Comparison of the YOLOv3 and Mask R-CNN architectures’ efficiency in the smart refrigerator’s computer vision

M G Dorrer¹ and A E Tolmacheva²

¹Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarsky Rabochy Ave., Krasnoyarsk, 660037, Russian Federation
²Solution Factory, 10c3, Krasnoy Armii street, Krasnoyarsk, 660001, Russian Federation

E-mail: mdorrer@mail.ru

Abstract. The article deals with the computer vision system of the smart refrigerator "Robimarket". The equipment of the working area of the refrigerator, the selection of a set of chambers, the collection of a training sample for the computer vision system are described. The choice of the artificial intelligence architecture of the computer vision system was made by comparative testing of the YOLOv3 and Mask R-CNN architectures. The comparison was made on one hardware platform, one training set and a set of test cases. As a result, a comparison table was created for the speed and quality values of each model. As a result, the Mask-RCNN architecture was chosen, which showed a significantly higher detection accuracy in the video stream with acceptable performance for this task.

1. Introduction

Nowadays, it is necessary to scarify something in the object classification task in real-time. 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.

The authors [1] solve the image recognition problem in various types of markup noise conditions: false negatives, false positives, and inaccurate borders. In [2] Berman et al. the neural architecture search (NAS) and MRF methods to select the optimal model architecture suitable for the current tasks were used. The article [3] describes the solution of the problem of estimating the trajectory of a tennis ball based on video stream capture. In the article [4], we considered a problem similar to the one discussed in [3] – tracking the puck in the game.

Thus, both the review of the performed works and the author’s practical experience confirm the relevance of the task of comparing different ANN architecture. The comparison was made on the "smart refrigerator" artificial vision system’s data implemented in the framework of the "Robimarket" project [5].

Robymarket, based in Krasnoyarsk, has moved away from developing smart refrigerators as an end product. Its goal is to develop software and hardware systems that are cheap for the client, allowing to fully automate the trading system, any point of sale: be it a refrigerator, a rack, or a retail space.

Figure 1 shows a refrigerator with a computer vision module based on the YOLOv3 architecture, developed by the Robymarket company, located in the Grabli restaurant in Moscow.
The advantages of Robymarket products lie in several parameters:

- round-the-clock work of the outlet;
- automatic accounting and analysis of trade;
- remote business management;
- recognition of buyers' faces;
- automatic loyalty programs.

The combination of the applied technologies of computer vision, deep neural networks and radio frequency tags allows you to minimize all the risks of a standard vending machine.

The initial idea of developing a computer vision module in this company assumed the use of OpenCV mathematical algorithms to detect goods with QR codes. But in practice, low accuracy in identifying goods was found, since users did not particularly care about the quality of applying QR stickers. Mathematical algorithms did not allow high accuracy to determine the type of goods by a damaged QR sticker, which hindered the sales flow. To solve this problem, it was decided to use artificial neural networks as a detector.

In the process of developing a computer vision system for a smart refrigerator, a large amount of work was carried out to adapt the YOLOv3 technology - optimization of training samples, improving conditions in the working area of the refrigerator (lighting, product placement), fine-tuning the architecture. All these actions, unfortunately, did not help to get rid of a system error that appears when analyzing a shelf densely filled with monotonous goods. At the edge of the shelf, there was a "blinking" error because the system could not account for the physical "blind spots". The error occurred at the moment when the detection percentage was close to the visible threshold. To fix this error, it was decided to experiment with the Mask R-CNN architecture, such as semantic segmentation. Since, unlike...
YOLOv3, this architecture looks through every pixel, it was initially expected that it would lose out to the first in speed. However, due to the same factors, it was expected that it would show higher quality indicators in the recognition process. To switch to the Musk R-CNN architecture, one training and test samples were prepared, augmentation and markup operations were performed for the standards of each architecture. For the experiment, the transfer learning method was not used, since it was necessary to first take into account the qualitative characteristics of the two architectures with the default settings.

2. Methods

2.1. How a smart refrigerator works

A smart refrigerator is a vending machine that combines the advantages of vending machines and shops without a cashier. It integrates with the application for mobile phones, allows you to form personal orders. Suitable for the sale of drinks, ready-made meals, bulk products and goods in fragile containers.

The main features of this type of commercial equipment:

- Access to the refrigerator by application / RFID card;
- CCTV camera with a neural network;
- Collecting analytics about consumers;
- Protection against theft and access without authorization;
- Personalized menu, loyalty system.

Figure 1 shows the functional requirements model for a Robimarket smart refrigerator.

![Figure 1](image1.png)

Figure 2. Functions of a smart refrigerator (UML Use-Case).

The user can access the products in four ways:

- Through the Robymarket app:
  - opening the nearest kiosk based on geolocation received from GPS;
  - opening by face identification;
  - opening by QR-code, created during customer registration.
• An alternative option is to pay by credit card or a device that supports NFC using a card reader.

The “access to goods” function is carried out using a software module based on deep neural network technologies.

2.2. Applied hardware solutions in the development of a smart refrigerator
At the request of the customer, a farm was assembled from six Nvidia 1060 cards and 64 GB of RAM. This made it possible to solve the problems of the training queue of several models of independent clients or to increase the model training speed globally.

Ubuntu 16.04 64-bit is used as the standard operating system since it is more flexible in setting up the server side and neural network modules.

To work in different devices, it was decided to supply two types of computational modules: an ARM module and a video card similar to a card intended for the training stage.

Among the modules provided, a series of experiments on speed characteristics were carried out.

Based on the results of testing all hardware platforms, it was found that among the CPUs in terms of image processing speed, HiKey970 is the best. Taking into account the use of CUDA scalar cores, the 1070ti video card showed the highest performance. But since the task was to choose the cheapest and most satisfying solution for future tasks, Jetson Nano was chosen for use in GPU mode.

According to the results of a comparative analysis of platforms in CPU mode, it was revealed that the Jetson Nano is 2 times more efficient (faster), the neural network is processed, compared to the HiKey 970.

Neural Compute Stick in recognition mode is 10 times faster than Jetson Nano. It should be noted that the Neural Compute Stick does not allow training models, unlike all other solutions presented in this article.

Based on the results of comparative tests [6] for the operation of the refrigerator with a neural network module, it was decided to choose the Neural Compute Stick as the hardware platform.

2.3. Choosing to light for the refrigerator shelf
During the experimental work, several shortcomings of the initial test bench of the automated refrigerator were revealed, namely the initial area of the lighting installation, the technical characteristics of the lighting, the insufficient quality parameters of the cameras and their location. There were problems with detecting and counting items along the edges of the shelves.

In the first version of the test refrigerator, the lighting in the form of an LED strip wrapped around the chamber was in the center without a matte filter. This lighting option affected the detection quality of tall objects under the camera and in the first row around it. A glare formed on the area identifying the product (lid, box surface), hiding the characteristic features of the neural network.

The decision was made to remove the lighting from the center and install white LED strips along the top of the shelf, adding a matte filter to diffuse the lighting without distorting the color parameters of the images received from the cameras.

In the original assembly, four 2-megapixel cameras were installed on the shelf.

After preparing test models to assess the behavior of the architecture of convolutional ANNs, it was revealed that it was necessary to obtain a better image. The use of four cameras led to difficulties in processing the overall frame of the shelf.

The decision was made to replace the four 2-megapixel cameras with one 5-megapixel wide-angle camera. The distortion effect was removed by the algorithms of the OpenCV library. These algorithms cut off distorted edges and display an aligned image.

After replacing the cameras, the image quality of objects improved: noises and pixelity disappeared. Small objects are seen much more clearly.
2.4. Selection of annotations for the client's assortment
During user behavior testing, it was found that small catering establishments were interested in this project. But at the same time, they offered different dishes in one package as an assortment, for example, a transparent container and the appearance of a dish. In this case, the weekly assortment could change several times.

In order not to teach each time about the appearance of the product (cheesecakes, pancakes), it was decided to use unique packaging areas - “annotations” of the product. The cheapest and least labor-intensive way is to create a set of stickers according to the number of types of food that will be in the refrigerator. This allows the client not to build a complete list of the refrigerator assortment and calmly change the appearance of the product and its packaging at his discretion, adding a small unique label to the surface.

However, there is another possibility of annotating a product assortment when a customer is selling a strongly typed product. For example, soda, chocolate, set lunch containers, etc.

In such a situation, the restaurant can easily change the contents of the container and inform the customer about it in the menu tab of the smart refrigerator or by directly displaying the product on the refrigerator shelf.

The preparation of the annotation is under the needs of the client - the trading network that will operate the vending machine. If a client changes the sales model from a strict appearance to unique tags or combines these two sub-methods, then the model has to be retrained for a new product class or added to an old one, but at the same time take into account that another product with the same unique tag cannot be found with a new one. goods.

2.5. Creating training samples
At the first stage, the original graphic material was augmented. To obtain an augmented sample, the albumentations library is used.

For the neural network to be able to separate an object from a background similar in texture, the training sample was supplemented with image fragments of an empty shelf and artificially generated object textures. The image data is blank and is a local negative sample.

If, when preparing a set of images for training the model, the client refused to provide a product mockup, then an artificial mockup image was used for the minimum viable model. Most often, this method was used when training a model for unique labels.

If the client could provide models of the goods being sold, then the collection of materials for the training sample was collected from the chambers of the refrigerator test stand. Since the cameras on different shelves worked somewhat differently, the initial sample turned out to be varied. This sample was subsequently supplemented with materials obtained from the chambers of the working version of the refrigerator assembled for the client.

Further, to improve the quality of the model, data on any errors were collected during a week of operation of the refrigerator in the mode of selling goods. During this time, the model was retrained every two days and updated until the client was satisfied with the minimum system error.

Automatic marking of the training sample of object samples was performed using the Canny operator.

2.6. YOLO architecture
YOLO or You Only Look Once [7] is a currently very popular CNN architecture, which is used to recognize multiple objects in an image. Figure 3 shows the YOLOv3 architecture.
To improve the accuracy of recognizing product classes in a smart refrigerator, the YOLOv3 architecture was supplemented with a transfer learning method.

2.7. Description of the architecture of the MaskRCNN convolutional neural network

The concepts underlying in Mask R-CNN [8] were gradually developed through the architecture of several intermediate neural networks that solved various tasks from the above list. Let's look at these stages sequentially.

Figure 4 shows the Mask R-CNN Architecture.

3. Results

Figure 5 shows an Ubuntu terminal with Nvidia 1070 graphics card load parameters during the YOLOv3 architecture training stage.
Figure 5. Technical video card’s indications used for learning YOLOv3.

Figure 6 shows the Ubuntu terminal with the load parameters of the Nvidia 1060 video card during the MaskRCNN architecture training stage.

Figure 6. Technical video card’s indications used for MaskRCNN training.

Loads of the video card during architecture training show that the MaskRCNN architecture takes up 18% fewer resources than the YOLOv3 architecture.

The sample used for the experiments included 800 images taken from the smart refrigerators’ cameras at the Tortberry Bakery. The images were checked for originality and augmentation.

In testing, 70 images with 430 samples of all product classes found on them were used.

In both models, the output is represented by six classes under the names: red, orange, blue, green, violet, yellow.

Testing in terms of the speed of solving the problem was carried out on an Nvidia 1060 video card on the Ubuntu 16.04 operating system. Figures 7 - 9 show YOLOv3 architecture’s results.
Figure 7. The result of the analysis of test image 1.

Figure 8. The result of the analysis of test image 2.
Figure 9. The result of the analysis of test image 3.

Table 1 shows the network results for each class sample obtained from test images processed by the YOLOv3 architecture model.

| Figure number | Class name | Accuracy % | Video processing |
|---------------|------------|------------|------------------|
| Figure 5      | Red        | 98         |                  |
|                | Blue       | 97         |                  |
|                | Yellow     | 87         |                  |
|                | Orange     | 89         |                  |
|                | Green      | 97         |                  |
|                | Violet     | 80         |                  |
| Figure 6      | Red        | 78         |                  |
|                | Blue       | 62         |                  |
|                | Yellow     | 62         | 30FPS            |
|                | Orange     | error      |                  |
|                | Green      | 45         |                  |
|                | Violet     | 67         |                  |
| Figure 7      | Red        | 97         |                  |
|                | Blue       | 95         |                  |
|                | Yellow     | 42         | 30FPS            |
|                | Orange     | error      |                  |
|                | Green      | 92         |                  |
|                | Violet     | 74, error  |                  |

Figures 10-12 show the results of the MaskRCNN architecture.
Figure 10. The result of the analysis of test image 1

Figure 11. The result of the analysis of test image 2
Table 2 shows the network results for each sample class obtained from test images processed by the Mask R-CNN architecture model.

**Table 2. Results of the Mask R-CNN architecture model’s analysis.**

| Figure number | Class name | Accuracy % | Video speed processing |
|---------------|------------|------------|------------------------|
| Figure 8      | Red        | 99         | 10FPS                  |
|               | Blue       | 99         |                        |
|               | Green      | 100        |                        |
|               | Violet     | 98         |                        |
| Figure 9      | Red        | 99         | 10FPS                  |
|               | Blue       | 100        |                        |
|               | Yellow     | 99         | 10FPS                  |
|               | Orange     | 95         |                        |
|               | Green      | 100        |                        |
|               | Violet     | 100        |                        |
| Figure 10     | Red        | 100        | 10FPS                  |
|               | Blue       | 100        |                        |
|               | Yellow     | 98         | 10FPS                  |
|               | Orange     | 98         |                        |
|               | Green      | 100        |                        |
|               | Violet     | 99         |                        |

4. **Discussion**

A comparative analysis of tables 1 and 2 shows that Mask R-CNN architecture has significantly higher accuracy in determining classes in the video stream. It can be seen that on the same classes and on the same images, the accuracy of assigning objects to classes for Mask R-CNN does not drop below 95% anywhere, but mostly amounts to 98-100%. At the same time, for the Yolo architecture, it failed to recognize one of the Classes at all (see the data in table 1), and for several others, the accuracy of solving the problem was 42-45%, which is unacceptable in practice.
At the same time, in all experiments, the performance on the same hardware platform for the YOLO architecture turned out to be in three times higher than for Mask R-CNN.

Also, despite the low image processing speed, the Mask R-CNN architecture on test samples showed high detection accuracy for each class sample. This fact confirms, in addition to the high accuracy of the Mask R-CNN architecture, the sufficiency of the applied training sample size for this model.

5. Conclusion
Thus, the Mask R-CNN architecture using was considered promising for the use of Robimarket smart refrigerators in the computer vision system [10]. The key factor here was the high accuracy of product identification, which is directly related to the commercial risks of operating a vending machine. At the same time, the speed of 10 FPS, which is not enough for gaming applications and real-time motion control, turned out to be quite sufficient for timely invoicing to the buyer.

The result of this work shows the high efficiency of using convolutional deep neural networks (especially the YOLOv3 and Mask R-CNN architecture models) when solving problems of controlling vending machines in real-time.

As a result of the research, the following research and applied results were obtained:

- Based on the original test bench, the design flaws of the smart refrigerator prototype were found and eliminated;
- An experimental computer vision module was implemented and successfully tested in a Moscow restaurant for a month;
- A manual was created for preparing a training sample for non-specialists in the field of machine learning;
- Based on the neuro module Neural Stick Computer 2, a working prototype of a shopping kiosk was created, which has 18 classes in the knowledge base and a training sample for 200 thousand unique images;
- Modules were developed to accelerate data collection at the training stage, as well as the augmentation module and auto-generation of additional samples in the training set.

References
[1] Gao J, Wang J, Dai S, Li L-J and Nevatia R 2019 NOTE-RCNN: NOise Tolerant Ensemble RCNN for Semi-Supervised Object Detection in 2019 IEEE/CVF International Conference on Computer Vision (ICCV) 9507-16 doi: 10.1109/ICCV.2019.00960
[2] Berman M, Pishchulin L, Xu N, Blaschko M B and Medioni G 2020 AOWS: Adaptive and optimal network width search with latency constraints [Online] Available: http://arxiv.org/abs/2005.10481
[3] Deepa R, Tamilselvan E, Abrar E S and Sampath S 2019 Comparison of Yolo, SSD, Faster RCNN for Real Time Tennis Ball Tracking for Action Decision Networks in 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE) 1–4 doi: 10.1109/ICACCE46606.2019.9079965
[4] Tolmacheva A E, Ogurcov D A and Dorrer M G 2019 Puck tracking system for aerohockey game with YOLO2 in Journal of Physics: Conference Series 1399(3) doi: 10.1088/1742-6596/1399/3/033116
[5] Robimarket https://startupvillage.ru/exhibitors/263
[6] Tolmacheva A E, Ogurtsov D A and Dorrer M G 2020 Justification for choosing a single-board hardware computing platform for a neural network performing image processing in IOP Conference Series: Materials Science and Engineering 734(1) 10.1088/1757-899X/734/1/012130
[7] Redmon J, Divvala S, Girshick R and Farhadi A 2015 You Only Look Once: Unified, Real-Time Object Detection [Online] Available: http://arxiv.org/abs/1506.02640
[8] He K, Gkioxari G, Dollár P and Girshick R 2017 Mask R-CNN [Online] Available: http://arxiv.org/abs/1703.06870