Software Requirements Specification

for

AEye

Vidhu Chaudhary

Department of Computer Science

Banasthali Vidyapith

07-02-2021

Table of Contents

1. Introduction

This software requirement specification (SRS) report is a detailed description for the system called AEye which is planned to be developed by our project team. AEye is basically a mobile application project whose aim is to detect common day to day objects in one's vicinity in real time. Although, it's end users are mainly specified as visually impaired people, anyone having a mobile phone with an android operating system will be able to use this application.

Visually Impaired People confront many problems in moving from one place to another. Vision is human's power to notify him of the obstacles in his way. A solution which is easily available is needed to solve the problems of visually Impaired people.This project tries to transform the visual world into the audio world with the potential to inform visually Impaired people about the objects in their local environment.Objects detected from the scene are represented by their names and converted to speech.

Mobile applications are software designed applications that are mainly designed to run on several devices like smartphones, tablets or computers. In spite of the fact that mobile phones have been around us for nearly fifty years, Android operating systems can be thought of as a new system and a development environment by having eleven years of lifetime. However, the number of applications which were developed for the use of disabled people is too limited. Therefore, our aim was to develop an application which will be used by visually impaired people for detecting objects easily around them.

Our system consists of several modules. Video is captured with a inbuilt android camera on the client side, and is streamed to the server for real-time image recognition with existing object detection models (YOLO).The 3D objects are then determined by bounding boxes from the detection algorithm in addition to an audio speech to recognize the objects with their names. Audios containing the name of the object are played at the interval of a few seconds, when the recognized object differs from the previous one.

1.1 Purpose

The purpose of this SRS report is to provide a detailed description of an object detection System. This document will present aims, features and all the functional and non-functional requirements of the system and the comprehensive explanations of these requirements. Moreover, this report will also contain the constraints under which this system will be operating.

Thus, in this SRS a model has been proposed which makes the use of smartphones and technology to make an application which can help the visually impaired users detect objects in their surroundings.

1.2 Intended Audience and Reading Suggestions

Intended audience for this SRS can be divided into several categories which can be listed as developers of the project, stakeholders which are the visually impaired people, other users and the academic staff of the Computer Science Engineering Department in the Banasthali Vidyapith.

1.3 Product Scope

“AEye”, is the name of the project which is planned to be developed by our team. The product will be an android application designed for the detection of multiple objects in real time. The system is an aided technology for the assistance of visually impaired people.

The aim of the project is to develop an application that can detect the objects in the user's surroundings. It identifies objects familiar to the user in his daily life, and then tells the user of these detected objects to aid him in his daily life routines. The reason it is more reliable is because it is developed on the Android operating system and Android-based smartphones are very common and highly available almost everywhere. In fact, it's one of the most used mobile operating systems. This makes the application convenient to get. However, it is focused on identifying the most common things in one’s day to day life, which are thought to be essential by the team.

The application will not require from the users any personal information to be inserted to the application. There will not be any login page or password requirements to use the application. However, to download the application from Google Store, a Gmail account is required. This obligation is applied by Google, not by the team developing the application. Also basic permissions, like to access the camera, will be required.

1.4 References

/* References cannot be cited because we have prepared the report ourself. */

2. Overall Description

Millions of people live in this world with incapacities of understanding the environment due to visual impairment. Although they can develop alternative approaches to deal with daily routines, they also suffer from certain navigation difficulties as well as social awkwardness. For example, it is very difficult for them to find a particular room in an unfamiliar environment. And blind and visually impaired people find it difficult to know whether a person is talking to them or someone else during a conversation.Computer vision technologies, especially the deep convolutional neural network, have been rapidly developed in recent years. It is promising to use the state-of-art computer vision techniques to help people with vision loss. In this project, we want to explore the possibility of using the hearing sense to understand visual objects. The sense of sight and hearing sense share a striking similarity: both visual object and audio sound can be spatially localized.

2.1 Product Perspective

There exists multiple tools to use computer vision technologies to assist visually impaired people. For example,The mobile app TapTapSee uses computer vision and 1 crowdsourcing to describe a picture captured by blind users in about 10 seconds. The Blindsight offers a mobile app Text Detective featuring optical character recognition (OCR) technology to detect and read text from pictures captured from the camera. Facebook is developing image captioning technology to help blind users engaging in conversations with other users about pictures. Baidu recently released a demo video of a DuLight project. No further details of the product is available at the moment. However, the product video suggests concepts of describing scenes and recognizing people, money bills, merchandises, and crosswalk signals. However, these products were not focusing on enabling general visual sense for people and did not use the sound techniques to further enhance the user experience.

Since our application is an android application, it is independent and totally self-contained. AEye will mainly focus on visually impaired people however other people can use this application. There is no need for an attractive interface for visually impaired people and therefore, our application focuses on providing an interface which is easy to use and navigate through. On the other hand, other people can also use the functionalities of the application through interfaces.

The below diagram is the architectural / conceptual diagram of our system. It is a four layered architecture. Through this conceptual design we are trying to show the interaction between the different layers of our system. All the requests of the users are taken by the Android application and are given to the API for processing. The API uses a dataset which contains thousands of labelled images and compares the current image with the images in the dataset. After classification and identification, the API sends the label of the current image to the application.

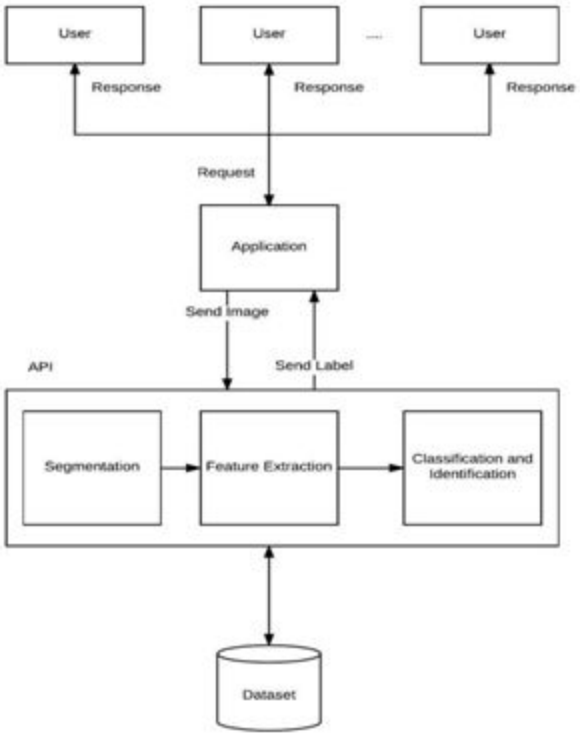

Fig.1 Architecture Of The System

2.2 Product Functions

In our project, we build a real-time object detection system with the goal of informing the user about surrounding objects.The system to be developed will be trained about object information. Feature extraction is also a part of the process.

The core specialty of the system is to detect multiple objects in an image. That is, it is a system where N object detectors are trained for N different objects. When an image is sent to the system, all object detectors do their work. If an object is found by a detector, it will mark its boundary and label the object name. After the process completes for all N detectors, the image is displayed with all the tags and the user is prompted one by one with audios containing the name of the objects detected.

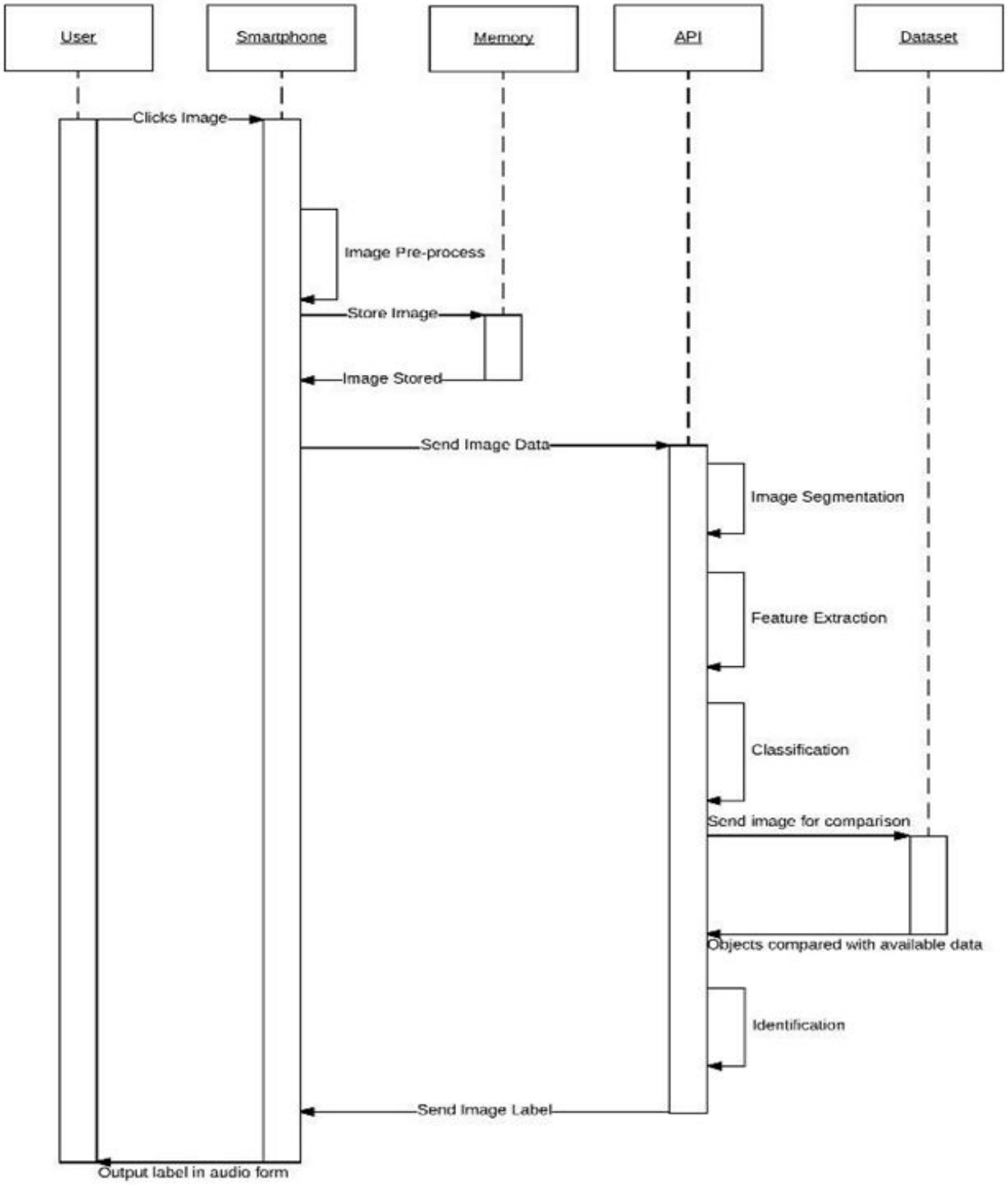

Fig. 2. Sequence Diagram

The above sequence diagram of our framework clarifies the stream of the framework, that is, what action takes place first and what action will follow the previous action. First, the user starts

the application and captures the image of the surrounding in front of him or of the object in front of him which he wants to identify. The application digitizes and stores the captured image in the memory. It is then used to detect objects in other images. This image will be processed by using libraries like OpenCV and Google Cloud Vision API.

The OpenCV library contains many Image Processing algorithms and Google Cloud Vision API has the power to compare the input image with millions of other images using Microsoft's COCO Dataset. The API receives image data, performs image segmentation, feature extraction, classification & identification functions and uses the COCO dataset to get the image label. The application then converts the label into audio and sends it via the speaker.

2.3 User Classes and Characteristics

There are two main user types in AEye. These are visually impaired people and normal people. In other words, normal people can also use the features of the system designed for visually impaired people. There are no restrictions to be a user for the AEye: anyone, who has a platform on which Android runs, can download and use the AEye. But to run the application, one must have connection to the internet and system requirements should be satisfied by their device.

2.4 Operating Environment

The output of the system is in audio form, which is easily understandable for a visually impaired user. The application being developed can detect the objects in the user's surroundings. It identifies objects familiar to the user in his daily life, and then tells the user of these detected objects to aid him in his daily life routines. The reason it is more reliable is because it is developed on the Android operating system and Android-based smartphones are very common and highly available almost everywhere. In fact, it's one of the most used mobile operating systems. This makes the application convenient to get.

For Image Processing purposes, we use the OpenCV Library. Apart from OpenCV the well-known Google Cloud Vision API which is used for interpreting the contents within a photo has the power to have our pictures compared with images from Microsoft's COCO Dataset. The information about external modules used for creating the application are as follows:

A. OpenCV

OpenCV (Open Source Computer Vision) is a library of programming capacity which for the most part went for constant PC vision. Initially created by Intel, it is currently kept up by Itseez and supported by Willow Garage. The library is cross-platform and free for use under the open-source BSD license. OpenCV bolsters the Deep Learning Structures Torch/PyTorch, Caffe & TensorFlow.

B. Google Cloud Vision API

Google Cloud Vision API empowers experts to comprehend an object in an image or a photo by speaking to fruitful machine learning models in a simple way to utilize REST API. It rapidly characterizes pictures into a large number of classifications (e.g., "sailboat", "Eiffel Tower"), distinguishes singular questions and faces inside pictures, and finds printed words contained inside pictures. One can build metadata on picture index, direct hostile content, or empower new advertising scenarios through picture assessment investigation. It analyzes images uploaded in the request or integrate with your image storage on Google Cloud Storage.

C. Microsoft COCO Dataset

The COCO dataset stands for Common Objects in Context, and is designed to represent a vast array of objects that we regularly encounter in everyday life.The COCO dataset is labeled, providing data to train supervised computer vision models that are able to identify the common objects in the dataset. Of course, these models are still far from perfect, so the COCO dataset provides a benchmark for evaluating the periodic improvement of these models through computer vision research. COCO is a large-scale object detection, segmentation, and captioning dataset. It has several features such as object segmentation, recognition in context, superpixel stuff segmentation, 330K images (>200K labeled) and 1.5 million object instances.

2.5 Design and Implementation Constraints

2.5.1 Design

The application mainly uses Android along with many supported libraries. The camera on an Android smartphone will be used to capture an image of the surrounding which will be stored in Android's memory. This image will be processed by using libraries like OpenCV and Google Cloud Vision API. Google Cloud Vision API uses Google Cloud. The image is sent to the cloud through the internet. It uses the COCO dataset to compare the input image with millions of other images. The process gets completed and the objects are identified in the image. The user gets informed about the identified objects present in his surroundings via an audio output.

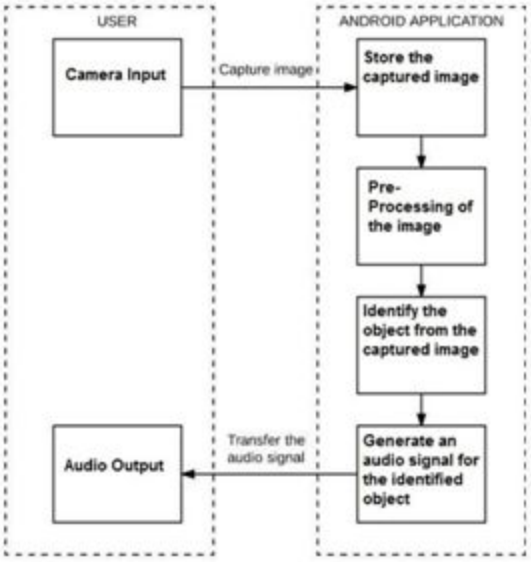

Fig. 3. Module of System

2.5.2 Implementation Constraints

- The user shall have enough free memory in order to download the application.
- The mobile device shall have at least 1.2GHz CPU and 1 GB RAM.
- The device shall have at least 5 MP camera resolutions.
- The operating system of the target device must be Android with greater than or equal to version 4.2.
- The application shall be available after downloading and connecting to the internet from the mobile device.
- Only one language i.e. english will be supported for the application interface in AEye.

2.6 User Documentation

A guide for the user to navigate through the application and use the application would be provided in the form of a document or a tutorial video within the application itself. A button representing this feature would be made available on the main page of the application.

2.7 Assumptions and Dependencies

AEye will depend on the internet connection. Without an internet connection, the application will not be started. For the correctness of the system, it is assumed that the camera is capturing the object as a whole. The users of the system are assumed to understand the supported language by the application i.e. english.

3. External Interface Requirements

3.1 User Interfaces

There will be one graphical user interface for the use of people who are not visually impaired. There will be two basic pages belonging to this GUI. This GUI will be responsible for providing communication between the user and the application. Before mentioning the specific features of each page, there are some common features of all pages in the user interface of AEye. Firstly, links to all these pages will be displayed on the screen via buttons. The contents of the pages will be displayed in english language.

Properties of each page of the application are listed below:

Main Page

Basic graphical user interface for the main page of the application will be in English as mentioned above.There will be a start button on this page which on clicking will redirect the user to the second page of the application which is the object detection page. In addition, buttons for navigating to other pages of the application such as tutorial, user manual and brief description about the application will also be present on the main page. Moreover, a pop up window requesting access to the camera will be prompted at the start of the application for the user to allow the functioning of the object detection feature.

Object Detection Page

As soon as the user is directed to the second page of the application the camera is launched at the page and it starts detecting the objects that are captured. As the surrounding objects are detected one by one the application will prompt the user about the identity of the object in an audio form. The page will also provide a stop button to stop the object detection being processed by the application.

Events

startApplication: All users of the application will be able to start the application via clicking the application icon on the device.

exitApplication: All users will be able to exit and close the application via a physical button of the device.

startObjectDetection: The users will be informed about the occurrence of an object captured in the camera via speech.

stopObjectDetection: The users will be able to stop the object detecting component without having to exit the application.

3.2 Hardware Interfaces

The system will work on an android mobile phone with internet connection. The Android version should be greater than 4.2.0. The phones having Android OS below this version may have some problems. It is expected that the CPU speed of the device and the internet connection speed shall be enough to respond to users' requests in less than a second.

3.3 Software Interfaces

As written in section 3.2 a mobile phone which has Android OS version above 4.2.0 with internet connection will be able to run the application. On the other hand, Windows operating systems will be used during the development process. The system will be implemented by using Android Studio IDE.

4. System Features

For each use case, detailed information about the description of the use case is discussed in this section. Notice that, since there is no interface for visually impaired people, use cases for visually impaired people are explained according to states instead of interfaces.

4.1 Start The Application

4.1.1 Description

This use case explains that the users of this application starts the application by clicking on its icon.

4.1.2 Actors

Users of the application are visually impaired people or normal people who wish to make use of this application.

4.1.3 Stimulus

When the users click on the icon of the application the program is started and the application interface is opened.The application will now be managed through these interfaces.

4.1.4 Functional Requirements

4.1.4.1 Functional Requirement 1.1

The system shall be opened with clicking its icon on the general interface of an Android device.

4.1.4.2 Functional Requirement 1.2

The system shall allow the application to access the camera for taking inputs.

4.1.4.3 Functional Requirement 1.3

In case of starting the application successfully by clicking its icon, the main interface of the application shall be shown to the user so that they can understand the application has started successfully.

4.2 Start Object Detection

4.2.1 Description

This use case explains that the users of this application will be informed if there is an object in their surrounding that has been captured by the camera.

4.2.2 Actors

Users of the application are visually impaired people or normal people who wish to make use of this application.

4.2.3 Stimulus

When the users click on the start button of the main page they are redirected to the object detection page and the object detection component is hence initiated.The application starts detecting the objects in the surrounding.

4.2.4 Functional Requirements

4.2.4.1 Functional Requirement 2.1

The system shall detect multiple objects by using the camera of the device if the object detection component is available.

4.2.4.2 Functional Requirement 2.2

The system shall give a warning message with voice stating that no objects are detected if that is the case.

4.2.4.3 Functional Requirement 2.3

After finding the object the system shall respond with speech giving the name of the detected objects one by one.As soon as the frame of the input changes it will start detecting new objects.

4.3 Stop Object Detection

4.3.1 Description

This use case explains that the users of this application can stop the object detection

4.3.2 Actors

Users of the application are visually impaired people or normal people who wish to make use of this application.

4.3.3 Stimulus

When the user clicks on the stop button of the object detection page they are redirected to the main page of the application thus terminating the object detection component.

4.3.4 Functional Requirements

4.3.4.1 Functional Requirement 3.1

The system shall require the user to click on the stop button, if the object detection component is required to stop working.

4.3.4.2 Functional Requirement 3.2

The system shall stop the object detection component if the concerning event is detected by the application.

4.4 Exit The Application

4.4.1 Description

This use case explains that the users of this application can exit from the application.

4.4.2 Actors

Users of the application are visually impaired people or normal people who wish to make use of this application.

4.4.3 Stimulus

When the user closes the application as per the android device's functioning the application stops working.

4.4.4 Functional Requirements

4.4.4.1 Functional Requirement 4.1

The system shall be closed by device's physical buttons like in case of all interface controlling applications.

5. Other Non-Functional Requirements

5.1 Performance Requirements

- The application should load and be usable within 3 seconds.
- The system shall be able to detect the objects within 5 seconds of appearance.
- The system shall convert text to speech and vice-versa in less than 3 second.

5.2 Safety Requirements

- The user should be aware of providing permissions to the camera.
- In any case, if the user feels the detected data is being shared then file a complaint directly to responsible authority.

5.3 Security Requirements

- The data of objects being detected by the users who download this application should be kept as secret.

5.4 Software Quality Attributes

Software quality attributes are discussed under this section.

5.4.1 Reliability

- The system's reliability mostly depends on the other software tools (Google Cloud Vision API, Microsoft COCO Dataset, OpenCV) used for the system development. Their reliability mostly meets the reliability of this system.

5.4.2 Availability

- The system must be available on a device with Android Operating System.

5.4.3 Maintainability

- There should be valid documents in requirements specification, design and implementation.
- Diagrams should be provided in the documents in order to increase the understanding of users and designers.

5.4.4 Portability

- The system must be portable to any android device that has an Operating System greater than 4.2.0.
- The system should not need any software installation.

5.4.5 Usability

- The interface should be easy to learn without a tutorial and allow users to accomplish their goals without errors.

5.4.6 Correctness

- The application should be able to detect the objects with an accuracy of 75%.///

6. Appendices

There is no available appendix for this SRS report.