Smart eco-friendly refrigerator based on implementation of architectures of convolutional neural networks

Anna E. Alekhina[1, 2], *Mikhail* G. Dorrer[1*], and *Alexander* G. Ovchinnikov[3]

[1]Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarsky Rabochy Ave., Krasnoyarsk, 660037, Russian Federation
[2]Siberian Federal University, 79 Svobodny pr., Krasnoyarsk, 660041, Russia
[3]Solution Factory, Krasnoy Armii, 10c3, Krasnoyarsk, 660001, Russian Federation

Abstract. The article discusses the solution to the problem of choosing the architecture of a convolutional neural network for use in the computer vision of a smart vending refrigerator. Comparative tests decided the architectures of convolutional neural networks YOLOv2, YOLOv3, YOLOv4, Mask R-CNN, and YOLACT ++ on a standard MS COCO dataset, and then on datasets formed from images of typical smart refrigerator products. As a result of comparative tests, the best performance was demonstrated by the YOLOv3 architecture, trained based on a normalized dataset, supplemented with examples with complex intersections of samples without preprocessing examples. The obtained results substantiated the architecture used in computer vision of serially produced "smart" vending machines.

1 Introduction

This article is devoted to presenting the results of R&D on the development of the computer vision module of the cloud software platform for the robotic retail trade. R&D was carried out within the framework of the grant agreement of the Fund for the Promotion of Innovations, agreement No. 641GRNTIS5 / 63440 of 10.12.

The task set within the project framework is to find objects in the image in real-time. At the same time, a limitation on the power of the hardware that analyzes images is imposed on the system's operation because this hardware is located on-site, directly at the point of sale. Moreover, the transmission of video information to the server for analysis can lead to problems in unstable communication and will require significant expenses for Internet traffic.

2 Related works on the topic of methods for identifying and classifying objects in a video stream

To solve the problems of finding objects in the image, two main parallel developing approaches are currently used:

* Corresponding author: dorrer_mg@sibsau.ru

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

1. Two-stage methods, they are also "region-based methods" - an approach divided into two stages. Such methods include Faster R-CNN, Mask R-CNN, etc.
2. One-stage methods - an approach that does not use a separate algorithm for generating regions. The one-step methods include different versions of YOLO.

These methods are quite well-known, are used in many computer vision tasks, and showed high accuracy and speed indicators on the MS COCO dataset.

Currently, in the tasks of object classification in real-time, it is necessary to sacrifice something. For example, the model works slowly, but it classifies objects qualitatively, or the recognition operation is performed quickly, but the definition quality drops sharply.

For each task, it is necessary to make this choice and select the optimal architecture. In the course of the described work, the following methods and architectures were applied:

- YOLOv2 [1];
- YOLOv3 [2];
- YOLOv4 [3];
- Mask R-CNN [4].

To improve image analysis, the model needs to preprocess the image to increase the brightness and saturation of bright areas of objects. This problem is solved using various contrasting methods [6].

Another way to increase the contrast of images is K-means clusterization [7], [8], which is an unsupervised machine learning algorithm. The purpose is to divide N observations into K clusters, in which each observation belongs the cluster with the nearest mean. A cluster refers to a collection of data points that are aggregated together due to certain similarities. For image segmentation, areas of different colors are considered as clusters.

The authors of this report also came across various methods of computer vision for processing the video stream of different technical systems in their work. Thus, in [9], the Mask-CNN method was used to detect cells in micrographs. The paper [10] considered a problem similar to that considered in the article [11] - tracking the puck in the game. Work [12] is devoted to a comparative analysis of the effectiveness of hardware platforms for video detection tasks using solutions based on the YOLO architecture. In [13], the application of instant segmentation methods for creating training samples for training neural networks based on the YOLOv3 architecture was investigated.

3 Comparative evaluation of methods and architectures

To determine the correspondence of the above-considered deep learning methods in terms of their applicability to R&D tasks, comparative testing of architectures was carried out on a test sample MS COCO (Microsoft Common Objects in Context).

Table 1 describes the results of comparative testing of models for the sample MS COCO.

Table 1. Comparison of model results on COCO-test.

Task	Method	Backbone	FPS	AP %
Detection	YOLOv2	Darknet-19	171	21.6
Detection	YOLOv3	Darknet-53	35	33.0
Detection	YOLOv4*	Darknet-53	31	45.5
Instance segmentation	Mask R-CNN	Resnet-101	8.6	46.1
Instance segmentation	YOLACT++	Resnet-101	33.5	29.8

According to the test results, YOLOv3, YOLOv4, and Mask R-CNN showed the best results.

4 Conducting test experiments on the analysis of a video stream in real-time using selected methods

Currently, the models are being analyzed on the Intel Neural Computer Stick 2 neuro module, which will be included in the standard kit for robotic retail outlets. This neuro module is a compact computational module designed to perform artificial intelligence and machine learning tasks. The device is based on the Intel Movidius Myriad X Vision Processing Unit (VPU) - a specialized SoC containing 16 general-purpose computing cores and hardware components to accelerate the inference of neural networks and computer vision.

Main characteristics of the device:

- Supported frameworks: TensorFlow, Caffe;
- Connection: USB 3.0 Type-A;
- Dimensions: 72.5 x 27 x 14 mm;
- Compatible OS: Ubuntu 16.04.3 LTS (64 bit), CentOS 7.4 (64 bit), Windows 10 (64 bit).

All models that showed the best results were sequentially compiled on a Neural Computer Stick 2, followed by testing experiments.

The next stage in selecting architectures for the computer vision of the cloud software platform for robotic retail trade was testing the platforms in conditions close to the operating mode in real conditions, both in terms of the hardware platform and the physical parameters of the working area. To formulate the final conclusions, it was necessary to conduct test experiments with its dataset, taking into account the features and limitations of the real object.

Limitations are imposed by taking into account several factors. First, the described R&D develops the scientific and technical groundwork previously formed by the executing company to create "smart" trade showcases. Although smart showcases are at the stage of implementation, from economic efficiency, it was decided to use in the R&D process the same components (cameras and other hardware) that are already used in showcases.

1. Camera parameters:
   - Fixed image size: 1920x1080 pixels;
   - Lens parameters.
2. Environmental conditions:
   - Distance between lens and shelf: 310 mm;
   - Shelf dimension: 350x350 mm;
   - Shelf color: white uniform solid color;
   - Type of lighting: internal, along the top of the shelf;
   - Warmth of lighting: cool white;
   - Availability of lighting filters: matte.
   - Input image size: 720x480.
   - Hardware resources: Neural Computer Stick 2.

For the experiments, seven target classes and one additional technical class were selected, conditional "garbage" – goods with damaged packaging and other objects. In addition, representatives of the main product categories were chosen as target classes: bread, cheese, mayonnaise, chocolate, carbonated drinks, vegetables, cereals. The initial set of images of class data, not subjected to preprocessing, is after this referred to as "zero datasets".

Due to the limitations imposed by the selected hardware, the work within the experiments was focused on working with a dataset and image preprocessing to obtain the best results.

For an objective assessment of the results of experiments from the generally accepted list of metrics for assessing the "success" of the machine learning model, we selected four parameters: Precision, Recall, IoU, F1-score [16]. In total, 18 experiments were carried out, the parameters of which are described in **Ошибка! Источник ссылки не найден.**, and the results are shown in **Ошибка! Источник ссылки не найден.**.

5 Experiment plan for choosing the optimal architecture

The plan of experiments for selecting the optimal architecture is included in the comparative assessment of combinations of the studied architectures and sets of training and test samples shown in Table 2. The experiment index is formed from the code of the tested architecture, where Y3, Y4, M - YOLOv3, YOLOv4, and Mask R-CNN, respectively, and the numbers after the "-" sign are the number of the experiment on the given model.

Table 2. Experiment plan.

№№	Experiment ID	Experiment parameters	
		Dataset	Image preprocessing
1	Y3-0	zero model, input image / video	does not apply
2	Y4-0	zero model, input image / video	does not apply
3	M-0	zero model, input image / video	does not apply
4	Y3-1	normalized, 1000 labeled samples per class	does not apply
5	Y4-1	normalized, 1000 labeled samples per class	does not apply
6	M-1	normalized, 1000 labeled samples per class	does not apply
7	Y3-2	normalized, each class has 1500 marked samples, complex intersections added	does not apply
8	Y4-2	normalized, each class has 1500 marked samples, complex intersections added	does not apply
9	M-2	normalized, each class has 1500 marked samples, complex intersections added	does not apply
10	Y3-3	normalized, each class has 1500 marked samples, complex intersections added. Added augmentation filters: black and white, contrast + saturation, blur	does not apply
11	Y4-3	normalized, each class has 1500 marked samples, complex intersections added. Added augmentation filters: black and white, contrast + saturation, blur	does not apply
12	M-3	normalized, each class has 1500 marked samples, complex intersections added. Added augmentation filters: black and white, contrast + saturation, blur	does not apply
13	Y3-4	normalized, each class has 1500 marked samples, complex intersections added. Added augmentation filters: black and white, contrast + saturation, blur	k-means, contrast
14	Y4-4	Zero model	k-means, contrast
15	M-4	Zero model	k-means, contrast
16	Y3-5	Zero model	k-means, contrast
17	Y4-5	Zero model	k-means, contrast
18	M-5	Zero model	k-means, contrast

6 Experimental results

Considering Table 3, one can see that, taking into account the limitations imposed by the hardware, according to the comparison results, experiment Y3-2 demonstrated the best performance based on a normalized dataset with added complex intersections of objects, without image preprocessing.

Table 3. Comparison Matrix of Experiment Results.

Rating	Experiment ID	Precision	Recall	IoU	F1-score	TP	FP	FN
	Y3-0	0.98	0.99	78.15	0.98	5285	131	62
3	Y4-0	0.97	0.99	83.23	0.98	5295	151	52
	M-0	0.85	0.76	75.3	0.80	5360	1000	2012
	Y3-1	0.99	0.98	84.20	0.98	8946	111	189
	Y4-1	0.94	0.99	78.34	0.96	8999	591	136
	M-1	0.93	0.96	85.13	0.94	8953	653	421
1	Y3-2	0.99	0.98	82.41	0.98	10083	138	238
	Y4-2	0.93	0.98	76.74	0.95	10120	784	201
2	M-2	0.98	0.97	90.5	0.97	10213	212	314
	Y3-3	0.97	0.96	81.48	0.96	28024	907	1239
	Y4-3	0.95	0.95	81.23	0.94	28073	1189	1256
	M-3	0.97	0.95	90.34	0.96	28163	782	1405
	Y3-4	0.97	0.90	73.28	0.93	229	8	26
	Y4-4	0.85	0.90	69.93	0.87	229	40	26
	M-4	0.97	0.90	65.74	0.93	229	6	26

7 Discussion and conclusions

Considering the rating of the results of experiments for the selected indicators, the YOLOv3 architecture was chosen as the basis for the computer vision of the cloud trading platform, which is trained based on a normalized dataset, supplemented with examples with complex intersections of samples without preprocessing examples.

However, in the system's functioning, the detection and classification of the object and the tracking of its movements play an important role. An automated sales outlet client can move with the goods in his hands through the outlet, move the goods on the shelf, or move it to another shelf. Tracking of these actions should take place to minimize the likelihood of erroneous write-offs. At the same time, it is important to reduce the computational load on the system.

Therefore, the direction of further work on the project is the use of object tracing methods for post-processing of the video stream.

References

1. J. Redmon and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
2. J. Redmon and A. Farhadi, YOLOv3: An Incremental Improvement arXiv:1804.02767 [cs.CV] (2018)
3. A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection arXiv:2004.10934 [cs.CV] (2020)
4. K. He, G. Gkioxari, P. Dollár and R. Girshick, Mask R-CNN arXiv: 1703.06870 [cs.CV] (2017)

5. D. Bolya, C. Zhou, F. Xiao and Y. J. Lee, YOLACT++ Better Real-Time Instance Segmentation *IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 44, Issue: 2, 01 February 2022)* (2019)
6. J. C. Russ and J. C. Russ, *The Image Processing Handbook* (CRC Press, 2002)
7. X. Zheng, Q. Lei, R. Yao, Y. Gong, Q. Yin, EURASIP J. Image Video Process **2018**, 68 (2018)
8. M. G. Dorrer and A. E. Alekhina, J. Phys. Conf. Ser. **1889**, 22103 (2021)
9. A. Yakimov, A. Morgun, A. Salmina, M. Dorrer, A. Tolmacheva, D. Ogurtsov, J. Phys. Conf. Ser. **1399** 033089 (2019)
10. A. E. Tolmacheva, D. A. Ogurcov, and M. G. Dorrer, J. Phys. Conf. Ser. **1679** 032089 (2019)
11. R. Deepa, E. Tamilselvan, E. S. Abrar, and S. Sampath, in *2019 Int. Conf. Adv. Comput. Commun. Eng.* (IEEE, 2019), 1-4
12. A. E. Tolmacheva, D. A. Ogurtsov, and M. G. Dorrer, IOP Conf. Ser. Mater. Sci. Eng. **1679** 032089 (2020)
13. M. G. Dorrer and A. E. Tolmacheva, J. Phys. Conf. Ser. **1679**, 032089 (2020)