Comparative Performance Analysis of Neural Network Real-Time Object Detections in Different Implementations

Alexey V. Stadnik[1],⋆, *Pavel S.* Sazhin[1],⋆⋆, and *Slavomir* Hnatic[2,3],⋆⋆⋆

[1] *OOO «Videointellect», Skolkovo Innovation Centre,*
42 Bolshoy boulevard, 143026 Moscow, Russian Federation
[2] *The Laboratory of Information Technologies, JINR,*
6 Joliot-Curie, 141980 Dubna, Moscow Region, Russian Federation
[3] *Institute of Experimental Physics, Slovak Academy of Sciences*
Watsonova 47, 04001 Košice, Slovak Republic

Abstract. The performance of neural networks is one of the most important topics in the field of computer vision. In this work, we analyze the speed of object detection using the well-known YOLOv3 neural network architecture in different frameworks under different hardware requirements. We obtain results, which allow us to formulate preliminary qualitative conclusions about the feasibility of various hardware scenarios to solve tasks in real-time environments.

1 Introduction

Nowadays, computer vision technologies are utilized in various branches of industry such as analysis of medical images, facial recognition in biometrics, text recognition in OCR, processing camera inputs in robotics, content filtering, situational analytics in security systems.

When applying computer vision to solve problems in real-time industrial environments it is crucial for the frame analysis to be as fast as possible. This is necessary to ensure the processing of maximum possible number of video streams coming from camera systems, which in turn saves computational resources and makes solutions more cost effective.

A significant amount of practical problems to be solved by the computer vision are governed by human action, therefore the related algorithms and methodologies are required to possess at least some degree of cognition.

This is because the availability of the information about the type and character of the objects of analyzed scene makes the decision process of the detection system considerably simpler. Moreover, such information considerably enlarges the number of possible applications of computer vision systems in general.

At the present time, most developments were done in the field of deep neural networks and those undoubtedly brought the computer vision industry into the age of cognition [1].

⋆e-mail: alexey.stadnik@intellect.video
⋆⋆e-mail: pavel.sazhin@intellect.video
⋆⋆⋆e-mail: drhnatic@drhnatic.com

2 Deep neural networks and their architectures

In practice, neural networks are used to solve the initial most general task of detecting and classifying the objects in the scene. The output of this process is so-called metadata, utilized at later stages to detect real scenarios: for example crowding, people counting, wrong parking up to more difficult, cascading algorithms. It is common to use deep neural networks in the first stage of frame analysis and as a metadata source for the analysis of the observed scene for the most effective application of computer vision in real life environments.

The question is, which neural network architecture is the most appropriate choice. There are several different architectures where the principles of deep learning are implemented. Currently, the fastest neural network architecture, applicable to real life conditions, is YOLOv3 [2]. It works by scaling the incoming frame to pixel size[1] 416 × 416, which is consequently divided into a net of 13 × 13 cells, where each cell has pixel size of 32 × 32. In this way, a coarse breakdown of the image is obtained, which is then used by a neural network [3]. The objective of neural network processing is to predict the probabilities of the appearance, type, location and size of the object in the frame. These probabilities are then averaged over cells to give the final result of neural network processing. The example of such processing is shown in the image below:

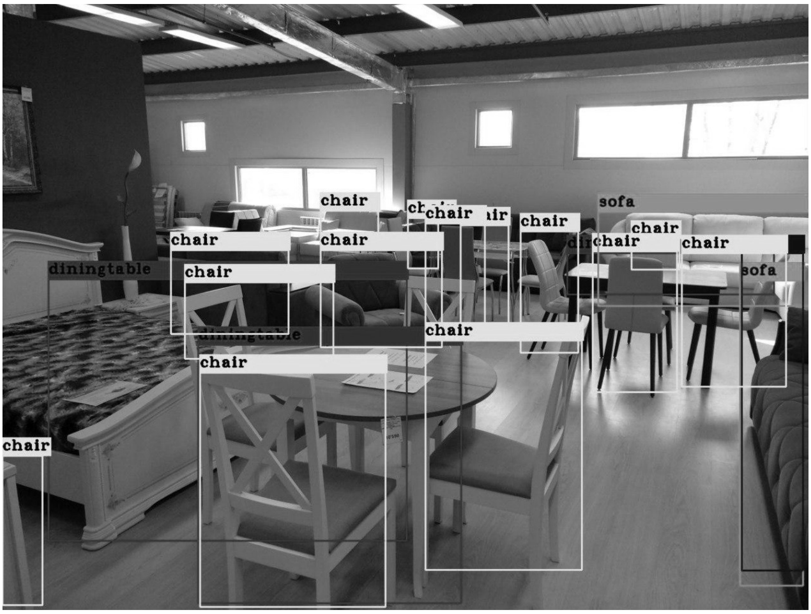

Figure 1. An instance of YOLOv3 detection on sample images.

Although YOLOv3 is fast, it is less precise in comparison with other neural network architectures. However, in practical cases of video detection, speed is more important than precision. In reality, the lack of precision is compensated by the nature of input data, as they are highly correlated. The typical video surveillance camera stream has FullHD resolution (1920 × 1280) at 25 frames per second (fps). Since the difference between adjacent frames is not that large, the failure to detect a certain object in a few frames is not critical. For example, in the time interval of one second, it might be possible to successfully detect and classify this object on average in 20 frames. Therefore, due to such balance

[1] This is the present status of the code, the original YOLOv3 publication [3] mentions 320 × 320 resolution.

of speed and precision, we consider YOLOv3 to be the most suitable neural network architecture for application in real life environments.

3 Performance analysis in different implementations

In the practical use of computer vision algorithms it is important to consider the combination of deep neural network architecture implementation and the hardware used in processing. There is a huge difference in the frame analysis time between CPU and GPU. Although slower, the implementations on CPU have their own additional benefits. For example, cloud videoanalytical services can be built using widespread CPU-architectures, instead of specially designed and currently expensive GPU-clusters.

When choosing the implementation, it is also necessary to take into account additional factors, such as the popularity of the framework, ease of its installation and others. This way one can obtain lightweightness of the solution. By utilizing transfer learning [4], i.e. training the already pre-trained model and using it in the framework one is left only with forward propagation in neural network processing. Then it is possible to make the code simpler, performance faster and speed up the procedures of installation and assembly.

By choosing the right framework, one can minimize the number of external dependencies, which in turn gives more freedom for spreading the applications utilizing deep neural networks. Also, using the appropriate framework significantly simplifies the procedure of creating hybrid algorithms, which combine the speed of commonly used computer vision algorithms with the remarkable generalization ability of deep neural networks.

From our experience, we consider the following two widely known frameworks for implementation of deep neural networks algorithms in real-time environments to be the best choice: Darknet [5] and OpenCV [6]. Darknet is an open source neural network framework written in C and CUDA. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. It is worth noting that OpenCV is strictly not a fully-fledged framework for deep neural networks per se. At present time, it has only the ability to output the result of deep neural network processing, however, in CPU implementation[2], this feature is sufficient for practical applications.

In our analysis we used the COCO dataset [7] to train YOLOv3. COCO is a large-scale object detection, segmentation, and captioning dataset. Neural network was trained for detection of 80 types of objects and the analysis was then performed on 200 images. The example of such detection is shown in Fig. 1. The results of the detection are hardware independent.

The results of average processing speed in various implementations and hardware scenarios are shown in Table 1.

As expected, the GPU implementation (Darknet) is significantly faster than the fastest CPU implementation – up to one order of magnitude, depending on hardware. However in purely CPU implementations OpenCV outperforms Darknet by a factor of 5.

[2] Running deep neural networks under GPU was not supported in OpenCV at the time of writing of this article.

Table 1. Computing time of YOLOv3 in various hardware scenarios in Darknet and OpenCV

Framework	CPU/GPU	HW	Computing time
Darknet	GPU	Tesla V100	0.06s
Darknet	GPU	Quadro M1000M	0.14s
Darknet	GPU	GeForce GTX 850M	0.20s
Darknet	GPU	Quadro K6000	0.26s
Darknet	CPU	core i7	2.53s
OpenCV	CPU	core i7	0.51s

4 Conclusion

We analyzed the performance of the YOLOv3 neural network architecture in OpenCV and Darknet implementations. YOLOv3 is the fastest deep neural network architecture for real-time camera detections in videoanalytics, Darknet and OpenCV are its fastest implementations.

Based on the obtained results, we found that the most sensible choice when dealing with the real-time detections is to use the Darknet Darknet framework for the computations on GPU, while in the case of CPU computations the better option would be to choose the OpenCV framework.

Acknowledgement

The work was supported by grant No. 146/33 (2019) of Plenipotentiary of Slovak Republic in Joint Institute for Nuclear research in Dubna.

References

[1] J. Schmidhuber, Neural Networks **61**, 85–117 (2015)
[2] https://pjreddie.com/darknet/yolo/
[3] J. Redmon, A. Farhadi, https://arxiv.org/abs/1804.02767 (2018)
[4] S. J. Pan, Q. Yang: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22** (10), 1345–1359 (2010)
[5] J. Redmon, *Darknet: Open Source Neural Networks in C*, https://pjreddie.com/darknet/, 2013–2016
[6] http://opencv.org, release 3.4.0.
[7] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, P. Dollar, https://arxiv.org/abs/1405.0312 (2014)