# Software Requirements Specification

# for

# Raspberry pi based Surveillance Robot for Real Time Intrusion Detection and Tracking

Prepared by Aniruddh Sanjeev Bhagwat, Chouhan Manav Kamlesh, Kshirsagar Rohit Suryakant, Prashant Kumar

Sinhgad College of Engineering

Table of Contents

Table of Contents [ii](#__RefHeading___Toc441230970)

1. Introduction [1](#__RefHeading___Toc441230972)

1.1 Purpose [1](#__RefHeading___Toc441230973)

1.2 Document Conventions [1](#__RefHeading___Toc441230974)

1.3 Intended Audience and Reading Suggestions [1](#__RefHeading___Toc441230975)

1.4 Product Scope [1](#__RefHeading___Toc441230976)

1.5 References [1](#__RefHeading___Toc441230977)

2. Overall Description [2](#__RefHeading___Toc441230978)

2.1 Product Perspective [2](#__RefHeading___Toc441230979)

2.2 Product Functions [2](#__RefHeading___Toc441230980)

2.3 User Classes and Characteristics [2](#__RefHeading___Toc441230981)

2.4 Operating Environment [2](#__RefHeading___Toc441230982)

2.5 Design and Implementation Constraints [2](#__RefHeading___Toc441230983)

2.6 User Documentation [2](#__RefHeading___Toc441230984)

3. External Interface Requirements [3](#__RefHeading___Toc441230986)

3.1 User Interfaces [3](#__RefHeading___Toc441230987)

3.2 Hardware Interfaces [3](#__RefHeading___Toc441230988)

3.3 Software Interfaces [3](#__RefHeading___Toc441230989)

3.4 Communications Interfaces [3](#__RefHeading___Toc441230990)

4. System Features [4](#__RefHeading___Toc441230991)

4.1 System Feature 1 [4](#__RefHeading___Toc441230992)

4.2 System Feature 2 (and so on) [4](#__RefHeading___Toc441230993)

5. Other Nonfunctional Requirements [4](#__RefHeading___Toc441230994)

5.1 Performance Requirements [4](#__RefHeading___Toc441230995)

5.2 Safety Requirements [5](#__RefHeading___Toc441230996)

5.3 Security Requirements [5](#__RefHeading___Toc441230997)

5.4 Software Quality Attributes [5](#__RefHeading___Toc441230998)

5.5 Business Rules [5](#__RefHeading___Toc441230999)

6. Other Requirements [5](#__RefHeading___Toc441231000)

Appendix A: Glossary [5](#__RefHeading___Toc441231001)

Appendix B: Analysis Models [5](#__RefHeading___Toc441231002)

Appendix C: To Be Determined List [6](#__RefHeading___Toc441231003)

# Introduction

## Purpose

This document details the working plan for “Raspberry pi based Surveillance Robot for Real Time Intrusion Detection and Tracking”. It also specifies it’s requirements, functioning, technology used and an overall description of the project to the reader.

## Document Conventions

There was no explicit set of standards or conventions in making of this document. The references are marked at each use.

## Intended Audience and Reading Suggestions

The document is intended for every stakeholder involved in the project from developer to end-user.

## Product Scope

The aim of the project is to create an autonomous robot capable of human detection and surveillance. The robot must have the ability to connect with the user over telegram app and send photo and video. It will have ability to follow a person, receive and follow command of user.

## References

[1] Rahul Madbhavi, Shreepad Potadar, Sylvester Jerome D’souza, “Object Follower and Barrier Escaping Robot Using Image Processing”, IJIRSET 2015.

[2] Kirti Bhagat, Sayalee Deshmukh, Shraddha Dhonde, Sneha Ghag, “Obstacle Avoidance Robot”, IJIRSET 2016.

[3] Rupa Gurram, SweathaSuresh.B., Sneha.B.R., Sushmitha.R, “Object Tracking Robot on Raspberry Pi using Opencv”, IJETT 2016.

[4] Chinmayi R, Yogesh kumar Jayam, Venkatesh Tunuguntla, Jaideep venkat Dammuru, Harshith Nadella, “Obstacle Detection and Avoidance Robot”, IEEE 2018

[5] https://towardsdatascience.com/ssd-single-shot-detector-for-object-detection-using-multibox-1818603644ca

# Overall Description

## Product Perspective

The product is a stand alone robot which is being developed to ensure security of a house. It is a raspberry pi based module with a camera ultrasonic sensor mounted on it. It is navigable, the wheels are moved by the motors installed which are powered by a battery.

## Product Functions

The major functions of the robot are:

1. Identification: Identifying whether the object seen in the camera is human or not. If it is human send it’s photo to user on telegram.
2. Following: Follow the person after sending photo.
3. Sending video: send a video as and when instructed by user.
4. Return to original position when instructed.

## User Classes and Characteristics

The model is easy to use implement and run. It requires to be executed only once, it runs till battery becomes low. Anyone with knowledge of using a mobile phone can use it.

## Operating Environment

It is a raspberry-pi 3b+ based model and has raspbian OS installed in it. The raspberry-pi based model is the robot unit. The telegram app serves as an user interface between model and the user.

## Design and Implementation Constraints

There are a few hardware limitations such as limited battery size, processing speed of raspberrypi and speed of following.

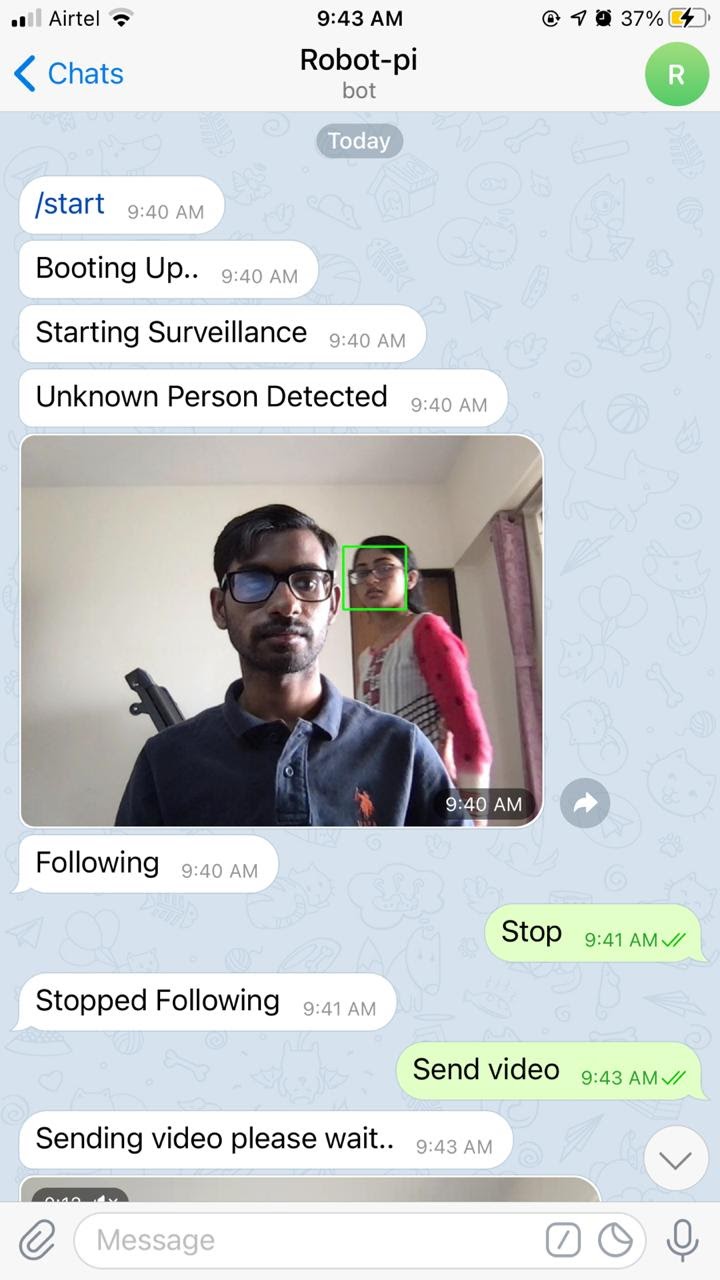
## User Documentation

The demonstration given during introduction of model shall be sufficient for user to understand the process of implementation.

# External Interface Requirements

## User Interfaces

Telegram serves as the user interface.it is the intermediary between user and raspberry pi.



User

Telegram

Processing unit

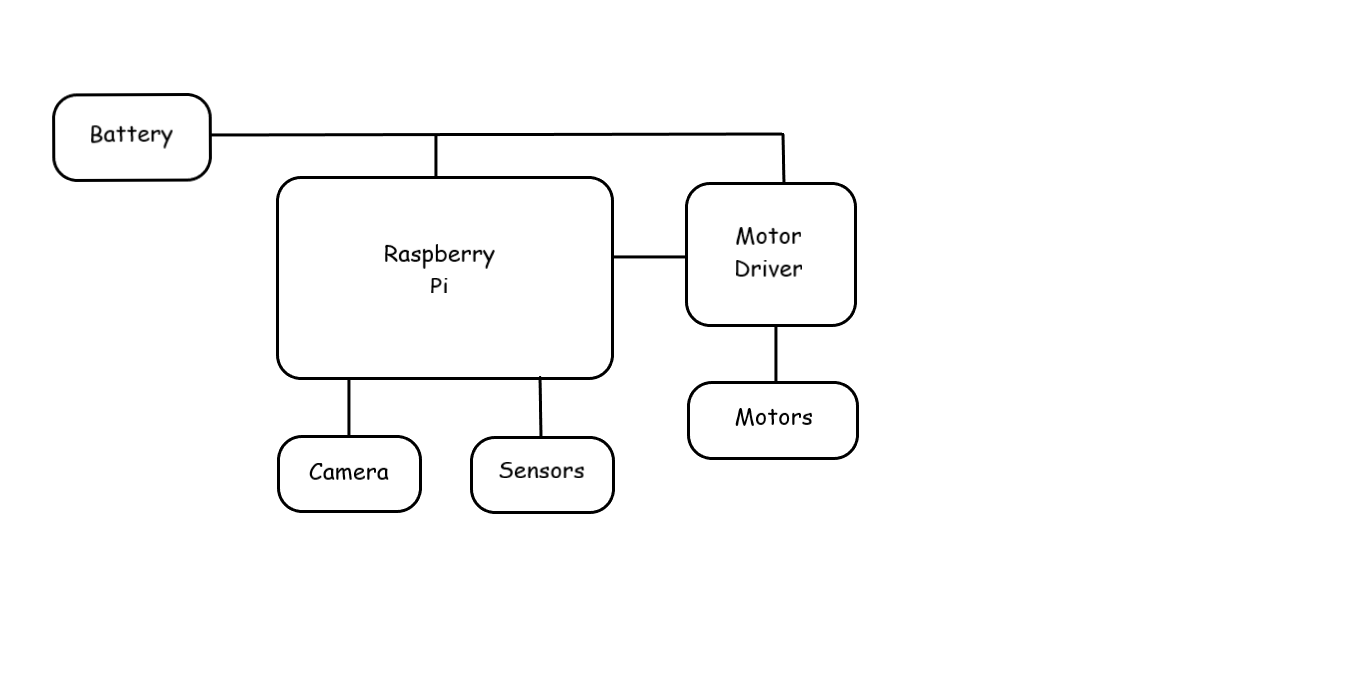
## Hardware Interfaces

The following are the hardware components in the robot:

Raspberry Pi 3B

1. Ultrasonic Sensor Module HC-SR04
2. ROBOT Chassis
3. Wheels, DC Motors
4. Raspberry pi 5MP camera
5. Bread Board and Resistor (1k)
6. Motor Driver L298 2A
7. Connecting wires and Power supply or Power bank

The physical architecture is as follows:



## Software Interfaces

The processing of the data takes place on the processing unit(in this case a laptop), the raspberry pi sends data to the processing unit. Various libraries of python namely cv2, telegram, socket and imutils are used by the processing unit to process the data. Cv2 library is used for face detection and recognition, telegram library is used for communication with the user. Socket library is used to establish UDP connection between raspberry pi and the processing unit. Imutils library is used to perform basic image processing functions such as translation, rotation, resizing, skeletonization, displaying.

## Communications Interfaces

There is a UDP connection between the raspberry pi and the processing unit(in this case laptop), the processing of the data takes place on the laptop.

# System Features

## Security

### Description

The main motive behind this entire project is to provide security to the user. The environment in which the robot is being deployed will be under constant surveillance. As this feature is of highest priority, any action that may lead to a compromise must be completely avoided.

**4.1.2 Functional Requirements**

REQ-1: System should be continuously on active state.

REQ-2: Alerting the user of a potential threat.

## Object & Human Detection

### Description

Object Detection plays a vital role in our system. This feature is responsible for invoking the detection algorithm. The detection algorithm is run on each input frame. This will inhibit the system from meeting its real time requirements. The location of the human targets is determined by tracking the detected humans using the tracking algorithm.

Human detection is a difficult task from a machine vision perspective as it is influenced by a wide range of possible appearance due to changing articulated pose, clothing, lighting and background, but prior knowledge on these limitations can improve the detection performance.

### Functional Requirements

The algorithm used for Object Detection is Single Shot Detection Algorithm. Single Shot detector like YOLO takes only one shot to detect multiple objects present in an image using multi box.

It is significantly faster in speed and high-accuracy object detection algorithm. A quick comparison between speed and accuracy of different object detection models on VOC2007

* SSD300 : 59 FPS with mAP 74.3%
* SSD500 : 22FPS with mAP 76.9%
* Faster R-CNN : 7 FPS with mAP 73.2%
* YOLO : 45 FPS with mAP 63.4%

High speed and accuracy of SSD using relatively low resolution images is attributed due to following reasons:

* Eliminates bounding box proposals like the ones used in RCNN’s
* Includes a progressively decreasing convolutional filter for predicting object categories and offsets in bounding box locations.

High detection accuracy in SSD is achieved by using multiple boxes or filters with different sizes, and aspect ratio for object detection. It also applies these filters to multiple feature maps from the later stages of a network. This helps perform detection at multiple scales.

## Tracking

### Description

The tracking of the object is based on division of the image into virtual grids.

The movement of the robot is based on the position of the object in the grid.

The path chosen may not always be the optimal path, but it provides faster response to unexpected obstacles. This system can simultaneously track a moving object and avoid obstacles in real time.

**4.3.2 Functional Requirements**

The robotic platform uses a visual camera to sense the movement of the preferred object and a range sensor to help the robot to notice and then avoid barriers in real time while continuing to track and follow the desired object.

## Communication

### Description

Robotic communication is an area of research that focuses on the human interaction with robots in a variety of contexts. Most commonly, this area of research is referred to as human-robot interaction (HRI). The purpose of HRI is to understand how humans and robots interact and the ways to improve the usability of robotic communication. Interaction through a screen, often a touch screen or a command, allows to convey quite a lot of information, either textual or visual.

To communicate with our system we have used Telegram as a medium for interaction. Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and inline requests. You control your bots using HTTPS requests to our Bot API.

### Functional Requirements

At the core, Telegram Bots are special accounts that do not require an additional phone number to set up. Users can interact with bots in two ways:

* Send messages and commands to bots by opening a chat with them or by adding them to groups.
* Send requests directly from the input field by typing the bot's @username and a query.

This allows sending content from inline bots directly into any chat, group or channel.

Messages, commands and requests sent by users are passed to the software running on your servers. Our intermediary server handles all encryption and communication with the Telegram API for you. You communicate with this server via a simple HTTPS-interface that offers a simplified version of the Telegram API. We call that interface our Bot API.

# Other Nonfunctional Requirements

## Performance Requirements

* The performance of the system depends highly on accuracy of the sentiment analysis and machine learning algorithms.
* Also, the system depends on efficiency of the hardware components.
* System is also dependent on the various user’s command for performing its further operations.

## Safety Requirements

## Security Requirements

* Only the authenticated user should be able to send commands to the system.
* The images/videos captured from the system should be sent to the appropriate user only.
* Anyone other than the concerned user should not be able to intercept the communication.

## Software Quality Attributes

* Adaptability: The system can adapt to it’s environment and work accordingly.
* Correctness: The system makes accurate calculations about the image obtained in the frame and can also operate according to the commands given by the user.
* Simplicity: The system is easy to understand and operate.
* Portability: Due to factors like lightweight and mostly wireless connections it is portable.
* Inexpensive: The cost of the building the system is relatively low.

## Business Rules

Only the user who is authenticated can handle and use the system. The scaling or extension of system is only done by developers according to the need of users.

# Other Requirements

Appendix A: Glossary

* SSD - Single Shot Detection
* CNN - Convolutional Neural Network
* R-CNN - Region based Convolutional Neural Network
* FPS - Frames Per Second
* YOLO - You Only Look Once
* HRI - Human Robot Interaction

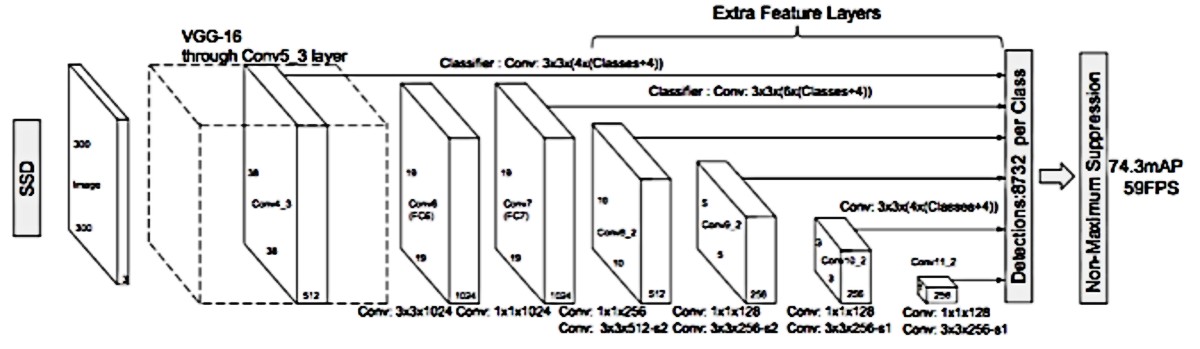
Appendix B: Analysis Models

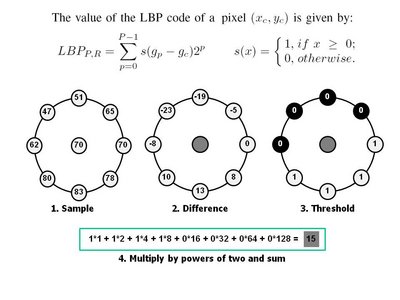
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Scoring** | **Different or Better** | **Delivers Value** | **Doable, Practical** | **Potential for early adopters** | **Score** |
| Person Identification, Communication & Tracking Robot | H | M | H | H | 11 |
| Domestic Help Robot | M | H | L | M | 8 |
| Home Automation Robot | L | H | M | H | 9 |

**Idea Matrix**

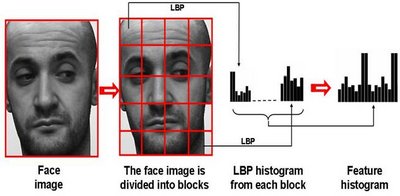
**Score Classification**

|  |  |
| --- | --- |
| **Score Category** | **Score** |
| H | 3 |
| M | 2 |
| L | 1 |

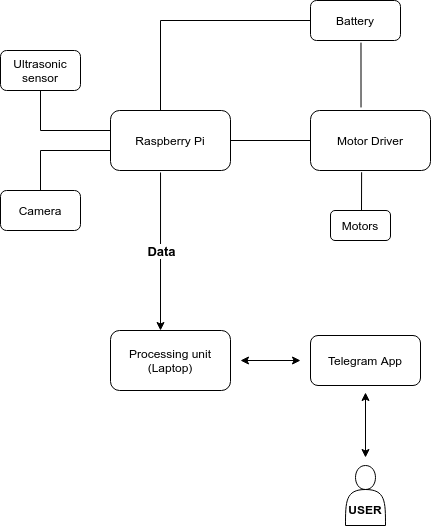


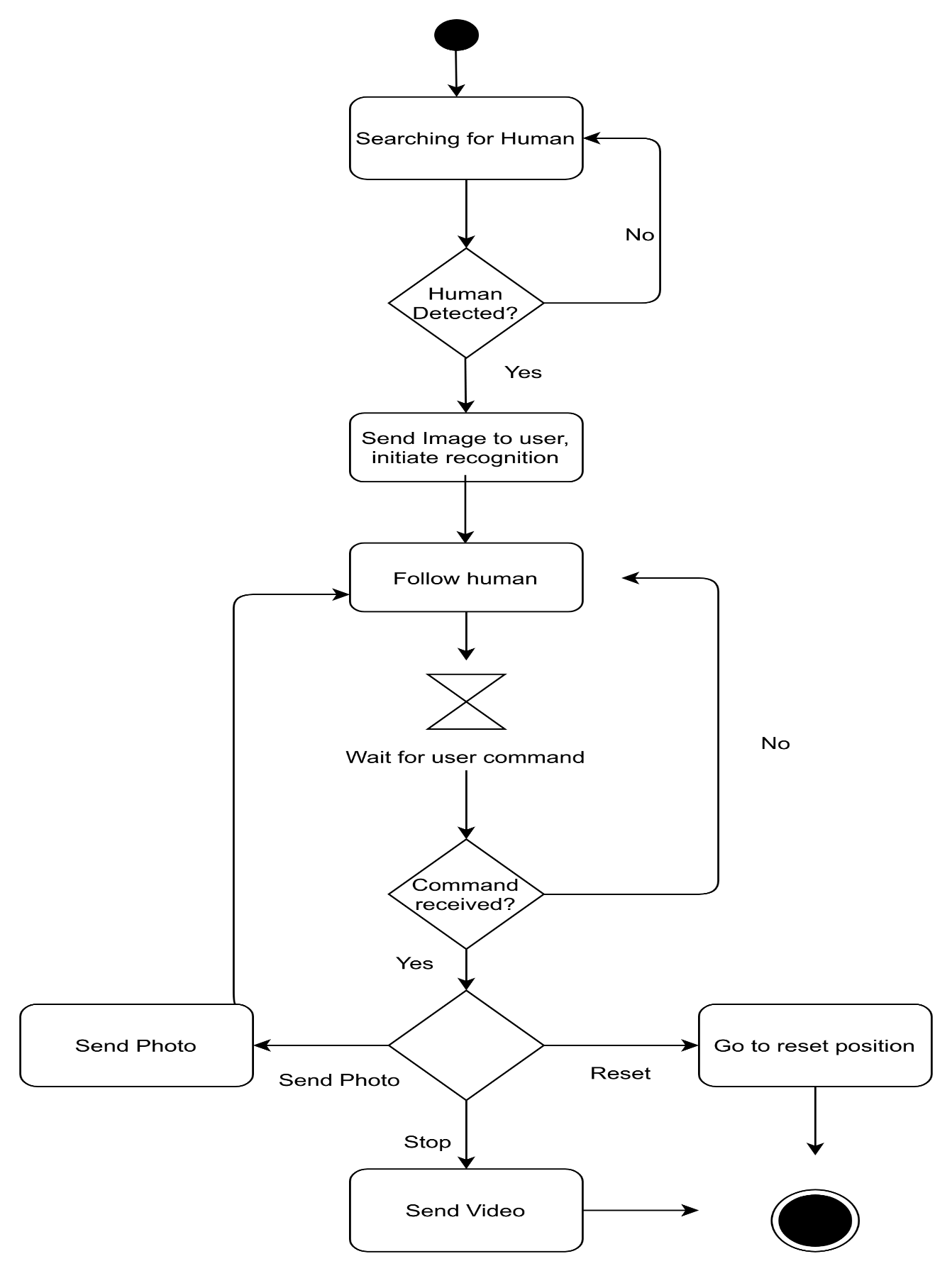
Architecture of single shot detection algorithm

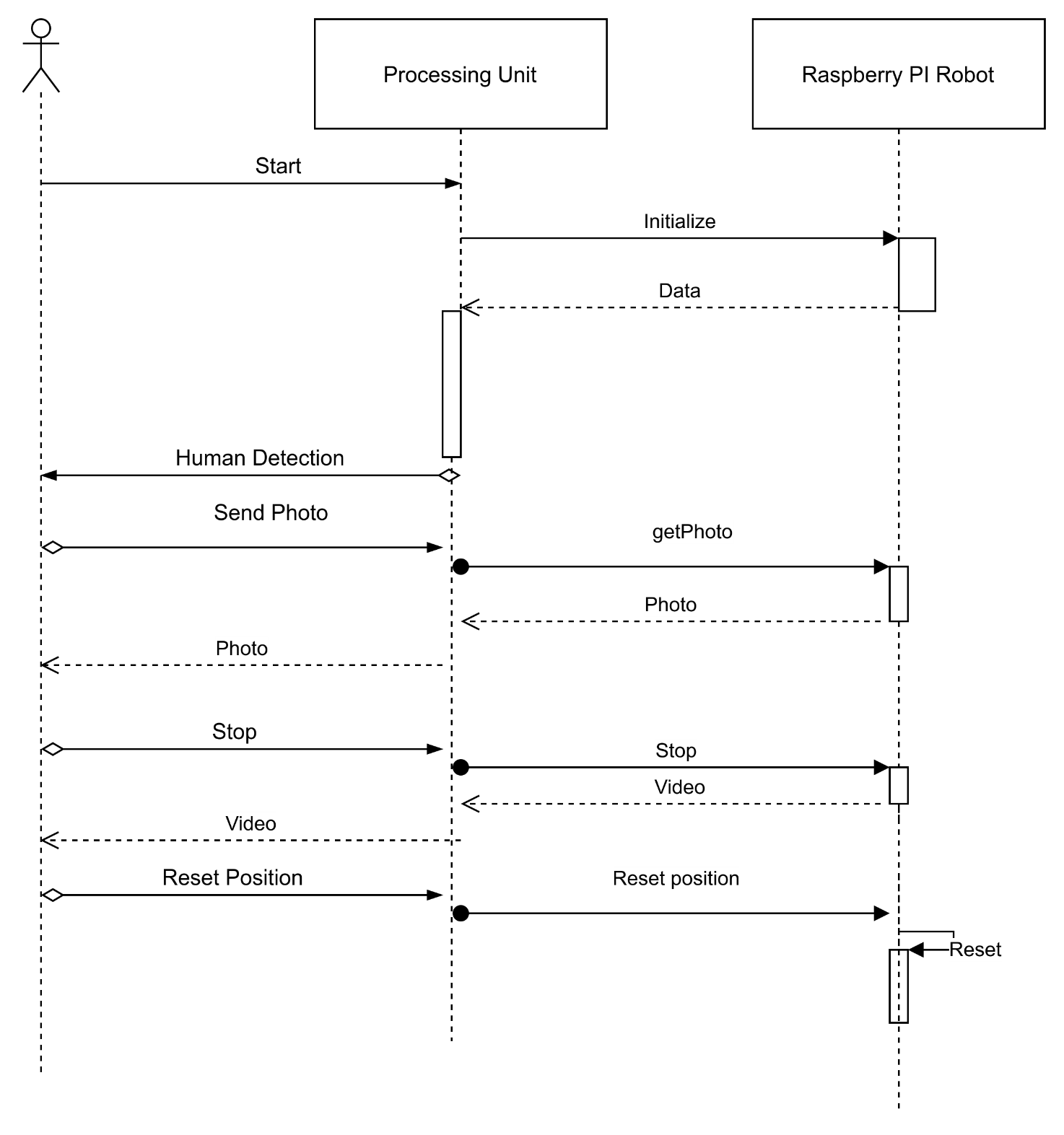
An example of LBP computation



Face description with local binary patterns







Appendix C: To Be Determined List

Source: http://www.frontiernet.net/~kwiegers/process\_assets/srs\_template.doc