57209451261, ORCID ID: 0000-0001-9825-0881, vsichkar@itmo.ru

Sergey A. Kolyubin — D.Sc., Associate Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 35303066700, ORCID ID: 0000-0002-8057-1959, s.kolyubin@itmo.ru

Авторы

Сичкар Валентин Николаевич — аспирант, программист, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 57209451261, ORCID ID: 0000-0001-9825-0881, vsichkar@itmo.ru

Колюбин Сергей Алексеевич — доктор технических наук, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 35303066700, ORCID ID: 0000-0002-8057-1959, s.kolyubin@itmo.ru