DIP End-Semester Report

Smart Glasses & Smart Stick for Blind People

Abstract:

Smart Glasses and Smart Stick independently serve the primary purpose to assist visually impaired people. These technologies are viewed as a way to encourage blind students to continue their education despite their challenges. Its main goal is to create a new way of reading texts for blind people and to make communication easier for them. The glasses' first job is to scan any text image and convert it to audio text, which the user can listen to through a headset linked to the glasses. The second step is to translate the entire text or a portion of it. Two units of Raspberry Pi 4 Model B will be used for all computing and processing functions. For obtaining the results, the amalgamation of OCR, face recognizer system for identification with the four-unit integrated sensor system in the stick facilitates high accuracy which demonstrates the ability of our product to detect & recognize the majority of obstacles.

Design & Simulation:

The objective of the product is to guide visually challenged people in their movement.

The design of the smart glasses includes an ultrasonic distance sensor and a camera module. The assembly of the glasses is shown in figure 2. The sensor and the camera will be controlled by a second Raspberry Pi.

The CAD model demonstrated in figure 1 shows the assembly of the smart stick. It is enabled with ultrasonic sensors for depth perception & water sensors to alert the user. With the placement of the sensors inclined at 120 degrees with each other, it offers a wide field of view. The static structural analysis of the stick was done using ANSYS and the results are shown in figure 3. The factor of safety of the design was calculated to be 7.6 and thus was considered to be safe.

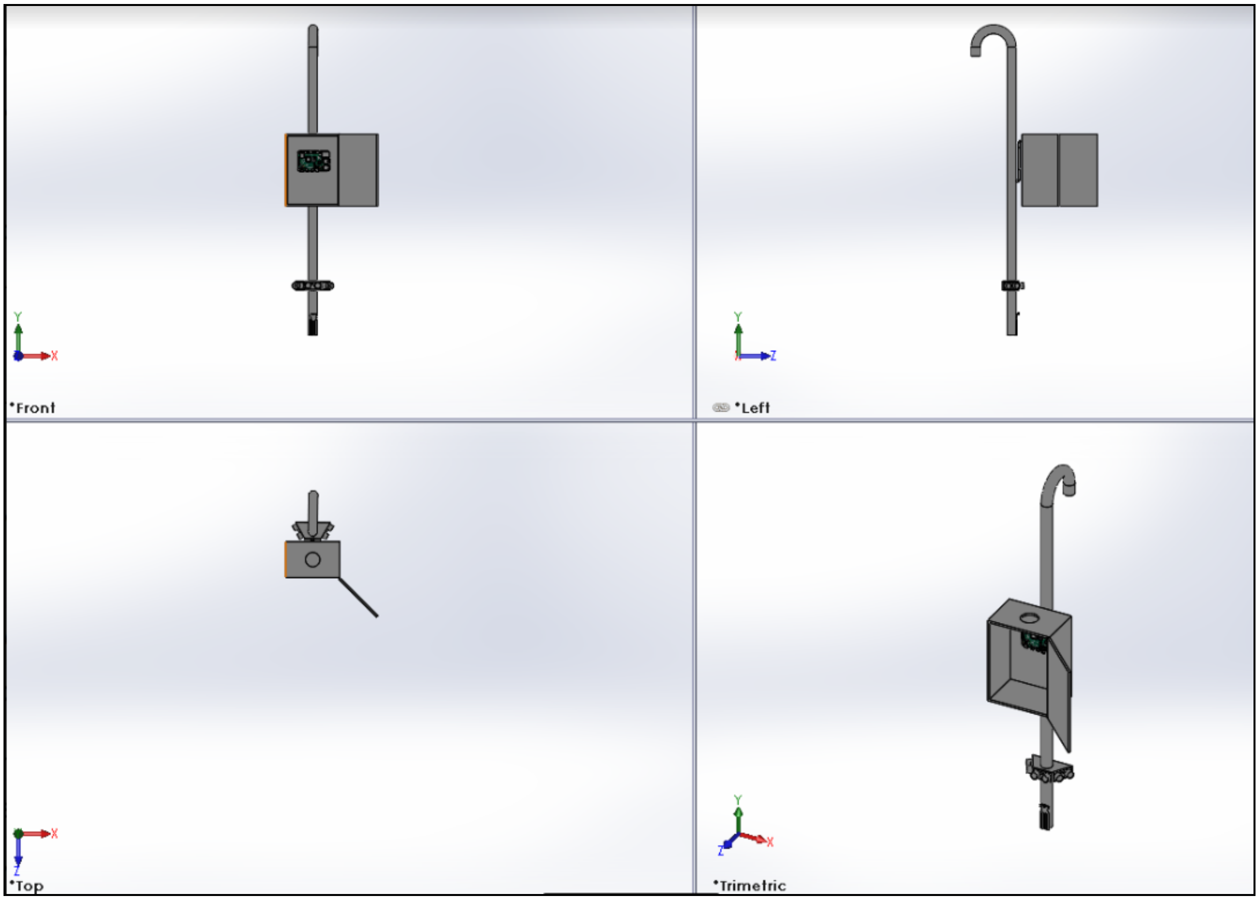


Figure 1: Front, top, bottom & side view of Smart Stick

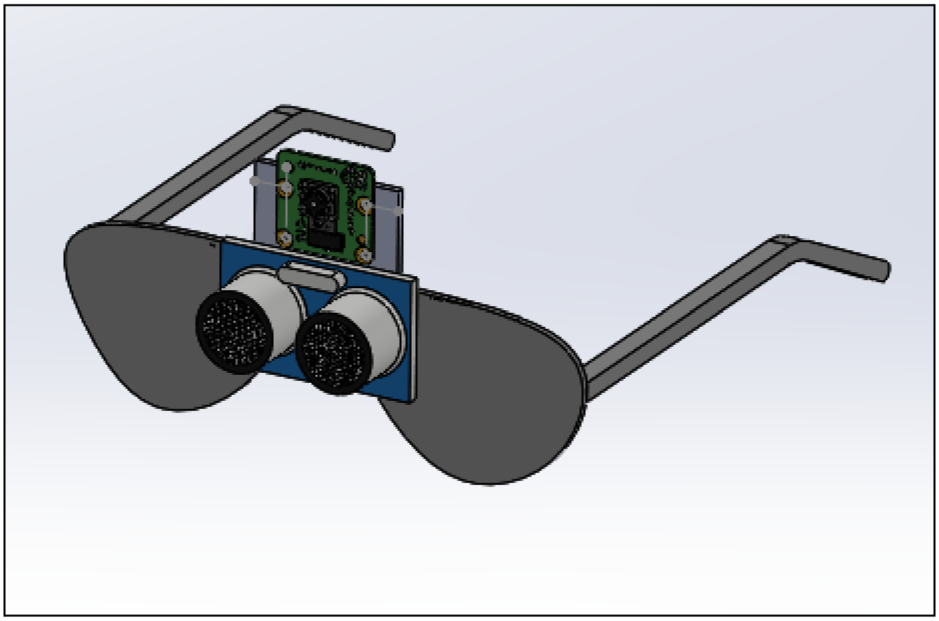


Figure 2: CAD Model of the Smart Goggles

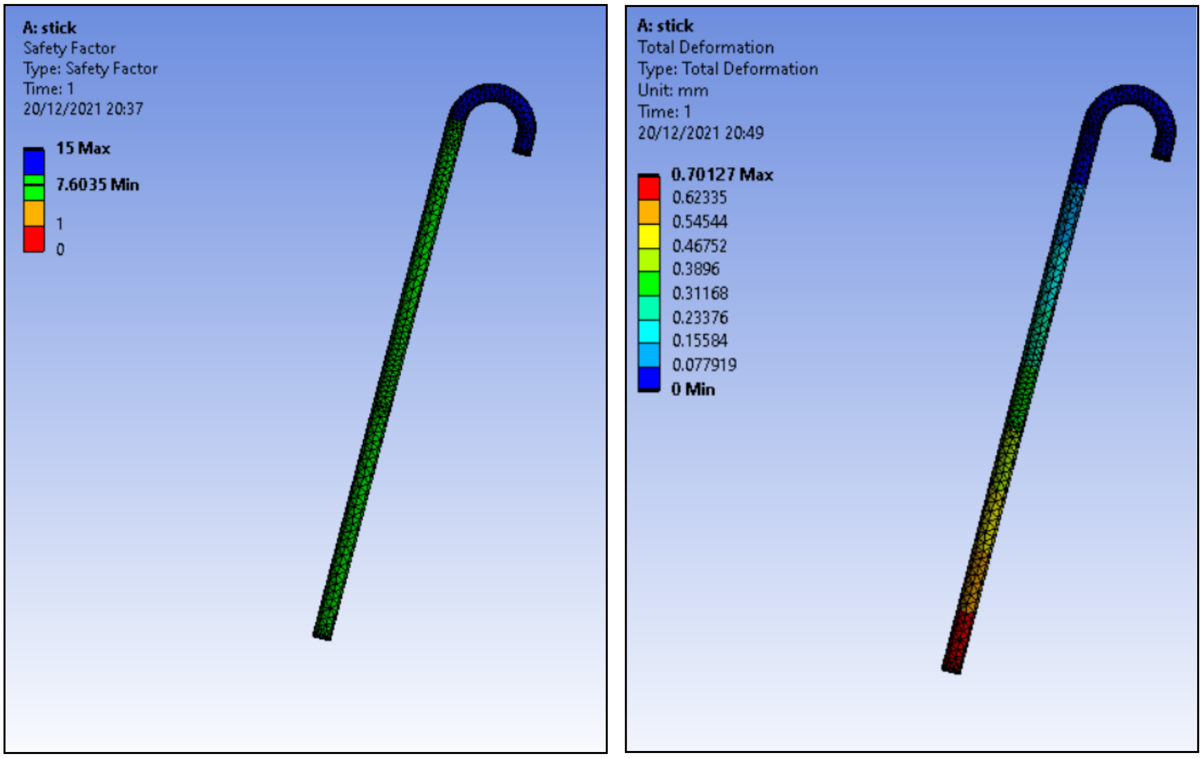


Figure 3: Deformation Analysis of Smart Stick

Bill of Materials:

LIST OF ITEMS REQUIRED FOR THE PROJECT	
Item	**Cost (Rupees)**
Glasses	1000
Raspberry Pi 5 MP Camera Module	1700
Power Bank Battery	2000
Raspberry Pi 4 Model B (x2)	6500 * 2
Sandisk 64 GB Memory Card	900
Samsung microSD Card EVO Plus 32 GB	1000
Ultrasonic Distance Sensor (HC-SR04 x4)	300 * 4
Earphones	800
Water sensor	400
Jumper Wires	900

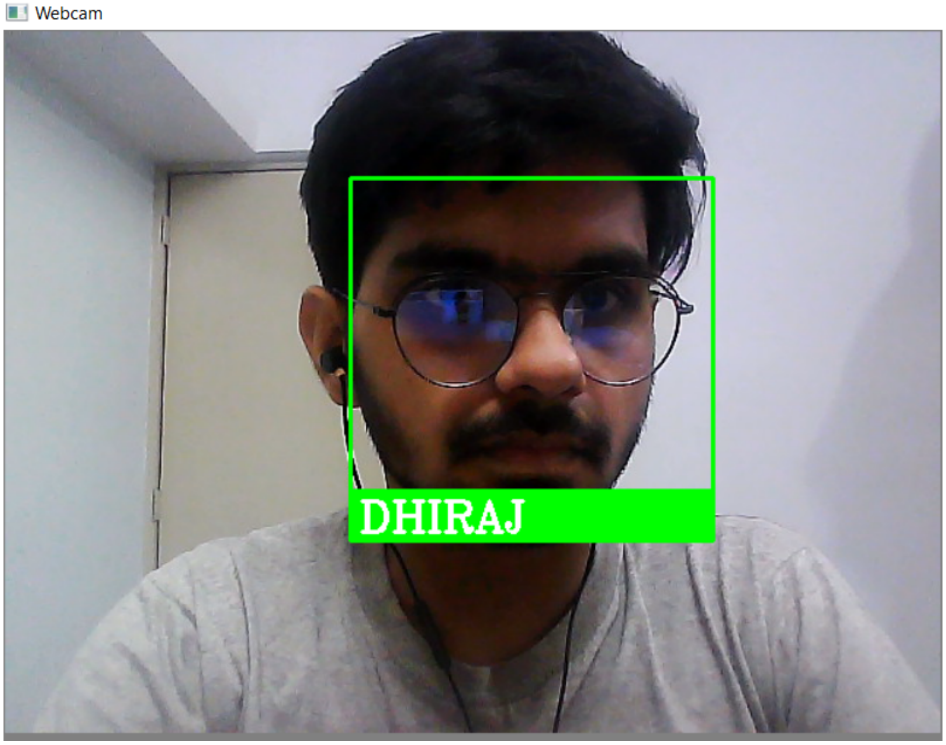
Raspberry Pi Starter Kit	6000
Walking Stick	300
Vibration Motor	200
TOTAL	29200

LIST OF EQUIPMENT REQUIRED FOR TESTING	
3-D Printer	Already Available in Campus
SOLIDWORKS	Already Available in Campus

Preliminary Results/Partial Outputs:

Face Recognition Using OpenCV:

We have currently developed a real-time face recognition model built using dlib’s state of the art face recognition built with deep learning. The code written, is integrable with the Raspberry Pi 4 Model B computer. We are currently using an HP High Definition camera for capturing real-time video, which will eventually be replaced by the Raspberry Pi 5 MP Camera Module. The highly optimized face recognition model requires only one image per face to be fed as training data, for real-time recognition.



The snippet of the Face Recognition output from the real-time video

Future Tasks:

Our next task is to give the recognized name of the face to the user as an output using the earphones attached to the goggles.
Subsequently, a python script is to be written that can automatically capture a real time frame of an unknown person (which a user wishes to add but is currently not in the database) and label it with the name of the person that the user feeds. The user will feed the new person's name via the microphone given, which will then be converted to text and fed to the training image folder of the model as the label for the real-time frame captured.

Libraries used for the Face Recognition model:

- Opencv-python
- Numpy
- face-recognition

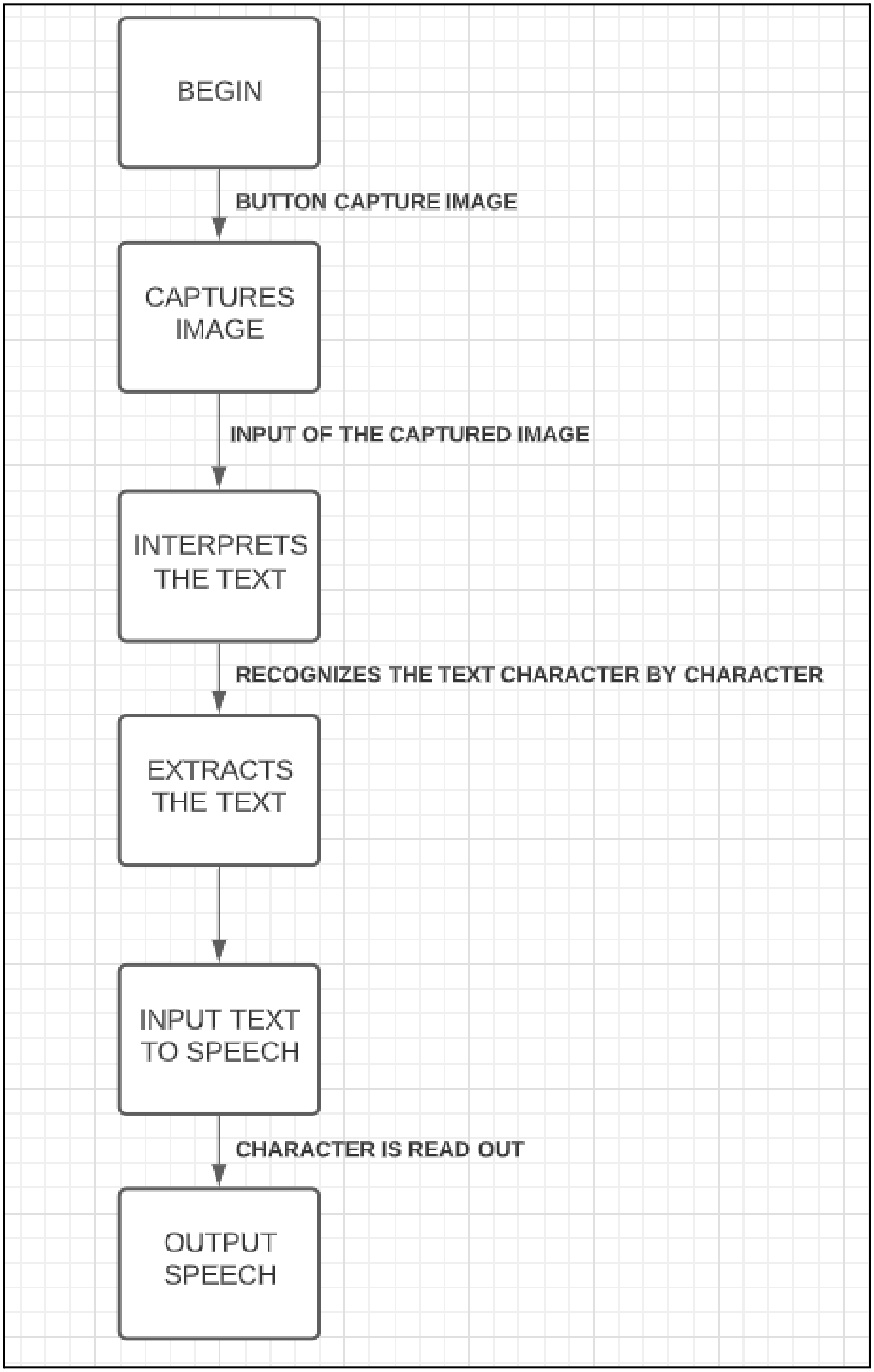
Image to speech:

We used OCR (Optical Character Recognition) which is a process of converting scanned images of machine or handwritten texts into a computer format text. Then we used gTTS (Google's text to speech) API to convert the computer format text into an mp3 file.

Libraries used in the Images to Speech Model:

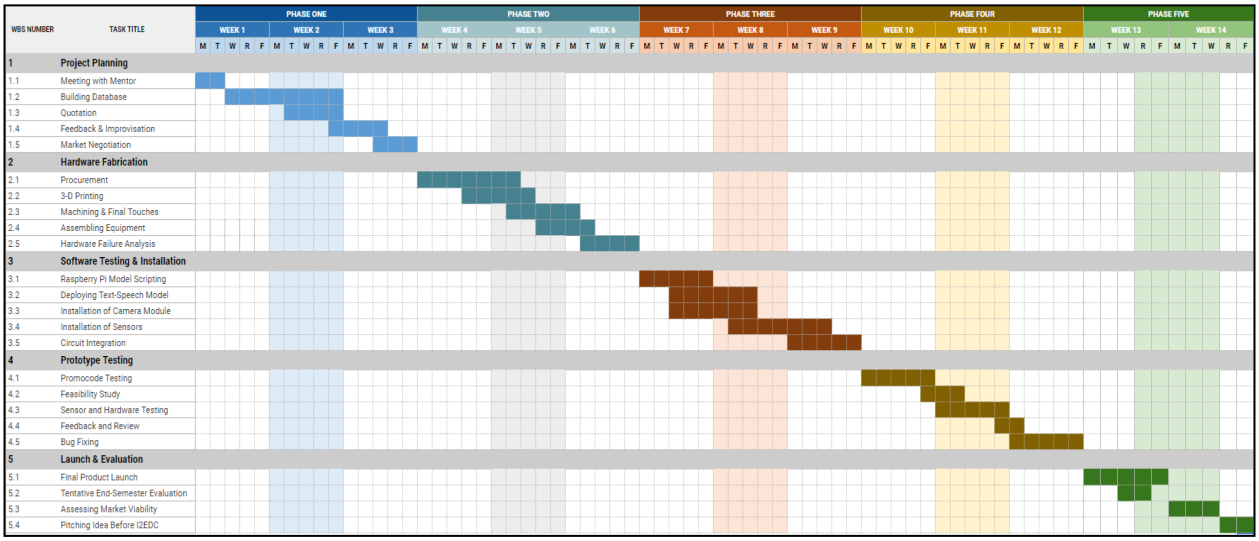
- PYTESSERACT
- gTTS
- PIL

Our workflow will be as follows:



Detailed Timeline:

The projected execution plan has been represented via a Gantt Chart to demonstrate precisely the timeline of our project and the various stages involved during the implementation.



WBS NUMBER	TASK TITLE	PHASE ONE			PHASE TWO			PHASE THREE			PHASE FOUR			PHASE FIVE	
		WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14
		M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F	M T W R F
1	**Project Planning**														
1.1	Meeting with Mentor	M T													
1.2	Building Database	W R F	M T W R F												
1.3	Quotation		T W R F												
1.4	Feedback & Improvisation		F	M T W											
1.5	Market Negotiation			W R F											
2	**Hardware Fabrication**														
2.1	Procurement				M T W R F	M T									
2.2	3-D Printing				R F	M T W									
2.3	Machining & Final Touches					T W R F	M								
2.4	Assembling Equipment					R F	M T								
2.5	Hardware Failure Analysis						T W R F								
3	**Software Testing & Installation**														
3.1	Raspberry Pi Model Scripting							M T W R F							
3.2	Deploying Text-Speech Model							W R F	M T W						
3.3	Installation of Camera Module							W R F	M T W						
3.4	Installation of Sensors								T W R F	M T W					
3.5	Circuit Integration									M T W R F					
4	**Prototype Testing**														
4.1	Promocode Testing										M T W R F				
4.2	Feasibility Study										F	M T			
4.3	Sensor and Hardware Testing											M T W R F			
4.4	Feedback and Review											F	M		
4.5	Bug Fixing												M T W R F		
5	**Launch & Evaluation**														
5.1	Final Product Launch													M T W R F	
5.2	Tentative End-Semester Evaluation													W R	
5.3	Assessing Market Viability														M T W
5.4	Pitching Idea Before I2EDC														R F