2024

Group ViitTomakieli

Kandel Sushila, Lipponen Seppo, Viitala Jussi & Varpula Teemu

[kandel.sushila242@gmail.com](mailto:kandel.sushila242@gmail.com)

[seppo.lipponen@outlook.com](mailto:seppo.lipponen@outlook.com)

[juzu.viita@protonmail.com](mailto:juzu.viita@protonmail.com)

[teemu.varpula@gmail.com](mailto:teemu.varpula@gmail.com)

Table of Contents

[Work Estimates 2](#_Toc160133392)

[Change Log 2](#_Toc160133393)

[Project Version and Deadline 2](#_Toc160133394)

[Introduction 3](#_Toc160133395)

[Project overview 3](#_Toc160133396)

[Hardware Requirements 3](#_Toc160133397)

[Software Requirements: 3](#_Toc160133398)

[Functional requirements 3](#_Toc160133399)

[General description 3](#_Toc160133400)

[Components and functions 3](#_Toc160133401)

[Image Capturing 3](#_Toc160133402)

[Image recognition 4](#_Toc160133403)

[Show state of recognition of image 4](#_Toc160133404)

[Show recognized characters to recipient 4](#_Toc160133405)

[Non-functional requirements 4](#_Toc160133406)

[Training of the model 4](#_Toc160133407)

[Raspberry Pi cooling 4](#_Toc160133408)

[Casing 4](#_Toc160133409)

[FlowChart 5](#_Toc160133410)

[Limitations 6](#_Toc160133411)

# Work Estimates

|  |  |  |
| --- | --- | --- |
| **Group Member** | **Hour estimate, Group Meetings / Individual work (h)** | |
|  | Version 1 |  |
| Viitala Jussi |  |  |
| Sushila Kandel |  |  |
| Varpula Teemu |  |  |
| Lipponen Seppo |  |  |

# Change Log

|  |  |  |
| --- | --- | --- |
| **Version of the document** | **Version description / Changes made** | **Date** |
| 1.0 | First version of document | 14.2.2024 |
| 1.3 | Table of contents added | 23.2.2024 |
| 1.7 | Flowchart diagram of the program added | 29.2.2024 |
|  |  |  |

# Project Version and Deadline

|  |  |  |  |
| --- | --- | --- | --- |
| Project Version | Deadline for System Completion | Relevant Dates | Scope and Deliverables |
| 1.0 | 1.3.2024 | 26.2.2024 | Raspberry Pi setup |
| 1.1 | 6.3.2024 | 29.2.2024 | Raspberry Pi image recognition |
| 1.2 | 8.3.2024 |  | Raspberry Pi working display |
|  |  |  | Getting image from camera to disk |
|  |  |  | Main program loop if image recognized show led light output |
|  |  |  |  |

Introduction

The aim of this project is to develop a system that can detect signs of Finnish sign language alphabet using OpenCV and Tensorflow running on Raspberry Pi. The system will use our own image collection to train the recognition model. First implemention is to recognise one Finnish sign language alphabet character. Rest of the alphabet will also be added if there is time.

# Project overview

Software is running on Raspberry Pi and signs to be recognized are taken with camera connected to it. Image is taken by pressing button, but later this may also work as video capturing toggle. In that case also mode switch between image and video capturing would be required. Recognition level will be indicated with leds to the signer and if sign is recognized, it will be show in text screen for the recipient. Led indicator could be done using different color Leds in a row. For example red, yellow and green color could be used to indicate the state.

Hardware Requirements:

* Raspberry Pi
* Camera module Breadboard
* LED lights
* Display for text ouput Buttons

## Software Requirements:

* Python
* OpenCV
* TensorFlow

Functional requirements

General description

System needs to recognize atleast one Finnish sign alphabet character and display it on the screen. The state of recognizion must be shown to the signer with leds.

Components and functions

Image Capturing

User presses button to capture image or video from camera attached to Raspberry. Live recognition is preferred if possible.

Image recognition

* Project uses machine learning to train the recognition model and computer vision to identify signs from images. The main steps how the recognition works.
* OpenCV to Preprocess the images to get better suited data for this task Images need to be labelled for sign detection.
* Training will be done with TensorFlow
* Sign detection from the image or video

Show state of recognition of image

Recognition level is shown to signer with leds and if display is available, they could also follow the recognition program running and indicating level of recognition.

Show recognized characters to recipient

Recognized characters are shown to recipient from 1602 LCD display. New characters are appended to the end when recognized.

Non-functional requirements

Training of the model

Training the AI model can take quite a long time and should be done on more powerfull pc than Raspberry Pi.

Raspberry Pi cooling

Considering image prosessing is quite a demanding task, processor might get hot without proper heatsink. Temperature should be monitored during use and specially during live video recognition.

Casing

Possibility of 3D printed case is considered.

FlowChart

A screenshot of a computer screen

Description automatically generated

Figure 1:Flowchart Diagram of Signlanguage Interpreter system

Limitations

Only characters from Finnish sign language are recognized and only stationary signs are supported.