Methods for object recognition and classification for tele-controlled service robots

Angel Ivanov¹, Nayden Chivarov¹,²
1 Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str., block 2, 1113 - Sofia, Bulgaria
2 European Polytechnical University, Pernik, Bulgaria
angel.b.ivanov@gmail.com, nchivarov@gmail.com

Abstract. The need of personal service robots has been increased in the past several years. The tasks assigned to the robots vary in complexity and degree of freedom. In order to assist senior citizens and people with disabilities with their day to day tasks, a greater degree of freedom is required – more autonomy, ability to fulfil more complex tasks and easier for use human-machine interface. In this paper we show some of the methods that can be implemented in personal service robots for object recognition and classification in order to achieve greater autonomy, hence greater degree of freedom.

1. Introduction
The recent growth of aging population and respectively people with disabilities lead to additional care for these people, done by large number of social workers and assistants. To face this issue, the society is forced to use robotics and methods for automation in order to ease the lives of the seniors and people with disabilities. One such solution is the use of household service robot.

The International Federation of Robotics (IFR) has defined in [1] the term “Service Robot” as robot that performs tasks, useful to humans. According to ISO8373 the robots require a degree of freedom, which means that it should have the ability to fulfil the assigned tasks without human intervention and based on current state and sensing. This degree can vary from partial autonomy – the tasks are done with human intervention, to full autonomy – the tasks are done by the service robot alone.

The IFR statistics show [2] that for 2018 the service robots, sold for domestic use and/or household tasks are 16.3 million units. They predict that by 2022, the domestic service robots production will drastically increase to 61.1 million units.

To achieve a higher degree of freedom, the service robot would require having a better “view” of the surrounding environment. At this point, the environment knowledge gathering is divided in 2 major problems:

- Environment mapping;
- Object detection and recognition

The environment mapping is usually done with the help of multiple sensors like ultrasound beacons, radars, GPS, etc. as they gather information about the current physical position of the robot and will help with the route planning and travel.

For object detection, recognition and classification, the only required sensing hardware is a camera. With the help of the camera, the service robot can actually “see” the objects that are in front of it. In order to do the actual recognition and classification, various methods can be used.
In this paper we will present and test 2 different machine learning architectures, with which we will do object recognition in the context of tele-controlled service robot for assisting elderly people and people with disabilities.

2. Object detection and classification

In the recent years a lot of attention is given to the problems of detecting objects in still images and/or video streams. This attention has led to a lot of new machine learning architectures and respectively revisions/improvements to existing ones. Traditional object detection and recognition methods rely on handcrafted features and shallow trainable architectures [3]. Some of the methods use Haar-like features to describe the desired object for recognition, however the task to build the Haar-like feature set is slow and cumbersome.

In [4] was suggested the use of convolutional neural network (CNN) for object detection which yields a very low error rate compared to the current state-of-the-art. In [5] an improvement in terms of speed was proposed with a R-CNN. This type of network scans the image from bottom-up and generates regions of interest (ROI). Each ROI is then parsed through a previously trained large CNN for object detection and classification. This method is said by the authors to improve the mean accuracy (mAP) by 50% compared to traditional CNN.

Although CNNs by themselves are great at object detection and classification in still images, this is not the case for video streams. This issue is easily resolved by adding a backpropagation connection into the neural network architecture. Such architectures are presented in [6] as Recurrent neural networks (RNN) and its special case Long Short-Term Memory (LSTM) networks.

The latest advancements [7,8,9] show the most successful in terms of speed and accuracy architectures. The YOLO and SSD are the architectures of choice in this paper to test the object recognition for service robots.

2.1. YOLO

You Only Look Once (YOLO) [7] is a state-of-the-art architecture that does not require a complex pipeline to process an image. Unlike the sliding window or region-based methods, it sees the whole image during training and test phases, and implicitly encodes contextual information about object’s class and appearance.

![YOLO Model](image)

**Figure 1. YOLO Model**

In figure 1 can be seen the model of the YOLO architecture. It first generates SxS grid on the input image. Then creates prediction bounding boxes B and assigns confidence scores to each box. At the same time creates a class probability map C. These predictions are finally encoded as $S \times S \times (B \times 5 + C)$ tensor.
The network architecture is shown in figure 2.

![YOLO Network architecture](image1)

**Figure 2.** YOLO Network architecture

The network contains 24 convolution layers, followed by 2 fully connected layers. The 1x1 convolution layers are used to reduce the feature set, generated by the previous layers.

The authors claim to achieve speed of 45 frames per second which is more than enough for realtime video processing.

The original YOLO architecture is fast, but it stays behind the competitors in terms of accuracy. Therefor the authors in [8] propose an updated version of the network. The new architecture is shown in figure 3.

![YOLOv3 feature extraction architecture](image2)

**Figure 3.** YOLOv3 feature extraction architecture
The network now consists of 53 convolutional layers, as opposed to 24 in the original architecture. This boosts the accuracy of the object classification with more than 30% while maintaining the same speed. On higher resolution images, the classification accuracy is improved even more.

For the experiments in this paper, we will use YOLOv3 architecture.

2.2. SSD

The Single Shot Multibox Detector (SSD) [9] is another state-of-the-art architecture that has received a wide publicity because of its speed and accuracy.

The SSD framework is shown in figure 4.

To train the network, it needs an image with ground truth boxes – figure 4a. Using a convolution layers, small set of default boxes with different aspect ratios are calculated. This is done for each location and in several feature maps with different scales – figure 4 b, c.

The model of SSD is shown in figure 5.

The SSD model is based on a feed-forward convolutional network and produces a fixed size collection of bounding boxes and scores. These scores represent the presence of an object class instance in the respective bounding boxes.

For the experiments in this paper, we will use SSD architecture.

3. Experiments and results

Our goal in this paper is to compare both SSD and YOLOv3 architectures in terms of accuracy and speed. The comparison is within the aspect of usage in autonomous service robots.

For the experiments, we use Intel RealSense D415 [10] stereo camera. For the sake of simplicity, we use only the RGB mode without the additional depth mode.
For video parsing we use the OpenCV library [11][12]. The OpenCV is an open-source computer vision library written in C/C++. It addresses the cases of object segmentation, detection, recognition and tracking. It also supports camera calibration, stereo vision and 2D/3D shape reconstruction.

The OpenCV library is available for use with Python wrappers which accelerates the prototyping speed.

Both SSD and YOLOv3 models are trained using the famous COCO dataset [13],[14]. This dataset contains over 300000 images of 91 object classes. All object instances are manually labelled using extensive crowd worker involvement and that concludes in over 2 mil labelled object instances. Some of the object categories include person, dog, car, bike, chair, sofa, bottle, etc.

Since both SSD and YOLOv3 are single-shot detectors, their performance in terms of speed in frames per second is similar – on CPU processing around 5-7 fps, on GPU processing around 30-40 fps.

Both SSD and YOLOv3 are able to recognize human with high confidence rate of at least 99,95% as shown in figure 6.

![Live Streaming YOLOv3](image1)

![Live Streaming SSD](image2)

**Figure 6.** YOLOv3 vs SSD on human detection

Both models can recognize common objects around a typical house with confidence rate of at least 70% as shown on figure 7.
The difference is observed when multiple objects are presented in front of the camera. In a test case with 10 objects, YOLOv3 is able to detect all of them, while SSD detects and recognizes only 4 that are nearest to the camera.

Depending on the image resolution parsed by OpenCV a wrong classification can occur from both models. Using smaller resolution increases drastically the performance but lowers the accuracy. Higher resolution decreases the performance, but the accuracy is much better. It is a matter of fine tuning for a specific use case to balance the speed/accuracy ratio.

![Live Streaming YOLOv3](image1.jpg)

![Live Streaming SSD](image2.jpg)

**Figure 7.** YOLOv3 vs SSD on common house objects

4. Conclusion
The rising need of service robots used as personal assistants for the elderly and people with disabilities require high degree of freedom, hence high level of autonomy. To achieve these requirements a combination of different sensing hardware is needed for navigation through the home of the patients. To achieve the higher autonomy the use of computer vision methods is required to be used.

In this paper a comparison of 2 state-of-the-art machine learning algorithms was made – YOLOv3 and SSD. Both methods are part of the family of single-shot detectors – they go only once through the image/frame.

While both methods provide similar speed/accuracy ratio, the experiments showed that YOLOv3 is able to detect and classify more simultaneous objects in a video stream.

In the use-case of service robots both methods can be used and will provide a similar practical outcome for object detection and classification, so it is up to the researcher to choose which model is more suited for the respective implementation.
5. Acknowledgement
The research presented in this paper was supported by the Bulgarian National Science Fund under the contract No. DN 07/23 – 15.12.2016 “Tele-controlled Service Robots for Increasing the Quality of Life of Elderly and Disabled”.

References
[1] International Federation of Robotics (IFR), https://www.ifr.org/service-robots/
[2] International Federation of Robotics (IFR), Executive Summary World Robotics 2019 Service Robots
[3] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu, Object Detection with Deep Learning: A Review
[4] Krizhevsky, A., Sutskever, I., & Hinton, G., ImageNet classification with deep convolutional neural networks
[5] Girshick, R., Donahue, J., Darrell, T., & Malik, J., Region-based convolutional networks for accurate object detection and segmentation
[6] Alex Sherstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network
[7] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection
[8] Joseph Redmon, Ali Farhadi, YOLOv3: An Incremental Improvement
[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, SSD: Single Shot MultiBox Detector
[10] Datasheet, Intel RealSense D400 Series Product Family
[11] Gary Bradski, The OpenCV Library
[12] Kari Pulli, Anatoly Baksheev, Kirill Kornyakov, and Victor Eruhimov, Real-Time Computer Vision with OpenCV
[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollar, Microsoft COCO: Common Objects in Context
[14] COCO Dataset Internet Site, http://cocodataset.org/