Traffic sign recognition using the mask R-CNN

Mykola Korablyov[1,†], Natalia Axak[1,†], Ihor Ivanisenko[2,3,*,†], Maksym Kushnaryov[1,†] and Igor Kobzev[4,†]

[1] *Kharkiv National University of Radio Electronics, Kharkiv 61166, Ukraine*
[2] *University of Jyväskylä, Jyväskylä 40014, Finland*
[3] *Kharkiv National Automobile and Highway University, Kharkiv 61002, Ukraine*
[4] *Simon Kuznets Kharkiv National University of Economics, Kharkiv 61166, Ukraine*

Abstract

Today, intelligent technologies are developing at a rapid pace, which, in turn, leads to the development of intelligent transport systems. Therefore, constructing traffic sign recognition systems using machine and deep learning technologies is urgent. Traffic sign recognition is a computer visualization problem that can be solved using convolutional neural networks. The analysis of the most effective models of convolutional neural networks of image processing was carried out to choose the most suitable one for recognizing traffic signs: R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN. The analysis showed that applying Mask R-CNN for traffic sign recognition is appropriate. It effectively detects objects in the image, creates a high-quality segmentation mask for each instance, and can be used in vehicle systems. Considering issues of traffic sign recognition using Mask R-CNN, the work consists of implementing relevant stages and components. The training of Mask R-CNN, which must learn to detect objects in the image and segment images, is considered. Experimental studies on Mask R-CNN for traffic sign recognition were conducted, for which a neural network training web application was created. Examples of training and testing of the work of Mask R-CNN on the recognition of traffic signs are presented, from which it is clear that Mask R-CNN, based on the trained classes, clearly finds and processes several traffic signs in the image. This makes it possible to expand the number of classes and objects for recognition and improve image processing quality.

Keywords

recognition, traffic sign, model, regional convolutional neural networks, learning, segmentation, dataset, web application

1. Introduction

Object detection in images is a key component of many deep-learning models and has undergone several significant transformations in recent years. Object detection algorithms are used in areas such as self-driving vehicles, security cameras, robotics, and almost all

MoMLeT-2024: 6th International Workshop on Modern Machine Learning Technologies, May, 31 - June, 1, 2024, Lviv-Shatsk, Ukraine

* Corresponding author.

† These authors contributed equally.

mykola.korablyov@nure.ua (M. Korablyov); nataliia.axak@nure.ua (N. Axak); ihor.i.ivanisenko@jyu.fi (I. Ivanisenko); maksym.kushnarov@nure.ua (M. Kushnaryov); ikobzev12@gmail.com (I. Kobzev)

0000-0002-8931-4350 (M. Korablyov); 0000-0001-8372-8432 (N. Axak); 0000-0002-2679-959X (I. Ivanisenko); 0000 0002-3772-3195 (M. Kushnaryov); 0000-0002-7182-5814 (I. Kobzev)

applications that involve visualization, such as medicine, as well as emerging areas such as self-service stores, cash registers, etc. The main problem was that many applications require object detection in real time. Today, this problem has been solved, as a whole set of methods, models, and algorithms have been developed, which can be used to detect objects in real time. A separate actual task of object detection in images is traffic sign recognition. Today, there is a significant increase in the number of vehicles on the roads, which leads to traffic jams and casualties. Therefore, improving the efficiency and safety of road traffic is an extremely important need of the hour. The attention of many car manufacturers is now focused on the creation of unmanned vehicles, which involves the introduction of a whole complex of software and hardware solutions, that work, including based on artificial intelligence technologies.

Automatic traffic sign recognition systems are widely used to increase the safety of motor vehicles. Every year, the need for an automatic traffic sign recognition system becomes more and more urgent. These systems are widely used in autopilots and driver assistants to increase the safety of motor vehicles. The systems can help to adhere to the established speed regime and observe travel restrictions and overtaking, which will help to significantly reduce accidents on the roads.

Today, intelligent technologies are developing at a rapid pace, which, in turn, leads to the development of intelligent transport systems. Currently, in various countries of the world, unmanned vehicles are widely used, which, to improve traffic efficiency, are equipped to significantly reduce the number of traffic accidents and reduce traffic jams. This is possible if autonomous vehicles are equipped with navigation aids and a traffic sign recognition system based on machine and deep learning technologies.

Traffic signs installed on the sides of the road provide navigational information and necessary warnings during trips. Therefore, vehicles must be equipped with automated systems that can analyze traffic events in real-time, and recognize and understand traffic signs placed on the roadsides while driving to ensure traffic safety. Established traffic sign recognition systems work with text signs or datasets containing clear traffic data, regardless of usage conditions. Therefore, the construction of traffic sign recognition systems using machine and deep learning technologies is an urgent task.

2. Analysis of approaches to traffic sign recognition

Image processing methods, models, and algorithms can solve the following tasks [1]:

- Classification of objects.
- Localization of objects.
- Object detection.
- Segmentation.

When classifying objects, the input data is the image, the output is the class of the object represented in the image. The number of classes is determined during system design.

Localization of objects means the determination of the location of the object in the image. As a rule, the object is highlighted in the image by a rectangle.

Detection is understood as a set of localization and classification operations performed sequentially. If several objects are detected in the image, each of them is classified separately. There are two main approaches to detection:

- Bounding box detection.
- Landmark detection.

Detection using bounding frames is based on the selection using some rectangle of the part of the image in which the object is located, having the coordinates of the center, height, and width. Segmentation refers to the process of dividing an image into several segments by establishing. identical visual characteristics of pixels that belong to the same type of object in the image. The result of the algorithm is a set of segments covering the image.

Traffic sign recognition is a computer imaging task that involves identifying a specific object, scene, or other feature in an image. There are two main approaches to image recognition [2]:

1. An approach based on signs.
2. Learning-based approach.

A feature-based approach is to first extract certain features from an image, which are then used to classify the image. Features can be low-level, such as color, texture, or shape, or high-level, such as context or semantics. A training-based approach is to train a model to recognize certain objects or scenes using a labeled image dataset.

The use of color segmentation is a common method that can be used to identify traffic signs in real-time using simple, inexpensive equipment. In [3], the use of color-based segmentation in the detection stage is considered, while taking into account the difference in RGB components ensures the reliability of the results. Color segmentation works when analyzing images with small signs or low-resolution traffic signs. After pixel-level color-based segmentation, traffic signs are classified using the Support Vector Machines (SVM) [4]. In traffic sign recognition, the SVM method is widely used together with other methods such as decision trees, random forests, and directed gradient histograms [4-6].

Today, there are many approaches to traffic sign recognition that use machine learning techniques [7-9]. The general approach to traffic sign recognition consists of two main stages: detection and classification. A large number of tasks for the detection and classification of objects in the image are solved using neural networks with different architectures [10]. At the same time, the use of different types of neural networks in combination with other methods allows for obtaining high-accuracy indicators. Many different neural network architectures are used for traffic sign recognition. Such variants of Convolutional Neural Networks (CNN) as R-CNN, Fast R-CNN, and Faster R-CNN have been most actively implemented, the use of which eliminates the need for manual feature extraction [11,12]. It was shown in [12] that using Faster R-CNN for traffic sign recognition increased the speed of the model compared to Fast R-CNN. To detect objects in the image, the newly developed Mask R-CNN is used [10, 13], which is an extension of Faster R-CNN and, in addition to the class label, provides an additional object mask and coordinates for bounding frames, and also increases the accuracy of traffic sign recognition.

Thus, to solve the problem of traffic sign recognition, we will use one of the models based on the use of one or another neural network.

3. Selection of a neural network model for traffic sign recognition

The most common models of neural networks, which are an effective means of object recognition, include CNN, autoencoders, transformers, global models, etc. It should be noted that most models based on neural networks provide real-time mode, reliable functionality, and a high level of accuracy. We will analyze the most effective models of neural networks for image processing to choose the most suitable one for recognizing traffic signs.

The basis of Regional Convolutional Neural Networks (R-CNN) is obtaining a set of regions that probably contain objects for classification, and then their further processing by a convolutional neural network [10]. Representatives of such networks are R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN, which is one of the latest models in the family of object detector algorithms [11-13].

R-CNN accepts an image as an input and forms up to 2000 regions of different sizes on it using a selective search algorithm. A region is a part of an image where there is a high probability of finding target objects. Each region is assigned a certain class and bounding box. In the next step, R-CNN uses a large CNN to compute features for each previously proposed region. At the final stage, each region is classified by the SVM and linear regression to obtain the most accurate coordinates of the object. The disadvantage of the R-CNN model is its slowness and energy consumption.

Since R-CNN processes each CNN region independently, this slows down the model significantly. To solve this problem, Fast R-CNN performs image processing once on the entire image. Fast R-CNN, based on the selective search algorithm, processes the entire image in parallel with a regular CNN to obtain features, which ensures the receipt of proposals for the regions of object placement. Then the obtained features and regions are processed in the region subsampling layer (RoI pooling), in which the region is transformed from image coordinates to feature map coordinates, obtaining a feature vector of fixed length as an output.

Each feature vector is fed into fully connected layers (FC), the result of which is then output to two output layers:

- Softmax – to assess whether an object belongs to a class.
- Regressions – to specify the coordinates of the object bounding box.

The first layer, using the softmax function, determines the possibility of assigning the object to one or another class, taking into account the background class of the entire image. The second layer outputs real numbers describing the position of the bounding box for each object. Thus, the main differences of Fast R-CNN are as follows:

- During processing, a set of features is generated for the entire image at once, and not for each frame, from which features for parallel obtained regions are then extracted using a special layer.
- The SVM and linear regression are not used by using additional layers of the full neural network.

In Faster R-CNN, the selective search algorithm is replaced by the Region Proposal Network (RPN) for searching regions. Fast R-CNN is used for detection. The object detection algorithm is based on predicting the object category and the deviation from the true bounding box for a large number of generated keyframes, followed by their filtering. The RPN receives features from the CNN as input, based on which it forms a set of proposals of regions for the placement of objects with some evaluation. To reduce the number of regions, the Non-Maximal Suppression (NMS) algorithm is used, which significantly reduces the number of regions. The received data is fed into the Fast R-CNN algorithm. Due to the use of the same convolutional layers in both networks, the speed of operation increases significantly and the object detection model can work in a mode close to the real-time mode.

Among the R-CNN models, the Mask Region-based Convolutional Neural Network (Mask R-CNN) should be singled out, which has many advantages and effectively detects objects in the image. Mask R-CNN is a type of CNN and is an extension of Faster R-CNN, specially designed for solving tasks of semantic segmentation of objects in images. The main idea is to add the layer to the Faster R-CNN architecture to generate a binary mask of each selected object. Mask R-CNN predicts the position of the mask covering the detected object, solving the problem of segmentation of image instances at the pixel level, which significantly improves the recognition accuracy.

Mask R-CNN, unlike Faster R-CNN, which effectively finds objects in the image, can create a high-quality segmentation mask for each instance. That is, Mask R-CNN can not only determine the location of objects in the image but also accurately outline the shape of each object. The architecture of Mask R-CNN, which is shown in Figure 1, consists of convolution layers, Region Proposal Networks (RPNs), and Fully Connected Networks (FCNs) [13].

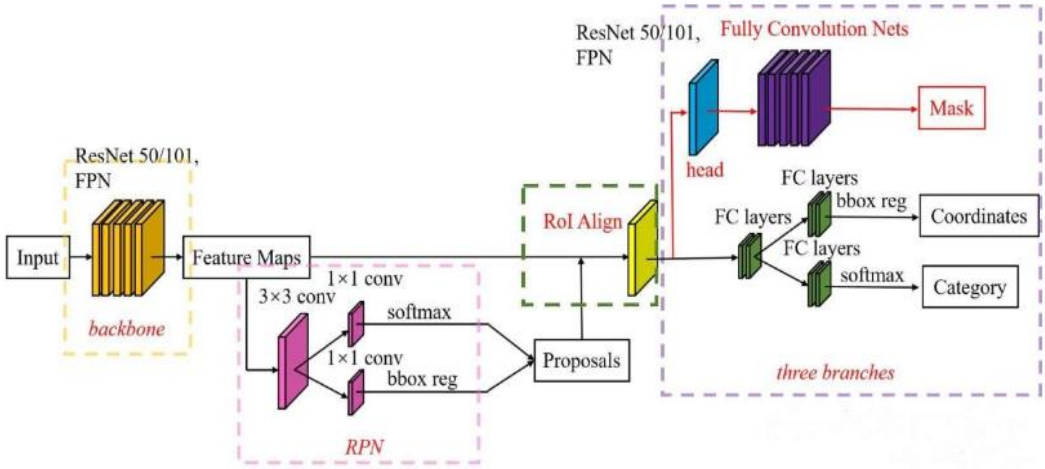

Figure 1: Mask R-CNN architecture.

In the Mask R-CNN architecture, two stages can be distinguished:

- Region supply network for object search.
- Head networks for object classification and segmentation mask prediction.

The first step in processing the input image is a pre-trained CNN, such as ResNet, which extracts high-level features from the image that are important for finding the complex patterns required for object detection. A feature pyramid network (FPN) by combining features from different layers of the backbone creates a multi-level feature pyramid that includes objects with different spatial resolutions, covering both high-resolution objects containing semantic information and low-resolution objects that provide more accurate spatial details of objects.

The RPN plays a crucial role in identifying potential objects in the image. Using a sliding window method, RPN scans the image to identify areas that may contain objects. By creating object-bound frames, RPN narrows the scope of interest. These suggestions are then refined and used in subsequent stages for more detailed analysis. The integration of RPN in Mask R-CNN allows for real-time processing and lower computational costs compared to stand-alone object detection methods.

The Region of Interest (ROI) ALIGN solves the problem of spatial discrepancies caused by the quantization process. ROI Align uses bilinear interpolation to accurately extract feature maps for each proposed feature region. This method ensures that the obtained features exactly match the objects, resulting in more accurate segmentation and classification. The classification and bounding box regression are Mask R-CNN components that simultaneously perform object classification and refine bounding boxes. For each region proposed by the RPN, the network predicts an object class by distinguishing different types of image objects. In addition to classification, the coordinates of each proposal of the bounding box are adjusted, specifying its size and position for more accurate coverage of the object.

Using a Fully Convolutional Network (FCN) for each ROI, a binary mask is generated that outlines the exact shape of the object. Mask prediction is done pixel by pixel, which allows for detailed and accurate segmentation. This is especially important for tasks that require a detailed outline of objects, such as the task of recognizing traffic signs.

Thus, using Mask R-CNN allows you to detect objects in the image while creating a high-quality segmentation mask for each instance. Mask R-CNN is easy to learn, it is easy to generalize it to other tasks, for example, to estimate human pose in the same structure, etc. Mask R-CNN provides an opportunity to obtain not only a library of regions but also accurate masks for objects in the image. This makes it very effective for tasks where the accuracy of determining the outline of an object is important.

All these factors make Mask R-CNN a powerful tool for solving various computer vision problems, as well as object segmentation in images.

Thus, the analysis of neural networks with the aim of their application for traffic sign recognition showed that for these purposes it is appropriate to use Mask R-CNN, which effectively detects objects in the image, creates a high-quality segmentation mask for each instance, and can be used in systems motor vehicles.

4. Implementation of traffic sign recognition using Mask R-CNN

The work of Mask R-CNN on the recognition of traffic signs consists of the following main stages and components:

1. Selection of RPN that contains objects.
2. Extracting signs. The images and regions selected by the RPN are fed into a convolutional neural network for feature extraction.
3. Main branch. Includes feature submission for classification and regression, similar to Faster R-CNN.
4. Mask head. An additional layer that is responsible for generating binary masks for objects. This layer has its convolutional architecture and is used to accurately define the shape and position of each object in the image.
5. Loss and learning function. A loss function is used, which takes into account both classification and regression losses, as well as losses relative to mask generation.

Training a Mask R-CNN can be a challenging task. This is because the neural network must learn to perform two tasks: object detection in the image and image segmentation. These tasks are quite complex and the neural network must be large enough to perform them. Mask R-CNN training consists of the following stages:

1. Data preparation. In this step, you need to collect a dataset of human-labeled images. Descriptions should include the coordinates of the contours of objects in the image, as well as their class.
2. Data conversion. At this stage, you need to convert the data into a format that can be used for neural network training.
3. Setting the neural network parameters. At this stage, you need to configure the neural network parameters, such as learning rate and batch size.
4. Neural Network Training. At this stage, the neural network is trained on a set of image data.
5. Experimental evaluation. In this step, you need to evaluate the accuracy of the neural network on an image dataset that was not used for training.

The image dataset for training the Mask R-CNN should include images with different objects to be detected. Descriptions must be accurate and consistent. If the descriptions are not accurate, the neural network can learn to detect false objects.

Many different datasets can be used to train a Mask R-CNN. Some of the more common datasets include:

- COCO. This dataset includes 80,000 images with 80 different object classes.
- PASCAL VOC. This dataset includes 11,000 images with 20 different object classes.
- MS COCO-Stuff. This dataset includes 118,000 images with 171 different object classes.

Once the image dataset is collected, it needs to be converted into a format that can be used to train the neural network. For this, you can use special tools such as COCO API or

PASCAL VOC API. Training rate and batch size are two important parameters that need to be adjusted before starting to train a neural network. The learning rate determines how quickly the neural network will update its parameters. The batch size determines how many images will be used for one neural network update. Training a neural network can take a long time. It depends on the dataset size, batch size, and training speed. Once the neural network is trained, it needs to be evaluated on an image dataset that was not used for training. This will help determine how well the neural network performs on images it has not seen before.

5. Experimental results

When using Mask R-CNN sequence of actions must be performed. It is necessary to prepare a dataset with images of the object that needs to be recognized. Importantly, the more images of an object are based in the dataset in different angles, backgrounds, and colors, the more accurately the neural network will be able to recognize subsequent images and objects. Next, it is necessary to perform annotations for the dataset, which consist of the following steps:

Step 1 – you need to add many different images, in different positions, under different lighting, among other objects, and against different backgrounds.

Step 2 – transition to annotations, i.e. manual determination of object position and label assignment. This can be done using the open-source project "www.makesense.ai".

Step 3 – create a project in the web application and upload the collected dataset with images.

Step 4 – Create a marker for further use to determine to which class the object found in the image belongs.

Step 5 – Manual selection of objects in the images using a special tool "polygon" (polygon) so that Mask R-CNN learns with the help of the initial dataset and can recognize similar objects already in the images that are not located in the dataset.

Step 6 - after manually selecting an object in the image (in some cases, there may be several of them in one image), it is necessary to assign the created labels to each selected object in the image.

Step 7 - all subsequent images must be processed according to the previous steps.

Step 8 – after processing all the images in the dataset, it is necessary to export the "JSON" file in the "SOSO" format. With this file, which contains the coordinate data of all selected points in each image, Mask R-CNN can be trained.

To obtain recognition results, a neural network training application was created. First, all modules and libraries must be downloaded for Mask R-CNN to work. After compilation, the system creates a trained Mask R-CNN model. Next, it is necessary to upload the dataset archive in ".zip" format and the downloaded file with annotations in "JSON" format to the files of the created project. Extracting the image archive from the dataset and annotation and assigning values to the variables was done using the web application. The results of extracting manually processed images of the dataset from the "JSON" file, which was made using the web service, are presented in Figure 2, which shows the original images of objects (left) and their images with a neural network mask (right).

training.

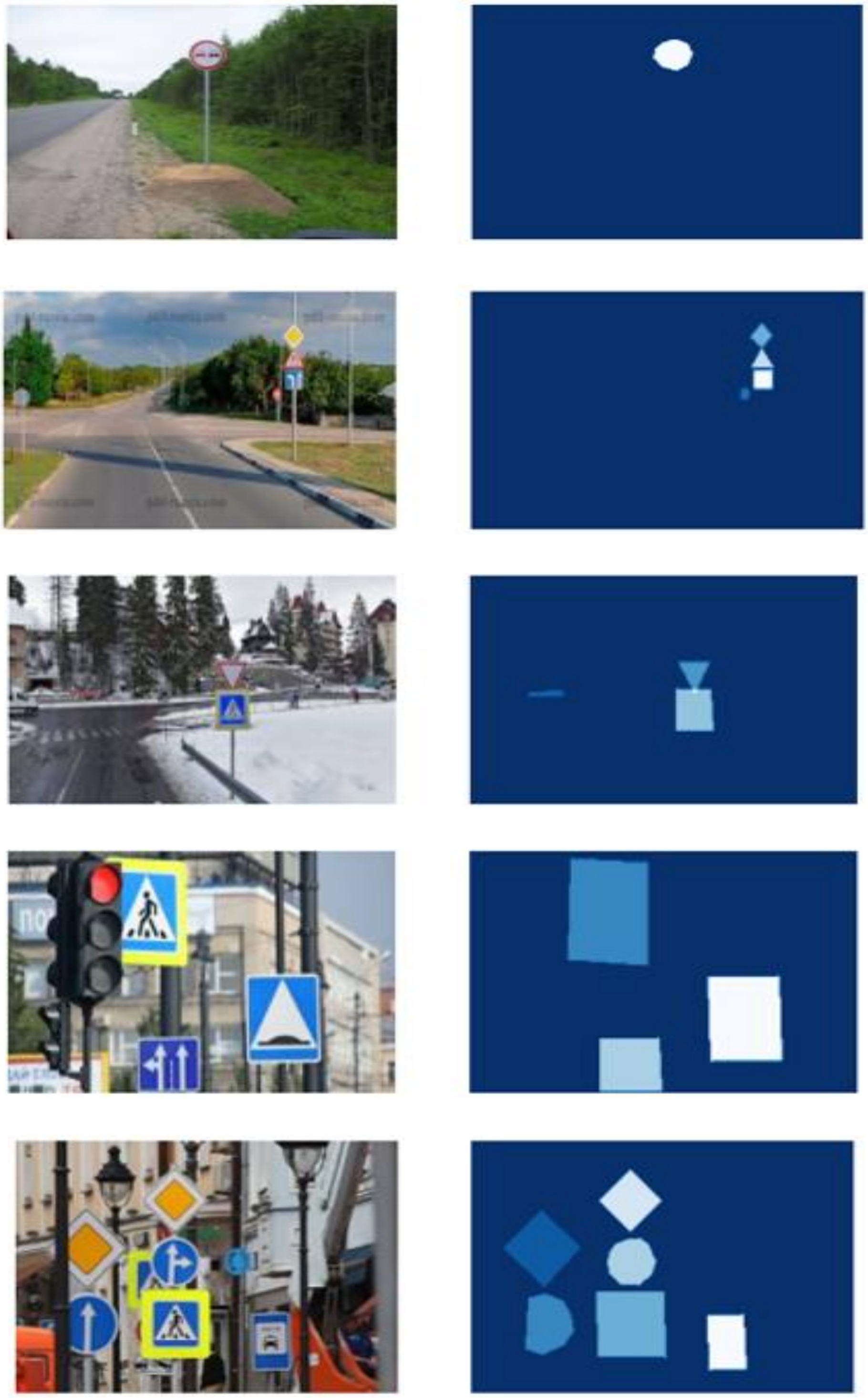

Figure 2: Image of traffic signs with a neural network mask.

Then, the following actions were performed in two stages. In the first stage, the number of images was checked and preparatory processing was performed for Mask R-CNN training.

In the second stage, the neural network itself was trained directly based on the prepared annotations and dataset. In the “logs” folder, files in “.h5” format were created, which are the results of Mask R-CNN training. Then, for subsequent image processing based on the trained model for traffic sign recognition, the last trained model, which is the most accurate of all the previous ones, was loaded.

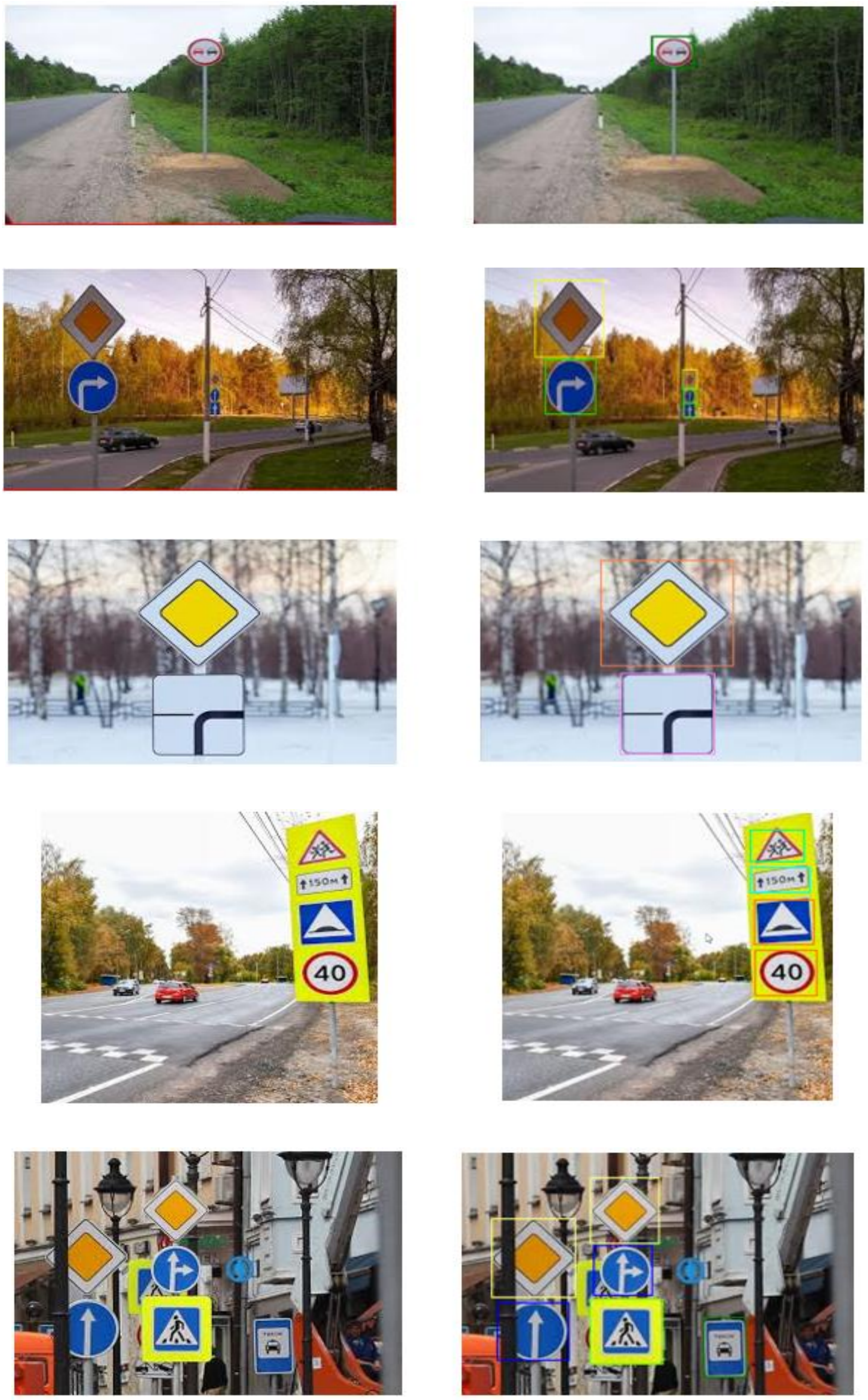

Figure 3: Examples of original images (left) and processed images (right)

Considered examples of testing the work of Mask R-CNN on the recognition of traffic signs on random images. Figure 3 shows examples of original images of objects with traffic signs (left) and examples of images processed by the Mask R-CNN network (right). From the given examples, it can be seen that Mask R-CNN based on trained classes finds and processes several traffic signs in the image. This allows you to expand the number of classes and objects for recognition and improve the quality of image processing with Mask R-CNN. In general, Mask R-CNN is a powerful tool for object segmentation, which significantly improves the capabilities of computer vision systems in various fields of application, in particular, in traffic sign recognition systems.

6. Conclusions

Every year, the need for automatic traffic sign recognition systems becomes more and more urgent. These systems are widely used in autopilots and driver assistants to increase the safety of motor vehicles. The systems can help to adhere to the established speed regime and observe travel restrictions and overtaking, which will help to significantly reduce accidents on the roads.

An analysis of approaches to traffic sign recognition as a task of object detection in the image was carried out. Since road sign recognition is a computer visualization task, both a feature-based approach and a learning-based approach can be used to solve it. The most effective approach to traffic sign recognition is the use of machine and deep learning technologies, in particular, convolutional neural networks.

The analysis of the most effective models of convolutional neural networks of image processing was carried out to choose the most suitable one for recognizing traffic signs: R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN. Their analysis showed that it is appropriate to use Mask R-CNN for traffic sign recognition, which effectively detects objects in the image, creates a high-quality segmentation mask for each instance, and can be used in vehicle systems.

The implementation of the task of recognizing traffic signs using Mask R-CNN is considered, the work of which is presented as a sequence of execution of the relevant stages and components. Mask R-CNN training is focused on the tasks of image object detection and image segmentation and consists of the following stages: data preparation, data conversion, neural network parameter settings, neural network training directly, and experimental evaluation of its effectiveness.

Experimental studies on the use of Mask R-CNN for traffic sign recognition were conducted, for which a corresponding dataset was prepared. Annotations were performed for the dataset, consisting of the implementation of the corresponding steps. To obtain recognition results, a neural network training web application was created, with the help of which images of traffic signs with a neural network mask were obtained using downloaded relevant modules and libraries.

Considered test examples of the work of Mask R-CNN on the recognition of traffic signs on random images, which showed that Mask R-CNN based on trained classes finds and processes several traffic signs on the image, allowing to expansion of the number of classes and objects for recognition and improve image processing quality.

References

[1] R. Archana, P.S. Eliahim Jeevaraj. Deep learning models for digital image processing: a review. Artificial Intelligence Review (2024) 57:11. https://doi.org/10.1007/s10462-023-10631-z.

[2] R. Szeliski. Computer Vision: Algorithms and Applications, 2nd Edition. Springer Nature, (2022) 925. https://cord.isir.upmc.fr/pdfs/courses/rdfia/SzeliskiBook_draft.pdf.

[3] A. Ruta, Y. Li, and X. Liu. Real-time traffic sign recognition from video by class-specific discriminative features. Pattern Recognition, 43(1) (2010) 416–430. doi: 10.1016/j.patcog.2009.05.018.

[4] F. Zaklouta, and B. Stanciulescu. Real-time traffic sign recognition using tree classifiers. IEEE Transactions on Intelligent Transportation Systems, 13(4) (2012) 1507–1514. doi:10.1109/TITS.2012.2225618.

[5] A. Ellahyani, M. El Ansari, and I. El Jaafari. Traffic sign detection and recognition based on random forests. Applied Soft Computing, 46 (2016) 805–815. doi: 10.1016/j.asoc.2015.12.041.

[6] J. Greenhalgh, and M. Mirmehdi. Real-time detection and recognition of road traffic signs. IEEE Transactions on Intelligent Transportation Systems, 13(4) (2012) 1498–1506. doi:10.1109/TITS.2012.2208909.

[7] H.H. Aghdam, E.J. Heravi, and D. Puig. A practical approach for detection and classification of traffic signs using convolutional neural networks. Robotics and Autonomous Systems, 84 (2016) 97–112. doi: 10.1016/j.robot.2016.07.003.

[8] M.M. William, P.S. Zaki, B.K. Soliman, K.G. Alexsan, M. Mansour, M. El-Moursy, and K. Khalil. Traffic Signs Detection and Recognition System using Deep Learning. Ninth International Conference on Intelligent Computing and Information Systems (ICICIS) (2019). 160-166. doi:10.1109/ICICIS46948.2019.9014763.

[9] A. Karne, R. Karne, K. K. Vaigandla, and A. Arunkumar. Convolutional Neural Networks for Object Detection and Recognition. Journal of Artificial Intelligence Machine Learning and Neural Network, vol.3, no 2 (2023) 1-13. doi:10.55529/jaimlnn.32.1.13.

[10] A. Barade, H. Poornachandran, K.M. Harshitha, E.D. Shiloah, R.R.C. Sunil. Automatic Traffic Sign Recognition System Using CNN. International Journal of Information Retrieval Research, IGI Global, Vol. 12, Iss. 1 (2022) 1-14. https://ideas.repec.org/s/igg/jirr00.html.

[11] G. Zhang, Y. Peng, and H. Wang. Road Traffic Sign Detection Method Based on RTS R-CNN Instance Segmentation Network. Sensors (2023) 23, 6543. https://doi.org/10.3390/s23146543.

[12] Z. Zuo, K. Yu, Q. Zhou, X. Wang, and T. Li, Traffic signs detection based on Faster R-CNN. IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW) (2017) 286-288, doi:10.1109/ICDCSW.2017.34.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. IEEE International Conference on Computer Vision (ICCV) (2017) 2980-2988. doi:10.1109/ICCV.2017.322.