Scene Classification for a Mobile Interactive Robot

by

Laura Donadoni

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday December 17, 2019 at 10:00 AM.

Student number:	4749324	
Project duration:	December 1, 2018 – December 17, 2019	
Thesis committee:	Prof. Dr. M. Neerincx,	TU Delft, supervisor
	Dr. Ir. D.J. Broekens,	TU Delft
	Prof. Dr. Ir. D.A. Abbink,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft Delft University of Technology

IR

Preface

This project aims to achieve awareness for a social robot through scene classification by analysing the environment and the context of the surroundings. This thesis is the result of my graduation project for the completion of the Master degree in Embedded Systems and my passion for technology and robotics.

I would first like to thank my thesis advisor dr. ir. Joost Broekens for the advice during the research and development, and the support; my supervisor prof. dr. Mark Neerincx that made my project possible, and dr.ir. David Abbink for being part of my committee.

I would also like to thank Bas Hazelzet, Diony Tadema, Jurjen Brouwer and Martijn Folmer from Interactive Robotics. Without your professional help, the development of my project could not have been successfully conducted.

I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them.

I could have not been successful in the process of researching and writing this thesis without the support of Rutger, which I am thankful for believing in me no matter what and for always having chocolate for cheering me up.

Finally, I must express my gratitude to my friends. Special thanks go to Melek and Mary Ann for the uncountable cups of tea taken while comforting each other; to Carlo, for being my fixed point in this changing environment; to Irene, my companion of trips and adventures; to Alex and Alessia, my oldest and dearest friends. Without your support, I would have not been able to reach this point.

Laura Donadoni
Delft, December 2019

Abstract

The use of social robots increased in the past few years. Current technology, however, lacks in deploying a single robot for different applications without the help of a human being. Current solutions are time-consuming, labour intensive and hard to generalize. Being aware of its surroundings, in terms of environment and context, the robot can select the appropriate application that the situation needs. We propose a multi-modal, knowledge-based hybrid scene classification method for applying awareness to the robot. As scene we refer to the combination of the environment and the context of the surroundings; a study on how to describe a scene has been done through knowledge-engineering methods that comprehend an anonymous online questionnaire and observations. The method inputs features of the type of objects, audio, and human detection and understanding; and outputs the probabilities of the possible social roles for the robot (*Receptionist*, *Tutor* and *Waiter*). The classification is based on a hybrid approach and trained and validated on a real-time multi-modal data-set collected by a mobile robot. The training experiment aimed to collect the data-set, to select the features that describe different roles and to calculate their weights. The validation experiments aimed to measure the performance and the generalization of the method. Results show that the robot was able to successfully classify the Receptionist role with an accuracy of 83.4%; the Tutor role with 82.7%; and finally, the Waiter role with 55.9%. On average, the method generalizes for 74% of unseen data.

List of Figures

List of Tables

Contents

1

Introduction

Human-Robot Interaction (HRI) is the field of study dedicated to understanding, designing and evaluating robotic systems for use by or with humans [32]. HRI is a challenging field of research at the intersection of psychology, cognitive science, social sciences, artificial intelligence, computer science, robotics, engineering, and human-computer interaction [22].

According to Dautenhahn et al. [22], much research in this field must be done, with a special focus on using robots in education, therapy, rehabilitation and supporting the elderly; and detecting and understanding human activity. Research has been done in sub-fields of HRI, such as education, assistive and service industry. These sub-fields have been targeted for experiments in which the robot has to act in a service role in a public place and performing predetermined tasks. We will refer to robots that perform tasks in these sub-fields as *social robots*.

For example, the robot of Kang et al. [34] had to assist cardiac patients in a health care center. Similarly, in the study of Krishnamurthy et al. [35], daily duties, such as carrying equipment and documents, were performed by a robot. It is possible to find various other examples of robots deployed in the health care field: robots for improving the quality of life of children in a hospital [38], or for elderly care [28]. In education, robots have been used for tutoring [16, 44, 48]. In the service industry robots have been programmed for being a receptionist [40] or a barista [36]. According to the International Federation of Robotics, the total number of professional service robots sold in 2017 rose considerably by 85% since 2016 and it is expected to increase between 2019-2021 [3].

It is notable that in all the cited examples, the robots were programmed for specific tasks, which were determined beforehand. The robot does not have to be aware of what is its surroundings, because this information is set in advance. Thus, using a single robot in multiple roles is unfeasible if it cannot determine the tasks it should perform, given its surroundings.

That leads to Situational Awareness (SA), which is the knowledge of what is going on around you [24]. SA can be decomposed in three distinct levels [24]. The first level, perception, states that perceiving cues is fundamental since, without this step, the odds of forming an incorrect picture of the situation increase dramatically. The second level is comprehension, in which integration of the information gathered at the first level happens for determining the goals of the agent. Finally, the last level is projection where the agent is able to use current events and dynamics for anticipating future events.

Up until now, studies have been performed on how to give and/or evaluate SA for a human in an HRI [23, 52] or for navigation planning of a robot [31], but not for a social robot. We will discuss the importance of achieving SA at level two for a social robot in the following section.

1.1. Motivation and related work

Different applications rely on the use of social robots. However, the role of the robot is set a priori meaning that the robot is not able to adapt itself to the possible different situations. Solutions to this problem consist of programmers, or employees, manually changing the role of the robot for adapting it to the specific situation. This is time-consuming and labour intensive, furthermore, it is hard to upfront establish when a role change can be done. This can be the case of the robot moving from a receptionist area to the canteen, it is not possible to change its role without prior planning. Alternatively, the programmer can also use location-methods for adapting the robot's behavior based on where it is. Yet, these solutions are not ideal since it is time-consuming to map the building and it is poor to generalize and/or replicate. If, for example, a room is used for meetings and sometimes for catering events, requiring thus different roles, the robot does not know which role is the most suitable one since the room might have been categorised for a specific role.

Awareness is the key to being able to properly react to external inputs, or more broadly, to what is happening around us. We believe that SA of level two applied to a social robot would be a contribution to the HRI field since it would allow the robot to decide which social role is the most suitable one based on the context of its surroundings. In this project, we are going to discuss how and if it is possible to obtain SA of level 2 for a social robot through scene classification.

As discussed in the previous section, for achieving SA, cues are needed. According to Endsley et al. [24], cues can be received through visual, acoustic, tactile, olfactory or taste receptors. Humans are able to quickly recognize a scene using visual and aural senses. Inspired by the human ability we will focus on detecting visual and acoustic cues. The others are not of significant influence on the project.

From these cues, we are interested in understanding two main pieces of information: *'Where'* and *'What'*. *Where* refers to understanding the environment in which you are in, while *What* expresses the context of your surroundings and what people around you are doing. We will refer to scene classification, the ability to integrate cues related to *Where* and *What* for achieving awareness.

Related works on scene classification using aural and visual cues are discussed in the following subsections.

1.1.1. Acoustic scene classification

Acoustic scene classification (ASC) refers to the task of classifying environments from the sounds they produce by associating a semantic label to an audio stream that identifies the environment in which it has been produced [18].

Sub-fields of the acoustic scene classification are the so-called computational auditory scene recognition (CASR) and the computational auditory scene analysis (CASA).

CASR refers to computational algorithms that attempt to automatically perform acoustic scene understanding by using signal processing and machine-learning methods. CASA refers to the computational analysis of an acoustic environment and the recognition of distinct sound events in it. Interpretation of individual events is the point that differentiates the scene recognition problem from the scene analysis.

Different auditory scene classifiers can be found in the literature [17, 20, 21, 25, 30, 33, 45]. However, the available classifiers based on acoustic input are not accurate enough for indoor scenes, thus not suitable for our project.

ASC represents an interesting problem that both humans and machines are only able to solve to a certain extent [18]. Different interpretations can derive from the task of semantic labeling of an acoustic scene or soundscape, as there is not a comprehensive taxonomy encompassing all the possible categories of environments. Generally, the problem is approached by researchers by defining a set

of categories, recording samples from these environments, and treating ASC as a supervised classification problem within a closed universe of possible classes. Furthermore, even within predefined categories, the set of acoustic events or qualities characterizing a certain environment is generally unbounded, making it difficult to derive rules that unambiguously map acoustic events or features to scenes[18].

1.1.2. Visual scene classification

Visual scene classification refers to the task of classifying environments from images or videos. Siagian et al. defined different methods for scene classification through visual information [49]. The first method is *Object-Based Scene Recognition* that bases the classification on the identification of a set of landmark objects known to be present in the scene. Espinace et al. [26] implemented a generative probabilistic hierarchical model that is able to identify indoor scenes by looking at objects such as doors and furniture. The recognized scenes were the following: *Living-Room, Dining-Room, Bedroom, Kitchen, Bathroom*. The authors were able to define discriminatory objects which were able to describe the scene.

The second method defined by Siagian et al.[49], is the *Region-Based Scene Recognition* that, instead of using objects, segments images into regions. Their configurational relationships, then, form a signature of a location. Early approaches of this methodology were breaking the input image into local blocks or patches for analysing their local features. The block or patch is processed by the classifier and then a voting strategy combines the results of every block or patch [50]. Other approaches used a mixture of probabilistic classifiers [43] instead of a voting strategy. Unfortunately, these methods suffer from generalization capabilities [26].

An alternative of the previous method is based on image segmentation for the identification of local image regions, such as vegetation or sky, through geometrical features called eigenregions [29]. For labelling each segment region, classifiers are used. A drawback of this method is related to the poor performance of segmentation algorithms, which problem is particularly relevant for the case of indoor scenes, where the presence of a large number of objects usually produces scenes with significant clutter that are difficult to segment [26].

The third method defined by Siagian et al.[49], is the *Context-Based Scene Recognition* that considers the input image as a whole and extract a low-dimensional signature that compactly summarizes the image's statistics and/or semantics. Early approaches of this method used colour/texture properties of the image, for instance, a forest scene presents highly textured regions (trees), a mountain scene is described by an important amount of blue (sky) and white (snow), or the presence of straight horizontal and vertical edges denote an urban scene [19].

Ulrich et al. [51] used color histograms as the image signature and a k-nearest neighbor scheme for classification. The method was meant for the topological localization of an indoor mobile robot. Unfortunately, the method has to be trained again for each specific indoor environment. Poor generalization beyond training sets is one of the problems for holistic methods based on global image features [26].

More robust approaches use semantic representations. Farinella et al. [27] developed a scene descriptor based on the statistics of the discrete cosine transform coefficients. The method was intended for limited memory and low computational resources devices. The scene classification was able to classify the scenes described in the MIT-67 data-set [47] that contains 67 different indoor scenes. Siagian et al. [49] were able to differentiate outdoor scenes using a multi-scale set of early-visual features, which capture the "gist" of the scene into a low-dimensional signature vector.

Madokoro et al. [37] used an autonomous robot for gaining data for indoor place recognition. The robot was able to walk in a corridor lab that was split into four parts that represented different scenes. The classification is performed by using an unsupervised scene classifier based on the context of features for semantic recognition. The method aims to represent spatial relationships among categories for mapping neighborhood units on category maps.

The main drawback of a visual scene classification method is that it is limited by the environment features. In an everyday situation, spaces can be used for different purposes, for example, people sitting at the canteen at university might having their meal or having a meeting, thus the contexts are different. The methods just cited are not able to differentiate the two different contexts.

In addition, Region- and Context-based methods might lack poor generalizations beyond training sets and require time for creating the training and validation sets. Furthermore, we cannot use existing data-sets for indoor scene recognition due to the diversity of view between a picture taken with a camera by a human and a picture taken by the embedded camera of a robot. Usually, robots do not have high resolution nor a wide field of view cameras. Not disposing of a high-resolution camera might give problems with recognizing objects, while not disposing of a wide field of view camera leads to the inability of capturing the whole meaning of the surroundings. The problem could be solved by reconstructing the surroundings by using multiple pictures, but this would be processing intensive.

To overcome the just-mentioned problems, we believe that using an Object-based method would be beneficial to our project since it will allow us to use knowledge-based assumptions and thus avoid the need for a huge amount of data which is one of the biggest problems for data-driven approaches.

1.1.3. Combined scene classification

Other approaches for scene classification have been done by fusing visual and acoustic data.

O'Connor et al. [41] proved that a multi-modal approach performs better than uni-modal by using a machine learning technique to automatically classify YouTube videos for social events detection. Events such as interviews, parties, weddings, and sporting events were detected by analyzing features, including color, brightness, volume, and silence.

As discussed before, in scene classification on a robot, the limitations come from the sensors that the robot dispose of. Nigam et al. [39] developed a context-based perception method for scene classification to be used by a 'home-made' mobile robot. The robot has been used for collecting real-world, multi-modal data after interacting with the human for appropriateness interaction detection from multi-use locations inside the campus library. The spaces where data was collected were used for different activities, that they classified into *Study, Lobby, Dining*.

The multi-modal and uni-modal classifications have been achieved through machine learning algorithms. Classifiers were trained using GIST features [42] for visual data. The scene is, in this way, seen as a representation of dimensions, such as naturalness, openness, roughness, expansion, and ruggedness to capture the spatial structure. While acoustic features used were volume mean, volume standard deviation, silence ratio, frequency centroid, frequency bandwidth, and energy. Together with these two features, the authors used the results from the interaction with the human that reflected the appropriateness for being bothered by the robot.

The work of Nigam et al. [39] is the closest project to our idea. However, in their work, the collection of data happened after interacting with the human being, while we believe that understanding the surroundings has to happen before interaction happens. Furthermore, they used a context-based method with poor generalization beyond training sets, as discussed before, and they used training and validation data from the same environments. The data-set is not available for being used for further researches and the 'home-made' style of the robot makes reproducibility hard to achieve. As a last remark, the work does not consider human detection and activity but rather uses appropriateness of interaction as a metric for understanding the context. We will further discuss why human detection and understanding play a key role in scene classification in the next sub-section.

1.1.4. Detecting and understanding human activity

In the previous sections, we have discussed how it is possible to analyse the environment through acoustic and visual cues. Most of the works cited are not able to differentiate scenes based on human activity.

For example, for a tutor robot, if we take a picture of an empty and of a crowded classroom, most of the existing technology might classify the two pictures in the same way since based on the environment. No additional information is gathered about what is happening in that scene. In the case of a social robot, the classification should instead give different outputs because in the first case (empty classroom) no specific action is expected by the robot. While in the other case, the algorithm should detect that a social role is needed.

Furthermore, the same location can be sometimes used for different purposes, as we discussed in the previous sections. Nowadays algorithms are not able to differentiate the different social contexts since the information about what people are doing is missing. For example, the university canteen spaces might be used for having a meal but also for studying or having meetings. Thus the scene classification should detect which particular context the robot is seeing and then react as a consequence.

For these reasons, we believe that cues about humans are necessary for obtaining awareness for a social robot. Features such as the number of people in the environment, their facial expression, and their body language can be used for augmenting the visual and acoustic cues we discussed in the previous sections for better understanding the context. For example, the work of Pereira et al. [46] showed that with a gesture recognition method, the robot is able to properly react by providing the most suitable way of helping the person.

2

Research Question

As discussed in the previous chapter, the increment of the use of social robots and the lack of nowadays technology in the deployment of a single robot for different applications without the help of a human being, lead us to explore a young research topic. The current solutions to this problem are time-consuming, labour intensive and hard to generalize. The problem could be solved by applying Situational Awareness of the second level through scene classification to a social robot. Being aware of its surroundings, in terms of environment and context, the robot can set its goal (the social role it has to play).

In related researches, studies have focused mainly on classifying the environment of a scene by processing images excluding the information related to the context. Most of the related works in visual scene classification used the MIT-67 [47] indoor scene data-set, that does not contain people in any picture. However, human detection and understanding is fundamental in the understanding of the context, in particular for robots that interact with people in different ways.

Other than the missing human activity information in the visual scene classification methods, the context- or region-based algorithms do not generalize to unseen data outside the training data-sets and no data-set for indoor scene classification for robots is available. Thus the need for creating one in order to use one of the two methods arises with the known drawback of being a time-consuming task.

Instead of processing images for obtaining gist features as in the related works, we will use object-based methods. Using a list of objects that a scene can contain will allow us to start from knowledge-based assumptions of the environment. This has two main advantages, the first one is related to limitations lead from the robot's sensor: recognizing objects does not need a wide view of field camera nor intensive processing for reconstructing the entire scene from different pictures. The second advantage is related to time and cost constraints: the method does not need to collect a big data-set for training. With an object-based method, we can use an open-source library for object recognition and easily detect objects, saving us time.

Furthermore, using multi-modal (visual and acoustic) data has been proven to be more accurate for classifying the environment of a scene. Only the work of Nigam et al. [39] is considered relevant in this field. However, we believe that classification should happen before interaction.

The aim of our project is then to define and classify scenes for a social robot through a list of cues gained through an object-based method, audio, human detection and activity.

This leads us to the following research question:

What is the performance of a multi-modal, object-based scene classification using a knowledge-based hybrid model for the classification, and how does the classification generalize towards unseen scenes?

With hybrid, the combination of symbolic, probabilistic and pattern recognition approaches is meant.
We believe that our newest scene classification method could lead the robot to be aware of its surroundings and thus autonomously detecting that a change of settings is required for the scene.

We will refer to the cues as features, to the specific sets of tasks that the robot should perform in a role as settings, whilst the role is based on the overarching social context of the robot's surroundings, such as receptionist or tutor, which we aim to classify.

In order to be able to answer the research question, it is needed to investigate the following sub-questions:

What features are distinctive of different typically indoor scenes?
How to develop a data-set from a first-person perspective?

The latter question arises from the fact that a first-person data-set for an indoor scene is not available.

2.1. Hypotheses

We will try to prove or disprove the following hypotheses:

H1: It is possible to define feature sets that are sufficiently distinct to classify scenes.

H2: Multi-modal knowledge-based hybrid scene classification generalizes to unseen data and is able to correctly classify scenes after manually setting the probabilities.

3

Method

The aim of this thesis project is to try to model different scenes by processing visual and audio information gathered by a mobile robot. We discussed in Chapter 1 that a scene refers to the surroundings and the context. Furthermore, in Chapter 2, we introduced the concepts of role, setting, and feature. The two definitions are linked by the fact that by classifying the scene, we output the role that the robot should play in that moment.

The sets of roles and settings to be described through features are further discussed in Section 3.1. The classifier is then described in Chapter 4. In order to classify the scene, training and validation phases need to happen. The training phase allows us to refine the set of features we want to use and their weights for the classifier. After that, the validation phase measures the performance of the classifier. These two phases will be discussed in Chapter 6 and in Chapter 7. The conclusions are then discussed in Chapter 8.

3.1. Roles and settings

Different roles and settings have been chosen inspired by the applications for social robots existing nowadays. They have been listed below.

- *Receptionist*, the robot is expected to welcome people, check and make appointments, giving directions.
  The settings belonging to the receptionist role are the following:
  - *Entrance building*, the robot stands at the entrance of the building waiting. The interaction may be started by the human or by the robot if it detects that someone needs help.
  - *Info desk*, the robot is expected to be at the info desk and wait for interaction
- *Tutor*, in this role, the robot mainly has the task to provide information.
  The settings for the tutor role are:
  - *Lecture*, the robot gives a lecture by explaining a specific topic.
  - *Meeting*, the robot takes part in a meeting by actively presenting data or it can be asked to search for information.
- *Waiter*, the robot is expected to act as a waiter and thus welcome clients, ask for orders, take care of the clients.
  The following settings are part of the restaurant service role:
  - *Busy restaurant*, this setting is meant to represent the restaurant activities during open hours. The robot then welcomes customers, take their orders and generally take care of them.
  - *Empty restaurant*, this setting is meant to represent the activities that take place while the restaurant is close to the public. These can include preparing the tables, cleaning, matching the tables with the reservations.

3.2. Features

In the following section, assumptions for defining the sets of features are explained. The final set of features is derived after training, discussed in Chapter 6.

3.2.1. Questionnaire on how to describe a scene

How to describe a scene is the first challenge faced in the project. From Chapter 1 and Chapter 2, we assumed that a scene is described by a set of features of the following types: object, audio, and human detection and understanding. For making our assumption stronger, an anonymous and online questionnaire for describing different scenes through features was distributed. The questionnaire aimed to collect information about which features humans are looking to when classifying scenes.

Eleven master students were asked to describe the following situations that correspond to the settings introduced in Section 3.1 : *Building entrance, Info desk, Lecture, Meeting, Empty restaurant, Busy restaurant*. More specifically, the request was to describe the scene from a point of view of a receptionist, tutor, and waiter. For not biasing their answers, no images where shown, but rather an example was made about a nurse in a hospital. Responses from the questionnaires have been sorted and analysed.

Examples of features in the responses were object cues, like 'Laptop', 'Paper', 'Pen', 'Door'. The participant used cues linked to human detection and understanding type such as 'People sitting around a table', 'People walking' or 'Conversation'. Also, audio cues were used in describing scenes, 'Noisy' or 'People talking' are examples of them. Finally, features that are not part of any of the previous types were used, like 'Black dress', 'Orders', 'Customers' or 'Students'. These features cannot be included in the other categories since trying to explain a concept more complex than stating an object.

Further documentation on the questionnaire can be found in appendix A.

3.2.2. Features derived by observations and the results of the questionnaire

Next to the questionnaires, features were studied by doing observations. Different buildings were visited for annotating features. Of these buildings, pictures were taken and shown in Appendix B.

Combining the observations of the different roles and settings with the data collected in the questionnaires, a list features has been created. The list of determined features sorted by role and feature can be found in Table 3.1. This is not the final set of features we aim to use for the classification, but rather a first approach to the problem on how to describe the settings.

Role	Settings	Features
Receptionist	Building entrance	Building entrance door Couch, sofa Chairs People entering from the entrance door People looking around People passing by Low level of noise (average) 1:1 Communication
	Info desk	Info desk Building entrance door Laptop, computer desk Paper, pen Couch, sofa Chairs People entering from the entrance door People passing by Low level of noise (average)

		1:1 Communication
Tutor	Lecture	Lecture room or Auditorium Chairs Desks Papers, Pens Presence of people People sitting People looking at the robot Projector Monitor Laptops Low level of noise 1:N Communication
	Meeting	Desks, Table Chairs Laptops Papers, Pens Presence of people People sitting People talking 1:N Communication High level of noise
Waiter	Busy restaurant	High level of noise Food Chairs, Tables Glasses Dishes, Cutlery, Tableware Cooker, Fridge, Oven Cash register Shelves Presence of people People passing by People entering from the main door People talking People sitting
	Empty restaurant	Low level of noise Chairs, Tables Glasses Dishes, Cutlery, Tableware Food Cooker, Fridge, Oven Cash register Shelves Low number of people present (<5)

Table 3.1: List of the features for each role and setting that characterizes the context and the surroundings

3.2.3. Features ranking

Features of different settings have been listed in Table 3.1, but for the purposes of this thesis, a deeper analysis is required. It is notable that some features appear in multiple settings, we might consider them as not discriminative because detecting them will not allow concluding with a high enough probability that we are in a specific role or setting. Further analysis of the features based on how they discriminate

between different scenes must be done.

Furthermore, the technology needed to obtain the relevant data has not yet been discussed. However, the available technology does introduce a limiting factor in our research. There might be some features that even though they are discriminative, are difficult to detect or to implement.

What follows is a study on the specificity and difficulty of each feature. Using these two measures for obtaining an overall rank on the feature will give us a first indication of how to proceed in the implementation.

The specificity of the feature is calculated as the number of the settings that use the feature over the total amount of settings:

$$Feature_{Specificity} = \frac{\#Settings}{tot_{Settings}} \tag{3.1}$$

The specificity can assume a value in [0.17, 6]. The lowest the specificity is, the more specific a feature is.

The difficulty can assume a discrete value in the interval [1,3] based on the availability of application programming interfaces (APIs) or library as described below:

$$Difficulty = \begin{cases} 1 & \text{API or library available} \\ 2 & \text{Semi implemented} \\ 3 & \text{To be implemented} \end{cases} \tag{3.2}$$

In Equation 3.2 we refer with *Semi-implemented*, all the libraries or APIs that require additional work to be adapted to our project, while with *To be implemented* the ones that have to be implemented in order to detect the specific feature.

Finally, the rank is calculated as described in Equation 3.3. The higher the rank is, the easier to detect and the more specific the feature is. The rank can be a value in the interval [0.33, 6], where the maximum is achievable when the difficulty is 1 and the specificity is 0.17; while the minimum is when the difficulty is 3 and the specificity is 1.

$$Total = \frac{1}{Difficulty} * \frac{1}{Feature_{Specificity}} = \frac{1}{Difficulty} * \frac{tot_{Settings}}{\#Settings} \tag{3.3}$$

All the features can now be ordered by decreasing *Total* ranking. Results are shown in Table 3.2.

Feature	Settings specificity	Difficulty	Total
Low number of people present	0.17	1	6
People looking at the robot	0.17	1	6
Projector	0.17	1	6
Monitor	0.17	1	6
Lecture room / auditorium	0.17	1	6
Couch, sofa	0.33	1	3
Food	0.33	1	3
People talking	0.33	1	3
High level of noise	0.33	1	3
Glasses	0.33	1	3
Dishes, Cutlery, Tableware	0.33	1	3
Fridge, Cooker, Oven	0.33	1	3
Cash register	0.33	1	3
Shelves	0.33	1	3
People looking around	0.17	2	2
People passing by	0.5	1	2
Desks	0.5	1	2
Laptop, Computer	0.5	1	2
Monitor	0.5	1	2
Table	0.5	1	2
Papers, Pens	0.5	1	2
Presence of people	0.5	1	2
People sitting	0.5	1	2
Low level of noise	0.67	1	1.5
Entrance door	0.33	3	1
Chair	1	1	1
People entering	0.5	1	1
1:1 Communication	0.33	3	1
1:N interaction	0.33	3	1

Table 3.2: Features ordered by decreasing *Total* ranking value.

With this analysis, it is possible to filter out some features listed in Table 3.2. Features in Table 3.2 that have a *Total* equal to 1 have been discarded since implementing an algorithm for detecting them would have been time requiring, or because not specific enough. Furthermore, 'People looking around', even if a valuable sign of the receptionist role, it has been discarded due to time constraints. In Chapter 6, further study on the features in terms of discrimination and frequency for each setting is done. Answering in this way one of the sub research questions stated in Chapter 2.

4

Tools

The tools used in the project are described in this chapter. Firstly, the chosen robot is described in Section 4.1. This is followed by Section 4.2 about the used APIs. Finally the Section 4.3 explains the Bayesian approach used for the classification.

4.1. Pepper Robot

One of the most popular interactive robots nowadays is Pepper from Softbank Robotics [13]. With its human-like shape and the set of sensors of which it is equipped, it makes eligible to be used in social roles. Pepper is frequently used in social roles, such as tutor [15], ice-cream seller [10], waiter [12] or receptionist [11].

Its height (121cm) and the two 2D cameras together with the 4 microphones located on the head of Pepper make it a valuable tool for us. The height allows us to take pictures and detect sounds from a human-like point of view. That is important since all the features have been thought through knowledge based on human perception, and suitable for collecting the data-set from a first-person perspective.

Furthermore, Pepper can autonomously move by using the three wheels at the base of it. For further information on Pepper and its sensors can be found in appendix C.

Figure 4.1: Picture of the Pepper Robot [14]

4.2. APIs

As discussed in the section *Features ranking*, APIs that allow us to detect the features listed in Table 3.1 were searched. The chosen APIs are discussed in details in the coming sub sections.

4.2.1. Cloud Vision API

The Goggle Cloud's Vision API allows developers to easily integrate vision detection features within applications, including image labelling, face and landmark detection, optical character recognition (OCR), and tagging of explicit content [4]. The API uses pre-trained machine learning models for analyzing the images and classify them into millions of predefined categories. Even though the API offers different analysis for images, in this project the labelling and object recognition functionalities are used since they allow to detect some of the features listed in Table 3.1.

Labels can identify general objects, locations, activities, animal species, products, and more [5]. With the labels, it is also possible to detect some human activity such as *Sitting, Eating, Conversation, Walking* that are useful for our project.

Object localization identifies multiple objects in an image and provides a *LocalizedObjectAnnotation* for each object in the image [6]. Each *LocalizedObjectAnnotation* identifies information about the object, the position of the object, and rectangular bounds for the region of the image that contains the object.

An example of label and object localization can be found in Figure 4.2 and in Figure 4.3. In Figure 4.2, it is shown that through labels it is possible to detect 'Eating', 'Food', 'Dish', 'Table'. While in Figure 4.3, through objects it is possible to detect 'Table', 'Person' and 'Tableware'.

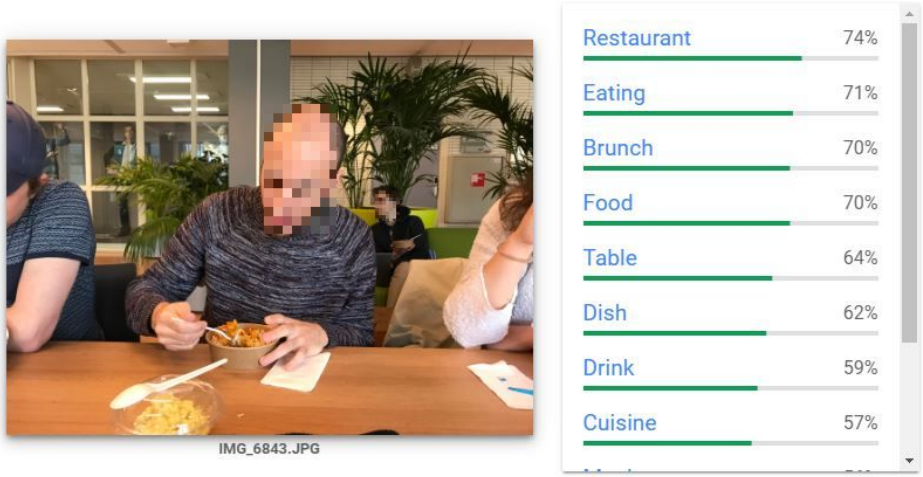


Figure 4.2: Example of output with the label detection by Google Cloud's vision API

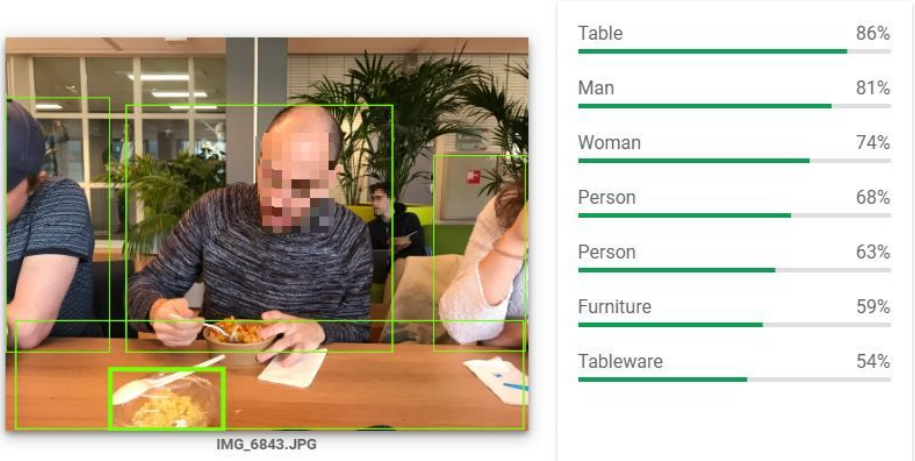


Figure 4.3: Example of output with the object localization by Google Cloud's vision API

Every time the API is used, it sends back a JSON file containing all the information relative to the labels and objects recognized. Information contains the *Title* and the *Confidence*. The information is then analyzed for extracting the features interested.

4.2.2. NAOqi APIs

Together with the robot, Softbank Robotics provides a set of APIs [7]. The APIs are thought to facilitate programmers in actuating, sensing and commanding the robot. In fact, the APIs are grouped in the following categories: *Core*, for core functionalities such as memory, *Interaction engines* for basic interactive and alive behavior, *Motion*, *Audio*, *Vision*, *People Perception*, and *Sensors and LEDs*.

We will mainly use APIs from the *Audio*, *Vision*, *People Perception* categories.

Audio APIs

Basing our implementation on the features of Table 3.1, the two main audio features we are interested in are the level of noise and the speech detection.

The speech detection, the *ALSpeechRecognition API*, sub-API of the APIs belonging to the audio category, raises an event every time the speech recognition engine has detected a voice activity. Unfortunately, this API did not work, so we were not able to use this information. We rather used the labelling from Google for detecting *Converstion*.

While for the level of noise the *ALAudioDevice API* is used. This module is able to calculate the energy of the sound detected from the microphones by processing a 170ms buffer for each microphone.

People Perception APIs

The big challenge of this project is being able to understand the context based on cues on human detection and activity. Softbank Robotics already provides APIs for people perception. The cues are mainly detected through cameras and lasers

The *ALGazeAnalysis* module allows analyzing the direction of the gaze of a detected person, in order to know if he/she is looking at the robot [8]. It also detects whether the person's eyes are open or closed. The module raises an event every time the list of people looking at the robot changes.

The *ALPeoplePerception* module is an extractor which keeps track of the people around the robot and provides basic information about them [9]. The robot maintains in memory a list of people seen in the environment. The module raises an event time the list of persons in the current population changes by adding or removing one or more persons. The associated data is a list of people IDs (integers) in the current population of visible and not visible people. A person is added if seen in the camera, while a person is removed if not seen for a timeout of 60s.

Furthermore, during testing, Google Cloud Vision API was able to label someones gesture such as *Conversation*, *Sitting*, *Walking* and *Eating*, as explained in the previous subsection. These have been then used for People Perception.

Interaction engines API

In the *Interaction engines* APIs, the *ALAUtonomousLife* module keeps the robot alive through different abilities, for example, blinking the eyes. One of the abilities is *ALBasicAwareness* that allow the robot to establish and keep eye contact with people. When the module is enabled, the robot can process stimuli coming from its surrounding environment. The stimuli can be a human detected by the camera, a perceived sound, touch and movement. The module has been used for maintaining track of the people in the environment.

4.3. Bayesian Network

Once the features are detected, they need to be analysed in order to estimate the role the robot is in. That is done by using a *Bayesian Network* (BN). Before proceeding to a detailed explanation of this step, it is necessary to first introduce a few concepts.

4.3.1. Features as observations

It is quite clear that in order to estimate, statistics is necessary. The probability of being in the *Receptionist* role, for example, is calculated over the features detected. A well known probabilistic tool is used,

the *Bayes' Theorem*, that states the probability of an event based on prior knowledge of conditions that might be related to the event.

$$P(A|B) = \frac{P(A,B)}{P(B)} \tag{4.1}$$

Equation 4 explains Bayes' Theorem: the probability of A knowing that B happened is the conjunct probability over the probability of B. Now, in the specific case we are studying, we are interested in the probability to be in a role, after observing some features, then we can treat the probability of the latter as in a binary sense: either they are true or they are false. What we are interested in, is if the feature is present in the environment, so, for example, when we are looking for the feature *Desk*, that can be translated to ' *A desk is present in the environment* '. The concept is quite straightforward, either a desk is present or it is not and that comes from direct observation. Thus, the probability of the feature over time can only be either 1 or 0. But since the vision module returns a certainty value after detecting objects, this value is used for determining if effectively the feature has been detected. That is simply done by checking a probability threshold that is set at 0.5 as shown below.

```
if certainty >= threshold:
    update_feature_value
else:
    pass
```

4.3.2. Directed acyclic graph - DAG

A Bayesian Network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG) [2].

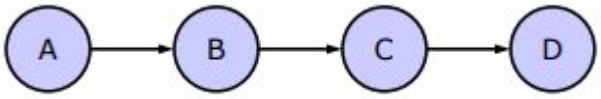


Figure 4.4: Simple example of a Bayesian Network

In Figure 4.4 a simple example of Bayesian Network, where the arrows represent the conditional probability and nodes represent the variables. Every node has associated a probabilistic table that expresses the probabilities of the variable based on the connected nodes.

A	P(B=True)	P(B=False)
True	0.1	0.9
False	0.8	0.2

Table 4.1: Example of probabilities table for node B of Figure 4.4

Table 4.1 contains the information about the values of the conditional probability that exists between A and B. Usually, Bayesian Networks are used for calculating the probability of the initial node, in the example node A, after observing some events that can be B, C or D. The case study is different from this, we want to start from A and get the probability of D. For that Bayesian inference is used.

4.3.3. Bayesian inference

Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available [1].

Going back to Figure 4.4, it is now possible calculate *P(d)* with Equation 4.3.

$$\begin{aligned} P(d) &= \sum_{ABC} P(a,b,c,d) = \sum_{ABC} P(d|c)P(c|b)P(b|a)P(a) = \\ &= \sum_{C} P(d|c) \sum_{B} P(c|b) \sum_{A} P(b|a)P(a) \end{aligned} \tag{4.2}$$

If we want to calculate the probability of d after an evidence of a, the formula is the following:

$$P(d|a) = \sum_{BC} P(a,b,c,d) = \sum_{BC} P(d|c)P(c|b)P(b|a)P(a) = \\ = \sum_{C} P(d|c) \sum_{B} P(c|b)P(b|a)P(a) \tag{4.3}$$

5

Implementation

The tools used in the project were explained in the previous chapter. However, the description of how the tools work with each other is missing. This chapter aims to explore how the tools work alone and together. The algorithm developed is made by three main parts: the Bayesian Network, the Visual Module and the Pepper code.

5.1. Bayesian Network, the use case

In the study case, it is possible to identify 3 different layers of nodes. As it is shown in Figure 5.1, the first layer of nodes is made by the features, right after that there is the settings layer and finally the roles layer.

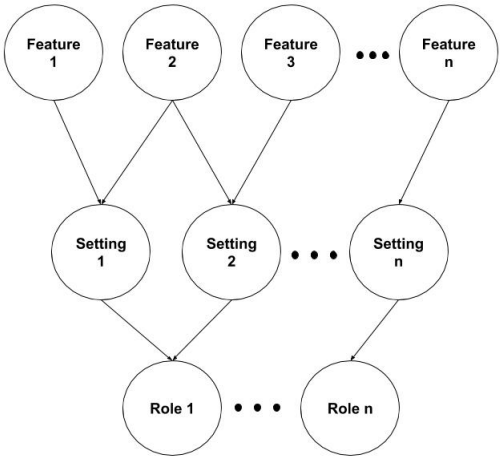


Figure 5.1: The use case BN is composed by three different layers: Features, Settings and Roles

The nodes that are part of the Feature layer have a simple table with only two columns determining the probability of the feature to be true or false. Going further in the network, the probability table increases in size leading to more complexity since it is proportional to

$$2^{numberfeatures}$$

. Using Equation 4.3 it is possible to calculate the probability of a role.

$$P(role) = \sum_{settings} P(role|setting) \sum_{features} P(setting|feature)P(features) \quad (5.1)$$

Since we assume that *P(features)* is 1, the Equation 5.1 can be written as

$$P(role) = \sum_{settings} P(role|setting) \sum_{features} P(setting|features) \tag{5.2}$$

Finally, we can express the probability of a role given a set of features as:

$$P(role|features) = \sum_{settings} P(role|setting)P(setting|features) \tag{5.3}$$

5.2. Algorithm

5.2.1. Bayesian Network

The Bayesian Network (BN) model and code have been entirely developed by us. Nodes are described using JSON files. An example of the feature, setting and role node can be found as follow:

```
{
"features_nodes": [
    {
      "id": 1,
      "title": "Door",
      "value" : "0",
      "probability" : "0.5",
      "link_to_settings_nodes_id": [
        "1",
        "2"
      ],
      "type" : "label-object",
    },
}
```

Listing 5.1: Example of feature node

A feature node is described through its ID, its title, the value that represent if the feature has been detected or not (0=False while 1=True), the probability threshold, the setting nodes that it is linked to, and how it is detected (label-object, audio or robot)

```
{
"settings_nodes": [
    {
      "id": 1,
      "title": "Building entrance",
      "link_to_roles_node_id": "1",
      "table": {
      }
    },
}
```

Listing 5.2: Example of setting node

The setting node is described by its own ID, the title, the link to the role nodes and the probabilities table.

Similarly to the setting nodes, the role nodes have the same objects with the only exception that it is not linked to any other node of the network. More information about the nodes can be found in appendix D.

The probabilities tables used for the setting and role nodes are stored in excel files and then loaded in the table section when the algorithm starts. An example of the probabilities table can be found in Figure 5.2.

	A	B	C	D	E	F	G	H	I	J
1	Person present	Low level of noise	Table	Desk	Event	Sitting	Laptop Computer	Projector Projection	TRUE	FALSE
199	1	1	0	0	0	1	0	1	0,51724138	0,48275862
200	1	1	0	0	0	1	1	0	0,62068966	0,37931034
201	1	1	0	0	0	1	1	1	0,68965517	0,31034483
202	1	1	0	0	1	0	0	0	0,51724138	0,48275862
203	1	1	0	0	1	0	0	1	0,5862069	0,4137931
204	1	1	0	0	1	0	1	0	0,68965517	0,31034483

Figure 5.2: Example of a probabilities table

Other than the JSON and Excel files, we developed the computational part of the BN in Python. The file loads the nodes and the tables. It is possible then to update the features and calculate the output of the BN.

5.2.2. Vision Module

For the Google Cloud Vision API, a dedicated module was used. The module is written in Python and mainly uses the open-source code that Google provides for the API. By giving the path of the images, the module sends the request to Google and returns a list of objects and labels. These lists are then checked and features are sent to the BN for updating.

5.2.3. Data sampling and scene classification on Pepper

Finally, the last main part of the algorithm is the part related to the robot. This module, written in Python, is responsible to connect to the robot and subscribe for events that we discussed in Section 4.2.2. On top of that, the module is responsible for collecting data, listening to events, requesting the objects and labels, and requesting the output of the BN. Indeed the module has an instance of the BN, so every time an event has triggered or labels/objects are received, it is possible to update the feature nodes by using the update feature function. The algorithm samples the output of the Bayesian Network every 5s, by calling the get probabilities function.

Once started, the algorithm collects data about the surroundings by using visual and audio features. It starts with the robot taking a picture every 45° in order to analyse the complete environment. Once all the needed pictures are taken, the visual module is requested to processes them while the robot starts analysing the audio signals captured by the front microphone. In total, 30s of an audio signal is examined. In the meantime, the NAOqi APIs are always active for detecting events. Every time a feature has been detected is updated in the Bayesian Network.

After taking the pictures, recording the audio and analysing both of them, the first output of the BN is calculated. Sequentially, the output is calculated every 5s up until 20 samples are taken. Then a new test starts over by taking pictures and recording audio and so on. The flow chart of the algorithm is shown in Figure 5.3.

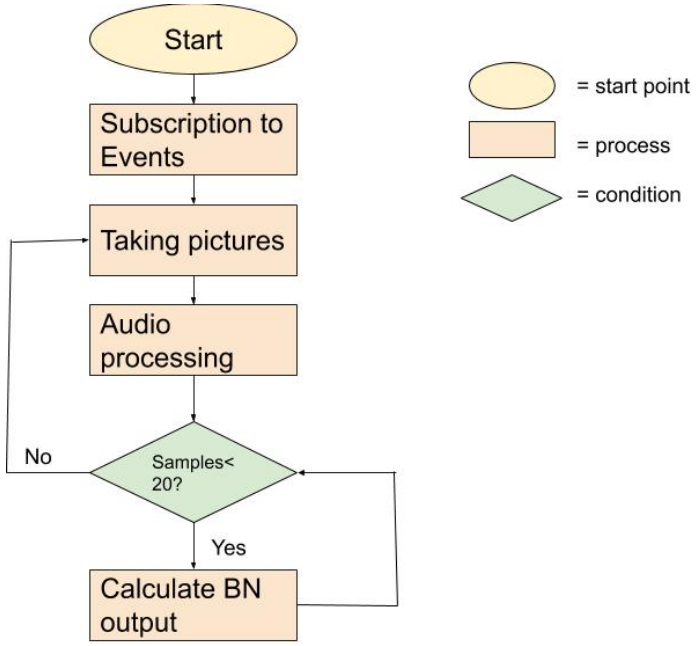


Figure 5.3: Flow chart

Furthermore, the algorithm can be stopped any time by pressing the tactile sensor on the right hand of the robot or can be started over by pressing the tactile sensor on the left arm. Information about the tactile sensors can be found in Appendix C.

5.3. Storing data

While the algorithm runs, data is collected. Pictures are taken and stored in the laptop that is connected to the robot, the same happens for the audio. Furthermore, at every experiment, features vectors are stored for offline simulation. The feature vector is a vector containing the value of the features at a particular moment in time. After every experiment, an Excel document is created containing all the instances of the features vectors seen. In Figure 5.4 is possible to see an example of 7 features vectors, one per each row.

Sofa Couch	People passing by	Low level of noise	Desk	Laptop Computer	People looking at the robot	Projector Projection	High level of noise	Food	Table
0	1	1	0	0	0	0	0	1	1
0	1	1	0	0	0	0	0	1	1
0	1	1	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1

Figure 5.4: Example of feature vectors with

5.4. Offline simulation

Having the features vectors for every experiment makes offline simulation possible. By giving as input the feature vector, the output of the BN is calculated. This is particularly important during training since, in this step, the feature sets will be formed for every setting, and their respective probability calculated. This process requires to run the algorithm different times for checking the output of the BN after every change has happened. Further explanation of the training procedure can be found in the next chapter.

6

Features selection and training

The first experiments were meant for collecting data in order to train the Bayesian Network. The procedure followed is shown in Figure 6.1. Data is collected as explained in Section 6.2 and further analysed in Section 6.3. After the analysis, the final set of feature for every setting is derived as described in Section 6.4. Finally, the weights of the classifier have been tuned until saturation of the results.

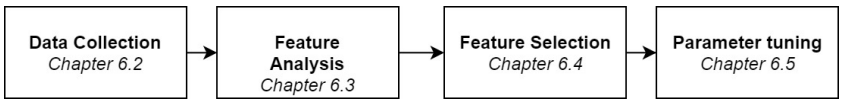


Figure 6.1: Classifier training process

Since while testing, data in which people can be recognized might be collected, a consent form and experiment information form have been used for collecting people's consensus. The two forms can be found in Appendix E. In the following paragraphs, we will discuss how we run the experiments, the challenges we faced during them and the results.

6.1. Challenges

The first limitations due to constraints of the embedded sensors of the robot were introduced in the previous chapters. As anticipated before, the low resolution and poor field of view placed the first challenges in the deployment of our methodology. Figure 6.2 and in Figure 6.3 show the differences between a picture taken with a smartphone and the pictures taken from the robot. Due to the poor resolution and the sensitivity to light of the robot's cameras, some features were not detected, even though present in the environment. Those features were mainly papers and dishes. Due to their white color, they were not detected on the table which color is white as well.

Figure 6.2: Picture taken from a camera (12Mega-pixel resolution).

Figure 6.3: Pictures taken from the first and second cameras of Pepper

6.2. Data collection and setup

The training experiments have been done in the RoboValley building located at Julianalaan 67A, 2628BC Delft. Experiments were run in the entrance/receptionist area, a room used for meeting or presentations and the canteen.

Experiment refers to data collection in a certain setting. During the experiments, the robot was placed in different locations and a test was run. These locations were beforehand selected with the reasoning of where the robot could be while carrying out its role. For example, the robot in a tutor role would be positioned in the position of the person giving the presentation.

Every test started by placing the robot in the desired location in the environment, then the algorithm was started and let running, but stopped after 20 samples. After that, the robot was put into *resting mode* and moved to the next location where the algorithm was started again. Further explanations and images on how the experiments have been done can be found in Appendix F,

6.3. Feature analysis

In this section, the data collected during the training experiments is analysed. The data is in the form of feature vectors, images and audio.

Feature vectors were used for analysing the detected features. Since the vectors collected in a

single test were similar to each other, we decided to select the most representative one. The most representative vector contains the features that occurred the most. An example on how to select the most representative feature vector is shown in Figure 6.4.

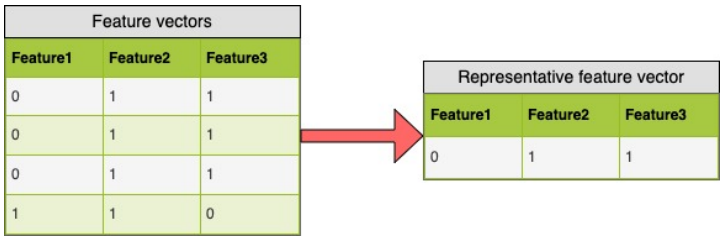


Figure 6.4: Example of the selection of the most representative feature vector form the collected ones

The images have been manually analysed with the Vision API for checking cases of wrong detection or labelling, that will be referred to as invalid features. At the same time, the analysis aimed to detect possible new features that were not listed in Table 3.1, but could be of additional value. The audio data was not further analysed, since its result was already contained in the feature vectors.

The invalid features have been removed since using them in the training could affect the behaviour of the classifier and it is not known beforehand how they would affect the classification of unseen data. The newly added features give the possibility of having a better final selection of features, which could positively influence the performance of the classifier.

The feature vectors collected from the experiments were analysed. Since the vectors were similar to each other, we decided to select the most representative one for every test for further analysis. The most representative vector contains the features that occurred the most.

Using the features and the amount of times they appear in the feature vectors have been listed as shown in Tables 6.1, 6.2 and 6.3.

Furthermore, different invalid features were detected due to a wrong classification from the Google Vision API, as said before, or due to poor performance of the NAOqi APIs. The invalid features will be discussed further in the following subsections.

6.3.1. Receptionist

Feature	#Occurrences
Door	6
Table	3
Room	6
Projector	1
People passing by	1
Person present	5
Low number of people	1
Low level of noise	4
High level of noise	2
Walking	1

Feature	#Occurrences
Desk	2
Laptop, Computer	1
Room	2
Sofa, Couch	2
Table	2
Person present	1
Sitting	1
Office chair	2
Low level of noise	2

Table 6.1: Features detected and amount of times they appear in the feature vectors during the Building Entrance (on the left, with a total of **6 tests**) and the Info Desk (on the right, with a total of **2 tests**) setting experiments

In Table 6.1 the features detected for the Building Entrance (left table) and for the Info Desk (on the right) are listed. From the experiments related to the Building Entrance, the feature 'People passing by' was expected to be occurred more times. An observation must be done at this point. During the experiments, only a few people passed by the hall. However, the 'Person present' feature is almost always 'True' during the experiments. This is a case of an invalid feature because what is triggering it

is the presence of another Pepper robot that is labelled as 'Person' by the Vision API. Another invalid feature is the 'Projector' that is not present in the environment.

As introduced before, we analysed the images for finding new possible features. 'Walking' and 'Office chair' were added to the list of features. The 'Walking' feature is detected by the Vision API if one of the pictures sent to the server contains a person walking. The Building Entrance description was one of the hardest to describe among the others due to the difficulty of finding features that any other setting or role might contain. Regarding the 'Office chair' feature, it was detected in all the tests, so it was added to the set of features for the Info Desk setting.

6.3.2. Tutor

Feature	#Occurrences	Feature	#Occurrences
Desk	4	Desk	6
Door	3	Door	6
Laptop, Computer	3	Laptop, Computer	5
Monitor, Display	1	Table	8
Table	5	Room	8
Room	5	Person present	8
Projector	2	People passing by	1
Person present	5	Low level of noise	8
People passing by	4	Sitting	7
Low number of people	2	**Event**	5
Low level of noise	3	Conversation	6
High level of noise	2		
Sitting	3		
Event	5		

Table 6.2: Features detected and amount of times they appear in the feature vectors during the Lecture (on the left, with a total of **5** tests) and the Meeting (on the right, with a total of **8** tests) setting experiments

Table 6.2 shows the features for Lecture (left) and Meeting (right). As regards the experiments of the Lecture setting, the 'Low number of people' became True 2 times out of 5. However, more than 10 people were attending the lecture, thus this is not representing reality. The cause of the invalid feature is the lost track of people of Pepper ending up with acknowledging a smaller number of people. The same phenomena lead to another case of the invalid feature: 'People passing by'.

Similarly happens during the tests of the Meeting setting, but with a lower frequency. Furthermore, we were expecting to find the features 'Paper' or 'Pen' for both the settings, but these objects were not detected by the Vision API even though present in the environment.

As it happened with the Receptionist analysis, another feature has been added for the Tutor's settings. The feature 'Event' has been detected with a high frequency in both the Lecture and Meeting setting as shown in Table 6.2. Providing information that something, from a social point of view, is going on. We believe that the feature can help in the classification since it will allow us to distinct, for example, between a secretary sitting at the info desk working at her/his laptop from people being in a lecture or a meeting.

6.3.3. Waiter

Feature	#Occurrences
Desk	2
Door	2
Kitchen appliances	1
Room	3
Shelf	2
Table	3
Low level of noise	3

Feature	#Occurrences
Desk	3
Door	1
Meal	2
Food	3
Table	4
Tableware	1
Paper	1
Room	4
Shelf	1
Person present	4
People passing by	1
Sitting	1
Conversation	3
Eating	1
Event	1
High level of noise	1
Low level of noise	3

Table 6.3: Features detected and amount of times they appear in the feature vectors during the Empty Restaurant (on the left, with a total of **3 tests**) and the Busy Restaurant (on the right, with a total of **4 tests**) setting experiments

In Table 6.3,the frequencies of the features detected during the Empty (left) and Busy (right) Restaurant are shown. During the tests of the Empty Restaurant setting, the robot did not detect any person in the canteen, even though we prepared the tables as we were in a restaurant. On the other hand, during the Busy Restaurant setting, most of the features have been detected.

Additional features have been added. 'Meal', 'Eating' and 'Event' were detected in the Busy Restaurant tests and added due to the additional value they can bring.

6.3.4. Sufficiently distinct sets analysis

Using the data collected, we can now build a feature/setting table that shows which features are contained in the set of a setting to study how distinct the sets are. The table in Figure 6.5 shows a different feature per row and a different setting per column. When the element of a cell $Cell(cr)$ is equal to 1 means that the feature of the row r is contained in the setting of the column c.

Features	Building Entrance	Info Desk	Lecture	Meeting	Empty Restaurant	Busy Restaurant
People passing by	1	0	0	0	0	1
Person Present	1	1	1	1	0	1
Low level of noise	1	1	1	1	1	1
Walking	1	0	0	0	0	0
Table	1	1	1	1	1	1
High level of noise	1	0	1	0	0	1
Desk	0	1	1	1	1	1
Laptop	0	1	1	1	0	0
Sofa	0	1	0	0	0	0
Sitting	0	1	1	1	0	1
Office chair	0	1	0	0	0	0
Monitor	0	0	1	0	0	0
Projector	0	0	1	0	0	0
Event	0	0	1	1	0	1
Conversation	0	0	0	1	0	1
Kitchen appliances	0	0	0	0	1	0
Shelf	0	0	0	0	1	1
Meal	0	0	0	0	0	1
Food	0	0	0	0	0	1
Tableware	0	0	0	0	0	1
Eating	0	0	0	0	0	1

Figure 6.5: Features/Settings table

From the table in Figure 6.5, it is possible to derive the features that are uniquely describing a setting. The table of 'unique features' is shown in Figure 6.6. When the element of a cell $Cell(cr)$ is equal to 1 means that the feature of the row r is uniquely describing the setting of the column c. For example 'office chair' uniquely describes the Info Desk setting.

Features	Building Entrance	Info Desk	Lecture	Meeting	Empty Restaurant	Busy Restaurant
People passing by	0	0	0	0	0	0
Person Present	0	0	0	0	0	0
Low level of noise	0	0	0	0	0	0
Walking	1	0	0	0	0	0
Table	0	0	0	0	0	0
High level of noise	0	0	0	0	0	0
Desk	0	0	0	0	0	0
Laptop	0	0	0	0	0	0
Sofa	0	1	0	0	0	0
Sitting	0	0	0	0	0	0
Office chair	0	1	0	0	0	0
Monitor	0	0	1	0	0	0
Projector	0	0	1	0	0	0
Event	0	0	0	0	0	0
Conversation	0	0	0	0	0	0
Kitchen appliances	0	0	0	0	1	0
Shelf	0	0	0	0	0	0
Meal	0	0	0	0	0	1
Food	0	0	0	0	0	1
Tableware	0	0	0	0	0	1
Eating	0	0	0	0	0	1

Figure 6.6: Features/Settings table, unique features

In the table shown in Figure 6.6 the settings Building Entrance, Info Desk, Lecture, Empty Restaurant, and Busy Restaurant have at least one 'unique' feature, that can be sufficient for distinguishing them from another setting. However, Meeting does not have any unique feature, so nothing can be concluded. The Meeting setting has features in common wit the Lecture and Busy Restaurant settings. The euclidean distance between the Lecture and Meeting settings, and the euclidean distance between the Meeting and Busy Restaurant are calculated for showing that the Meeting set is not a sub set of the other two but rather of the union of them. The distances are calculated as shown in Figure 6.7.

```
%% Arrays
Lecture = [0,1,1,0,1,1,1,1,0,1,0,1,1,1,0,0,0,0,0,0,0];
Meeting = [0,1,1,0,1,0,1,1,0,1,0,0,0,1,1,0,0,0,0,0,0];
BusyRestaurant = [1,1,1,0,1,1,1,0,0,1,0,0,0,1,1,0,1,1,1,1,1];

%% plot

distance1 = norm(Lecture - Meeting);
distance2 = norm(Meeting - BusyRestaurant);
```

Figure 6.7: Features/Settings table, unique features

The distance between Lecture and Meeting is 2.0, and the distance between Busy Restaurant and Meeting is 2.8, thus we can conclude that the Meeting set of feature is sufficiently distinct.

6.4. Features selection for every setting

The analysis of the features has been explained in the previous section. The next step is then to define a final set of features that the classifier will use and their weights.

Features that appeared in all of the settings, like 'Door' or 'Room', and all features that were wrongly detected were discarded. An exception was made in the Building Entrance setting: 'People present' was falsely triggered due to recognizing another Pepper robot in the room. This caused the robot to not trigger anymore at the presence of real people. The feature 'People present' was still seen as useful and not discarded because, in future experiments, the other Pepper robot would not be present. After discarding the features, the procedure was to select features that occur the most for every setting and are specific for the setting.

In Chapter 4, we discussed that the probabilities table increases in size proportionally to 2 to the power of the number of features. In order to not overload the system in terms of computation power and memory allocation, a limit on the number of the features was set to 6. However, an exception was made for the Busy Restaurant setting due to its poor performance during training and the number of features was increased to 7.

The list of features for every setting is summarised in Tables 6.4, 6.5 and 6.6.

6.4.1. Receptionist

Feature for Building Entrance	Feature for Info Desk
People passing by	Desk
Person present	Laptop, Computer
Low level of noise	Sofa, Couch
Walking	Person present
	Sitting
	Office chair

Table 6.4: Features selected for the Building Entrance (on the left) and the Info Desk (on the right) settings.

6.4.2. Tutor

Feature for Lecture	Feature for Meeting
Desk	Conversation
Laptop, Computer	Laptop, Computer
Projector	Low level of noise
Person present	Person present
Sitting	Sitting
Event	Event

Table 6.5: Features selected for the Lecture (on the left) and the Meeting (on the right) settings.

6.4.3. Waiter

Feature for Empty Restaurant	Feature for Busy Restaurant
Kitchen appliances	Meal
Food	Food
Shelf	Table
Table	Tableware
Low level of noise	Person present
Low number of people	Sitting
	Conversation

Table 6.6: Features selected for the Empty Restaurant (on the left) and the Busy Restaurant (on the right) settings.

For the Empty Restaurant setting, additional two features that were not detected during the tests were added since without them, the description would result poor. The two features are 'Food' and 'Low number of people'.

6.4.4. Classifier weights

In order to classify the scenes, the posterior probabilities to store in the probabilities table must be calculated. Since during the data collection, not all the possible combinations of the feature vectors were present, and due to the different number of tests done on the settings that would result in overestimating a setting, we decided to not calculate the posterior probability through conventional methods, but rather we used a combination of knowledge-based and frequentist approach to estimate it.

The knowledge-based approach, that consists in setting the posterior probability to a known value. That is the case of the Lecture and Meeting setting: the posterior probability was set equal to 0 if no people were present.

For the rest of the cases, the frequentist approach was used for estimating the posterior probabilities. The method consisted in attributing weights to the features for every setting, by looking at the number of occurrences described in the previous section. The weights in a setting are calculated as shown in Equation 6.1 where k refers to the feature and i to the setting. N is the number of occurrences of the feature k in setting i. M is the number of tests of the setting.

$$w_{ki} = \frac{N_{ki}}{M_i} \tag{6.1}$$

The weights are then used for estimating the posterior probabilities. The probability is calculated as shown in Equation 6.1, where F refers to the used feature vector, which contains n features $(F_1, F_2, ..., F_n)$, in setting i.

$$P(i|F) \approx \frac{\sum_{k=1}^{n} w_{ki} * F_k}{\sum_{k=1}^{n} w_{ki}} \tag{6.2}$$

We assume that the probabilities calculated in Equation 6.2 can be approximated to the posterior probability $P(setting|features)$. This method allowed us to make educated guess on how to tune the weights of the features while tuning the classifier.

6.5. Tuning of the parameters of the classifier

Tuning the classifier implied adjusting the weights described in Equation 6.1. Every time after adjusting the weights, the results were simulated as explained in Chapter 5.4. The tuning was stopped when the average error in the classification of the roles was minimised and no improvement was possible.

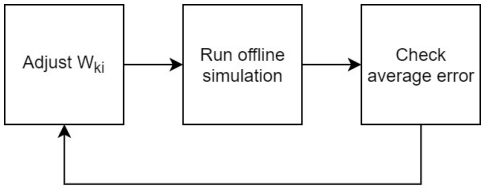


Figure 6.8: Classifier tuning process

6.6. Results

Following in this section is the outputs of the Bayesian Network after training, for every setting. A limited number of tests was run due to time constraint caused by participants and the testing location. In Appendix F it is shown in which rooms the experiments were conducted, by showing the blueprint and the position of the robot in the space. The test numbers refer to the position within this room and are also shown in Appendix F. Since the location of the robot in the room changes per test, the features detected by the robot also changes, causing the output probability to be different per test number.

In the following sections, the results calculated over the representative vectors are shown and discussed per setting.

Info Desk

For the Info Desk setting two tests have been done due to time constraint for using the room, and in both of them, the Receptionist role has the higher probability as shown in Figure 6.10.

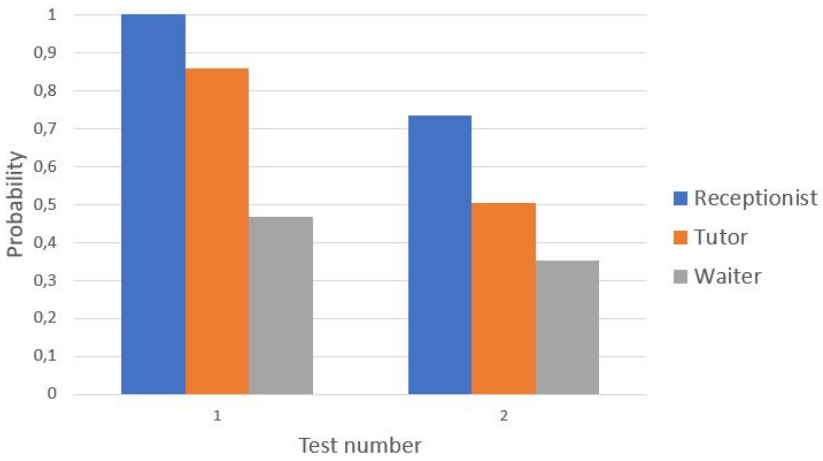


Figure 6.9: Results from the training Info Desk tests.

Building entrance

Six tests were done for the Building Entrance setting. Results in Figure 6.10 show that 66.7% of the tests successfully classify the scene. As discussed in Section 6.3.1, the description of this setting was found challenging due to the lack of discriminate features.

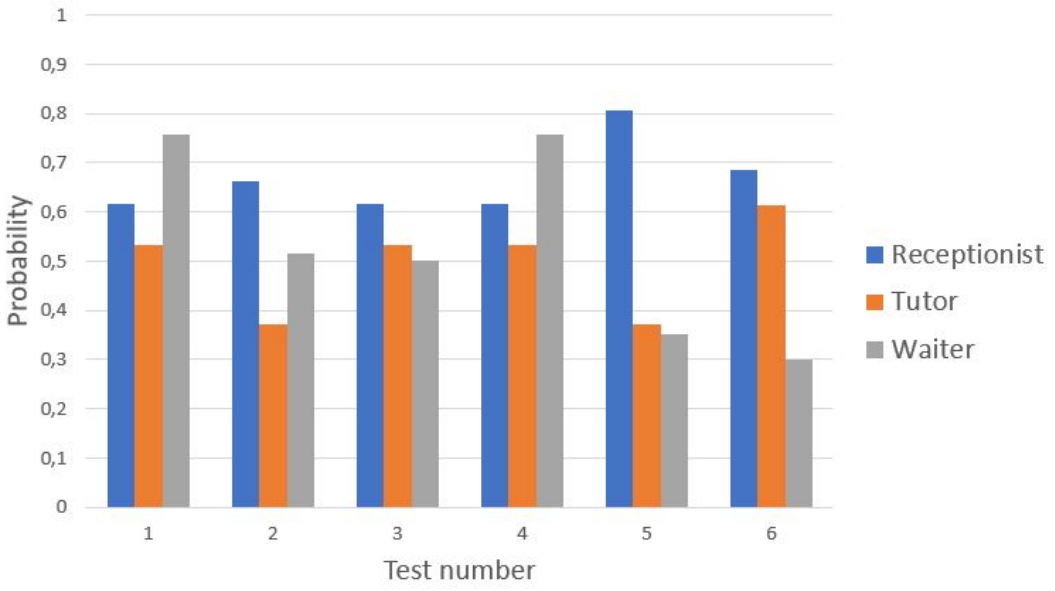


Figure 6.10: Results from the training Entrance Building tests.

Lecture

For the Lecture setting, people have been invited on the 16th October 2019 to assist a lecture at Robovalley. After training, the Tutor role is classified in 100% of the cases as shown iin Figure 6.11.

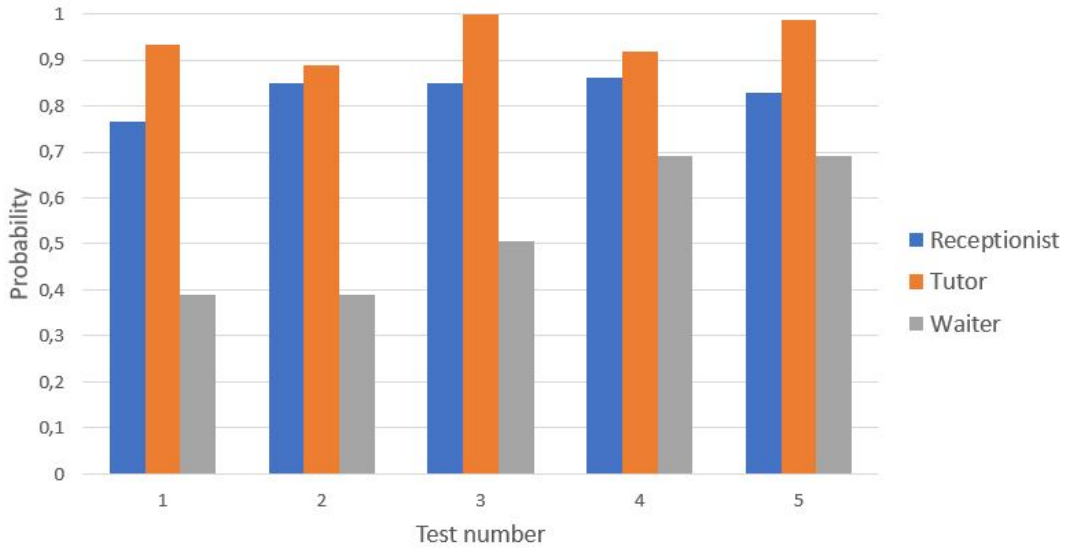


Figure 6.11: Results from the training Lecture tests.

Meeting

The same people invited for the Lecture were then asked to participate in a discussion on different topics. The Tutor role is recognized 100% of the times as shown in Figure 6.12.

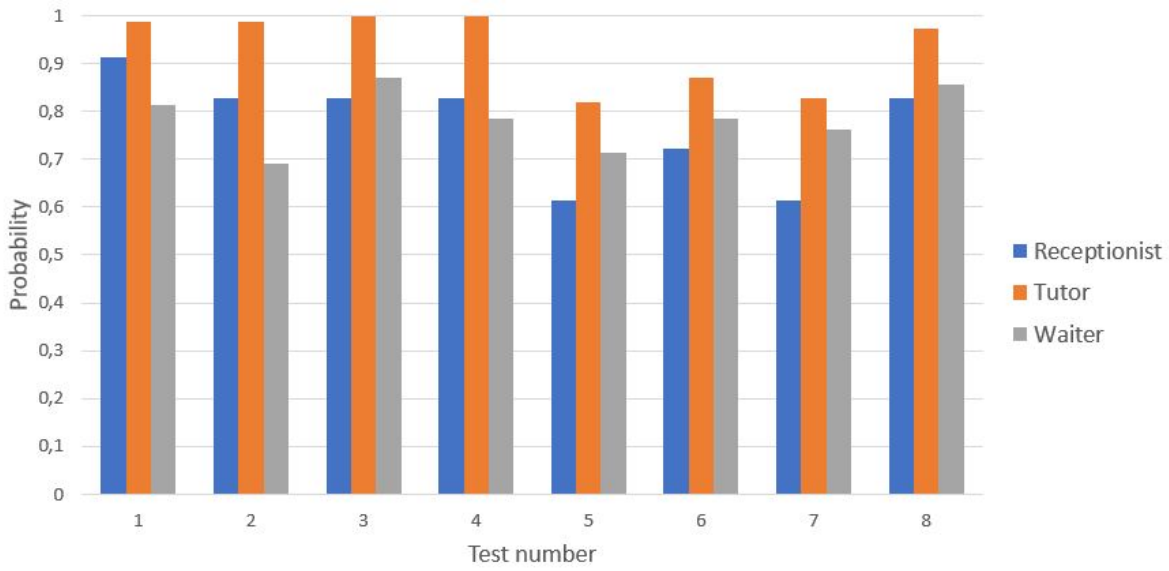


Figure 6.12: Results from the training Meeting tests.

Empty Restaurant

The Empty Restaurant experiments have been done in the canteen of the building outside peak hours for meals. During the tests, the robot did not detect any person in the canteen, even though we prepared the table as we were in a restaurant. For this reason, the Tutor role has 0% of probability as shown in Figure 6.13. The BN classifies the scene under the Waiter role in all the cases as shown in Figure 6.13.

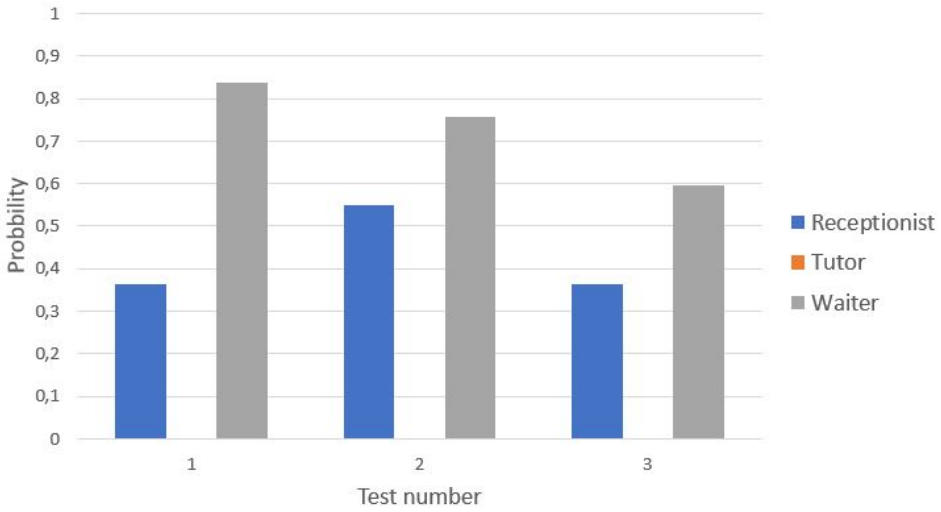


Figure 6.13: Results from the training Empty Restaurant tests.

Busy Restaurant

For the Busy Restaurant tests, the people invited for the Meeting and Lecture were afterwards welcomed in the canteen of the building where food was provided for them. Figure 6.14 shows that 50% of the tests were successfully classifying the scene while the remaining 50% of the tests were classified as in a Tutor role. Some considerations were done about the tests. The first test has been done before food was served, missing this feature the BN was not able to classify correctly the scene. The last test was able to recognise that food was present in the context but it did not label as a meal, giving 0.3 points more to the Tutor rather than the Waiter.

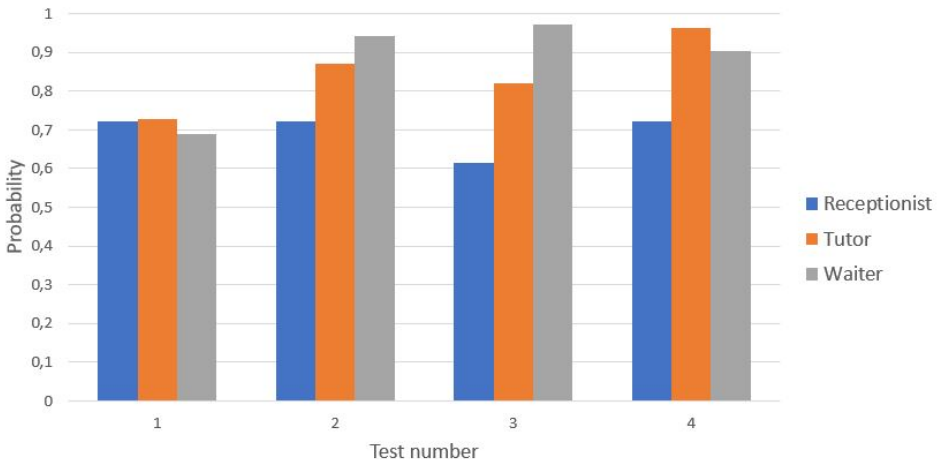


Figure 6.14: Results from the training Busy Restaurant tests.

7

Validation

This chapter discusses the experiments made for the validation part. During validation, the aim is to test the performance and the generalization of our method. Similarly to the training experiments, data in which people can be recognized might be collected, a consent form and experiment information form have been used for collecting people's consensus. The two forms can be found in Appendix E.

7.1. Experimental setup

The validation experiments have been done in a different building from the training ones. That is important since unseen data is analysed and performance analysis on the model can be done. The chosen building is the number 28 of the Delft University of Technology, Mekelweg 5, 2628 CD Delft. Similar to the training experiment, three different locations were chosen: the entrance/receptionist area, a room used for meeting or presentations and the canteen. Every test started by placing the robot in the desired location in the environment, then the algorithm was started. Differently from the training tests, during the validation ones the algorithm was not stopped between one test to another, but rather the robot was moved autonomously or manually depending on the situation. The choice of never stopping the algorithm was due to having a collection of continuous data allowing us deeper analysis. Results of the experiments are discussed in the following subsection.

7.2. Results

Results are discussed in the following way: first, the number of samples and their classification over time are visualised in a graph. Together with that, the total percentage of the classification is shown. A discussion about the classification over time follows, explaining why the output changes and the errors occurred.

The following types of errors will be discussed:

1. **Feature detection**, the error is the result of the detection of a feature that is not present, or the miss of detecting a feature that is present. For example, the detection of a projector, even though it is not present in the environment.
2. **Robot positioning**, the error occurs when the position of the robot results in missing features or losing information about its surroundings. For example, if the robot is in a corner and keeps looking at the wall, therefore, is not able to detect features.
3. **Predicting model**, the error is the result of a design choice of the model. For example, not including a feature, even though it contributes to describing a setting.
4. **Conceptualisation of the setting**, the error occurs when the setting changes to a different one than the intended setting for the experiment. For example, if people are having a meeting in a restaurant, the setting changes.

Info Desk

The results of the tests made in the info desk settings are shown in Figure 7.1. The output of the classifier is stable since not much is happening at the info desk of the building. A total of 60 samples

have been collected all of them are classified correctly. However, due to the lack of variation of the location, no conclusions about the generalization performance can be made.

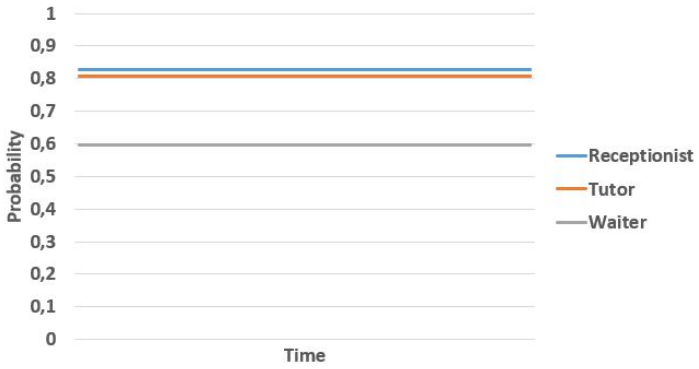


Figure 7.1: Results from the validation Info Desk tests.

Building entrance

In Figure 7.2, it is possible to notice that during the 160 samples recorded, the accuracy of the three roles fluctuates in time. However, 79% of the classifications are correct as shown in Figure 7.3

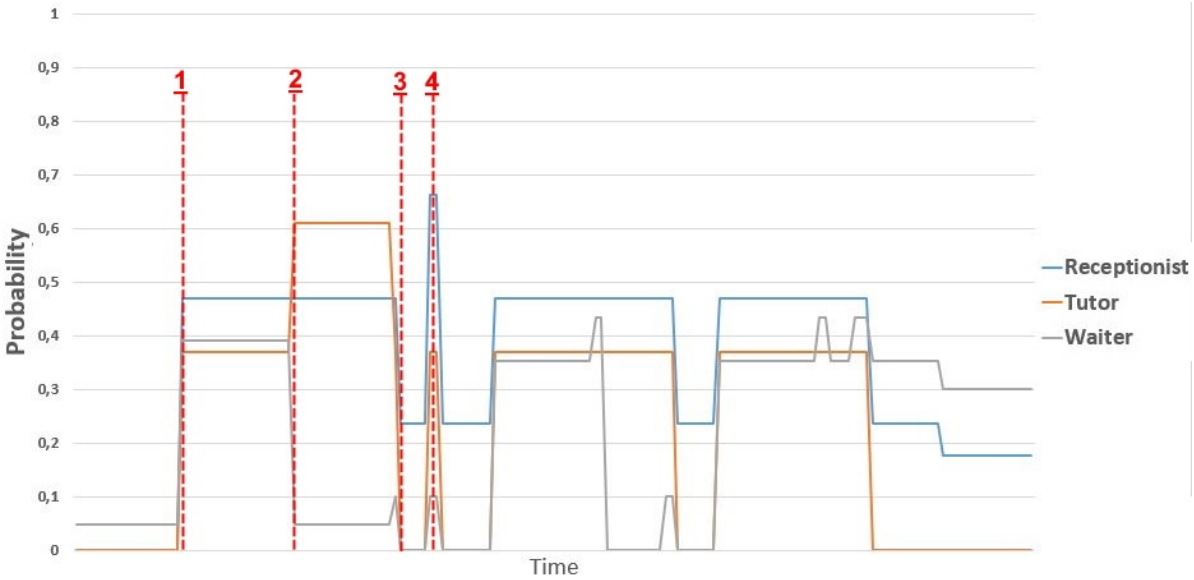


Figure 7.2: Results from the validation Building Entrance tests.

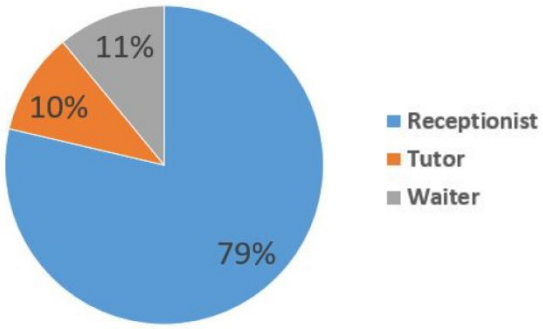


Figure 7.3: Percentages of the outputs for the Building Entrance tests.

The fluctuations in the graph in Figure 7.2 were analysed for understanding what features caused the classifier to change its output. The graph has been split into sub-regions for better understanding during the explanation. The first transition happens when the three percentages suddenly become higher. This is due to the fact that at the beginning of the experiment no one was entering or exiting the building, so the robot did not detect any human activity or presence. As soon as the robot detected the presence of people, at point 1, the probabilities increase.

The person that the robot saw is not actually part of the scene. As shown in Figure 7.4, a study room is right next to the entrance. The glass doors of the room allowed the robot to see what is not really part of the scene.

At point 2, the probability of the Tutor and the one of Restaurant change, this is due to the feature 'Event', that once again is an invalid feature, triggered by a poster in the room. However, an important event happens between the point 2 and 3 in Figure 7.2: a person enters the door, gets the attention of Pepper and then leaves, making the 'People passing by' feature true. So, when the next person comes in at point 4, the BN gives almost 70 percent of probability for being in a Receptionist role.

After point 4, the probabilities are almost constant, the fluctuations are due to people entering and then leaving during the test, triggering the 'People passing by' feature. Not so many people passed by, so when the feature became true, no more people passed by the entrance, leaving the Receptionist role at almost 50%.

However, if no person is detected, the robot does not necessarily need to act as a receptionist. This could be a conceptualisation error of the setting and could be overcome by changing the model or to add a threshold in the outcome that if the percentages are lower than it, then no role is required.

Figure 7.4: The study room next to the entrance of the building

Lecture

Similarly to the training experiments, for the Lecture tests, people have been invited to attend a presentation in the building. Results of the 120 samples are shown in Figure 7.5, where the Tutor role is recognized most of the time. Figure 7.6 displays the percentages of the output of the BN for the three roles, the Tutor has the highest percentage of 71%.

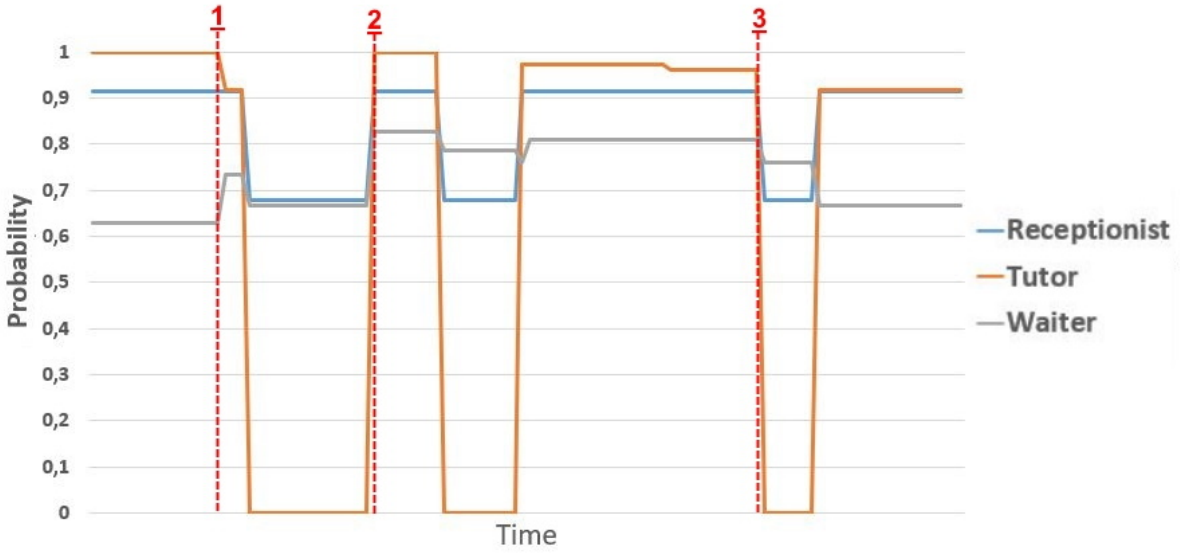


Figure 7.5: Results from the validation Lecture tests.

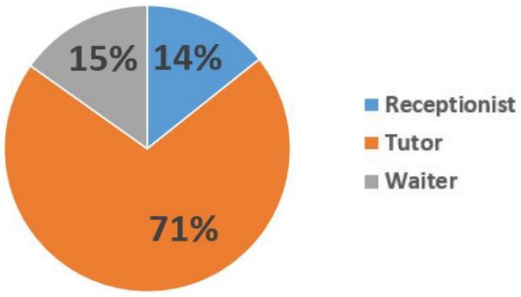


Figure 7.6: Percentages of the outputs for the Lecture tests.

An analysis of the results follow. Similarly to the Building Entrance analysis, the results were divided into sub-regions by indicating points. At point 1, after the output has been stable, the Tutor and Waiter outputs slightly change due to the missing of two features, 'Projector' and 'Event'. The two features were not detected because the robot stopped rotating during the collection of the pictures.

Right after, the robot lost track of the people in the room resulting in a drastic decrease of the Tutor percentage to 0%. We noticed during testing that the room was deceiving the robot: the sound of people talking was bouncing on the walls and making the robot turning in the wrong direction and never turning back until the next test. In NAOqi APIs, if a person is not seen for a predefined timeout, then the person is removed by the current population list of Pepper. Looking to the wall, Pepper lost track of every person in the population, leading to the incorrect 'false' value of the feature 'Person present'. This shows the importance of the position and the direction of view of the robot during the tests. The same behaviour can be seen other two times in the graph.

The Waiter percentage has some fluctuations during the tests. It increases when the 'Conversation' is detected at point 2, together with the 'Person present' again. Then fluctuates since the feature 'Low

number of people' is True due to the losing track of the people until point 3. The error is due to the position of the robot that results in detecting a wrong feature (e.g. the robot saw all the people present but when finishing to scan the surrounding it can see only a part of them). However, even with this error, the output of the classifier is correct. Only when the 'People present' error occurs, as discussed before, the output is wrong.

Meeting

The same people invited to attend the lecture were then invited to discuss topics as they were in a meeting. Results of the 140 sample of the Meeting experiments are shown in Figure 7.7 and the percentages of the roles as the output of the BN are shown in Figure 7.8. Most of the time (93%) the output of the BN is displaying the correct role, the Tutor one.

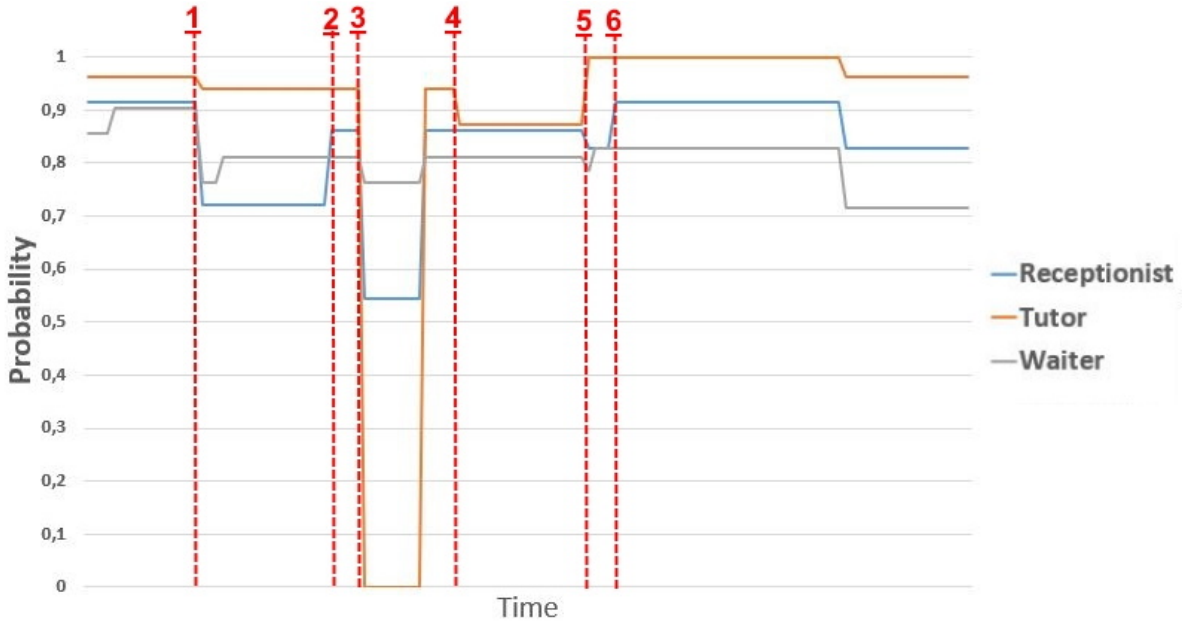


Figure 7.7: Results from the validation Meeting tests.

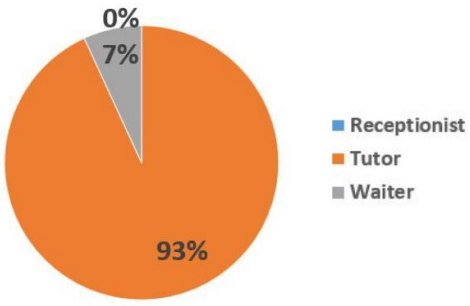


Figure 7.8: Percentages of the outputs for the Meeting tests.

Similar to the previous settings, a detailed analysis of the graph in Figure 7.7 is done.

Initially, between point 1 and 2, the line of the Tutor role is slightly oscillating between 0.96 and 0.94, this is because the robot first recognized the presence of a 'Desk' in the environment and after that, it did not. More interesting is what is happening with the other two roles: the Receptionist starts with an accuracy of 0.91 that at a certain point drops to 0.72, because the feature 'People passing by' is not true anymore. However, it must be said that the 'True' value of the feature at the beginning is wrong since no one left the room. The fluctuation of it during tests is caused by this feature. On the

other hand, also the Waiter role fluctuates because the feature 'Low number of people' gets True and False due to inaccuracy of the tracking of Pepper. In both cases, the errors are due to the position of the robot that results in detecting wrong features (e.g. the robot saw all the people present but when finishing to scan the surrounding it can see only a part of them). The same happens for the rest of the test. However, even if these errors occurred, the classifier is still able to correctly detect the Tutor role.

At point 3 in the Figure 7.7, the same behaviour explained in the lecture experiments happens: the robot lost track of the people in the room by triggering the 'False' to the 'Person present' feature. This error causes the classifier to fail in the classification.

At point 4, the percentages of the Receptionist and Tutor become really close since the feature 'Laptop' is not seen in the scene anymore.

At point 5 'Event' makes the Tutor percentage increasing while the other two drops. 'People passing by' increases the percentage of the Receptionist at point 6.

Empty Restaurant

The results for the experiments for the Empty Restaurant setting are shown in Figure 7.9. For this experiment, 160 samples were collected. Contrarily with what happened in the previous cases, the total percentage graph of Figure 7.10 shows that 55% of the time the output of the BN is wrong.

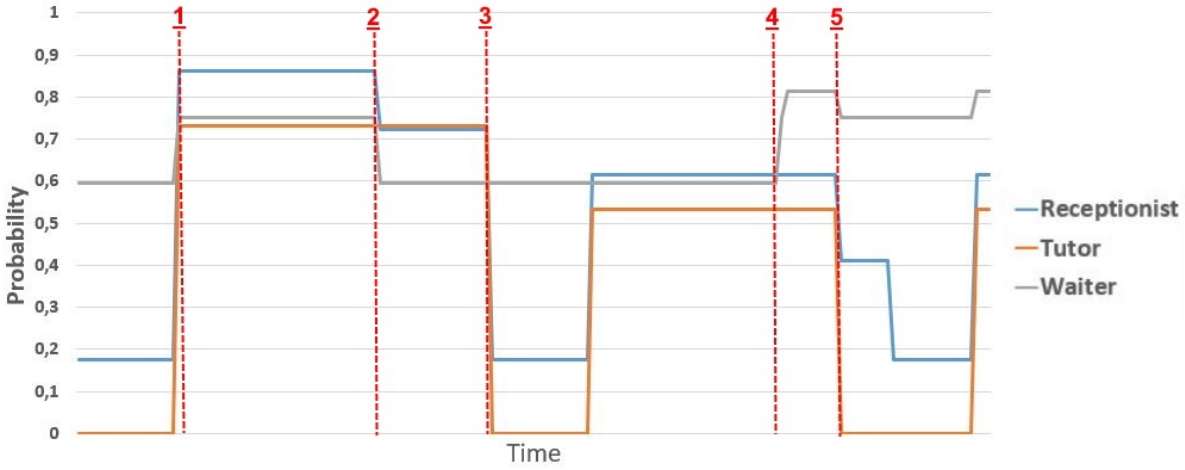


Figure 7.9: Results from the validation Empty Restaurant tests.

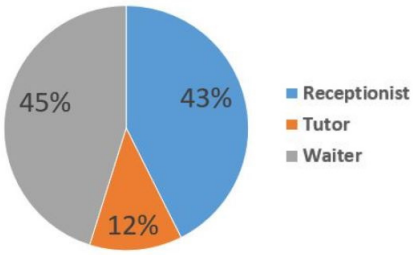


Figure 7.10: Percentages of the outputs for the validation Empty Restaurant tests.

Initially, the output of the model classifies the scene as a Waiter role scene with a 60% of probability until point 1. The only features that are detected up until that moment are: 'Low level of noise' and 'Table'. No additional features are detected, even though the robot has been placed really close to the snacks shelf, it was not able to recognise the food next to it as shown in Figure 7.11. However, the output of the classifier is correct.

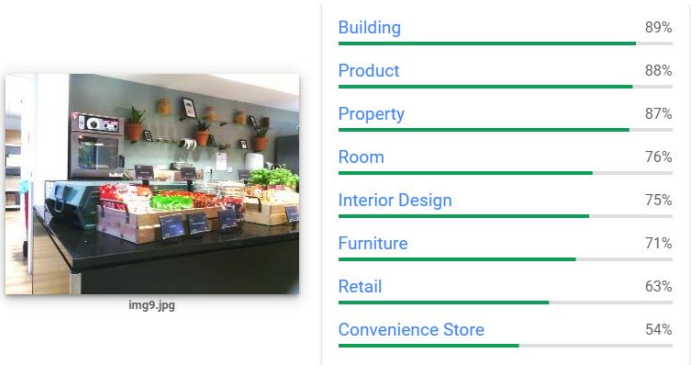


Figure 7.11: The Vision API is not able to detect the food next to the robot, probably due to the poor quality of the picture.

Between point 1 and 2, the robot was able to detect much more: 'Food', 'People' and also 'People passing by', since the woman working at the canteen that day went next to the robot and back to the kitchen. Next to that, it detects the presence of a 'Laptop' that was not present. Furthermore, the robot was not able to detect the kitchen appliances present in the environment such as the oven and the cooker. The errors for 'Laptop' and 'Kitchen appliances' are both due to errors in the feature detection. This results in the classifier giving a higher probability to be in a Receptionist role.

After point 2, the Receptionist and the Waiter values drop. This is because the robot did not see anyone passing by and the robot has reached a place where it is not possible to see food anymore. That shows that 'Passing by' could be included in the feature set of the Empty Restaurant setting. This could be seen as an error in the model.

The percentages right after point 3 drop again since no one was there and so the 'People present' becomes false. Between point 3 and 4 the robot reaches a position that is in the corridor that faces the entrance of the building but it is, still, next to the counter of the canteen. In Figure 7.12 and 7.13, pictures taken from the tests run between point 3 and 4 are shown.

(a) The entrance of the building seen during the test (b) The counter of the canteen seen during the test

Figure 7.12: Picture taken from a test run between point 3 and 4 of Figure 7.9

(a) The counter of the canteen seen during the test

Figure 7.13: Picture taken from a test run between point 3 and 4 of Figure 7.9

Initially, the classifier gives a higher probability to the Waiter role until the presence of people is detected again. This could be seen as an error in the model and that 'People present' could be an addition to the model of the Empty Restaurant. However, this could also be the result of an error in the conceptualisation of the setting and, thus, the model correctly classifies it as a Receptionist role when people are present.

After point 4, the Waiter percentage increases due to the robot detecting the presence of Food. This results in the correct classification of the scene.

Busy Restaurant

The last experiments are the ones related to the Busy Restaurant setting in which 180 samples were collected. Results are shown in Figure 7.14. The prevalent role during the test is the Restaurant one with 67%, as shown in Figure 7.15.

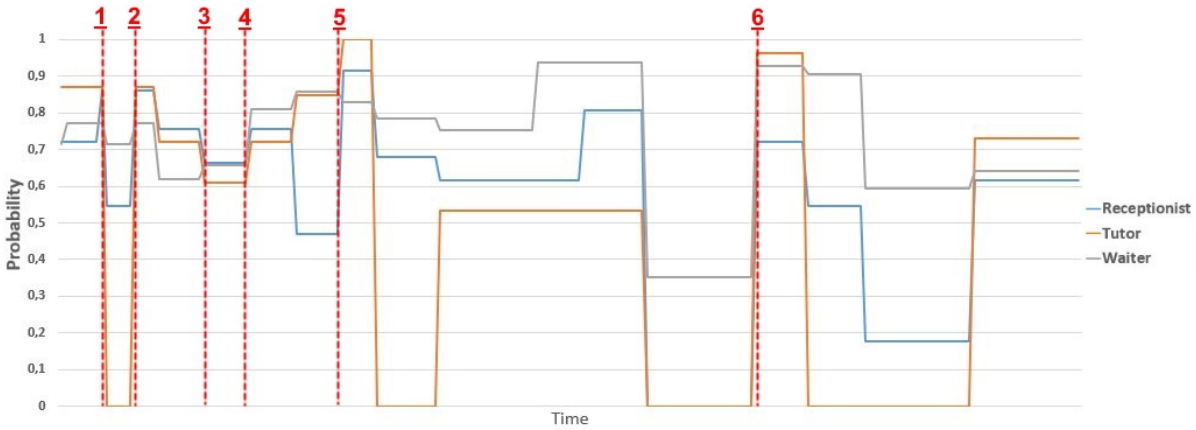


Figure 7.14: Results from the validation Busy Restaurant tests.

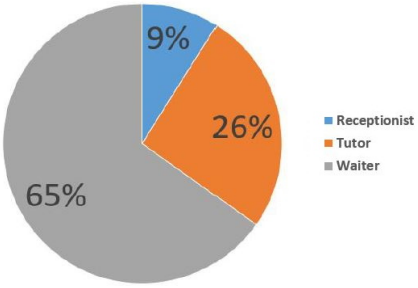


Figure 7.15: Percentages of the outputs for the validation Busy Restaurant tests.

After analysis, the graph is split into sub-regions as shown in Figure 7.14. In the beginning, the Tutor role has a high percentage. The robot was able to detect that there were people in the context, that they were having a conversation, but no food or meal was detected due to a feature detection error. This error gave the Tutor role a major impact on the output. However, during the test, some people were having a meeting and some were having their meal as shown in Figure 7.16. This could be a conceptualisation error of the setting. The same behaviour is seen between point 2 and 3.

Figure 7.16: Image from the test. Two people are working on their laptop while other people are having a meal.

At point 1 the robot lost track of people. Consequentially, all the probabilities decrease rapidly, before then going up once a person was detected again in point 2. However, the Waiter role is the one that is affected the least from this error, resulting in being the classified role.

At point 3, after detecting 'Food', the Waiter probability gets closer to the Receptionist role that has decreased together with the Tutor role due to not seeing a 'Desk' anymore.

Only when the robot detects 'Meal', the Waiter probability becomes the highest of the three, at point 4. The Tutor role overtakes it at point 5 due to some two feature detection errors, 'Projector' and 'Laptop'. After the peak of the Tutor, its probability drops and the following in tests no feature detection errors occur. Only at point 6, the Tutor has a peak to almost 100% of probability due to the fact that almost all the features of the Meeting setting have been true except to the presence of a laptop. This can be seen as an error in the conceptualisation of the setting since it is not clearly defined whether it is still a restaurant or a meeting setting.

8

Conclusions

With this thesis project, we aimed to study the performance and the generalization of a multi-modal, object-based scene classification that uses a knowledge-based hybrid model for classification. By scene, we refer to the environment and the context of the surroundings. The input for the classification are features that can be of three types: objects, audio, and human detection and understanding. While the output is the percentage of the roles the robot can play, that are *Receptionist*, *Tutor* and *Waiter*. To our knowledge, we are the first to do so.

Initially, a study on how to describe a scene by using a set of features was done. The results of an anonymous online questionnaire were used for finding out potentially relevant features. After narrowing the possible features, detecting and choosing the tools for the project, the design of the implementation happened. The features are thought through knowledge-based assumptions on human perception. Therefore, a data-set from a first-person perspective was needed for the experiments. Real-time multi-modal data of different locations and scenes were collected using a Pepper robot by Softbank. The data-set includes pictures of the surroundings (one picture every 45 degrees until reaching 360 degrees), audio and people perception cues.

With the data collected for training, the sets of features to be used for classifying the scenes were finalised. Training the classifier implied heuristically adjusting the weights. Tuning was stopped when the average error on the probabilities in the classification of the roles was minimised. A summary of the results from the training experiments is shown in the confusion matrix of Table 8.1. The Receptionist role was difficult to uniquely describe its settings, especially for the Building Entrance. This caused it to be often be confused with the Waiter role. The Tutor role, on the other hand, has been correctly classified in all the tests. Finally, the Waiter role was correctly classified the lowest amount of times, because it was often confused with the Tutor role.

These results and the results on the analysis of the sets of Section 6.3.4, support our first hypothesis, which states it is possible to define sets of features that are sufficiently distinct for classifying scenes.

	Receptionist	Tutor	Waiter
Receptionist	0.750	0.000	0.000
Tutor	0.000	1.000	0.285
Waiter	0.250	0.000	0.715

Table 8.1: Confusion matrix from the training experiments

The last phase of our project was the validation one, where we collected real-time multi-modal data from a different location than the training experiments. A summary of the results achieved from this phase can be seen in Table 8.2. Four different types of errors occurred during validation (feature detection, robot positioning, predicting model or conceptualisation of the setting). The accuracy of the Tutor and Waiter reduced during validation, while the Receptionist accuracy increased becoming the most accurate role classified, compared to the training results. Probably the increased accuracy

of the Receptionist role is due to the fact that the validation locations were more dynamic than the training ones. Furthermore, the Receptionist role, when not recognized, it is confused for most of the time with the Waiter role, similar to what happened during training. Also, the Tutor role is most of the time confused with the Waiter role. However, as discussed in Chapter 7, the errors that occurred for that to happen were due to a wrong position of the robot. The Waiter role is now confused most of the time with the Receptionist role. This was caused by feature detection errors and changing the prediction model could have helped to increase the performance of the classifier. This, combined with a poor conceptualisation of the setting, contribute to the poor results of this experiment. In Chapter 7 the results for every setting were discussed by showing the output of the classifier over time. By moving around the robot, it was able to detect different features in the environment, proving the second hypothesis of Section 2.1.

	Receptionist	Tutor	Waiter
Receptionist	0.839	0.066	0.246
Tutor	0.078	0.827	0.195
Waiter	0.083	0.107	0.559

Table 8.2: Confusion matrix from the validation experiments

Showing that our multi-modal, knowledge-based hybrid scene classification generalises to unseen data, support our second hypothesis. This could be a first step into enabling the future deployment of a single robot in multiple applications.

Furthermore, we excluded from the results of Table 8.2 all the classifications that were due to avoidable errors that are of the type: robot positioning and conceptualisation of the setting. Table 8.3 shows the new confusion matrix, which gives a better representation of the performance of the classifier removing avoidable errors. The accuracy of the three roles has increased, especially for the Tutor role. This shows that if, in future, these errors are taken into consideration, the performance of the classification can increase.

	Receptionist	Tutor	Waiter
Receptionist	0.878	0.000	0.153
Tutor	0.122	1.00	0.212
Waiter	0.000	0.000	0.635

Table 8.3: Confusion matrix from the filtered validation experiments

8.0.1. Contributions

With our project we contributed to:

- The description of a social scene through a set of features.
- Combining objects, audio, and human understanding into a set of features.
- Collecting real-time multi-modal data from a mobile robot.
- Using a heuristic approach for classification.
- Classification of a social role for a mobile robot.
- Future possible deployment of a robot adjusting its role in multiple scenes.

8.1. Limitations and future work

During the development of the project, we faced some problems that are mainly related to the APIs used and the sensors which Pepper is equipped with.

During the experiments, four types of errors were detected:

1. **Feature detection**: during training, some features, like 'Paper', 'Pen' or 'Dish', were not recognised due to the poor resolution of the cameras. To overcome the problem, the features were discarded. Similar errors occurred during the validation experiments as discussed in Chapter 7. Possible solutions could be by either using better sensors or researching how to overall increase the accuracy of the feature detection. A possible approach could be to change the threshold for which a feature is defined true. This could help avoidance of false positives, but could also increase the number of false negative outputs, or vice versa.
2. **Robot positioning**: during both training and validation, features were not detected due to wrong positioning of the robot's field of view. Research could be done on how to position the robot in a way that maximizes the classifier performance.
3. **Predicting model**: during validation, some errors in the prediction model occurred. The features set of the Empty Restaurant lacked in performance, probably due to the poor generalization of the training data. Taking data from a true restaurant could improve the description and analysis of this setting. Other than that, more data from different locations and scenes should be collected for this reason. More research could be done by exploring the performance of the classifier with varying the features and the number of features.
4. **Conceptualisation of the setting**: during the research, it was noticed that the settings were sometimes not clearly defined. For example, a restaurant was simultaneously used for having a meeting and having a meal. This meant it was not clear what the output of the classifier should be and therefore whether it was correct or not. Research could be done for better understanding scenes, or clearer defined scenes could be chosen for the experiments (for example a true restaurant).

Apart from the errors just mentioned, we found that a deeper analysis of features should be done. Future works might include the temporal aspects of the description of the scenes by analysing how the features change over time in the same scene (For example during the busy restaurant, we noticed changing features).

Another approach that is suggested is to use the negation of some features for better differentiate between roles (for example using the fact that 'Meal' has not been detected for emphasises the other roles other than the waiter one), after proper analysis of data. The number of features for setting can be increased, with the possible drawback of reducing performance.

Furthermore, for future research, we suggest the following dos and don'ts:

- Do have an algorithm that takes into account or avoids Pepper turning and facing the wall
- Do use Pepper robot for reproducibility reasons
- Do collect data from a real restaurant
- Do add the feature 'People passing by' and 'People present' in the Empty restaurant setting
- Do not collect a different amount of data per setting
- Do make use an offline simulator for simulating the classifier

Looking at the future, we believe that our project can be a valuable addition to the social robotics field. Other than studies on how to describe scenes, it is also suggested that future works explore different tools (APIs and classifiers), or even different topologies of the Bayesian Network.

A

Questionnaire on how to describe a scene

During our research we wanted to study how successfully describe a social scene. For this reason we decided to take inspiration by how humans do that. We asked some people to answer an anonymous online questionnaire where they were asked to describe the scene from a point of view of a receptionist, tutor and waiter in different occasions. For the receptionist we asked to describe the scene thinking at first to standing at the entrance of the building, and then being at the info desk. For the tutor, we asked to describe the scene thinking to be in a lecture and sequentially in a meeting. Finally, for the waiter role, we asked to describe the scene when the restaurant is empty (before and after opening hours) and then during opening hours, representing the busy restaurant setting. The questionnaire was an anonymous online questionnaire made with Google Form. We now show the answers to the questionnaire.

A.1. Responses

A.1.1. Building entrance

R1	R2	R3	R4	R5
Desk Glass windows Pen Entrance Door Roof Noisy People passing by	Men/Women People standing Smiles Stylish dresses	People walking Tablets People standing	Doors Halls People walking by	Noisy Confused people Random questions

Table A.1: Questionnaire responses for the building entrance

R6	R7	R8
People that need information Desk Laptop Papers	People coming in People passing by People stopping for asking questions Noisy environment Info desk Security guards Security doors Decorations	Reception desk People passing by Noisy

Table A.2: Questionnaire responses for the building entrance

A.1.2. Info desk

R1	R2	R3	R4
Desk	Men/Women	People standing in line	Desk
Glass windows	People standing	Colleagues sitting next	Phone
Pen	Smiles	Telephone	Direction signs
Entrance Door	Stylish dresses	Headset	People asking questions
Roof		Computer	
Noisy		Office objects	
People passing by		Ticket machine with display	

Table A.3: Questionnaire responses for the info desk

R5	R6	R7	R8
People that need information	Laptop	Phone	Computer screen
Desk	Flowers	Computer	Random questions
Laptop	Decorations	People waiting in line	Borders between
Papers	Pencils	Info desk sign	customers and receptionist
	Paper	Noisy	
	Notebook		
	Phone		
	Tea or coffee		
	Drawers		
	Chair		
	Water bottle		

Table A.4: Questionnaire responses for the info desk

A.1.3. Lecture

R1	R2	R3
Person in opposite direction	Standing	Students
of the audience	Group of people sitting	Cell phones
Speaking standing	Board/Screen	White board
Projector screen	Tables and chairs	Books
People looking at the teacher	Books	Notebooks
Standing	Notes	Pens
Only one voice	Laptop/Computer	Bags
Laptop	Microphone	Coffee cups
		People having snacks
		Laptops

Table A.5: Questionnaire responses for the lecture

R4	R5
Quiet	Clarity
People listening	Board audience
People looking at the screen	Noise
People taking notes	
Screen	

Table A.6: Questionnaire responses for the lecture

R6	R7	R8
One person talking People listening Teacher looking at the students Teacher standing Big room	Projector screen Projector Laptop Books Notebooks Desk Decorations Chairs Group of people sitting People coming in or out Quiet Sometimes noisy	Teacher speaking More than 2-3 people Audience looking at the teacher Board Projection screen People raising their hand for questions Crowded in the 5 minutes before the lecture

Table A.7: Questionnaire responses for the lecture

A.1.4. Meeting

R1	R2	R3	R4
Looking at each others Sitting at the table Multiple voices Whiteboard	Coffee Sitting Pen Papers Mobile phones Laptop Tablet	Talking Looking at each others Taking notes Conversation	Arguments Quarrels Wide range of topics

Table A.8: Questionnaire responses for the meeting

R5	R6	R7	R8
Laptops Papers Pens IPad Presentation board People wearing black clothes	Conversation Private environment Sitting Small room	People sitting around a table Chairs Laptops/IPads Phones Notebooks Pencils, pens Decorations Projection screen Projector Cookies or snacks Coffee Talking	Conversation People sitting around a table

Table A.9: Questionnaire responses for the meeting

A.1.5. Empty restaurant

R1	R2	R3	R4	R5
Waiter clothing Cleaning Setting up tables No approaching people	Waiters Other employees Cleaning Mobile phone	Working alone Making tables Cleaning Calm Quiet	Reservation Cleaning	Food Plates Glasses Cleaning Empty tables Chairs

Table A.10: Questionnaire responses for the empty restaurant

R6	R7	R8
Talking with colleagues Private conversations Chill mood	Empty chairs Cleaning Setting up tables Empty bar Kitchen area few people cleaning Other employees walking around and talking Quiet	Cleaning Moving stuff Few people

Table A.11: Questionnaire responses for the empty restaurant

A.1.6. Busy restaurant

R1	R2	R3	R4	R5
Waiter clothing Menus Setting up tables Approaching people People sitting Noisy	Taking notes Paper, pen Food, drinks Empty plates Silverware Wiping the table Paying machine Cash	Taking orders Walking around Bringing orders Talking to customers Rush Kitchen Noisy	Customers Complaints Noise	Crowded environment Couples Friends Families Shopping bags Food, drinks Kids running People hungry

Table A.12: Questionnaire responses for the busy restaurant

R6	R7	R8
Dressing code Funny/superficial conversations Discussion about working organizations Serious mood	People sitting Noisy People walking around Food, drinks Orders Some empty chairs Busy bar People walking in and out	Walking Waiters carrying dishes Waiter taking orders Background music

Table A.13: Questionnaire responses for the busy restaurant

B

Features annotated by observations

In the study of how to describe a scene, observations of the different scenes were made. Different buildings (number 20 and 36) in the campus of the university of TU Delft were visited to annotate features seen. Pictures of the different scenes are shown below.

(a) Picture taken at the canteen of building 36

(b) Picture taken at the canteen of building 36

Figure B.1: Pictures taken during observations

(a) Picture taken at the canteen of building 36

(b) Picture taken at the canteen of building 36

Figure B.2: Pictures taken during observations

(a) Picture of a person eating taken at the canteen of building 36

(b) Picture of a person eating taken at the canteen of building 36

Figure B.3: Pictures taken during observations

(a) Picture taken at the canteen of building 20

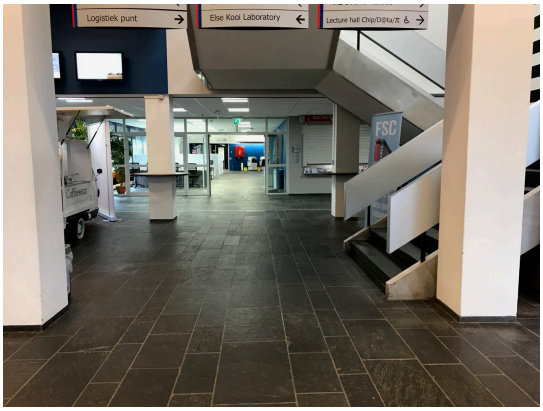
(b) Picture taken at the canteen of building 20

Figure B.4: Pictures taken during observations

(a) Picture taken at the canteen of building 20

Figure B.5: Pictures taken during observations

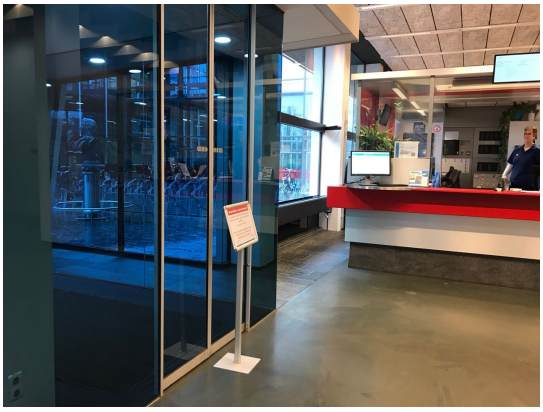
(a) Picture taken at the entrance of building 36



(b) Picture taken at the entrance of building 36

Figure B.6: Pictures taken during observations

(a) Picture taken at the entrance of building 36



(b) Picture taken at the entrance of building 36

Figure B.7: Pictures taken during observations

(a) Picture taken at the entrance of building 36

Figure B.8: Pictures taken during observations

(a) Picture taken at a lecture room in building 36

(b) Picture taken at a lecture room in building 36

Figure B.9: Pictures taken during observations

(a) Picture taken at a lecture room in building 36

(b) Picture taken at a lecture room in building 36

Figure B.10: Pictures taken during observations

(a) Picture taken at a lecture room in building 36

Figure B.11: Pictures taken during observations

C

Pepper's sensors

Further information about the sensors of Pepper is explained in this appendix. Pepper has various motors that allow it to move fluidly recreating the movements of a human being. In Figure C.1, all the motors on Pepper are shown.

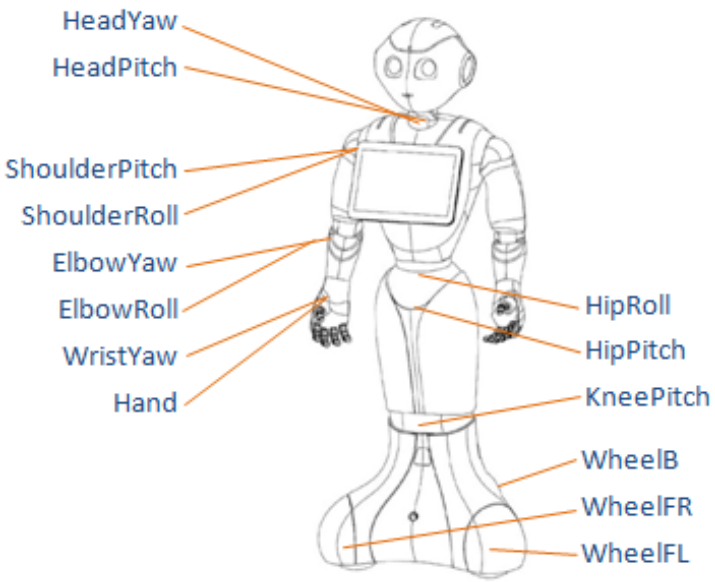


Figure C.1: Motors on Pepper

What we are mainly interested are the cameras and the microphones for detecting our features. Pepper has two 2D-cameras shown in Figure C.2 which specification are listed in Figure C.3.

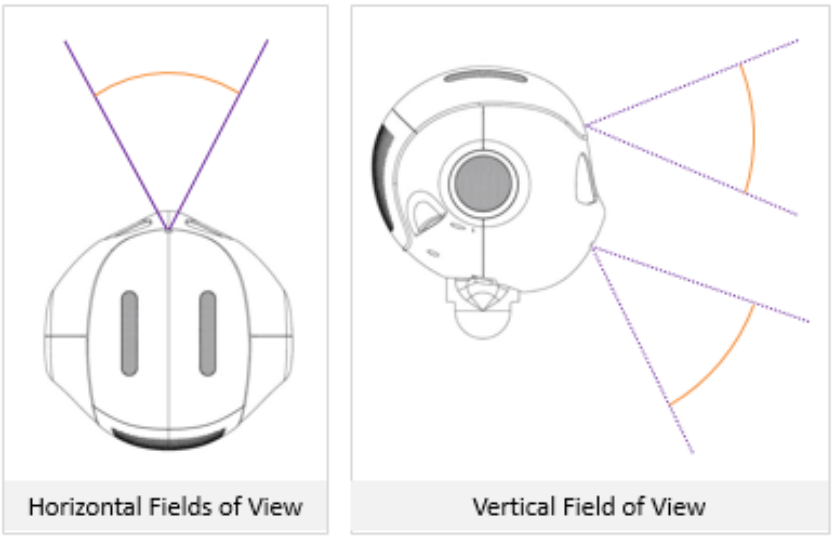


Figure C.2: Cameras of Pepper

Specifications

Camera	Model	OV5640
	Type	System-on-a-chip (SoC) CMOS image sensor
Imaging Array	Resolution	5Mp
	Optical format	1/4 inch
	Active Pixels (HxV)	2592x1944
Sensitivity	Pixel size	1.4µm*1.4µm
	Dynamic range	68db@8x gain
	Signal/Noise ratio (max)	36dB (maximum)
	Responsivity	600 mV/Lux-sec
Output	Camera output	640*480@30fps or 2560*1920@1fps
	Data Format	YUV and RGB
	Shutter type	Rolling shutter
View	Field of view	67.4°DFOV (56.3°HFOV,43.7°VFOV)
	Focus type	Auto focus

Figure C.3: Specifications of the cameras

Furthermore, Pepper has 4 microphones, which position is shown in Figure C.4. And the specifications are listed in Figure C.5.

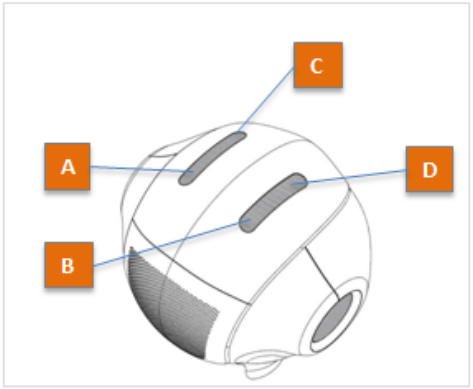


Figure C.4: Microphones of Pepper: A is the rear left one; B is the rear right one; C the front left and D the front right.

Specification

Microphones	x4 on the head
Sensitivity	250mV/Pa +/-3dB at 1kHz
Frequency range	100Hz to 10kHz (-10dB relative to 1kHz)

Figure C.5: Specifications of the microphones

Finally we use tactile sensors located in the left and right arm for stopping or resetting the algorithm. This two sensors are labelled as B in Figure C.7

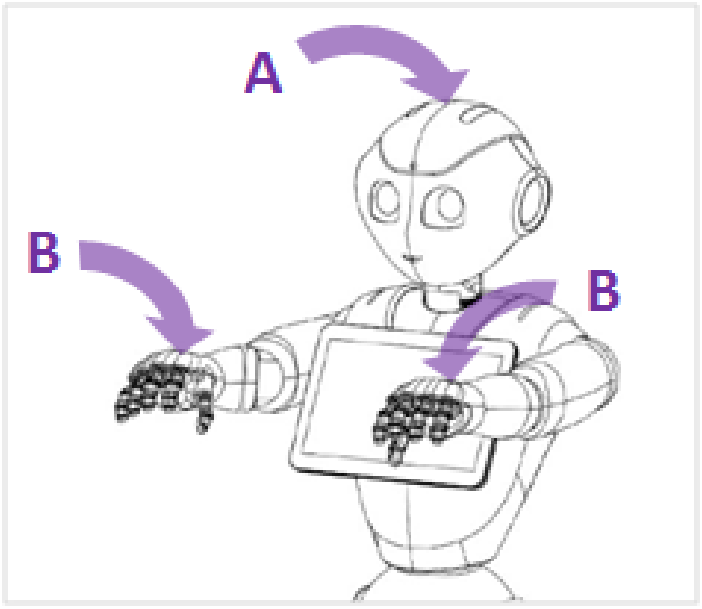


Figure C.6: Picture of the tactile sensors of Pepper.

Figure C.7: Picture of the hand tactile sensors of Pepper.

D

Bayesian Network Nodes

D.1. Feature nodes


```
{
  "features_nodes": [
    {
      "id": 1,
      "title": "Door",
      "value" : "0",
      "probability" : "0.5",
      "link_to_settings_nodes_id": [
        "1",
        "2"
      ],
      "type" : "label-object",
    },
    {
      "id": 2,
      "title": "Chair",
      "value" : "0",
      "probability" : "0.5",
      "link_to_settings_nodes_id": [
        "1",
        "2",
        "6"
      ],
      "type" : "label-object",
    },
    {
      "id": 3,
      "title": "Sofa Couch",
      "value" : "0",
      "probability" : "0.5",
      "link_to_settings_nodes_id": [
        "1",
        "2"
      ],
      "type" : "label-object",
    },
    {
      "id": 4,
```

```
    "title": "People passing by",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "1",
      "2"
    ],
    "type" : "robotAPI",
  },
  {
    "id": 5,
    "title": "Low level of noise",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "1",
      "2",
      "4",
      "6"
    ],
    "type" : "robotAPI",
  },
  {
    "id": 6,
    "title": "Desk",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "2",
      "3"
    ],
    "type" : "label-object",
  },
  {
    "id": 7,
    "title": "Laptop Computer",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "2",
      "3",
      "4"
    ],
    "type" : "label-object",
  },
  {
    "id": 8,
    "title": "Paper",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "2",
      "3",
      "4"
    ],
    "type" : "label-object",
```

```
},
{
  "id": 9,
  "title": "Pen",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "2",
    "3",
    "4"
  ],
  "type" : "label-object",
},
{
  "id": 10,
  "title": "People looking at the robot",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "1",
    "2",
    "3"
  ],
  "type" : "robotAPI",
},
{
  "id": 11,
  "title": "Projector Projection",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "3"
  ],
  "type" : "label-object",
},
{
  "id": 12,
  "title": "Monitor Display Screen",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "3"
  ],
  "type" : "label-object",
},
{
  "id": 13,
  "title": "High level of noise",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "5"
  ],
  "type" : "robotAPI",
},
{
```

```
    "id": 14,
    "title": "Food",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "5"
    ],
    "type" : "label-object",
  },
  {
    "id": 15,
    "title": "Dish Cutlery Tableware",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "5",
      "6"
    ],
    "type" : "label-object",
  },
  {
    "id": 16,
    "title": "Table",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "3",
      "5",
      "6"
    ],
    "type" : "label-object",
  },
  {
    "id": 17,
    "title": "Glasses",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "3",
      "4",
      "5",
      "6"
    ],
    "type" : "label-object",
  },
  {
    "id": 18,
    "title": "Fridge Cooker Oven Microwave Dishwasher",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "6"
    ],
    "type" : "label-object",
  },
  {
```

```
    "id": 19,
    "title": "Shelf shelving",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "6",
      "7",
      "8"
    ],
    "type" : "label-object",
  },
  {
    "id": 20,
    "title": "Low number of people",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "6"
    ],
    "type" : "robotAPI",
  },
  {
    "id": 21,
    "title": "Room",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "3",
      "4",
      "6"
    ],
    "type" : "label-object",
  },
  {
    "id": 22,
    "title": "Person present",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "1",
      "3",
      "4",
      "5"
    ],
    "type" : "label-object",
  },
  {
    "id": 23,
    "title": "Event",
    "value" : "0",
    "probability" : "0.5",
    "link_to_settings_nodes_id": [
      "3",
      "4"
    ],
    "type" : "label-object",
```

```
},
{
  "id": 24,
  "title": "Sitting",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "3",
    "4",
    "5"
  ],
  "type" : "label-object",
},
{
  "id": 25,
  "title": "Conversation",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "4",
    "5"
  ],
  "type" : "label-object",
},
{
  "id": 26,
  "title": "Eating",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "5"
  ],
  "type" : "label-object",
},
{
  "id": 27,
  "title": "Meal",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "5"
  ],
  "type" : "label-object",
},
{
  "id": 28,
  "title": "Walking",
  "value" : "0",
  "probability" : "0.5",
  "link_to_settings_nodes_id": [
    "2"
  ],
  "type" : "label-object",
},
{
  "id": 29,
```

```
      "title": "Waiting",
      "value" : "0",
      "probability" : "0.5",
      "link_to_settings_nodes_id": [
        "2"
      ],
      "type" : "label-object",
    },
    {
      "id": 30,
      "title": "Office chair",
      "value" : "0",
      "probability" : "0.5",
      "link_to_settings_nodes_id": [
        "1"
      ],
      "type" : "label-object",
    }
  ]
}
```


D.2. Setting nodes


```
{
  "settings_nodes": [
    {
      "id": 1,
      "title": "Building entrance",
      "link_to_roles_node_id": "1",
      "table": {
      }
    },
    {
      "id": 2,
      "title": "Info desk",
      "link_to_roles_node_id": "1",
      "table": {
      }
    },
    {
      "id": 3,
      "title": "Lecture",
      "link_to_roles_node_id": "2",
      "table": {
      }
    },
    {
      "id": 4,
      "title": "Meeting",
      "link_to_roles_node_id": "2",
      "table": {
      }
    },
    {
      "id": 5,
      "title": "Busy",
```

```
      "link_to_roles_node_id": "3",
      "table": {
      }
    },
    {
      "id": 6,
      "title": "Empty",
      "link_to_roles_node_id": "3",
      "table": {
      }
    }

  ]
}
```


D.3. Role nodes


```
{
  "roles_nodes": [
    {
      "id": 1,
      "title": "Receptionist",
      "table": {
      }
    },
    {
      "id": 2,
      "title": "Tutor",
      "table": {
      }
    },
    {
      "id": 3,
      "title": "Waiter",
      "table": {
      }
    }
  ]
}
```

E

Information form

Study of Scene Classification for a Mobile Interactive Robot

Delft,

The research aims to achieve awareness for a mobile interactive robot (Pepper robot by SoftBank Robotics) by understanding the surrounding context, this would lead the robot to dynamically change its behaviour.

Pictures of the surrounding will be taken for extrapolating information such as object recognition and labelling and the presence of people in the surroundings.

Main part of the research involves people perception, that can be achieved through APIs. Built-in APIs are available on the robot platform itself, while for object recognition and labelling from images, Google Cloud Visual API is used. Both platforms use visual data, with the difference that data processed by Google must be uploaded for processing.

The study aims to train the scene classifier.

Collecting data means that the pictures taken during the experiments will be stored and will be used internally at the TU Delft for research. In case of publishing a dataset for scientific purpose, permission will be asked again via the email address provided in this form.

If for any reason the participant decides to withdraw consent, images containing that person will be immediately erased.

No personal information such as name or address of the participants will be published or shared. For any further details the researcher Laura Donadoni can be contacted on the mobile at +31645269916 or via email at L.Donadoni@student.tudelft.nl.

Consent Form for study of Scene Classification for a Mobile Interactive Robot

Please tick the appropriate boxes	Yes	No
Taking part in the study		
I have read and understood the study information dated _____, or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	○	○
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason. In that case, the image containing myself will be immediately erased.	○	○
I understand that taking part in the study involves images collection that will form the dataset. This dataset can only be used internally for research at the TU Delft.	○	○
Use of the information in the study		
I understand that personal information collected about me (pictures) that can identify me, will be used internally at the TU Delft for the study.	○	○
Future use and reuse of the information by others		
I give consent to use my email address to contact me for possible future use of the collected data such as publishing a dataset.	○	○

Email adress:_____

Signatures

_____	_____	_____
Name of participant [printed]	Signature	Date

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

_____	_____	_____
Researcher name [Laura Donadoni]	Signature	Date

Study contact details for further information: [*Laura Donadoni, +31645269916, l.donadoni@student.tudelft.nl*]

F

Training experiments

In this appendix we show where the robot was placed and the direction of the sight of it during the tests. In the following drawings that represent the planimetry of the location in which the test took place, the furniture are represented by rectangles with labels, walls are represented by thick dark grey lines while entrances is when the dark grey line is missing. When people are present in the location, a light grey circle represents the location of the person, while the red symbol is the robot and the arrow indicates the direction in which the robot is looking at.

Receptionist

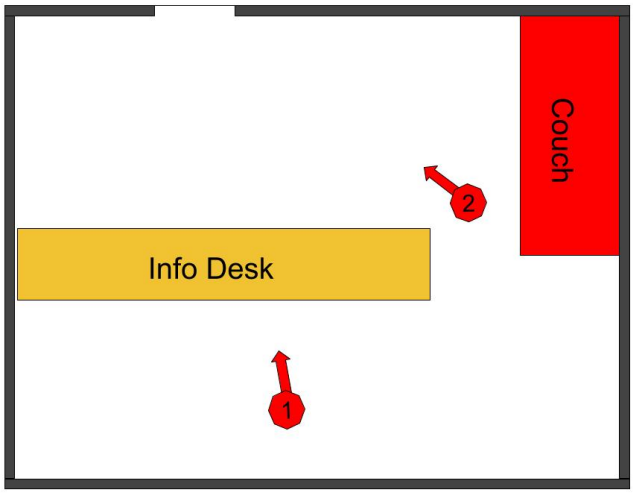


Figure F.1: Building Entrance experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represents the table, while walls are the dark grey lines on the top and bottom.

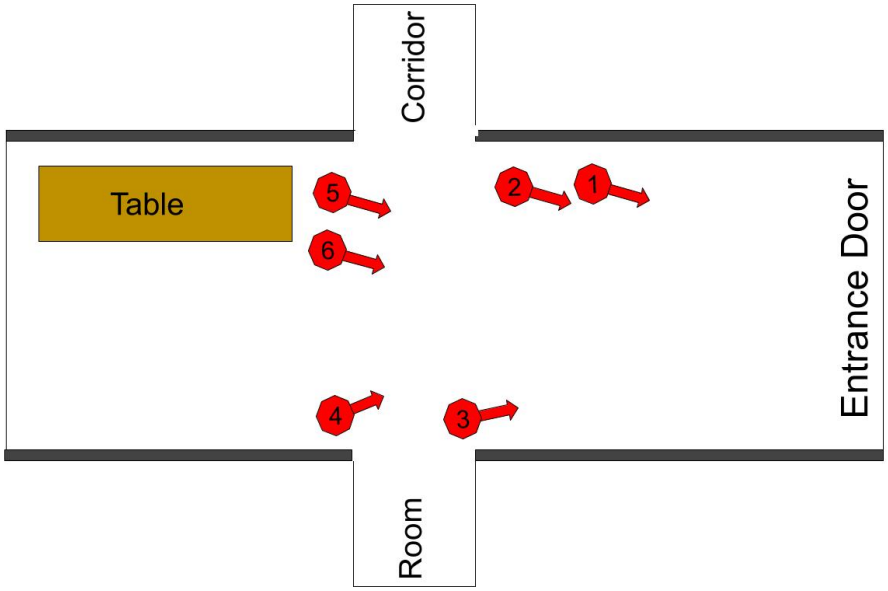


Figure F.2: Building Entrance experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represents the table, while walls are the dark grey lines on the top and bottom.

Tutor

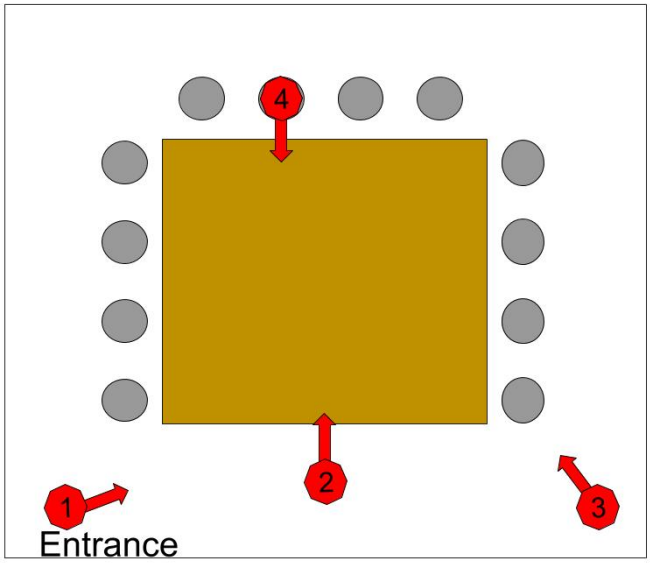


Figure F.3: Lecture experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represents the table while the grey circles are the people.

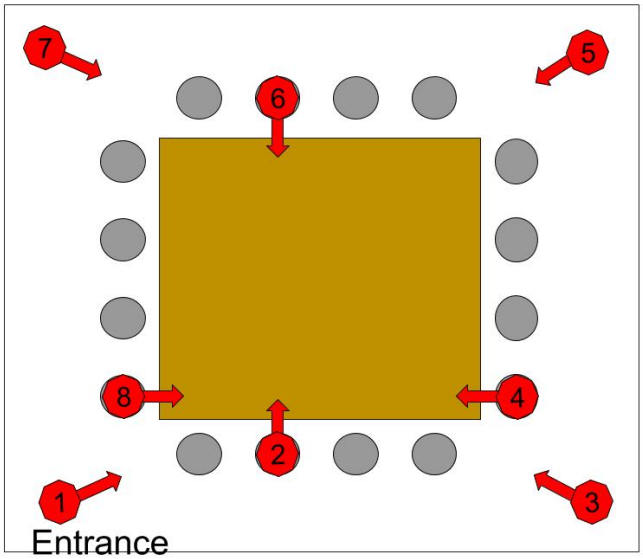


Figure F.4: Meeting experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represents the table while the grey circles are the people.

Waiter

Figure F.5: Empty restaurant experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represent the tables, while the dark grey rectangles represent the kitchen furniture.

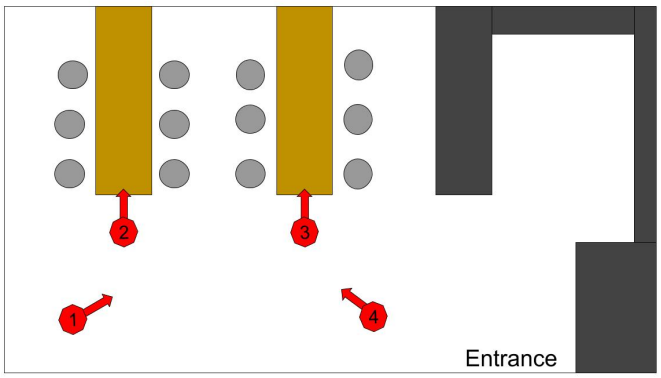


Figure F.6: Empty restaurant experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represent the tables, while the dark grey rectangles represent the kitchen furniture.

G

Data from results of the training

In this appendix we will show the results of the Bayesian Network after training. The following tables show the percentages with which every role has been classified. The data is graphically shown in Chapter 6.5.

Receptionist - Info desk

Results from the single tests:

Receptionist	Tutor	Restaurant
1.00	0.86	0.47
0.74	0.50	0.35

Table G.1: Results from the tests in the entrance building setting

Receptionist - Entrance building

Results from the single tests:

Receptionist	Tutor	Restaurant
0.61	0.53	0.76
0.66	0.37	0.51
0.61	0.53	0.50
0.61	0.53	0.76
0.81	0.37	0.35
0.68	0.61	0.35

Table G.2: Results from the tests in the entrance building setting

Tutor - Lecture

Results from the single tests:

Receptionist	Tutor	Restaurant
0.76	0.93	0.39
0.85	0.89	0.39
0.85	1.00	0.50
0.86	0.92	0.69
0.83	0.99	0.69

Table G.3: Results from the tests in the lecture setting

Tutor - Meeting

Results from the single tests:

Receptionist	Tutor	Restaurant
0.91	0.99	0.81
0.83	0.99	0.69
0.83	1.00	0.87
0.83	1.00	0.79
0.61	0.82	0.71
0.72	0.87	0.79
0.61	0.83	0.76
0.83	0.97	0.86

Table G.4: Results from the tests in the meeting setting

Waiter - Empty restaurant

Results from the single tests:

Receptionist	Tutor	Restaurant
0.37	0	0.84
0.55	0	0.76
0.36	0	0.60

Table G.5: Results from the tests in the empty restaurant setting

Waiter - Busy restaurant

Results from the single tests:

Receptionist	Tutor	Restaurant
0.72	0.73	0.69
0.72	0.87	0.94
0.61	0.82	0.97
0.72	0.96	0.90

Table G.6: Results from the tests in the busy restaurant setting

H

Validation test data

In this appendix we show where the robot was placed and the direction of the sight of it during the tests. In the following drawings that represent the planimetry of the location in which the test took place, the furniture are represented by rectangles with labels, walls are represented by thick dark grey lines while entrances is when the dark grey line is missing. When people are present in the location, a light grey circle represents the location of the person, while the red symbol is the robot and the arrow indicates the direction in which the robot is looking at.

Receptionist

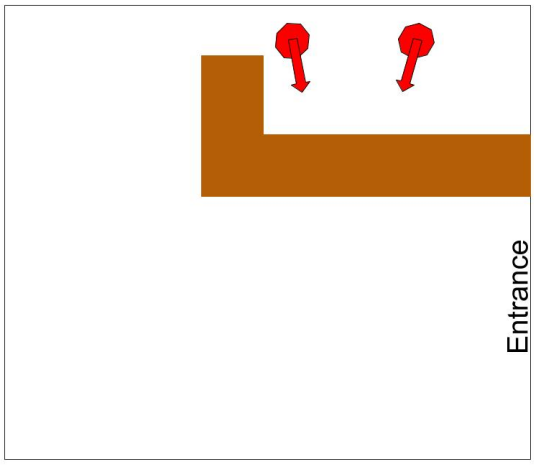


Figure H.1: Building Entrance experiments. The red symbols represent where the robot has been placed for the experiments. The brown rectangle represents the desk.

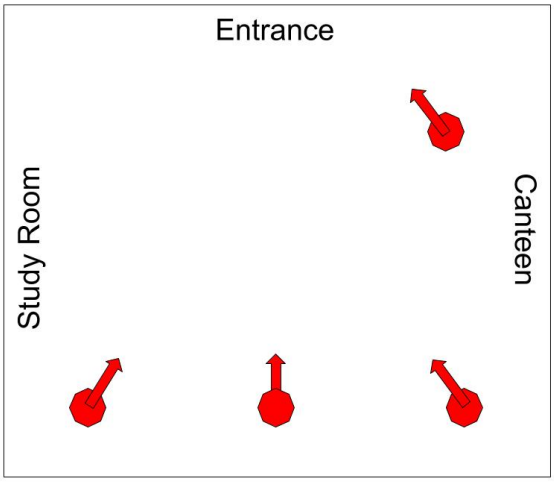


Figure H.2: Building Entrance experiments. The red symbols represent where the robot has been placed for the experiments. The labels indicates the glass doors.

Tutor

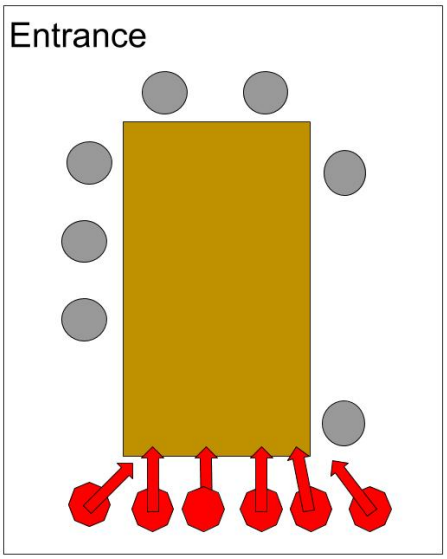


Figure H.3: Lecture experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represents the table while the grey circles are the people.

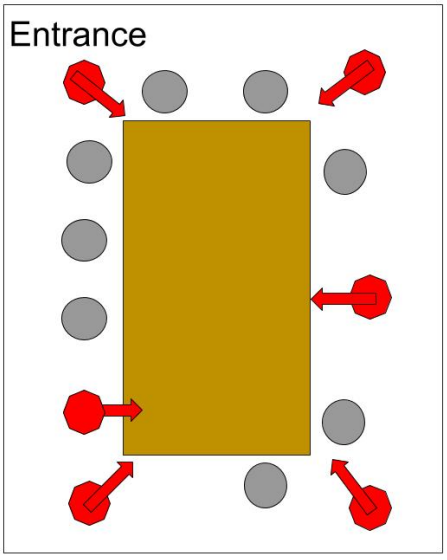


Figure H.4: Meeting experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represents the table while the grey circles are the people.

Waiter

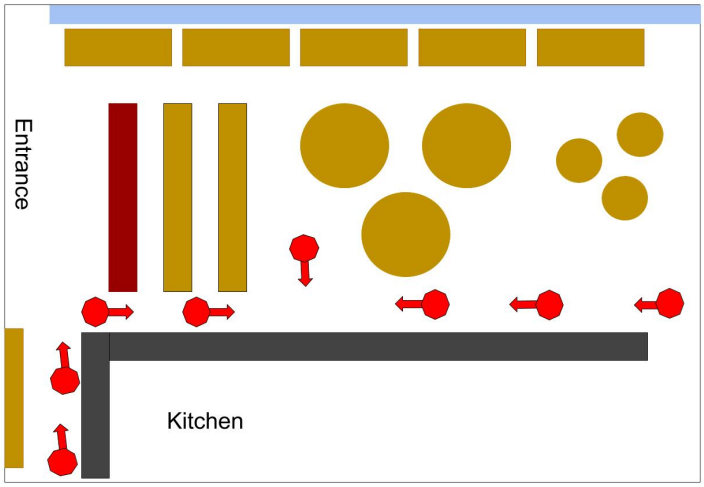


Figure H.5: Empty restaurant experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represent the tables, while the dark grey rectangles represent the kitchen furniture.

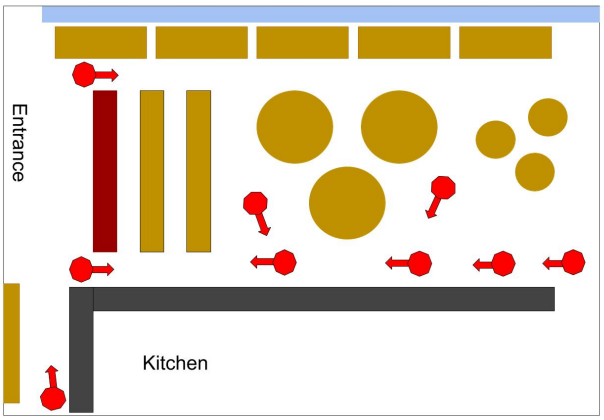


Figure H.6: Empty restaurant experiments. The red symbols represent where the robot has been placed for the experiments. The yellow rectangle represent the tables, while the dark grey rectangles represent the kitchen furniture.

Bibliography

[1] Bayesian inference. https://en.wikipedia.org/wiki/Bayesian_inference, . URL https://en.wikipedia.org/wiki/Bayesian_inference.

[2] Bayesian network. https://en.wikipedia.org/wiki/Bayesian_network, . URL https://en.wikipedia.org/wiki/Bayesian_network.

[3] Executive summary world robotics 2018 service robots. https://ifr.org/downloads/press2018/Executive_Summary_WR_Service_Robots_2018.pdf. URL https://ifr.org/downloads/press2018/Executive_Summary_WR_Service_Robots_2018.pdf.

[4] Google cloud vision api. https://cloud.google.com/vision/docs/, . URL https://cloud.google.com/vision/docs/.

[5] Google cloud vision api label detection. https://cloud.google.com/vision/docs/labels, . URL https://cloud.google.com/vision/docs/labels.

[6] Google cloud vision api object localization. https://cloud.google.com/vision/docs/object-localizer, . URL https://cloud.google.com/vision/docs/object-localizer.

[7] Naoqi apis. http://doc.aldebaran.com/2-5/naoqi/index.html, . URL http://doc.aldebaran.com/2-5/naoqi/index.html.

[8] Naoqi algazeanalysis api. http://doc.aldebaran.com/2-5/naoqi/peopleperception/algazeanalysis.html, . URL http://doc.aldebaran.com/2-5/naoqi/peopleperception/algazeanalysis.html.

[9] Naoqi alpeopleperception api. http://doc.aldebaran.com/2-5/naoqi/peopleperception/alpeopleperception.html, . URL http://doc.aldebaran.com/2-5/naoqi/peopleperception/alpeopleperception.html.

[10] Pepper robot as ice-cream seller. https://www.youtube.com/watch?v=HlHN1re4_MA, . URL https://www.youtube.com/watch?v=HlHN1re4_MA.

[11] Pepper robot as receptionist. https://www.robotsoflondon.co.uk/pepper-the-receptionist, . URL https://www.robotsoflondon.co.uk/pepper-the-receptionist.

[12] Pepper robot as waiter. https://www.mirror.co.uk/tech/pizza-hut-hires-robot-waiters-8045172, . URL https://www.mirror.co.uk/tech/pizza-hut-hires-robot-waiters-8045172.

[13] Pepper-documentation. http://doc.aldebaran.com/2-4/home_pepper.html, . URL http://doc.aldebaran.com/2-4/home_pepper.html.

[14] Pepper robot image. https://www.softbankrobotics.com/emea/en/pepper, . URL https://www.softbankrobotics.com/emea/en/pepper.

[15] Pepper tutor. https://www.youtube.com/watch?v=tBDI6kjj4nI, . URL https://www.youtube.com/watch?v=tBDI6kjj4nI.

[16] Samer Al Moubayed, Jonas Beskow, Bajibabu Bollepalli, Joakim Gustafson, Ahmed Hussen-Abdelaziz, Martin Johansson, Maria Koutsombogera, José David Lopes, Jekaterina Novikova, Catharine Oertel, et al. Human-robot collaborative tutoring using multiparty multimodal spoken dialogue. In *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 112–113. IEEE, 2014.

[17] Sumair Aziz, Muhammad Awais, Tallha Akram, Umar Khan, Musaed Alhussein, and Khursheed Aurangzeb. Automatic scene recognition through acoustic classification for behavioral robotics. *Electronics*, 8(5):483, 2019.

[18] Daniele Barchiesi, Dimitrios Giannoulis, Dan Stowell, and Mark D Plumbley. Acoustic scene classification: Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, 32(3):16–34, 2015.

[19] Anna Bosch, Xavier Muñoz, and Robert Martí. Which is the best way to organize/classify images by content? *Image and vision computing*, 25(6):778–791, 2007.

[20] May Chum, Ariel Habshush, Abrar Rahman, and Christopher Sang. Ieee aasp scene classification challenge using hidden markov models and frame based classification. *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[21] Brian Clarkson, Nitin Sawhney, and Alex Pentland. Auditory context awareness via wearable computing. *Energy*, 400(600):20, 1998.

[22] Kerstin Dautenhahn. Methodology & themes of human-robot interaction: A growing research field. *International Journal of Advanced Robotic Systems*, 4(1):15, 2007.

[23] Jill L Drury, Jean Scholtz, and Holly A Yanco. Awareness in human-robot interactions. In *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, volume 1, pages 912–918. IEEE, 2003.

[24] Mica R Endsley, Daniel J Garland, et al. Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, 1:24, 2000.

[25] Antti J Eronen, Vesa T Peltonen, Juha T Tuomi, Anssi P Klapuri, Seppo Fagerlund, Timo Sorsa, Gaëtan Lorho, and Jyri Huopaniemi. Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):321–329, 2005.

[26] Pablo Espinace, Thomas Kollar, Nicholas Roy, and Alvaro Soto. Indoor scene recognition by a mobile robot through adaptive object detection. *Robotics and Autonomous Systems*, 61(9):932–947, 2013.

[27] Giovanni Maria Farinella, Daniele Ravì, Valeria Tomaselli, Mirko Guarnera, and Sebastiano Battiato. Representing scenes for real-time context classification on mobile devices. *Pattern Recognition*, 48(4):1086–1100, 2015.

[28] Juan Fasola and Maja J Matarić. A socially assistive robot exercise coach for the elderly. *Journal of Human-Robot Interaction*, 2(2):3–32, 2013.

[29] Clement Fredembach, Michael Schroder, and Sabine Susstrunk. Eigenregions for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1645–1649, 2004.

[30] Jrgen T Geiger, Bjoern Schuller, and Gerhard Rigoll. Recognising acoustic scenes with large-scale audio feature extraction and svm. *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[31] Mohamed Walid Ben Ghezala, Amel Bouzeghoub, and Christophe Leroux. Rsaw: A situation awareness system for autonomous robots. In *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 450–455. IEEE, 2014.

[32] Michael A Goodrich, Alan C Schultz, et al. Human–robot interaction: a survey. *Foundations and Trends® in Human–Computer Interaction*, 1(3):203–275, 2008.

[33] Yoonchang Han and Kyogu Lee. Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation. *arXiv preprint arXiv:1607.02383*, 2016.

[34] Kyong Il Kang, Sanford Freedman, Maja J Mataric, Mark J Cunningham, and Becky Lopez. A hands-off physical therapy assistance robot for cardiac patients. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 337–340. IEEE, 2005.

[35] Bela Krishnamurthy and J Evans. Helpmate: A robotic courier for hospital use. In *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1630–1634. IEEE, 1992.

[36] Dilip Kumar Limbu, Yeow Kee Tan, and Lawrence TC Por. Fusionbot: a barista robot-fusionbot serving coffees to visitors during technology exhibition event. In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, pages 341–342. IEEE Press, 2010.

[37] Hirokazu Madokoro, Yuya Utsumi, and Kazuhito Sato. Scene classification using unsupervised neural networks for mobile robot vision. In *SICE Annual Conference (SICE), 2012 Proceedings of*, pages 1568–1573. IEEE, 2012.

[38] Ali Meghdari, Minoo Alemi, Mobin Khamooshi, Ali Amoozandeh, Azadeh Shariati, and Behrad Mozafari. Conceptual design of a social robot for pediatric hospitals. In *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, pages 566–571. IEEE, 2016.

[39] Aastha Nigam and Laurel D Riek. Social context perception for mobile robots. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3621–3627. IEEE, 2015.

[40] Ryuichi Nisimura, Takashi Uchida, Akinobu Lee, Hiroshi Saruwatari, Kiyohiro Shikano, and Yoshio Matsumoto. Aska: receptionist robot with speech dialogue system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1314–1319. IEEE, 2002.

[41] Maria Francesca O'Connor and Laurel D Riek. Detecting social context: A method for social event classification using naturalistic multimodal data. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 3, pages 1–7. IEEE, 2015.

[42] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[43] Seungyup Paek and Shih-Fu Chang. A knowledge engineering approach for image classification based on probabilistic reasoning systems. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532)*, volume 2, pages 1133–1136. IEEE, 2000.

[44] Chenoshu Park, Jaehong Kim, and Ji-Hoon Kang. Robot social skills for enhancing social interaction in physical training. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 493–494. IEEE, 2016.

[45] Vesa Peltonen, Juha Tuomi, Anssi Klapuri, Jyri Huopaniemi, and Timo Sorsa. Computational auditory scene recognition. In *Acoustics, speech, and signal processing (icassp), 2002 IEEE international conference on*, volume 2, pages II–1941. IEEE, 2002.

[46] Flávio Garcia Pereira, Raquel Frizera Vassallo, and Evandro Ottoni Teatini Salles. Human–robot interaction and cooperation through people detection and gesture recognition. *Journal of Control, Automation and Electrical Systems*, 24(3):187–198, 2013.

[47] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009.

[48] Raquel Ros, Alexandre Coninx, Yiannis Demiris, Georgios Patsis, Valentin Enescu, and Hichem Sahli. Behavioral accommodation towards a dance robot tutor. In *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 278–279. IEEE, 2014.

[49] Christian Siagian and Laurent Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE transactions on pattern analysis and machine intelligence*, 29 (2):300–312, 2007.

[50] Martin Szummer and Rosalind W Picard. Indoor-outdoor image classification. In *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*, pages 42–51. IEEE, 1998.

[51] Iwan Ulrich and Illah Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 1023–1029. Ieee, 2000.

[52] Holly A Yanco and Jill Drury. ” where am i?” acquiring situation awareness using a remote robot platform. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 3, pages 2835–2840. IEEE, 2004.