UNIVERSITY OF BIRMINGHAM

ITIC Internship Project Report:
Office Robot Prototype: Monitoring for Health Hazards

1 Introduction

In the *Office of the Future,* robots will play a crucial part. Robots deployments in manufacturing factories, warehouses, care homes etc. are on the rise. Soon we will have a world with robots roaming offices freely. Currently one of the most widespread use of robots in the office is to provide telepresence, where a robot can be remotely controlled by a person being their eyes, ears and mouth in the environment.

The aim of this project was to prototype more uses for robots in the office such as monitoring, mail or drinks delivery etc. Health and safety is vital to any working environment. It is important to make sure that health and safety rules are being followed and to keep certain checks. During daily office work, seemingly benign tasks can sometimes cause health and safety risks such as leaving a water bottle open next to a switch; if the bottle tips over and damages the switch, it could result in a short circuit or even a fire. It is easy to forget to screw on the cap of the bottle in a hurry. Therefore, it is important to be able to monitor for such anomalies.

2 Purpose

This project is a first step towards such monitoring. It aims to detect bottles and other objects on the floor that may be damaged or cause damage.

A video of the project can be found here: https://youtu.be/hCtqooVM48Y

3 Overview

3.1 Design

3.1.1 Hardware

1. Robot Base: Turtlebot[9] - The Turtlebot 2 is a mobile robot that consists of a Kobuki base and Microsoft XBOX Kinect.
2. Processor: Intel NUC[2] - The Intel NUC (NUC5CPYH) is a bare-bones PC which was equipped with an 8GB RAM and 64 GB USB drive for storage.
3. Battery: RAV Power Bank[1]

3.1.2 Software

1. Robot Operating System (ROS) Indigo[8] on Ubuntu 14.04 - ROS includes a wide set of algorithms useful for various aspects of robotics such as navigation and sensing. It makes it very easy to quickly set up a robot and get it moving.
2. TensorFlow[5] - TensorFlow is an open source library by Google for Machine Intelligence. This project uses the TensorFlow Object Detection API[6] for detecting objects.

Figure 1: Turtlebot 2 (source: [9])

Figure 2:Intel NUC (source: ple.com.au)

Figure 3: Logo for ROS Indigo (source: [4])

3.2 Combined Setup

The Turtlebot 2 was the main part of the hardware platform. The Kinect was used as a depth sensor for navigation and camera for object detection. Since the Intel NUC requires 12-19 VDC power and is not battery powered, the RAV Power Bank was used to power the processor. The Intel NUC served as the brain of the robot. All software required for the robot to move and sense was installed on the NUC. The robot then consisted of the NUC, Turtlebot and power bank.

A remote laptop was used to connect to the robot and most processes were run from the remote laptop while movement and sensing were done through the NUC.

ROS runs various programs as nodes where each node can communicate with others by publishing and subscribing to topics. ROS creates a master node that all other nodes can communicate with. This setup allows for remote connections and processes. An in-depth tutorial can be found at [3] or [4].

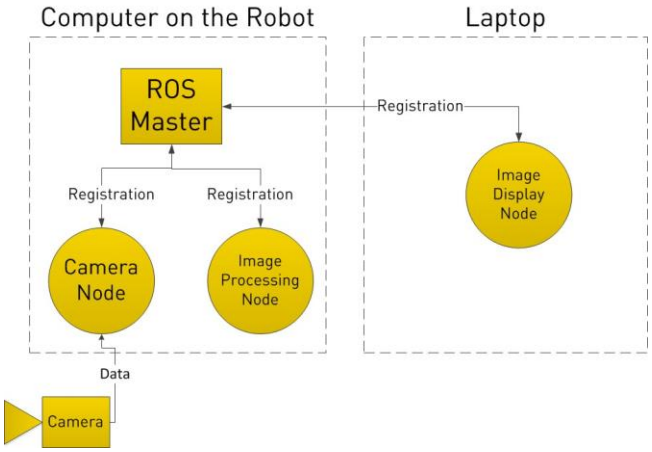

Figure 4: ROS with a robot and remote PC (source [3])

3.3 Functions

1. Visit specified locations in environment/office space
2. Collect pictures at said locations
3. Detect items at said locations
4. Send email alerts if any hazardous items found

Fatma Faruq (CS PhD Student, ITIC Intern)
fxf603@student.bham.ac.uk

4 Methodology

The system was first tested in simulation using ROS and Gazebo. Then the actual robot hardware was setup and the robot system was tested.

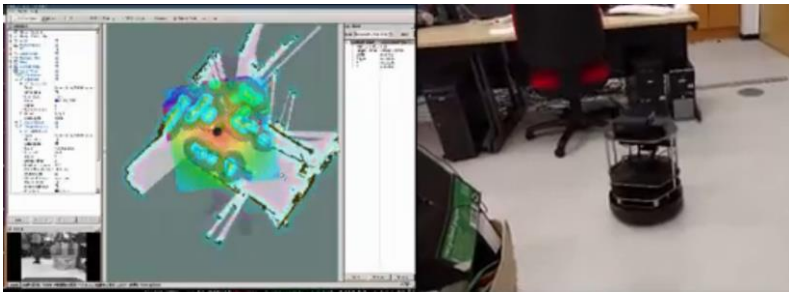

Figure 5: Left: Map of the environment, sensor data from the Kinect. Right: Robot in the real world

Figure 6 illustrates the system setup. To have the robot navigate from one place to another, a map of the environment was made using the ROS navigation stack. Once the map was made, it was saved and used as a base map for the robot. In ROS navigation, this is referred to as the global map.

Coordinates for specific locations were saved and the location scheduling node was used to have the robot visit each of these locations. At each location, the robot took two pictures which were used for object detection. Tensorflow's Object Detection API was used to detect objects. A pretrained convolutional neural network model (provided by Tensorflow) from the COCO dataset[7] was used. If objects were detected an alert was sent to a predefined email address with a picture of the location.

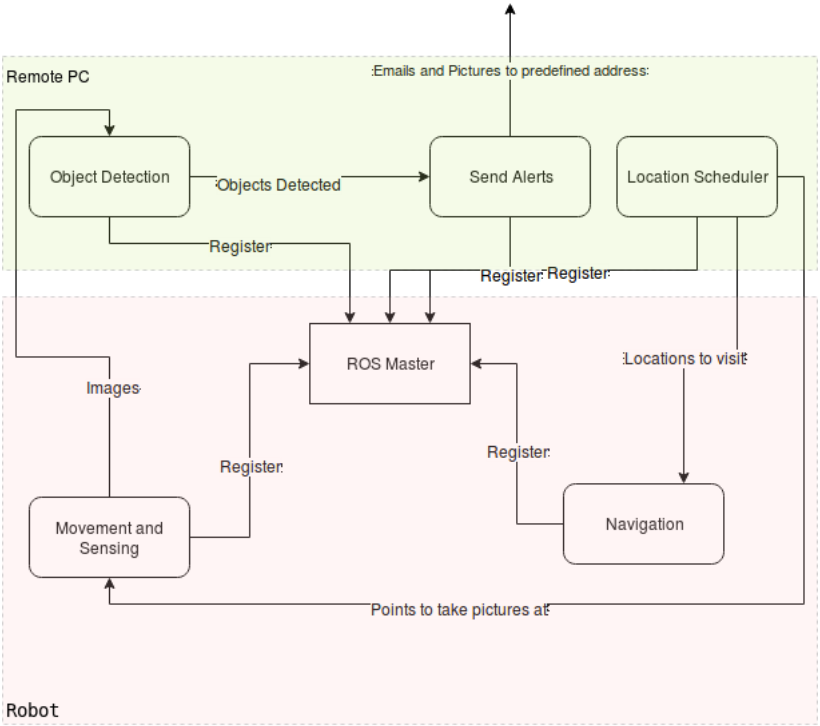

Figure 6: System Diagram

The ROS Navigation stack which is used to get the robot from one place to another consists of two planners- a global planner and a local planner. The global planner plans a path for the robot from point A to B using the global map which is the saved map. The plan is a path (as a series of points) the robot

Fatma Faruq (CS PhD Student, ITIC Intern)
fxf603@student.bham.ac.uk

should follow to get to location B. The local planner in this case was the Dynamic Window Approach(DWA) planner. The local planner considers dynamic obstacles and hence uses real time map information obtained from the Kinect. It plans the next step for the robot while trying to stay as close to the path the global planner has produced as possible. If an obstacle blocks the robot's path, it tries to move around the obstacle which may not be possible due to the robot's dynamics, the environment itself or the robot's speed profile parameters as defined by the user. If the robot cannot move around the obstacle it aborts going to its goal location and alerts the user. For each run the robot had to be localized in its environment. This meant that before the monitoring process began the robot had to be moved around so that it could determine where it was in the environment using sensor data and information from the saved map.

The Tensorflow Object Detection API provides a pipeline for object detection. It makes it easy to define the type of training data to be used, the model used as well as the output of the detector. For this project, the Tensorflow object detector for off the shelf inference was used. The detector consists of a Single Shot Detector (SSD) which uses a single feed forward convolutional neural network to predict which class an object belongs to[6]. The SSD model is trained on the COCO dataset which contains 80 object categories including various household objects such as bed, bottle etc, animals such as horse, cat etc and many others[7]. The result of the detector is a list of classes or categories which objects may belong to along with a score which is used to denote how likely it is that the said object belongs to said class. 0 means not likely at all while 1 means certainly belongs to said class. Along with the class scores the detector also gives a bounding box which can be used to determine the location of the object in the image. For this project if top three classes ordered by their score included the bottle class, it was considered that a bottle was detected. Furthermore, if an object was detected with a score of more than 0.9 (this is the reason the laptop was detected in the video).

5 Limitations and Improvements

Navigation: The navigation needs to be further tuned to consider dynamic obstacles. There are times when the robot will abort its mission if dynamic obstacle blocks its path for a long time, which is acceptable since the robot may have a specific time duration in which to fulfil its monitoring task. However, this can be further modified.

Object Detection: The types of objects that can be detected are limited to the ones in the COCO Dataset. For example, the robot was unable to detect coke cans since these were not in the dataset. Furthermore, false detections need to be excluded for example the robot detected a PC as a laptop on the floor. Context based detection could be used, meaning only objects that should not be on the floor should be detected. This could be as simple as creating a list of possible objects that can be excluded if detected.

Hardware: The size of the robot and the placement of the Kinect are such that the robot can only detect objects on the floor. Furthermore, the Kinect is placed such that some portion of the robot occludes its view. To detect objects at desk level or higher, another Kinect could be placed at a higher location.

Others: The project started with the idea of having multiple robots collaborate with each other to deliver items or bring photocopies to offices. However, due to hardware limitations (non-availability of robots able to grasp paper) this was abandoned in favour of robots that could be asked to bring objects from predefined locations. This would require that a human place required objects on the robot since the robot did not have any arms. Some work was done in this direction where Google speech recognition and CMU sphinx speech recognition were used to translate speech to text and this text was processed for certain words. The robot could then navigate to a location, ask for an object in those

locations and once the object was placed on the robot bring it back to its starting position. However, this required that the navigation be robust to dynamic obstacles and speech recognition work for various accents and could not be achieved in the limited time for this project.

In terms of hardware, the communication between the robot and PC had to be established on a separate router. For the purposes of this project a Google Nexus 5X was used as a WIFI hotspot which both the PC and robot connected to. Furthermore, to be able to send mails, the remote PC had to be connected to the Internet. This was accomplished by using a WIFI adapter connected to the remote PC. To localize the robot and create the map, it was important to move the robot around. It was easy to move the robot around using a Logitech joystick.

6 SWOT Analysis

STRENGTHS • An automated monitoring tool • Reduces need for human monitoring • Human resources can be used elsewhere • Data collection (in the form of pictures) • Reduced costs	OPPORTUNITIES • Checking for hazards is a mundane task that requires attention to detail which can be very tiring. • More detailed cataloguing and checking can take place • Robots can be deployed at any time and at multiple times in a day
WEAKNESSES • In development/prototype stage • No flexible decision-making system • False positives • Higher one-time investment	THREATS • Resistance to adoption of technology • Lack of trust in robot monitoring • Feasibility in terms of developmental costs and testing

7 Conclusion

The project is in its early stages and could prove to be a useful endeavour. Though there is a lot of room for improvement, there is a lot of scope for enveloping other tasks into the project as well such as intruder detection, mail or coffee delivery etc. Robots are poised to enter both the work place and the home, therefore work on such technology is at the cutting edge of technology and innovation but in infancy.

8 Acknowledgments

I would like to thank the people at the IT Innovation Centre particularly Nandy Milan for being so accommodating, for letting me incorporate elements that would be relevant to my research, for being ever ready with solutions and workarounds and her help and guidance. I would also like to thank Li Zhao for her advice, help and support and for being so approachable. I would like to thank my advisers at UoB (Nick Hawes and Dave Parker) for encouraging me to take on this topic and their input on ideas for the

project and their support. I'd like to thank BARC at UoB for giving me the confidence to work on this project on my own. Lastly, I'd like to thank my family and friends who always support me and put up with all the odd times.

Appendix I. Evaluation Matrix Scores

Area	Scoring System	Score	Reason
Maturity	1 = Idea 5 = Mainstream Product	2	Office robot is an emerging technology in the office of the Future
Technology (Adoption timescales)	1 = > 3 years 5 = < 3 months	1	Office robot is still in the prototying stage
Business Process (Adoption timescales)	1 = > 3 years 5 = < 3 months	1	Due to the immaturity of the technology, the business is unlikely to adopt it in the very near future.
Adoption Overview	1 = v long time 5 = very short	1	Emerging technology and the acceptance of the business will be a long process
Existing Technology (Impact)	1 = v large impact 5 = very little	2	It will bring large impact to the business both on technology and business models
Resources Required	1 = v large impact 5 = very little	1	To build a mature office robot will need investment on robot and software development
Scope	1=very difficult 5=very easy	1	At its current stage, it is difficult to be used in other areas.
Usability	1=very difficult 5=very easy	1	Limited usage
Security	1 = very poor 5 = excellent	1	No security measures has been considered in the prototype
Innovation Value	1 = low innov. 5 = high innov.	2	Emerging technology
Cost Effectiveness	1=very expensive 5=very cost effective	2	To build a more useful prototype, will need to invest in hardware and software development
Adoption Readiness Score	*<20 - not ready* *20-29 - emerging* *30-39 - Adoptable* *>39 Fully Ready*	*15*	*Office robot is an emerging technology and will take some time before it reaches maturity.*
***Note**: Rows that have no highlight colour indicate the score value is not added to the adoption readiness total. Instead, the overview score for that area is used as part of the total score.*			

References

[1] Charge faster charge smarter. https://www.ravpower.com/ismart.

[2] Intel R nuc kit nuc5cpyh. https://www.intel.co.uk/content/www/uk/en/products/ boards-kits/nuc/kits/nuc5cpyh.html.

Fatma Faruq (CS PhD Student, ITIC Intern)
fxf603@student.bham.ac.uk

[3] Ros 101. http://robohub.org/ros-101-intro-to-the-robot-operating-system/.

[4] Ros wiki. http://wiki.ros.org/.

[5] JJ Allaire, Dirk Eddelbuettel, Nick Golding, and Yuan Tang. *tensorflow: R Interface to TensorFlow*, 2016.

[6] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016.

[7] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[8] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[9] Turtlebot. www.turtlebot.com