Robust Real-Time Object Detection on Embedded Devices in @work Robot

Reza Ghasemi[1[0000-0003-0002-3846]] and Suhair Salehi[2[0000-0003-0040-3929]]

[1] Robotics Research Center, Chista Academy, Zand St., Shiraz, Iran
[2] Robotics Research Center, Chista Academy, Zand St., Shiraz, Iran
reza.ghasemi@chist.org
s.salehi@chist.org

Abstract. Deep Learning has long been used in machine vision to identify objects. Real-time object detection is used to cross obstacles, help navigate, move objects for the @Work robot. We use real-time RGB images with different lights, rotation in the body, different distances, etc. to create training datasets and train our network and compare our method with other methods of object detection. Real-time object recognition model training results are shown for both models. The You Only Look Once Algorithm version 3 (YOLOv3) is compared to the new network in terms of object recognition accuracy and computational costs. When using the YOLOv3 architecture and the new architecture, we have considered all the input experimental images under the same conditions. It has been shown that the new network has almost the same detection capability as the YOLOv3 algorithm. However, the amount of memory consumption for the new network is much less. We have reported the results that show the efficiency of this network for detecting objects in @Work robots.

Keywords: @work robot, Object detection, Object recognition, YOLO algorithm.

1 Introduction

Robotics is one of the main branches in industrial automation. Industrial robots can perform a wide range of human tasks and replace human operators. Industrial robots can be defined as follows: “Robot is any automatically controlled, reprogrammable multipurpose manipulator programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications.”[1] This is a comprehensive definition of industrial robots that fully explains all the features of a robot, such as the ability to repeat in the program - change the execution of tasks or change the path of the robot by making changes in the program.

According to the main performance, industrial robots can be divided into:[2]

- Manipulation Robots
- Technological Robots
- Universal Robots

- Special Robots

Industrial robots are equipped with a variety of sensors, many of which are intelligent that can help guide robots, detect obstacles and detect various objects in harsh environmental conditions or processes, and prevent possible damage to parts or the robot itself. One of these sensors is the camera, which sends its data to fast pattern recognition algorithms, and other cognitive sensors, which are often equipped with object recognition functions for advanced decision making. Due to the variety and complexity of detection, an efficient detection and inspection algorithm is essential for a robotic platform for use in object detection and displacement. When the machine is detecting an object, the challenges arise from complex machine tasks that require powerful computing capabilities, complex network model architectures, and powerful hardware. Due to advances in convolutional neural network (CNN) techniques, object detection performance has improved rapidly in recent years and is a good way to detect an object in real time. A major development in the CNN network is the use of several new activation functions that have enhanced model performance. One of the most effective activation functions is the Rectified linear unit (ReLU). The transition to ReLU from the sigmoid function solved its known problems, which hindered the learning of primary neural networks (NNs) on large datasets. ReLU can be mathematically represented as Eq. 1:

$$Z_{a,b,c} = max\ (k_{\ a,b,c}\ ,0) \tag{1}$$

where $k_{a,b,c}$ is the input of the activation function located at $(a\ ,b)$. Other variants include Leaky ReLU (and many other variants), ELU, Maxout, and Probout. For a better analysis of CNN components, you can refer to the following reference [3].

As mentioned, all of the above requires powerful hardware to do this. Now we want to do it at the same speed but with less powerful hardware. (For example, with Raspberry Pi). Do this because of the lightness of the network on less powerful hardware, but with the same speed and accuracy that the YOLO network has. All the above work has been done with the help of MATLAB software and we have compared the results with the YOLO network.

The remainder of this article is prepared as follows: Description of the object detection and Yolo network algorithm is presented in Sect. 2. The Method description is explained in Sect. 3. The efficiency of the suggested scheme is illustrated in Sect. 4. Some conclusions are finally included in Sect. 5.

2 Literature review

2.1 Object detection

Object detection is a phenomenon in computer vision that involves the detection of various objects in digital images or videos. Some of the objects detected include people, cars, chairs, stones, buildings, and animals. Object detection consists of various approaches such as fast R-CNN, Retina-Net, and Single-Shot MultiBox Detector (SSD). Although these approaches have solved the challenges of data limitation and modeling in object detection, they are not able to detect objects in a single algorithm run. YOLO algorithm has gained popularity because of its superior performance over the aforementioned object detection techniques [4].

2.2 YOLO Algorithm

Various deep learning architectures have been proposed since Krizhevsky et. al. [4] trained a neural network model of multiple convolutional and feedforward layers on large dataset of images for object classification, numerous deep learning architectures have been proposed. One family of these architectures is designed for object detection

which entails predicting bounding boxes that enclose objects of interest in a certain image. The state-of-the-art approaches for this task can be roughly divided into two categories. The first include two-stage models such as R-CNN [5], Fast R-CNN [6], Faster R-CNN [7] and SPP-net [8]. These models propose search regions then process and classify those regions. The second category comprises single-stage models such as Yolo [9] and SSD [10]. Two-stage object detection models achieve better accuracy but with slow inference due to demanding computations. On the other hand, single-stage models are faster with lower accuracy compared to two-stage models. In order to deploy the above models onboard of embedded devices, two important aspects must be taken into consideration. First, typical embedded devices have limited computational power. Second, a sequence of images (i.e., video) must be processed. Recent work aimed to address these constraints by creating light-weight versions of original models such as Tiny-Yolo1 and SSD300.8 Other approaches such as MobileNet [11] and ShuffleNet [12] optimize the base of a pre-trained network to have higher FPS. Lu et. al. [13] incorporated a Long Short-Term Memory (LSTM) model to make use of the spatio-temporal relation among consecutive frames in a video while Broad et. al. [14] added a convolutional recurrent layer to the SSD architecture to fuse temporal information.

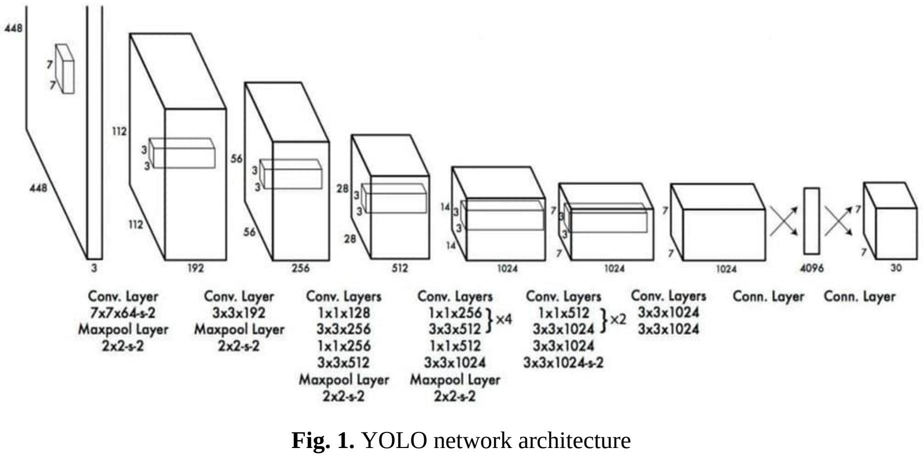

Fig. 1. YOLO network architecture

3 Method description

The process of identifying objects in @Work competitions is a very vital and important part. However, such a process is very difficult and time consuming without proper hardware. Traditionally, the most common way to identify objects in @Work competitions is to use image processing, which itself is based on a number of fixed and appropriate environmental factors (such as light, proper object angle, proper field, distance, etc.) to correctly identify and separate the object needs. However, the problem can be challenging due to continuously-changing camera viewpoint and varying object appearances as well as the need for lightweight algorithms suitable for embedded systems [15]. All of this requires the right hardware. As a result, we need programs and tools that, in addition to being lightweight, can be used on a variety of hardware. We need high accuracy in identifying objects in different situations so that we can easily and quickly identify objects despite the many changes that occur in the environment.

Here we have examined the neural network based on the YOLO network, but with the changes we have made in its hidden layers, we have been able to meet our demands in terms of identifying the objects needed to race with high speed and accuracy. Even if there are changes in the environment, we can easily identify objects. In addition, if a new object is added to the competition, the network will be trained at a higher speed.

The new changes are as follows: In the new network, a total of 16 layers have been used, which includes 4 Learning layers, 5 Activation layers, 2 Pooling layers and 2 FullyConnected layers (with two input and output layers). Accuracy and speed, we have reached a very low volume of the program, which reduces energy consumption and increases processing speed.

4 Simulation results

The proposed method is implemented in MATLAB software environment. In this method, we give the existing data to the input of the YOLO network to be trained and then with the same settings we give the datasets as input to the new network so that this network is also trained. Then we compare the accuracy and error rate of the two networks. The value of network output parameters occurs in different modes, which depends on the number of inputs, the number of network layers, etc., whose output includes learning time, network accuracy, loss, and so on.
Where the Accuracy is equal to (n here equals the number of correct detections in the images, m equals the total number of selected images) Eq. 2:

$$Accuracy = \frac{n}{m} \tag{2}$$

And the Loss formula is equal to (N is the number of instances. K is the number of classes. t_{ij} is the index whose instance i belongs to class j. y_{ij} the output for instance i for class j. i connect to class j) Eq. 3:

$$loss = \sum_{i=1}^{N} \sum_{j=1}^{K} t_{ij} \ln y_{ij} \tag{3}$$

As shown in Figures 2 and 3, both networks are fully trained.

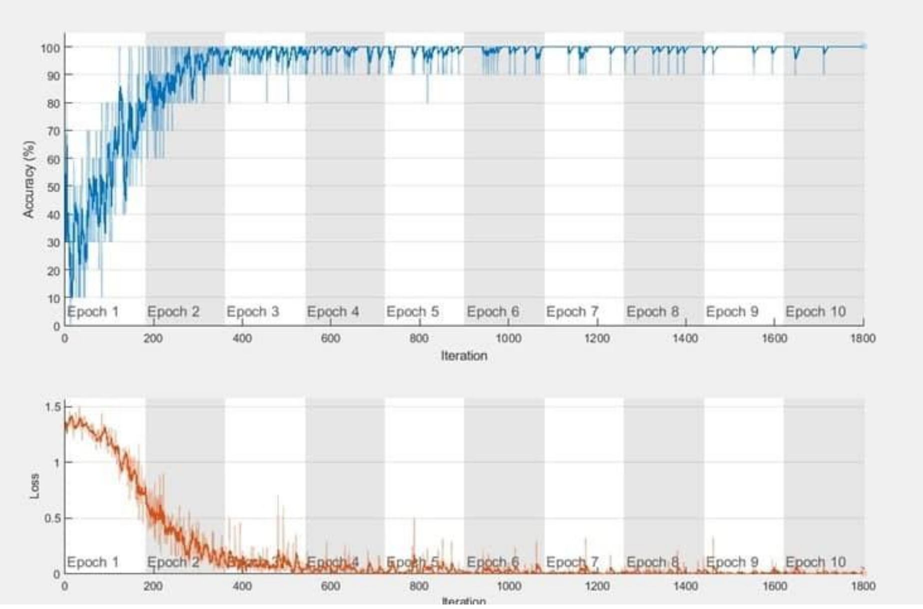

Fig. 2. The accuracy and amount of error when training a YOLO network

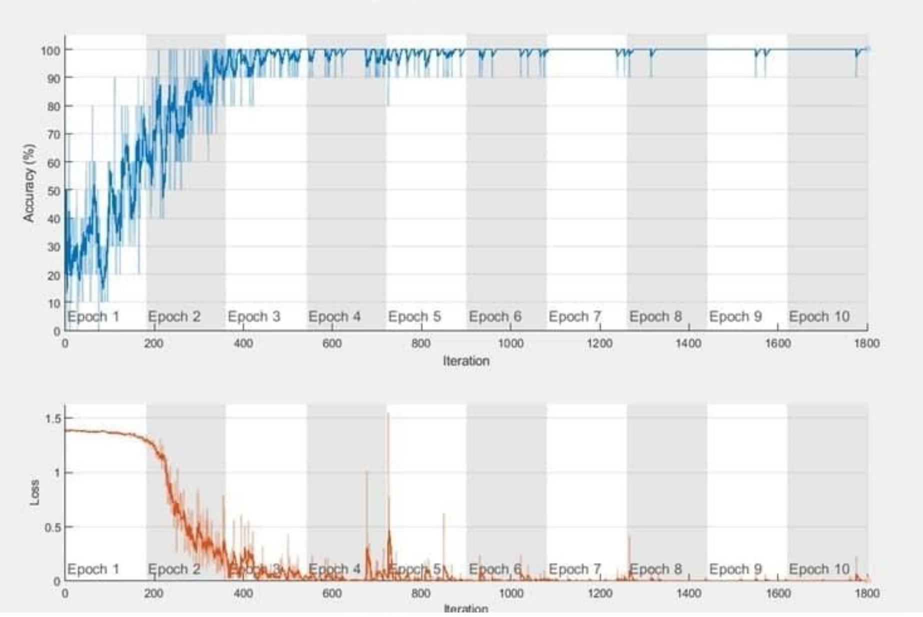

Fig. 3. The accuracy and amount of error when training a new network

These two networks have reached close to 100% accuracy and near zero error after completing 8 epochs, with the difference that the new network with fewer layers has a smaller volume according to Table 1. Performance evaluation of the proposed method using fewer layers is shown in Table 1.

Table 1. The units and values of the parameters in YOLO and newNet

	Established Time	Epoch	Iteration	Iteration Per Epoch	Hardware Resource	Network Size	Accuracy
YOLO Net	306 min 39 sec	10	1800	180	CPU	39908kb	**99.8311**
New Net	152 min 51 sec	10	1800	180	CPU	**5090kb**	99.4932

5 Conclusion and future work

In this study, using RGB images of objects in an environment with different backgrounds, overlapping objects and ambient light, this data set is taught and methods of object recognition are compared. Object detection results were reported with the YOLOv3 architecture and a new grid for new object image datasets. In this study, it was shown that by reducing the number of layers of the YOLO network as well as modifying and optimizing the remaining layers, we can reduce the size of the program, which in turn reduces the robot's hardware energy requirements and speeds up data processing in real time. It was also shown that reducing the number of layers does not have much effect on network recognition and training for this data sample. In addition, as the size of the program decreases, it meets our hardware needs. Further research should be pursued to improve the detection of objects when moving the robot arm and dim ambient light. One problem with our experiment is that only a few limited classes were used during the training. This problem can lead to overfitting and could be the reason why the new network works so well in the data set. To avoid over-adaptation, more classes and image changes should be used during training and testing. We intend to expand the number and type of images in the dataset and determine whether the new network (or other models) can exceed performance expectations during real-time object detection in low-light environments at higher speeds and more arm movements.

References

1. ISO 8373: 2012, Robots and Robotic Devices — Vocabulary. ISO - Online Browser, International Organization for Standardization, 2020, Available on: www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en.
2. Malega, P., Kuždák. V., 2012. Trendy vývoja liniek v automobilovom priemysle. Transfer inovácií 22/2012, 2012
3. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al., 2018. Recent advances in convolutional neural networks. Pattern Recogn. 77, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013.
4. Kiizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural
5. Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation," in [Proceedings of the IEEE conference on computer vision and pattern recognition], 580–587 (2014).
6. Girshick, R., "Fast r-cnn," in [Proceedings of the IEEE international conference on computer vision], 1440–1448 (2015).
7. Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks," in [Advances in neural information processing systems], 91–99 (2015).
8. He, K., Zhang, X., Ren, S., and Sun, J., "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE transactions on pattern analysis and machine intelligence 37(9), 1904–1916 (2015).
9. Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," in [Proceedings of the IEEE conference on computer vision and pattern recognition], 779–788(2016).
10. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "Ssd: Single shot multibox detector," in [European conference on computer vision], 21–37, Springer (2016).
11. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861 (2017).
12. Zhang, X., Zhou, X., Lin, M., and Sun, J., "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in [Proceedings of the IEEE conference on computer vision and pattern recognition], 6848–6856 (2018).
13. Lu, Y., Lu, C., and Tang, C.-K., "Online video object detection using association lstm," in [Proceedings of the IEEE International Conference on Computer Vision], 2344–2352 (2017).
14. Broad, A., Jones, M., and Lee, T.-Y., "Recurrent multi-frame single shot detector for video object detection.," in [BMVC], 94 (2018).
15. https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11605/1160529/Robust-real-time-pedestrian-detection-on-embedded-devices/10.1117/12.2587097.short