Computing Science and Mathematics
University of Stirling

Contextual Understanding for Intent Detection on an Embedded System

Daniel Rooney
2535156

Supervisor: Dr. Deepayan Bhowmik

Dissertation Outline

October 2019

Chapter 1

Introduction

The purpose of this project is to allow pre-existing software, based on an embedded system, to identify objects and their actions and determine their intent via a visual input. This is to be achieved with the use of advanced scene segmentation and intelligent neural networks. By definition, "Intent detection" is the ability to recognise, classify, and confirm an object/persons intent, whether it be through a set of actions, tone of voice or body language. Intent detection and computer vision are therefore inevitably linked. Computer vision applications using intent detection allow users to determine the direction of a situation before it begins, potentially creating safer conditions, increased efficiency, and easier accomplishment of tasks. There are several instances of Computer Vision applications being used in the present, such as face recognition, object tracking and standardised object identification. Applications specific to intent detection are discussed and analysed in the following *state of the art* section.

1.1 Background and Context

The problem to be addressed within the project is the inability of the current software to intuitively determine the behaviour/purpose of an identifiable object. This is something that cannot be achieved without building and identifying contextual elements (whether or not these come from pre-trained CNN with predetermined classes or data sets) around said objects. Potential issues that may arise within the project are primarily linked to the ability of the software to identify separate objects around a subject of interest, and successfully classify/categorise these objects individually or in a group using a pre-trained identifier from existing data sets. Furthermore the potential to be able to construct a possible model around a subject (3D or otherwise), to identify any "hidden objects" that may be essential to determining the behaviour or context of a subject and/or object within its surrounding environment, could prove exceptionally difficult within the projects scope, however feasible this potential method may be.

1.2 Scope and Objectives

1.2.1 Scope

The scope of this project is to be able to implement a method that allows the software to analyse and and identify certain contextual cues and behavioural sequences carried out by objects/individuals using activity recognition techniques and semantic scene segmentation, allowing the the software to make a conclusion on the intent of a specific person/object, determine the contextual surroundings of a situation and produce a result that can be categorised as either, no threat, potential threat or direct threat. All of this is to be carried out on an embedded system that can be ran at a relatively low level. The direct results gained from this project allow the operator quickly assess a situation that my be potentially dangerous, and react accordingly, potential preventing harm to themselves or equipment.

1.2.2 Objectives

The objectives that make up the scope of this project can be broken down into three main entities. Firstly, the software must be able to correctly identify and track an object or an individual. Secondly, the software must be able to conclude a contextual understanding of an identified object/person, either by classifying an objects surroundings in relation to that object, or identifying specific activities around an area of interest. Thirdly, the software must be able to recognise a sequence of behaviours or movements that an object or person(s) is undertaking and predict/evaluate the intent of said person within a short amount of time. Recognising activities based on training from previous data-sets, identifying potential objects of threat and comparing them to the context of the situation as well as the object/persons behaviours at that time.

Chapter 2

State-of-The-Art

Discuss the work of others in the same area as your project. Show critical awareness of what others have done, and how you hope to extend or complement existing capabilities.

It is important to write a *critical* literature review that identifies gaps in current solutions and that clearly shows how the project was driven to address these gaps. This chapter should therefore feed into well-defined requirements for the project. Avoid a banal description of related work that does not carefully analyse its strengths and weaknesses.

This section may also include a discussion of relevant technologies and a critical analysis of which technologies you plan to use for the project.

Give references to other work by using citations like [4]. Use the *cite* command to cite references. Books [1], standards [2], reports [3], journal articles [6], conference papers [4], and web pages [5] are conventionally presented in slightly different ways. If a web page does not have a date, you should give the date on which you consulted it.

Citations are created with a *thebibliography* environment and *bibitem* commands in a '.bbl' file. Unless you are willing to invest time in creating BibTEX bibliographies, you can do this by hand.

Chapter 3

Problem description and analysis

3.1 Problem Description

The existing platform is capable of identifying objects and labelling them when they are present in the camera view. However, these objects/individuals are all treated as a person of interest and their intent is not known. For example, an old person walking across a street and minding their own business is not considered a person of interest in this scenario. However, an individual peering from the top window of a building with an object in their hand(s) may incur a point of interest to the user. The classification of a potential threat can only be positively determined by context, or a sequence of behaviours that may amount to some sort of action(s) that could be deemed a threat based on previous datasets. Therefore the problem we are facing is the inability of the software to be able to understand the context of a situation and the several factors involved that may lead to a reasonable conclusion of what is happening or about to happen at that point in time.

3.2 Problem Analysis

3.3 Constraints

3.4 User Stories

3.5 Requirements

3.5.1 Functional Requirements

3.5.2 Additional Requirements

3.6 Assumptions

3.7 Approach

3.8 First Section

Subdivide your text into sections with the *section* command.

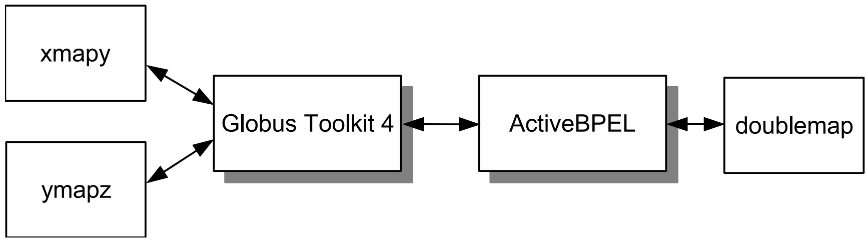

Figure 3.1: Highly Technical Diagram

3.8.1 First Subsection

If necessary, also use subsections. Subsections are entered using the *subsection* command.

First Subsubsection

If you really need subsubsections, enter these using the *subsubsection* command.

Figures are created with the *figure* environment, while tables are created with the *table* environment. They are identified by the *label* command, and are referenced by the *ref* command. Graphics are inserted with the *graphic* command. Captions are entered using the *caption* command. As an example of a figure, consider figure 3.1.

The native format for LaTeX graphics is EPS (Encapsulated PostScript). Graphical editors are usually capable of producing EPS. When outputting to PDF (Portable Document Format), the native graphics format is also PDF. Conversion of EPS to PDF is supported by a number of TeX toolsets.

Chapter 4

Project Plan

Give an updated plan for the remainder of the project. This should identify the remaining pieces of work (deliverables) and when you plan to complete them. In particular, describe what you will demonstrate when the final report is due.

References

[1] D. Greene and P. C. Williams. *Linear Accelerators for Radiation Therapy.* IOP Publishing Ltd., Bristol and Philadelphia, 1997.

[2] ISO/IEC. *Information Processing Systems – Open Systems Interconnection – LOTOS – A Formal Description Technique based on the Temporal Ordering of Observational Behaviour.* ISO/IEC 8807. International Organization for Standardization, Geneva, Switzerland, 1989.

[3] J. Jacobson and O. Andersen. Software controlled medical devices. Technical Report SP-Rapport 1997:11, European Network of Clubs for Reliability and Safety of Software, Apr. 1997. ISBN 91-7848-669-6.

[4] Ji He and K. J. Turner. Specification and verification of synchronous hardware using LOTOS. In J. Wu, S. T. Chanson, and Q. Gao, editors, *Proc. Formal Methods for Protocol Engineering and Distributed Systems (FORTE XII/PSTV XIX)*, pages 295–312, London, UK, Oct. 1999. Kluwer Academic Publishers.

[5] K. J. Turner. World-wide Environment for Learning LOTOS. *http://www.cs.stir.ac.uk/well/*, June 2000.

[6] K. J. Turner. Representing and analysing composed web services using CRESS. *Network and Computer Applications*, 30(2):541–562, Apr. 2007.