An Introduction to Image Classification and Object Detection using YOLO Detector

Martin Štancel[1(✉)] [0000-0001-6669-1439] and Michal Hulič[1] [0000-0002-2974-8050]

[1] Technical University of Košice, Košice, Slovakia
martin.stancel@tuke.sk , michal.hulic@tuke.sk

Abstract. Artificial neural networks have been proved to be the best and the most used solution for image classification and object detection tasks. Paper analyzes them as a tool that significantly improves the mentioned, very complicated computational calculations. In the paper there is a brief history of their development as well as the selected object detector that we used for our introductory experiment that is shown later in the paper. Also, there is introduced the idea of the future research that is going to be based on the conducted experiment and which is going to involve a new methodology for an automated generation of new domain-specific datasets that are essential in the training phase of the neural networks.

Keywords: Artificial neural network, Image classification, Object detection, Dataset, Pattern recognition, Computer vision, Machine learning.

1 Introduction

In the last two decades scientists and researchers in the fields of computer vision, machine learning and neural networks perceive an increasing popularity of these sectors of computer science due to the fact that technologically hardware as well as software components of today's computers have been significantly advanced. It has allowed us to do extensive algorithmic operations and work with a huge amount of data.

We analyzed artificial neural networks (in short neural networks), which is a subarea of the machine learning, that are the most suitable method for image classification and object detection tasks.

Neural networks use methodologies of the machine learning and computer vision. Computer vision takes care about image processing in a way so it also deals with noise reduction, brightness change, or image enhancement by various techniques. On the other hand, the machine learning is very flexible, because it can be used in computer vision, image processing as well as other sectors of computer science.

The paper also describes the history of the neural networks as well as the primarily used convolutional neural network which has become the most popular method at the image classification and object detection tasks.

According to the analyzed facts and the results from our empirically tested data, in the future we would like to design and implement optimized method for automated generation of domain-specific datasets that are essential in the training phase of the neural networks which is very necessary task to do for the neural networks to actually be able to learn and detect objects on the series of any new images.

2 Neural Networks

There are a lot of general methods that deal with a problem in a unique way in an optimal time consuming interval and nowadays neural networks have been one of them that become commercially popularized thanks to the fact that hardware as well as software are being significantly advanced on daily basis. Today, they have been widely used in many sectors of the computer science from arduino microcontroller interfaces [1] through authentication [2] or our researched image classification and object detection.

Neural networks consist of many interconnected groups of nodes that are called neurons. Variables from input functions from data are transmitted to these neurons as a multivariable linear combination, where the values are multiplied with each function variable (i.e. weights). On this linear combination there is later applied non-linearity that give the neural networks an ability to model complex non-linear relations. Neural networks can have more layers, where an output from one layer is the input for the other. Also, for the learning and detecting processes, neural networks use trained datasets (section 2.2).

Nowadays, there are a lot of algorithms with various types of neural networks. Their historical development is described in the next section 2.1.

2.1 History of the Neural Networks

For a few decades there have been simple approaches to create one of the firsts neural networks and its very first approach begun by Frank Rosenblatt in 1958 [3], who researched how information from physical world are stored in biological system so it could be used for detection or behavioral influences in the future.

Later, there were developed models with several successively non-linear layers of neurons that are dated back to the sixty's [4] and seventy's [5]. Gradient descent method was in the supervised learning [6] in discrete, differencional networks of an arbitrary depth called backpropagation [7] applied for the first time to a neural network in 1981.

With a huge amount of various layers, neural networks were too hard to develop at this time, because of that their development stagnated until the beginning of ninety's [8], when unsupervised learning [9] method was implemented.

In the ninety's and twenty's of the last century there were significant improvements in this kind of field. There was developed a new method of reinforced learning [10] that looks into an unknown environment and by using the trial and error method, agent learns about its surroundings and gets better every time it tries a new approach with its actions [11].

In the third millennium, the neural networks attracted a large amount of researchers for their application in many different sectors [12, 13] resulting among the best algorithms. Since 2009 the neural networks have won many competitions especially in a pattern recognition.

The pattern recognition was significantly improved when Alex Krizhevsky et al. in 2012 developed convolutional neural network for image classification task on ImageNet challenge [14]. He and his team won the challenge and created state-of-the-art image classification method that is also used today.

2.2 Datasets

Today, there are a lot of various datasets for the machine learning but we will take a closer look at image datasets that are essential for image classification and object detection tasks.

Creating image datasets is a relatively time-consuming operation, since their meaning is acquired when they contain a huge amount of data. The image datasets that are used in image classification and object detection are created by labeling objects and accurately locating them with a bounding box. Nowadays, there are no such tools that could perform fully automated objects labeling and locating.

We want to direct our research to domain-specific environments, so creating a method that automates the generation of these datasets is desired in the community.

We assume that it will be based on a convolutional neural network and an image object detector within YOLO architecture which we empirically tested on the series of our two experiments (section 3.2). Our idea is to collect images online that would consist of various types and colors of the same object classes, transparent or one-colored background and accurate name. Then, we could extract individual objects from the images and programmatically adjust their brightness, light settings, shadows, etc to get even more images for the training phase.

Our idea is to put those objects into randomly generated backgrounds with random location and overlapping as can be seen in the next figure (**Fig. 1**).

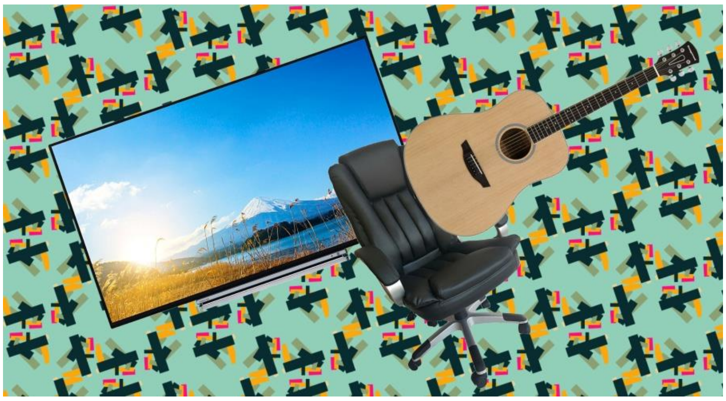

Fig. 1. Randomly generated background with random locating and overlapping objects.

3 Detector YOLO and the Experiments

YOLO is an object detector created by Redmon, J., et al. [16]. The YOLO authors state [17] that it is a state-of-the-art image object detector that achieves the best results in terms of accuracy and speed and that's why we used it in our research along with its neural network called Darknet.

3.1 Detector

YOLO divides each image into a grid of size *S x S* and each cell in the grid predicts *B* bounding boxes and their confidence. This confidence of an object reflects how reliable and accurate the bounding box that locates and classifies an object is. It defines the confidence of an object as follows:

$$PR(Object) * IOU_{pred}^{truth} \tag{1}$$

which means that the probability of the detected object is multiplied with an intersection over union (the intersection area divided by the union area for two bounding boxes) between the predicted boundary box and the ground truth box (i.e. hand labeled bounding box in a training data).

3.2 Experiments

With the detector YOLO we conducted two experiments on a pre-trained COCO dataset [18]. In the first one, we showed how the detector works on the image shown below (**Fig. 2**) and in the second one we tested the detector on the series of 500 images to empirically confirm its functionality.

Using the detector on the image in various resolutions. In this experiment we compared the image classification and object detection while processed on processor Intel Core i7-7700K (**Table 1**) and graphic card GeForce GTX 1070 (**Table 2**) while we used the same image for both of the components.

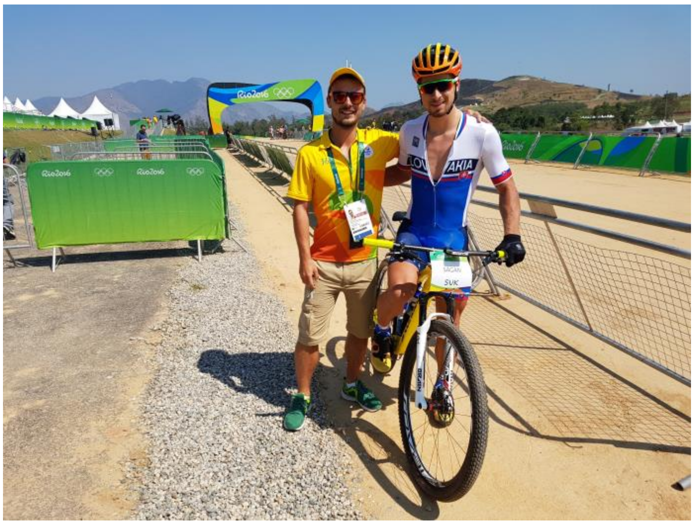

Fig. 2. Used Image for this experiment.

By comparing the two tables, we can see that the data processing, image classification and the object detection on the processor is noticeably slower than on the graphic card (approximately 8x slower). Also, with the increasing resolution, the number of detected objects is also increased, which is caused because of the better quality and clearer image.

Table 1. Objects Detection Testing on the Processor.

Resolution	Objects Detected	Time in ms
378x284	8	1639.114
756x567	9	1623.239
1008x756	11	1810.924
2016x1512	13	1679.288
4032x3024	14	1550.013

Table 2. Objects Detection Testing on the Graphic Card.

Resolution	Objects Detected	Time in ms
378x284	8	194.474
756x567	9	202.065
1008x756	11	202.817
2016x1512	13	198.224
4032x3024	14	194.799

Using the detector on the series of 500 images. We extended our first experiment to detect objects on the series of 500 images. Also, according to the results of our previous experiment, we didn't use various resolutions anymore, because it has no effect in time on the final detections and using the images in their original resolution provide more detected objects. For the comparison we chose the images with the fastest and the slowest detection time and the images with the most and the least objects detected. Also, we provided average time and average amount of detected objects per whole series of the images. Similarly we used processor and graphic card processing as in the first experiment. The results are shown in the next tables (**Table 3** and **Table 4**).

Table 3. 500 Images Objects Detection Testing on the Processor.

Property	Objects Detected	FLOPS	Time in ms
The fastest detection	20	65.864	2188.822
The slowest detection	20	65.864	1401.273
Average time	13.23	65.864	1563.503
The most objects	44	65.864	1505.41
The least objects	2	65.864	1572.51

Table 4. 500 Images Objects Detection Testing on the Graphic Card.

Property	Objects Detected	FLOPS	Time in ms
The fastest detection	20	65.864	220.742
The slowest detection	20	65.864	187.003
Average time	13.23	65.864	192.299
The most objects	44	65.864	189.685
The least objects	2	65.864	188.625

The speed of the detection on the series of 500 images is between 1401.273ms to 2188.822ms with the average time of the detection 1563.503ms on the processor and 187.003ms to 220.742ms with the average time of the detection 192.299ms on the graphic card.

From the results of this experiment we can conclude that the amount of objects detected doesn't affect the speed of detection (the fastest and the slowest processed images contain the same amount of objects) as well as the time of the most and the least objects detected images is almost identical.

4 Future Research

In the future, we would like to use the YOLO detector for processing a huge amount of images for a training phase of automated generation of domain-specific datasets. Based on our results, we will aim the processing on a graphic card. The card we used achieved 5 FPS.

The future research will also be aimed to design completely new methodology for the automated generation of domain-specific datasets. We assume that the method will be of a great importance in reducing time cost while creating new datasets, especially in the phase of the labeling where each object on an image must be precisely put into the bounding box. Nowadays this task is handmade by people and this approach should completely get rid of the human intervention during the labeling process. The method would also be applied in real-time detections as well as many other tasks like determining specific species of a certain kind or in education to learn specific objects in the same way as children learn from their very first moments of life.

The last thing I would like to point out is that creating such datasets is a serious problem since labeling and locating of the objects in the images is mostly a manual work. Our method would help researchers in many different areas to get significantly better results because, as is written in this papers [19, 20], often times their datasets are very limited and it could affect the results accuracy.

With our approach of automated generation of domain-specific datasets we could train the neural networks on specific environments which would significantly help with a determination not only of a class of some object but also its kinds and subclasses e.g. a detected flower would be more accurately detected as forget-me-not or a detected tree would be more accurately detected as baobab.

Acknowledgement

This work was supported by the Faculty of Electrical Engineering and Informatics, Technical University of Košice under the contract No. FEI-2018-59: Semantic Machine of Source-Oriented Transparent Intensional Logic.

References

1. Madoš, B., Ádám, N., Hurtuk, J., Čopjak, M.: Brain-computer interface and Arduino microcontroller family software interconnection solution. In: Proc. of the IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (2016), pp. 217–221, 2010.
2. Vokorokos, L., Danková, E., Ádám, N.: Task scheduling in distributed system for photorealistic rendering. In: Proc. of the IEEE 8th International Symposium on Applied Machine Intelligence and Informatics (2010), pp. 43–47, 2010.

3. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in Brain. In: Psychological Review, USA, 1958, vol. 65, iss. 6, pp. 386–407.
4. Ivakhnenko, G. A., Lapa, G. V.: Cybernetic predicting devices. USA. CCM Information Corp, 1965.
5. Werbos, P.: Beyond regression: new tools for prediction and analysis in the behavioral sciences. 1974.
6. Hardt, M., Price, E., Srebro, N.: Equality of Opportunity in Supervised Learning. In: Advances in Neural Information Processing Systems (2016), vol. 29, 2016.
7. Wang, L., Zengya, Y., Chen, T.: Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. In: Expert Systems with Applications. 2015, vol. 42, iss. 2, pp. 855–863.
8. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. In: IEEE Transactions on Neural Networks (1994), vol. 5, iss. 2, pp. 157–166, 1994.
9. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In: International Conference on Learning Representations (ICLR). 2016.
10. Marco, W., Van Otterlo, M.: Reinforcement Learning. 2012. ISBN 978-3-642-27645-3.
11. Chovanec, M., Chovancová, E., Dufala, M.: DIDS based on hybrid detection. In: IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA), Slovakia, pp. 79-83. 2014.
12. Vokorokos, L., Pekár, A., Ádám, N., Daranyi, P.: Yet Another Attempt in User Authentication. 2013, vol. 10, iss. 3, pp 37–50. Acta Polytechnica Hungarica (2013).
13. Hurtuk, J., Baláž, A., Ádám, N.: Security sandbox based on RBAC model. In: Proc. of the 11th International Symposium on Applied Computational Intelligence and Informatics (2016). pp. 75–80. 2016.
14. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems (2012), vol. 25, 2012.
15. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. In: Nature. 2015, pp. 436–444.
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2016). IEEE Xplore, pp. 779–788. 2016.
17. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. Tech Report. arXiv:1804.02767. 2018.
18. Tsung-Yi, L., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, L., Dollár, P.: Microsoft COCO: Common Objects in Context. In: European Conference on Computer Vision (ECCV), pp. 740-755. 2014.
19. Garcia, J., Barbedo, A.: Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. In: Computers and Electronics in Agriculture, vol. 153, pp. 46-53. Elsevier. 2018.
20. Vokorokos, L., Ennert, M., Čajkovský, M., Radušovský, J.: A Survey of parallel intrusion detection on graphical processors. In: Central European Journal of Computer Science, vol. 4, iss. 4, pp. 222–230. Open Computer Science. 2014.