# Cyclops - A Spatial AI based Assistant for Visually Impaired People

*OpenCV Spatial AI Competition Project Report*

Kaustubh Sadekar,Vishruth Kumar, Malav Bateriwala

October 31, 2020

# Contents

# 1 Problem statement

Cyclops is a spatial AI-based assistant for visually impaired people. It is assembled in a device similar to a VR headset and allows the wearer to ask if any objects matching the speech input are found nearby and provides active feedback when approaching found objects.

# 2 Problem description

We propose Cyclops as a spatial AI-based assistant for visually impaired people. It is a system assembled in a device similar to a VR headset. The user asks Cyclops to search for a particular object. For example, 'Cyclops, can you see bottle?' The device searches for the object and guides the visually impaired user using active audio feedback.

The overall problem statement consists of two main stages.

1. **Determining the object to be searched based on the user input.**

   This can be done using NLP based speech recognition methods. The input is provided in the form of a speech signal by the user, which is converted to a text signal using speech recognition methods. The generated text signal is used to determine the object to be searched.

2. **Guiding the user to the target object.**

   Object detection can be performed using the OAK-D neural inference functionality. If the object to be searched is detected, audio feedback will be generated, along with additional information about its location with respect to the user. e.g., "Object at 3 meters in the right" or "Object is far away in the left." The approximate distance of the target object will be calculated using the depth functionality of OAK-D.

   The audio feedback will be generated continuously based on the updated values from OAK-D until the user reaches the object.

   Finally, the user will be informed when the object is close enough to be held.

Refer Figure 1 that explains how the user interacts with the world using Cyclops. We share the product sketch in Figure 2, which illustrates the design for Cyclops if it to be developed into a complete product.

# 3 Implemented Solutions

Cyclops is developed using OAK-D and Raspberry-Pi3 and the code is developed in python. Processing of the code is divided into two threads using multi-threading. One thread takes care of taking audio input from the user and generating audio feedback. The second thread deals with the acquisition of the data from the OAK-D device. Refer figure 3 for better understanding.

Implementation details for different sections of the project are shared in the following subsections.

## 3.1 Speech To Text Conversion

Speech_recognition library is used to take user input as an audio signal and convert it to text. Several other speech-to-text libraries were explored, but we used this library because it supports several speech-recognition APIs and engines online and offline. For our experiments, we use the method that supports Google Cloud SpeechAPI to convert audio signal to text.

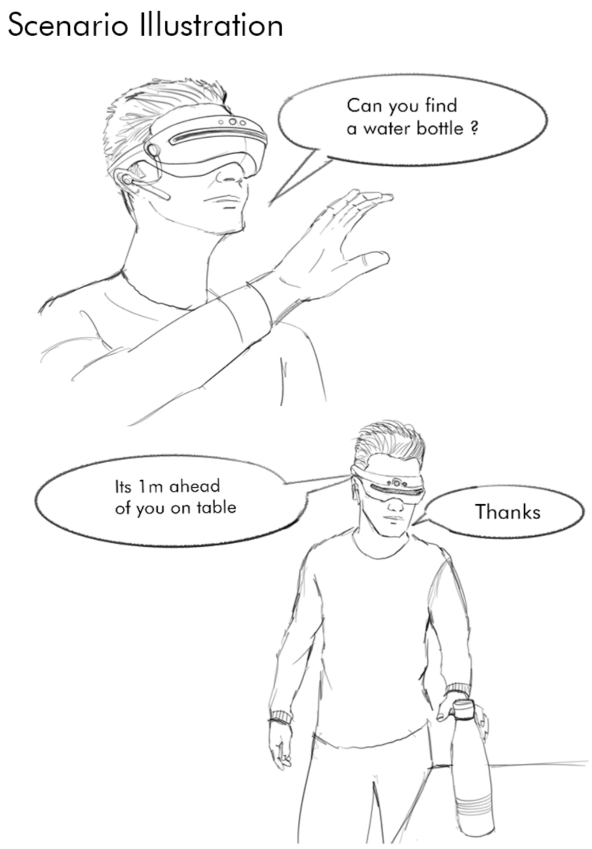

Figure 1: Scenario Illustration

The generated text is then used to determine the object to be searched. We make use of the identification phrase - ”Cyclops can you see. ” The word after this phrase is stored as the target object to be searched. Hence to search for a bottle, the user needs to say - ”Cyclops can you see bottle”.

## 3.2 Object Detection

Object detection is performed by the OAK-D device connected to the host processor, which is a Raspberry Pi 3B. The model number for OAK-D is bw108obc.

Python API - depthai is used to obtain the image data captured by the OAK-D camera as well as the predictions. We use the MobileNetSSD model for object detection. Pre-trained weights and files available on the github page of depthai were used for the project.

## 3.3 Audio Feedback

Once the target object is searched, audio feedback needs to be generated to guide the user if Cyclops detects the target object.

The output of object detection and the respective distance information is stored in global variables so it can be accessed by both the vision thread and the audio thread. Based on this information, the audio output is generated to guide the user. For example, if the object is detected and 2 meters away in front of the user, the audio feedback would say, ”Object at 2 meters in front”. This way, the users can orient themselves and move towards the target object.

Several options for generating the audio feedback were explored. Google Text-To-Speech (gTTS) library is used to generate audio files from strings, and mpg123 API is used to play the

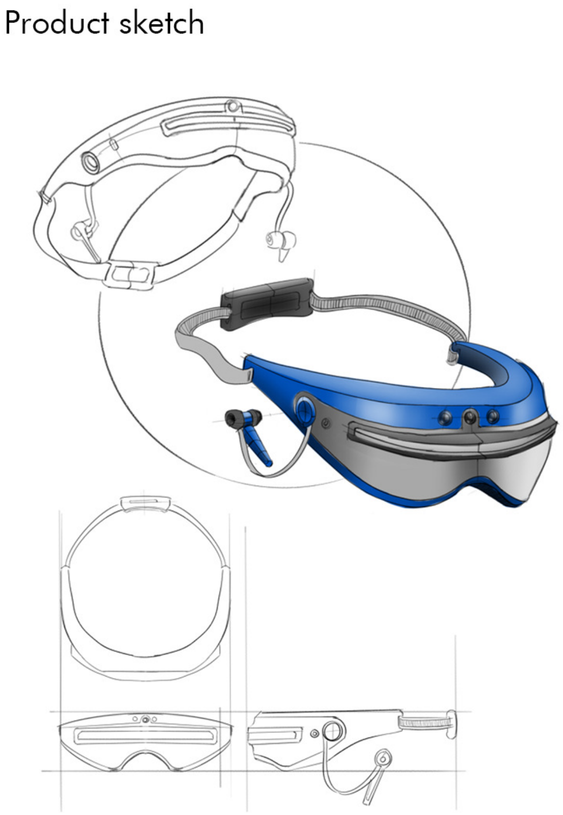

Figure 2: Product sketch for Cyclops

generated audio files.

# 4 Experiments and Technical Challenges

Transforming Cyclops from a conceptual idea to a working prototype was full of technical challenges and several experiments were performed to overcome the challenges. Details about the experiments and challenges are shared in the following subsections.

## 4.1 Device Assembly

There are several design aspects that we need to take into account when developing Cyclops as a fully functional product. Following is the list of hardware components used in the device:

1. OAK-D device
2. Power bank
3. USB Microphone
4. Headphones
5. Housing box

It is not easy to place all these components into a single device. The setup becomes bulky and is difficult to wear as a headset. Hence we only place the OAK-D device in the housing

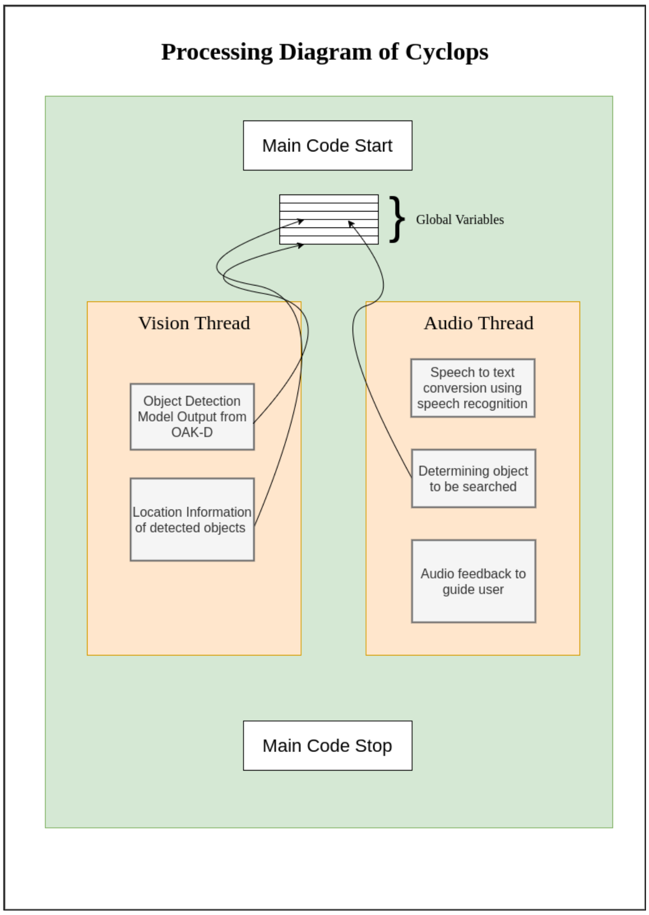
Processing Diagram of Cyclops
Main Code Start
Global Variables
Vision Thread
Object Detection Model Output from OAK-D
Location Information of detected objects
Audio Thread
Speech to text conversion using speech recognition
Determining object to be searched
Audio feedback to guide user
Main Code Stop

Figure 3: Product sketch for Cyclops

box, which can be worn easily. The remaining components are packed and kept in the pocket of the user.

Another challenge in developing the hardware assembly is that after running for a few minutes, there is a significant increase in the temperature of OAK-D and Raspberry Pi. It is a crucial point that should be considered when designing Cyclops at a commercial level. Refer figure 4 for images of the housing box along with OAK-D.

## 4.2 Audio Input With Raspberry Pi

Multiple options for audio input were explored. As Raspberry Pi has a single 3.5mm Jack and it was to be used for headphones, one more choice was to use a mini USB sound card, like the one shown in figure 5. Several configuration problems were faced to get the sound card running, and also, using a sound card was making the setup further bulkier. Hence we finally decided to use a USB microphone. For experiment purpose, we the microphone circuit of a USB webcam. It was easy to use, and had fewer configuration challenges.

Audio input was captured and stored using the Microphone class of the Speech$_{r}ecognition$ library.

## 4.3 Object Detection With Depth

Pre-trained weights and files available on the github page of depthai for MobileNetSSD object detection model worked quite well in terms of detection.

The critical limitations observed were related to the depth estimates for the detected objects. The minimum distance for accurate depth prediction varied depending on the size of the object.

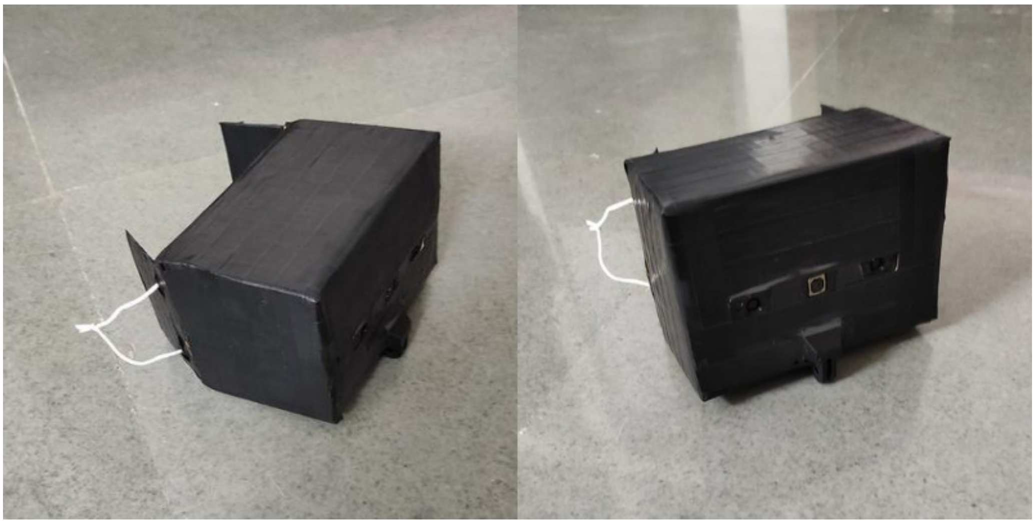

Figure 4: Prototype Assembly for Cyclops

For smaller objects like a bottle, the minimum distance was about 1.2m, and for larger classes like a person, the minimum distance was about 0.7m.

Addressing the fact that the users can easily stretch their hand to grab an object within a distance of 1 meter, this limitation did not significantly impact Cyclops' objectives.

### 4.4 Real Time Operation

The real-time operation of the device is an essential requirement for such a task. The power of spatial AI on chip provided by the OAK-D makes it possible to use a small pocket size low cost SOC like Raspberry Pi as a host and still achieve real time performance as the computational load for object detection and depth estimation is taken care by the OAK-D and not the host processor.

Although the object detection and depth estimation could be performed at more than 25-30 FPS, the major problem started when the detection functionalities were merged with the speech recognition and audio feedback generation functionalities.

Playing the audio files was observed to be the bottle neck part of the entire code. Hence multi-threading was used to divide the tasks related to audio and vision—this significantly improved performance.

However, an important observation is that sometimes the detection values are not displayed on the output window when the audio thread is running. The reason for this glitch is not yet known. This is also demonstrated in the first part of the demo video.

## 5 Limitations and Future Work

Cyclops' ultimate objective would be served when it is entirely robust and ready to be used by potential users to augment how they interact with the world. Several limitations need to be tackled to fulfill this objective.

### 5.1 Speech Understanding

In the present code, the speech to text conversion is very accurate, but the audio is sampled and processed in the background in intervals of 3 seconds. This interaction can be made more efficient and effective.

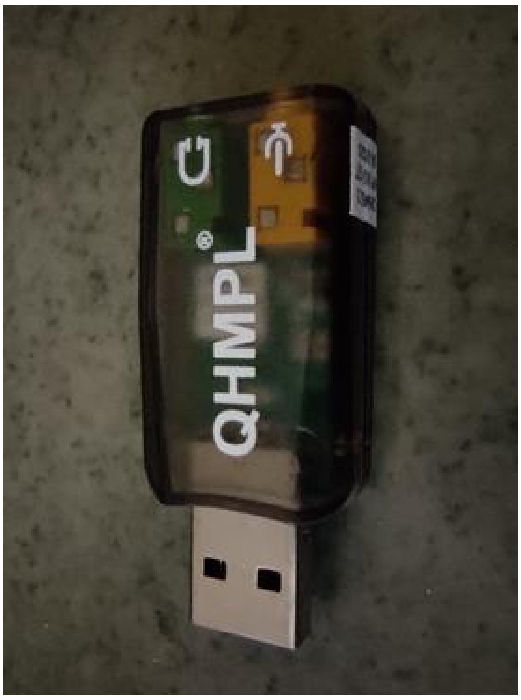

Figure 5: USB Sound Card

## 5.2 Audio Feedback Generation

The present code plays a set of pre-recorded audio files based on different conditions. Methods to generate custom audio files and playing them were explored, but they were very slow and not applicable for real-time operations. In the future, generating custom audio files and running them in real-time can be made faster and more efficient.

## 5.3 Custom Object Registration Pipeline

Presently the objects Cyclops can detect are limited to the object detection model. In real-life scenarios, some objects are not present in the object detection class list but are essential for the user.

A custom object registration pipeline and a training flow can be created to register the object and use Cyclops to search for it in the future.

## 5.4 Text Detection and Recognition

With a well designed robust pipeline, the functionality of text detection can also be incorporated. So the user can say, ”Cyclops read me what you see,” to read a book or some signs on the road.

## 5.5 General Environment Update

A custom model or the object detection model’s predictions can be used to generate a general statement giving a Short summary of the world around the user. NLP based models can be used for more detailed summaries. So the user can ask - ”Cyclops, what is going on around me,” and custom audio could be generated that provides a summary of the surrounding environment.

We would be open-sourcing this project, and all the future work ideas would be listed in the README file. The code would be continuously improved to make things more modular to encourage contributions from other enthusiasts.

# 6 Important Links

1. Link to the demo video.
2. Link to the GitHub repository.