**Face Sorter**

Software Requirements Specification

Alexander Joslin

May 4, 2020

Table of Contents

[1. Introduction 3](#_Toc39518266)

[1.1 Purpose 3](#_Toc39518267)

[1.2 Scope 3](#_Toc39518268)

[1.3 Overview 3](#_Toc39518269)

[2. Overall description 3](#_Toc39518270)

[2.1 Product Prospective 3](#_Toc39518271)

[2.2 Product Functions 4](#_Toc39518272)

[2.2.1 User Interaction 4](#_Toc39518273)

[2.2.2 Pickled Dataset File 4](#_Toc39518274)

[2.3 User Characteristics 4](#_Toc39518275)

[2.4 Constraints 4](#_Toc39518276)

[2.4.1 Operating system 4](#_Toc39518277)

[2.4.2 Hardware 4](#_Toc39518278)

[3. Specific Requirements 4](#_Toc39518279)

[3.1 External Interface Requirements 4](#_Toc39518280)

[3.1.1 Hardware Requirements 4](#_Toc39518281)

[3.1.2 Software Requirements 4](#_Toc39518282)

[3.2 Functional Requirements 4](#_Toc39518283)

[3.2.1 Event Handler 4](#_Toc39518284)

[3.2.2 Sorting Class 4](#_Toc39518285)

[3.2.3 Deployment Class 5](#_Toc39518286)

[3.2.4 Training Set 5](#_Toc39518287)

[3.2.5 Update Training Set 5](#_Toc39518288)

[3.3 Performance Requirements 5](#_Toc39518289)

[3.3.1 Training Set 5](#_Toc39518290)

[3.3.2 Sorting and Deploying Photos 5](#_Toc39518291)

[3.4 Software System Attributes 5](#_Toc39518292)

[3.4.1 Reliability 5](#_Toc39518293)

[3.4.2 Security 5](#_Toc39518294)

[3.4.3 Maintainability 5](#_Toc39518295)

[3.4.4 Portability 5](#_Toc39518296)

[4 References 5](#_Toc39518297)

[4.1 IEEE 5](#_Toc39518298)

[4.2 YouTube 6](#_Toc39518299)

# 1. Introduction

This SRS provides an overview of how the Face Detection software was planned during its requirements elicitation.

## 1.1 Purpose

The purpose of this document is to help illustrate the data flow of this system.

## 1.2 Scope

This software application is split into two parts. One is to train on data and the other is to operate on new data. In this case the data are photos.

## 1.3 Overview

The goal is to create software the organizes photos based on previous data given to it.

# 2. Overall description

This section describes how the system and its subsystems interact with each other.

## 2.1 Product Prospective

The system will use Python and Computer Vision Libraries to process images. The software will be interacting with the user as well with a training set.

## 2.2 Product Functions

This section will list 3 main functionalities of the software.

### 2.2.1 User Interaction

The user will be interacting with a file system to upload their photos and they will have the option to update the training set.

### 2.2.2 Pickled Dataset File

The software will retrieve data that has been previously trained on. This data can be in the form of a pickled training set.

## 2.3 User Characteristics

There are no notable user characteristics to list. Any user can use this software.

## 2.4 Constraints

There is 2 constraints the user or software may face.

### 2.4.1 Operating system

This software may not work for systems running Linux.

### 2.4.2 Hardware

Tablets and Phones will not be supported.

# 3. Specific Requirements

This section provides all the functional and non-functional requirements of the system.

## 3.1 External Interface Requirements

### 3.1.1 Hardware Requirements

This software is meant to be used only on laptops or desktops with macOS or windows.

### 3.1.2 Software Requirements

External libraries needed for the software include os, watchdog, threading, and pickle.

## 3.2 Functional Requirements

### 3.2.1 Event Handler

An event handler will be used to detect if photos have been updated.

### 3.2.2 Sorting Class

A class that properly classifies new photos uploaded to their appropriate categories.

### 3.2.3 Deployment Class

A class that properly places the categorized photos into a folder with that category name.

### 3.2.4 Training Set

Previous photos will be trained on so the software can properly sort new photos.

### 3.2.5 Update Training Set

A feature that will let users add photos to the training set to update the system.

## 3.3 Performance Requirements

### 3.3.1 Training Set

None specific requirements

### 3.3.2 Sorting and Deploying Photos

The photos should be sorted and placed into new folders quickly. For every 25 Photos, it should take no more than 5 seconds for this operation.

## 3.4 Software System Attributes

### 3.4.1 Reliability

The software should be able to handle any photo thrown at it. It should also not allow any video or gif files.

### 3.4.2 Security

Photos will not be shared with any 3rd parties. Users have complete control of their data. No photos will be stored by the software.

### 3.4.3 Maintainability

The software will use the chain of command design pattern so that classes can perform their individual functionality and new functionality can be added later by simply creating a new class.

### 3.4.4 Portability

This software will be available on Windows and MacOS only.

# 4 References

## 4.1 IEEE

IEEE (1998). IEEE Recommended Practice for Software Requirements Specifications *IEEE Std 830-1998*

## 4.2 YouTube

Kalle Hallden (August 16, 2019). *Super Quick Python Automation Ideas*. YouTube. https://www.youtube.com/watch?v=qbW6FRbaSl0&t=287s