POLITECNICO DI TORINO

MASTER'S DEGREE IN MECHATRONIC ENGINEERING

Title: Person tracking methodologies and algorithms in service robotic applications
Candidate: Anna Boschi [s253105]
Supervisor: Prof. Marcello Chiaberge

ABSTRACT

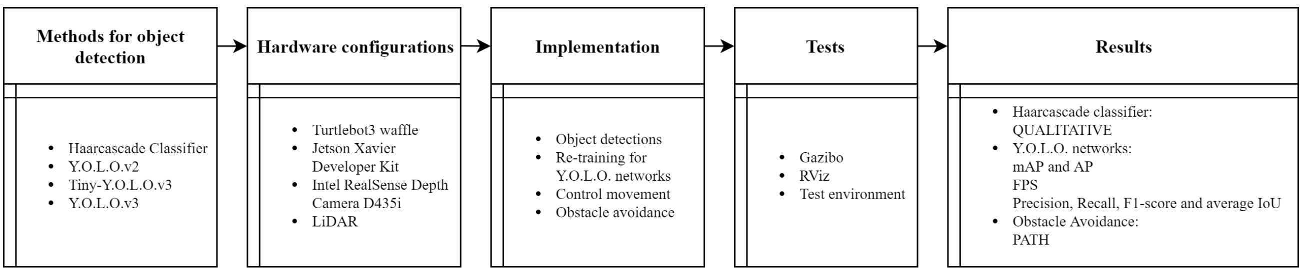
The vital statistics of the last century highlight a sharply increasement of the average life of the world population with a consequent growth of the number of elderly people. This scenario has caused new social needs that the research in the service robotics field is trying to fulfill. Particularly, the idea of this thesis is born at the PIC4SeR (PoliTo interdepartmental centre for service robotics) with the purpose of creating complex service robotics applications to support the autonomous and self-sufficient old people into their house in everyday life, avoiding the task of monitoring them by third parties.
This work represents the first steps of a broad project in which many other service tasks will be integrated.
The main argument of this thesis is to develop algorithms and methodologies to detect, track and follow a person in an indoor environment using a small wheeled rover and low cost and available sensors to monitor the target person. Several techniques are explored showing the evolution of these methods along the years: from the classical Machine Learning algorithms to the Deep Neural Network ones. Since the main requirement to be respected is the necessity of real-time results, only few of the analysed algorithms are developed for this project scope and at the end are compared in order to find the best solution with optimal outcomes. The detection and localization are the basis of the person tracking application, done by the robot on which it has been implemented a movement control algorithm and at last it has been introduced an obstacle avoidance algorithm to prevent collisions.

1. OBJECTIVES

In the last century the attention on the service robotics has focused on the assistive system in order to promote the aging-in-place and to make the independent indoor life easier. The objective of this thesis is to do *detection*, which implies both identification and localization of the person in the indoor environment using a stereo-camera set on a TurtleBot3 waffle and other suitable hardware according to the real-time requests and the type of algorithm used for object detection. The person detected is followed, remaining at a certain safety distance, by the robot on which it is implemented a movement control coherent to the person movements. At the end, an obstacle avoidance algorithm is integrated in order to prevent collisions. The purpose of this thesis is to set the basis for a broad project in which many other tasks to monitor elderly people will be added.

2. DEVELOPED WORKS

To correctly explain the work done, the result section will be divided in:



2.1 Methods for Object detection

According to the scope of the thesis the first thing to do is to detect the person in the indoor environment, so several methods have been explored and depending on the real-time request only few of them have been developed. The chosen techniques are referred to two different approaches: *Haarcascade Classifier* belonging to the classical Machine Learning algorithms and *Y.O.L.O.* networks belonging to the Deep Neural Network techniques.

The Haarcascade classifier algorithm consists in the construction of a cascade of classifiers in series to increase the detection results and to reduce the computation time. This is a great intuition because the use of smaller boosted classifiers reduces the complexity of the algorithm, which is considered to run real-time. The detection process can be explained with a degenerate decisional tree structure of classifiers commonly called "cascade". The process of detection consists in dividing the image in sub-windows and then analysing all of them with the first stage of classifier. The sub-windows that receive negative response are rejected, differently the ones that result positive are analysed using the second stage of classifier. The process continues until either all the areas of the image are rejected or there is a final positive response, so the object requested is detected in the image. In this way the time cost is reduced and the performance of the algorithm is high.

Y.O.L.O. (*You Only Look Once*) is one of the last neural networks used for creating bounding boxes over the detected objects. The operations done from the network are the following: i) the input image is divided into a grid; ii) each grid cell generates bounding boxes and predicts their confidence rate; iii) each grid cell has its class probability; iv) the total number of bounding boxes is minimized setting a minimum confidence rate and the output is the same image with the bounding box over the detected objects with the reference classes and the accuracy percentages. Y.O.L.O. has many versions and three of them are tested: Y.O.L.O.v2, Tiny-Y.O.L.O.v3 and Y.O.L.O.v3. Y.O.L.O.v2 is an important evolution of the original version including many features to increase the performance. Y.O.L.O.v3 is the last version of this network in which there is an improvement on the prediction of the classes. Tiny-Y.O.L.O.v3 is the lightweight version of the original with a reduced dimension of the network.

2.2 Hardware configurations

The problem of this thesis comes from the necessity of a real-time algorithm that is able to perform person detection and elaborate these data to move the robot and doing person tracking. Of course all of these operations must be as instantaneous as possible because the risk to be prevented is to lose the person to follow in case the robot moves away from the person's view. A set of possible hardware has been analysed in order to find the best solution for this specific application. The selected solution is using a TurtleBot3 waffle as robot platform with a Jetson Xavier Developer Kit, an Intel RealSense Depth Camera D435i and a LiDAR integration and the power supply is a Lipo 4S with a battery discharge sensor. These hardware components are supported by a ROS Kinetic Klame platform installed inside.

2.3 Implementation

Using the previously explained methods for object detection and the hardware chosen for this project it has been possible to localize the person in the space and using this information to generate a control movement algorithm to regulate the linear and angular velocities of the robot coherently with the movement of the person remaining at a certain safety distance from it. In case of no detection or multiple detection the robot stops.

The Haarcascade classifier algorithm has been developed following a double approach: using only the upper-body classifier or using the three listed classifiers in decreasing cascade: full-body, lower-body and upper-body trying to increase the global robustness of the person detection. However only the first approach results suitable for this project because the use of three classifiers in cascade makes the algorithm slow and not real-time.

The Y.O.L.O. networks are generally used pre-trained with the COCO (*Common Object in COntext*) dataset of 80 classes, but in perspective of the use of them for person detection a re-training could give an improvement in the performance of the networks. The re-training has been done using the *Transfer Learning* technique and for that the networks have been modified in their structure: the number of classes and other specific hyper-parameters as batch dimension, learning rate and input image size.

At the end, with the LiDAR support, an Obstacle Avoidance algorithm has been developed, belonging to the Local Motion Control, with a dynamic goal, set as the position of the person with a certain safety margin.

2.4 Tests

The implemented algorithms have been tested many times at the beginning in the simulation environment on Gazebo and on RViz, a 3D visualization tool of ROS, to regulate the control parameters and consequently the movement of the robot. Finally, the robot's behaviour has been tested in a real environment.

2.5 Results

The results obtained evidence that:

- the Haarcascade classifier algorithm is not suitable for this application because the false positive detections imply not fluid movement of the robot and the loss of the person in few time (fig.1a);
- the re-training of the Y.O.L.O. networks does not produce improvements in mAP (*mean Average Precision*) or AP (*Average Precision*) of the "big" networks, but only of the lightweight one (fig.2a). However, the FPS and other parameters as Precision, Recall, F1-score and average IoU (*Intersection over Union*) increase the performance of the pre re-training networks (fig.2b). Anyway, Y.O.L.O.v2 is not fitted for this application caused by many false positives (fig.1b); Tiny-Y.O.L.O.v3 (fig.1c) and Y.O.L.O.v3 (fig.1d) are extremely performant, so both suitable. However, the best one is the Tiny version because it reaches the same results with higher FPS, so a best performance with lower computational complexity;
- the Obstacle Avoidance tests highlight the correct behaviour of the robot according to the implemented algorithm (fig.2), however the not optimized trajectory and the long execution time needed make this implementation not suitable, it should be improved to correctly integrate it into the person tracking project.

(a) *Haarcascade classifier: **a**:Detection in Infrared mode; **b**:Detection in RGB mode; **c**:False positives.*

(b) *Y.O.L.O.v2*

(c) *Tiny-Y.O.L.O.v3*

(d) *Y.O.L.O.v3*

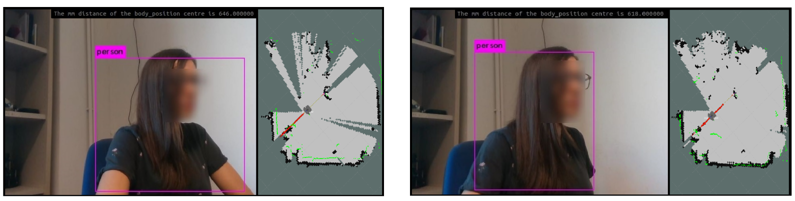
Figure 1: Outputs of the Haarcascade classifier and the Y.O.L.O. networks.

Architecture	**Dataset**			
	COCO		Person	
	AP	mAP	AP	mAP
Y.O.L.O.v2	56.81%	23.30%	54.65%	16.24%
Tiny-Y.O.L.O.v3	19.21%	5.16%	49.30%	8.17%
Y.O.L.O.v3	69.11%	28.99%	68.64%	28.46%

(a) *AP and mAP of each network related to COCO and Person datasets.*

Parameters	**Y.O.L.O.v2**		**Tiny-Y.O.L.O.v3**		**Y.O.L.O.v3**	
	COCO	Person	COCO	Person	COCO	Person
Precision	0.23	0.24	0.15	0.24	0.26	0.46
Recall	0.30	0.24	0.11	0.16	0.38	0.36
F1-score	0.26	0.24	0.13	0.19	0.31	0.40
average IoU	18.99%	20.29%	12.43%	19.99%	21.71%	38.52%

(b) *Re-training parameters related to IoU=75%.*



(a) *Detection of the person, identification of the goal.*

(b) *The goal's pose changes while the robot is moving towards it.*

Figure 2: Obstacle Avoidance results using Tiny-Y.O.L.O.v3 network.