VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI – 590018, Karnataka

INTERNSHIP REPORT

ON

“Virtual Assistant for Visually Impaired”

Submitted in partial fulfilment for the **award of the degree of Bachelor of Engineering in AI & ML of Visvesvaraya Technological University, Belagavi**

Submitted by:

G VENKATA SAI AKHIL

1RN20AI022

Conducted at
Compsoft Technologies Pvt Ltd

ESTD : 2001
An Institute with a Difference

Department of AI & ML

RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road,
Bengaluru-560 098
2023-2024

RNS INSTITUTE OF TECHNOLOGY

Channasandra, Dr. Vishnuvardhan Road, Bengaluru-560098

Department of Artificial Intelligence & Machine Learning

ESTD : 2001
An Institute with a Difference

CERTIFICATE

This is to certify that the Internship titled "**Virtual Assistant for Visually Impaired**" has been successfully carried out by **G VENKATA SAI AKHIL** bearing USN **"1RN20AI022",** a bonafide student of **"RNS Institute of Technology"** in partial fulfilment of the requirements for the 7th semester of **"Bachelor of Engineering in Artificial Intelligence and Machine Learning Engineering of Visvesvaraya Technological University"**, Belagavi, during academic year 2023-2024. It is certified that all corrections/suggestions indicated have been incorporated in the report.

The project report has been approved as it satisfies the academic requirements in respect of Internship prescribed for the course Internship / Professional Practice (18CSI85)

Signature of Guide	**Signature of HOD**	**Signature of Principal**
	Dr. Harsha S Professor & Head Dept. Of AI & ML	Dr. H S Ramesh Babu Principal RNSIT, Bengaluru

External Viva:

Name of the Examiner Signature with Date

1)__________________ _________________

2)__________________ _________________

DECLARATION

I, **G VENKATA SAI AKHIL**, final year student of Artificial Intelligence and Machine Learning, RNS Institute of Technology - 560 098 declare that the Internship has been successfully completed, in **COMPSOFT TECHNOLOGIES**. This report is submitted in partial fulfillment of the requirements for award of Bachelor Degree in Artificial Intelligence and Machine Learning, during the academic year 2022-2023.

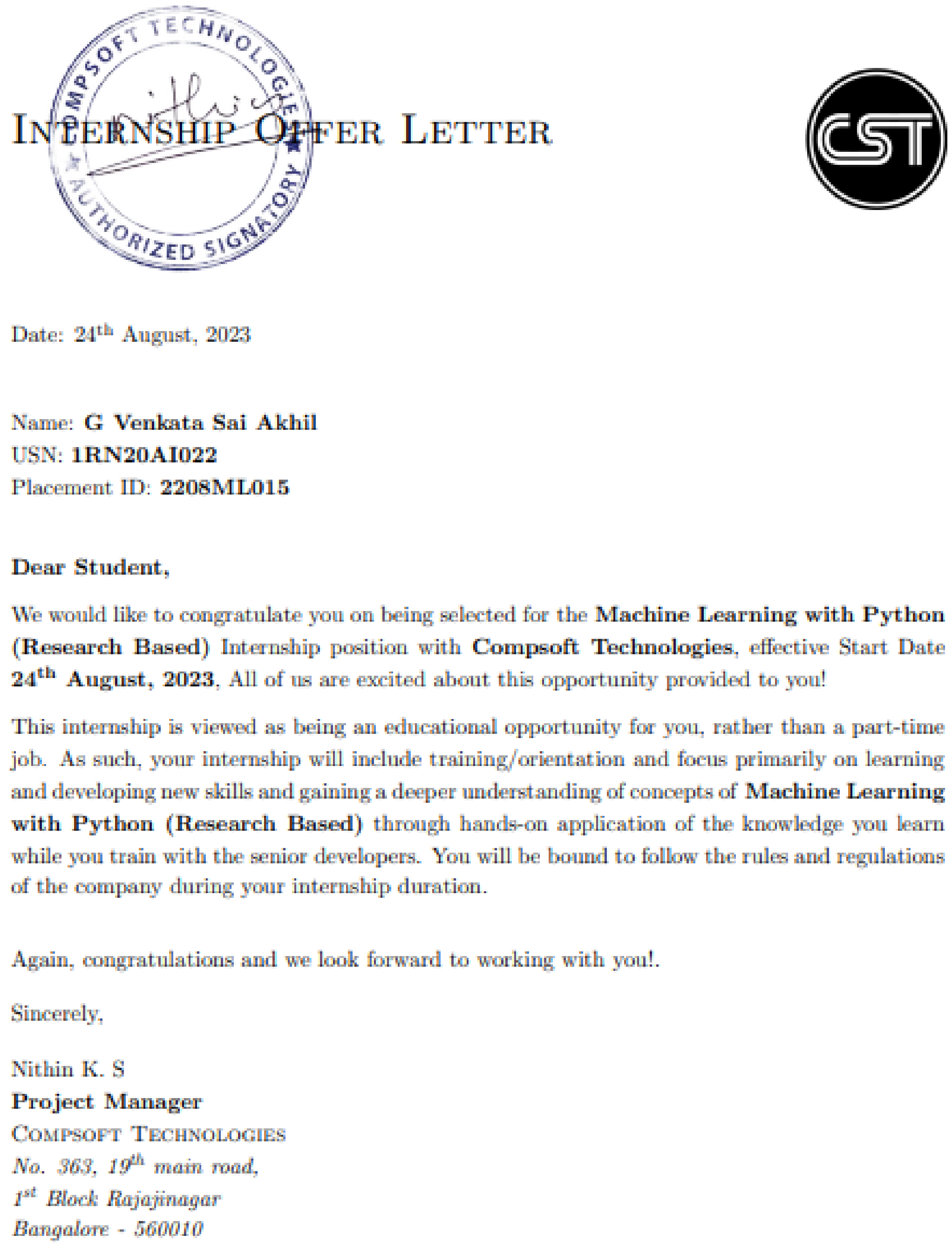
Date : 12/10/2023 :

Place : Bengaluru

USN : 1RN20AI022

NAME : G Venkata Sai Akhil

OFFER LETTER



COMPSOFT TECHNOLOGIES
AUTHORIZED SIGNATORY

Internship Offer Letter

CST

Date: 24th August, 2023

Name: **G Venkata Sai Akhil**
USN: **1RN20AI022**
Placement ID: **2208ML015**

Dear Student,

We would like to congratulate you on being selected for the **Machine Learning with Python (Research Based)** Internship position with **Compsoft Technologies**, effective Start Date **24th August, 2023**, All of us are excited about this opportunity provided to you!

This internship is viewed as being an educational opportunity for you, rather than a part-time job. As such, your internship will include training/orientation and focus primarily on learning and developing new skills and gaining a deeper understanding of concepts of **Machine Learning with Python (Research Based)** through hands-on application of the knowledge you learn while you train with the senior developers. You will be bound to follow the rules and regulations of the company during your internship duration.

Again, congratulations and we look forward to working with you!.

Sincerely,

Nithin K. S
Project Manager
Compsoft Technologies
No. 363, 19th main road,
1st Block Rajajinagar
Bangalore - 560010

ACKNOWLEDGMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities, have helped me in carrying out this internship. I would like to take this opportunity to thank them all.

I am grateful to the **Management** and **Dr. M K Venkatesha**, Director and **Dr. H S Ramesh Babu** Principal, RNSIT, Bangalore, for their support towards completing this mini project.

I would like to thank **Dr. Harsha S**, Head, Department of Artificial Intelligence and Machine Learning Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

I would like to thank all the teaching and non-teaching staff of the Department of Artificial Intelligence and Machine Learning Engineering, RNSIT, Bengaluru-98 for their constant support and encouragement.

Last but not the least, we would like to thank our parents and friends without whose constant help, the completion of Internship would have not been possible.

Signature:

G VENKATA SAI AKHIL

1RN20AI022

ABSTRACT

Jyoti, the virtual assistant designed exclusively for individuals with visual impairments, represents a remarkable advancement in assistive technology. This abstract delves deeper into the device's multifaceted capabilities, underscoring its potential to significantly enhance accessibility and independence for visually impaired users.

At its core, Jyoti is a responsive and intuitive device powered by state-of-the-art technologies. Voice recognition, natural language processing, and computer vision converge to create an indispensable digital companion for those facing visual impairments. Users can interact with Jyoti seamlessly through voice commands, enabling them to ask questions, request assistance, or access information effortlessly.

Jyoti excels in providing navigational support, utilizing GPS and environmental sensors to assist users in navigating both indoor and outdoor environments. It offers turn-by-turn directions and identifies nearby points of interest, fostering a sense of confidence and security during travel.

The implementation of Jyoti translates into tangible benefits for visually impaired individuals. It promotes enhanced independence and self-sufficiency across various domains, including education, employment, daily tasks, and leisure activities. By reducing barriers to information and social inclusion, Jyoti fosters a more inclusive society.

In conclusion, Jyoti stands as a pioneering solution in the realm of assistive technology, addressing the unique challenges faced by visually impaired individuals. Its amalgamation of artificial intelligence and sensory technologies holds the promise of redefining the lives of its users, fostering greater independence and inclusivity within society.

Table of Contents

CHAPTER 1

COMPANY PROFILE

1. COMPANY PROFILE

A Brief History of Company

Compsoft Technologies stands at the forefront of the relentless digital transformation era, where the global landscape is intricately interwoven through rapid advancements in internet technologies. As a beacon for businesses worldwide, we are committed to harnessing the power of real-time, event-driven applications to assist enterprises in responding swiftly and effectively to dynamic market opportunities. Our extensive portfolio spans across diverse industries, including financial services, transportation, energy, retail, healthcare, and gaming companies.

At Compsoft Technologies, we take pride in our proven record of establishing highly scalable, world-class consulting processes that deliver substantial business advantages to our clients. We achieve this through cost-effective solutions, tangible results, and the consistent delivery of projects across the globe. Our core mission centers on enhancing businesses by optimizing operational performance, developing and implementing strategic business plans to improve financial outcomes, and precisely defining and managing strategic goals.

Our vision revolves around going the extra mile to ensure consistent success for our clients. We are dedicated to delivering the right blend of resources, solutions, and services to meet the unique technological challenges and business objectives of each client. In doing so, we aim to optimize client satisfaction through the delivery of high-quality services and to offer the most efficient, top-tier solutions, leveraging leading technologies and industry best practices. At Compsoft Technologies, we are your trusted partner on the journey of digital transformation, empowering you to navigate the rapidly evolving digital landscape with innovation and efficiency, all while ensuring your distinctive goals and needs are met.

CHAPTER 2

ABOUT THE COMPANY

2. ABOUT THE COMPANY

The race towards digital transformation is in full swing, driven by the globally connected and constantly evolving on-demand landscape of internet technologies. Companies around the world face persistent pressure to infuse applications with cutting-edge real-time capabilities to seize emerging market opportunities. Across various sectors, including financial services, transportation, energy, retail, healthcare, and gaming, there's a universal drive to build event-driven, real-time applications.

The mission is centered on simplifying the development of innovative real-time applications while ensuring their efficient operation in production environments. With a well established track record, there is a history of creating highly scalable and world-class consulting processes that bestow significant business advantages upon clients. These advantages manifest in the form of substantial cost savings, definitive results, and unwavering project deliveries across the globe.

At the heart of these efforts lies a commitment to enhance businesses by providing a comprehensive array of competencies, encompassing improved operational performance, the formulation and application of business strategies to enhance financial reports, precise definition and management of strategic goals, and the measurement and careful oversight of these objectives. The ultimate goal is to empower businesses to thrive in the era of digital transformation.

Services provided by Compsoft Technologies:

- Web services and development
- Research and Development/Improvise of ML Models
- Branding and Design
- Search Engine Optimization
- Content Writing
- Research
- Embedded System and IOT

CHAPTER 3

INTRODUCTION

3. INTRODUCTION

Introduction to ML

Machine Learning (ML) is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computer systems to improve their performance on a specific task through learning from data, without being explicitly programmed. In essence, it's a method for teaching computers to make predictions or decisions based on patterns and information present in the data. These are the key concepts and principles of machine learning: Learning from Data, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Model Building, Training and Evaluation, Overfitting and Underfitting. Machine learning continues to advance rapidly and has the potential to revolutionize various fields by enabling automated decision-making and data-driven insights. It's a dynamic and evolving field with ongoing research and practical applications in diverse domains

Problem Statement

Visually impaired individuals confront substantial barriers to accessibility and independence, impeding their full participation in society. A significant issue they encounter is the limited access to information, including printed and digital content, hampering educational, vocational, and social inclusion. Navigating unfamiliar environments poses safety and mobility challenges, compromising their ability to independently access services and engage in daily life activities.

Existing assistive technologies, while valuable, often lack the comprehensive capabilities required to effectively address these multifaceted challenges. Thus, there is a critical need for innovative solutions that harness advanced technologies like natural language processing, voice recognition, computer vision, and navigational support. Integrating these technologies into a cohesive system, tailored to the unique needs of visually impaired individuals, can bridge the accessibility gap.

These technologies must incorporate advanced features such as voice recognition, natural language processing, computer vision, and navigational support to provide a comprehensive solution that enhances independence, accessibility, and inclusivity for this population

CHAPTER 4

SYSTEM ANALYSIS

4. SYSTEM ANALYSIS

1. Existing System

The existing system for assisting visually impaired individuals is characterized by traditional, non-integrated solutions:

1. **Assistive Tools:** Visually impaired individuals currently rely on a combination of traditional assistive tools and devices, such as screen readers, braille displays, white canes, and guide dogs.
2. **Limited Integration:** These tools typically operate in isolation from each other, making it necessary for users to switch between different devices for various tasks.
3. **Challenges with Digital Content:** While screen readers can convert digital text to speech, they may not work optimally with images, printed materials, or real-time environmental data.
4. **Lack of Mobility Support:** Traditional systems do not provide real-time navigation support or information about the immediate environment.
5. **Limited Interaction:** There is minimal interaction with smartphones and other digital devices for activities such as reading messages, accessing calendars, and responding to notifications.

2. Proposed System

The proposed system, a Virtual Assistant for Visually Impaired (VAVI), represents a comprehensive and integrated solution. VAVI harnesses advanced technologies like natural language processing, voice recognition, computer vision, and navigation support to provide a unified and adaptable platform. This holistic approach aims to empower visually impaired individuals with enhanced accessibility, independence, and inclusivity in various aspects of life.

The proposed system, " VisioVoice," is an innovative and integrated solution designed to significantly improve the quality of life for visually impaired individuals:

1. Advanced Hardware Setup: VisioVoice features a unique hardware setup consisting of wearable headphones interfaced to a Logitech webcam connected to a Raspberry Pi, enabling a seamless and portable experience.
2. Comprehensive Voice Assistant: VisioVoice serves as a versatile and comprehensive voice assistant that interprets user input in the form of speech and responds with spoken output.
3. Real-time Environment Description: VisioVoice provides a concise description of the user's surroundings, enhancing situational awareness.

4. Road Condition Information: The system delivers real-time updates on road conditions, offering safety and mobility support for outdoor navigation.
5. Location Identification: VisioVoice can identify common places, making it easier for users to recognize and navigate within familiar environments.
6. Object and People Detection: The system informs users about the number of people and objects within the camera's view, enhancing situational awareness.
7. "Find" Commands: Users can issue voice commands to find specific objects or entities, and Jyoti scans the camera's feed for their presence, providing feedback.
8. Text Detection and Reading: VisioVoice detects and reads text from images, making printed materials accessible. It can also summarize articles from newspapers.
9. Form Reading: The system can read out forms, particularly useful for handling documents related to banking and other purposes.

3. Objective of the System

The primary objective of the Virtual Assistant for Visually Impaired (VAVI) system is to profoundly enhance the lives of individuals with visual impairments by addressing their unique challenges and needs. The system is meticulously designed to achieve several crucial goals:

1. **Improving Accessibility:** VAVI aims to break down barriers by providing visually impaired individuals with seamless access to digital and printed information. This includes educational materials, job-related content, and online resources.
2. **Enhancing Independence:** VAVI is designed to promote greater independence in daily tasks and activities, from reading and navigating to accessing services and information without assistance.
3. **Fostering Inclusivity:** By offering real-time access to digital content and services, VAVI seeks to reduce social isolation and empower visually impaired individuals to participate fully in society, including social interactions, news, and entertainment.
4. **Adaptability and Evolution:** The system's objective is to remain adaptable and incorporate emerging technologies, ensuring that it continues to meet the changing needs of visually impaired individuals and remains relevant in a rapidly evolving technological landscape

CHAPTER 5

REQUIREMENT ANALYSIS

5. REQUIREMENT ANALYSIS

Hardware Requirement Specification

- Central Processing Unit (CPU)
- Graphics Processing Unit (GPU)
- Memory (RAM): 64GB
- Storage: SSD’s
- Cloud Computing
- Backup and Redundancy

Software Requirement Specification

- VSCode
- Jupyter Notebook
- Python 3.9 and above.
- Miniconda3
- Various python Libraries like Sklearn, Pandas, Numpy, TensorFlow and so on.

CHAPTER 6

DESIGN ANALYSIS

6. DESIGN & ANALYSIS

1. **User-Centric Interface:** Prioritize an intuitive, voice-driven interface for ease of use.
2. **Accessibility Features:** Include screen readers, voice prompts, and high-contrast text for accessibility.
3. **Natural Language Processing:** Implement NLP for seamless, conversational interactions.
4. **Computer Vision:** Integrate computer vision for object recognition and scene analysis.
5. **Navigation & Wayfinding:** Develop GPS-based navigation with obstacle detection and points of interest.
6. **Smart Device Integration:** Enable control of smart home devices using voice commands.
7. **User Testing:** Continuously gather feedback from visually impaired users for usability improvements.
8. **Usability Testing:** Regularly assess the system's usability to address pain points.
9. **Data Privacy & Security:** Ensure robust data security and user privacy safeguards.
10. **Scalability & Performance:** Analyze scalability and performance for responsiveness as the user base grows.
11. **Third-Party Integration:** Evaluate the reliability of third-party service integrations for information retrieval.
12. **Accessibility Standards:** Confirm compliance with accessibility standards like WCAG to cater to diverse visual impairments.

CHAPTER 7

IMPLEMENTATION

7. IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and it constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning.

Two major tasks of preparing the implementation are education and training of the users and testing of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

TESTING

The testing phase is an important part of software development. It is the Information zed system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. Software testing is carried out in three steps:

1. The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether theobjectives have been met. Errors are noted down and corrected immediately.

2. Unit testing is the important and major part of the project. So errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So unit testing is conducted to individual modules.

3. The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole.

Detecting code:

```
import io
from google.oauth2 import service_account
from google.cloud import vision
import dialogflow_v2 as dialogflow
import cv2

def detect_text(cam, engine):
    credentials = service_account.Credentials.from_service_account_file('aj.json')
    client = vision.ImageAnnotatorClient(credentials=credentials)
    ret, content = cam.read()
    cv2.imwrite('op.jpg', content)
    with io.open('op.jpg', 'rb') as image_file:
        content = image_file.read()
    image = vision.types.Image(content=content)
    response = client.text_detection(image=image)
    texts = response.text_annotations
    print(len(texts))
    print('Text:')
    textm = ""
    for i, text in enumerate(texts):
        engine.text_speech(text.description)
        textm += text.description
        textm = textm + " "
    print(textm)

def detect_form(cam, engine):

    credentials = service_account.Credentials.from_service_account_file('aj.json')
    client = vision.ImageAnnotatorClient(credentials= credentials)
    #content = cam.read()
    path = 'bank.jpg'
    with io.open(path, 'rb') as image_file:
        content = image_file.read()
    image = vision.types.Image(content=content)
    response = client.text_detection(image=image)
    texts = response.text_annotations

    print('Text:')
    textm = ""
    for i, text in enumerate(texts):
        if(i==0):
            engine.text_speech("The form is entitled as")
        if(i==1):
            engine.text_speech("The form asks about these details")
        engine.text_speech(text.description)
        if("Official" in text.description):
            break
        textm += text.description
        textm = textm+" "
    print(textm)

def describeScene(cam, model, engine):
    ret, frame = cam.read()
    cv2.imwrite('op.jpg',frame)
    credentials = service_account.Credentials.from_service_account_file('aj.json')
    client = vision.ImageAnnotatorClient(credentials= credentials)
    path = 'op.jpg'
    #path = 'road.jpg'
    with io.open(path, 'rb') as image_file:
        content = image_file.read()
    image = vision.types.Image(content=content)
    response = client.label_detection(image=image)
    labels = response.label_annotations
    engine.text_speech("Description of the view")
    stop = 2
    for i,j in enumerate(labels):
        engine.text_speech(j.description)
        if(i == 1):
            break

    checkRoad(labels, engine)
    tellObjects(client, image, engine)

def checkRoad(labels, engine):
    road = 0
    car = 0
    motor_vehicle = 0
    bicycle = 0
    classroom = 0
    truck = 0
    traffic = 0
    face = 0
    for i, label in enumerate(labels):
        if (label.description == "Highway" or label.description == "Lane" or label.description ==
"Road"):
            road += 1
        if (label.description == "Car"):
            car += 1
        if (label.description == "Motor vehicle"):
            motor_vehicle += 1
        if (label.description == "Bicycle"):
            bicycle += 1
        if (label.description == "Truck"):
            truck += 1
        if (label.description == "Face"):
            face += 1
        if (label.description == "Classroom"):
            classroom += 1
        if (label.description == "Traffic"):
            traffic += 1
    if (road >= 1):
        if (car >= 1 or motor_vehicle >= 1 or bicycle >= 1 or truck >= 1 or traffic >= 1):
            engine.text_speech(
                "It seems you are walking on a road with vehicles. Beware! Do you want me to find
people for help?")
        else:
            engine.text_speech("It seems the road you are walking on is quite safe. Yet beware.")
    if (classroom >= 1):
        engine.text_speech("You seem to be in a classroom!")


def tellObjects(client, image,  engine):
    objects = client.object_localization(
        image=image).localized_object_annotations
    print('Number of objects found: {}'.format(len(objects)))
    # engine.text_speech("I will tell you the objects near you")
    for object_ in objects:
        print('{} '.format(object_.name))
        # engine.text_speech(object_.name)
    lbldict = {}
    for i in objects:
        if i.name in lbldict:
            lbldict[i.name] += 1
        else:
            lbldict[i.name] = 1
    once = True
    length = len(lbldict)
    r = 0
    for i, j in lbldict.items():
        if once:
            if j != 1:
                engine.text_speech("There are")
            else:
                engine.text_speech("There is")
            once = False
        engine.text_speech("{} {}".format(j, i))
        r += 1
        if r != length:
            engine.text_speech("and")
    if (length == 0):
        engine.text_speech("No objects found")5
```

Functions code:

```
import cv2
import numpy as np
import wave
import pyaudio
```

```
def getBrightness(cam):
    ret, frame = cam.read()
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    avg = np.sum(frame)/(frame.shape[0]*frame.shape[1])
    avg=avg/255
    if(avg > 0.6):
        return ("Very bright", avg)
    if(avg > 0.4):
        return ("Bright", avg)
    if(avg>0.2):
        return ("Dim", avg)
    else:
        return ("Dark",avg)


def play_file(fname):
    # create an audio object
    wf = wave.open(fname, 'rb')
    p = pyaudio.PyAudio()
    chunk = 1024

    # open stream based on the wave object which has been input.
    stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
                    channels=wf.getnchannels(),
                    rate=wf.getframerate(),
                    output=True)

    # read data (based on the chunk size)
    data = wf.readframes(chunk)

    # play stream (looping from beginning of file to the end)
    while data != '':
        # writing to the stream is what *actually* plays the sound.
        stream.write(data)
        data = wf.readframes(chunk)

        # cleanup stuff.
    stream.close()
    p.terminate()
```

Main Function:

```
import functions
import yolopy
import speech
import cv2
import os
import detect
import datetime
```

```
os.environ["GOOGLE_APPLICATION_CREDENTIALS"]= "dfkey.json"

labelsPath = "yolo/coco.names"
weightsPath = "yolo/yolov3.weights"
configPath = "yolo/yolov3.cfg"
args = {"threshold":0.3, "confidence":0.5}
project_id = "blindbot-4f356"
#project_id = "blindbot-286ed"
engine = speech.speech_to_text()

model = yolopy.yolo(labelsPath, weightsPath, configPath)
listening = False
intent = None
while True:
    cam = cv2.VideoCapture(1)
    if not listening:
        resp = engine.recognize_speech_from_mic()
        print(resp)
        if(resp != None):
            intent, text = detect.detect_intent_texts(project_id, 0, [resp], 'en')
        if(intent == 'Jyoti' and resp!=None):
            listening = True

    else:
        engine.text_speech("What can I help you with?")
        intent = ''
        engine.text_speech("Listening")
        resp = engine.recognize_speech_from_mic()
        engine.text_speech("Processing")
        if(resp!=None):
            print(resp)
            intent, text = detect.detect_intent_texts(project_id, 0, [resp], 'en')
        if intent == 'Describe':
            detect.describeScene(cam, model, engine)
        elif intent == 'endconvo':
            print(text)
            listening = False
            engine.text_speech(text)
        elif intent == 'Brightness':
            engine.text_speech("It is {} outside".format((functions.getBrightness(cam))[0]))
        elif intent == "FillForm":
            detect.detect_form(cam, engine)
        elif intent == "Read":
            print("read")
            detect.detect_text(cam, engine)
        elif intent == "Time":
            currentDT = datetime.datetime.now()
            engine.text_speech("The time is {} hours and {} minutes".format(currentDT.hour,
currentDT.minute))
        elif resp != 'None':
            engine.text_speech(text)
    cam.release()
```

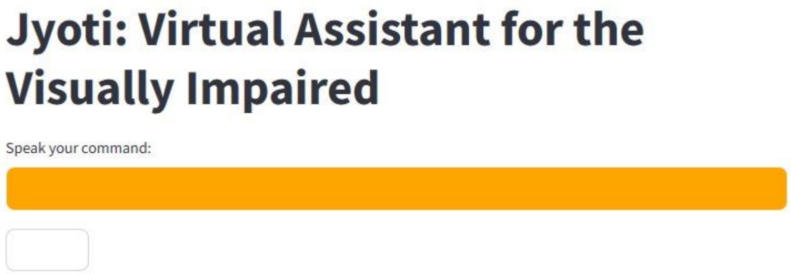
Speech function:

```
import speech_recognition as sr
import pyttsx3
from google.oauth2 import service_account

from nltk.stem.porter import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
import re
import nltk
from nltk.corpus import stopwords

from nltk.tokenize import RegexpTokenizer
from nltk.stem.wordnet import WordNetLemmatizer

class speech_to_text():
    def __init__(self):
        en_voice_id =
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-
US_ZIRA_11.0"
        ru_voice_id =
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_RU-
RU_IRINA_11.0"
        self.recognizer = sr.Recognizer()
        self.microphone = sr.Microphone()
        self.engine = pyttsx3.init();
        self.engine.setProperty('voice', en_voice_id)
        self.credentials = service_account.Credentials.from_service_account_file('api-key.json')


    def recognize_speech_from_mic(self):
        print("Start...")
        with self.microphone as source:
            self.recognizer.adjust_for_ambient_noise(source)
            audio = self.recognizer.listen(source)
        print("Found mic")
        response = {
            "success": True,
            "error": None,
            "transcription": None
        }
        try:
            response["transcription"] = self.recognizer.recognize_google(audio)
        except sr.RequestError:
            # API was unreachable or unresponsive
            response["success"] = False
            response["error"] = "API unavailable"
        except sr.UnknownValueError:
            # speech was unintelligible
```

CHAPTER 8

SNAPSHOTS

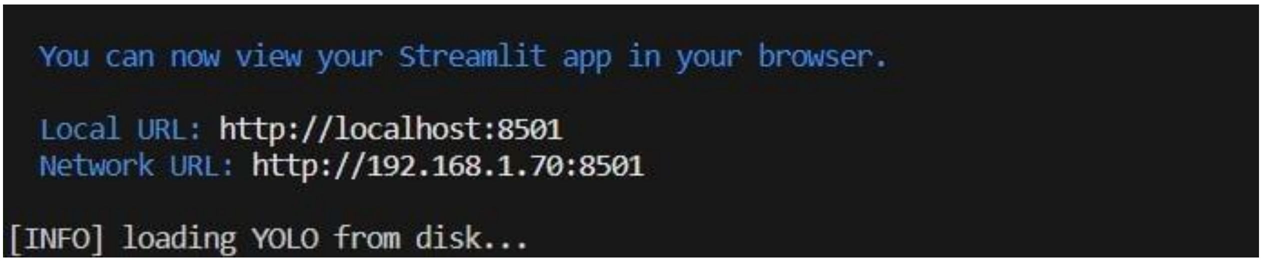
8. SNAPSHOTS



Jyoti: Virtual Assistant for the Visually Impaired

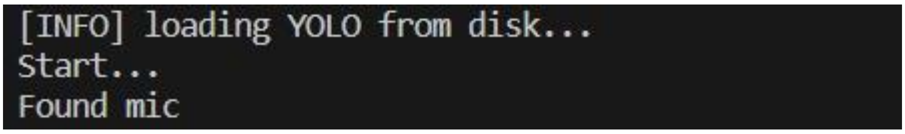
Speak your command:

Fig 8.1 Frontend of the project while started running



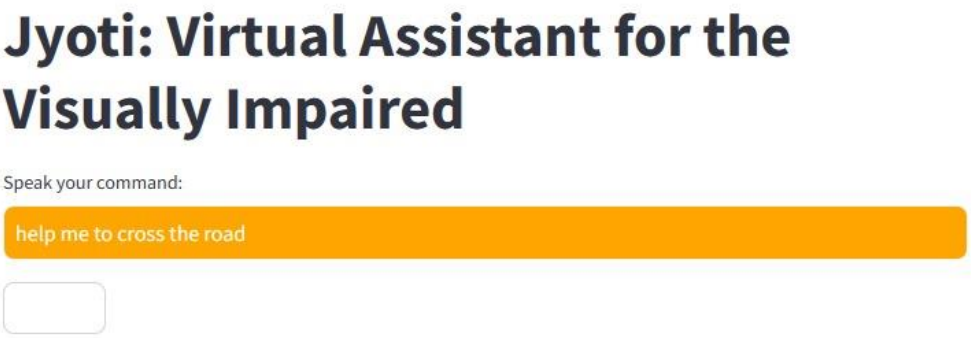
```
  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.70:8501

[INFO] loading YOLO from disk...
```

Fig 8.2 Backend of project loading YOLO from disk and running Streamlit app



```
[INFO] loading YOLO from disk...
Start...
Found mic
```

Fig 8.3 Backend of the project Started and listening for input command or asking to speak your command

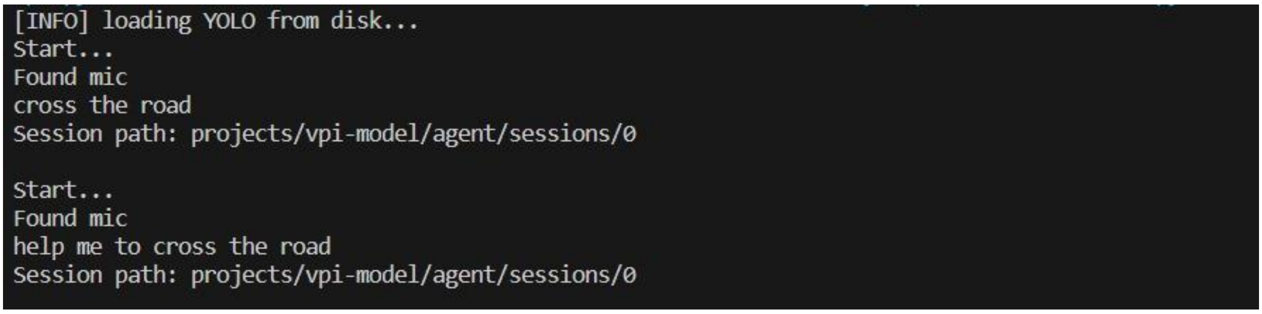


Jyoti: Virtual Assistant for the Visually Impaired

Speak your command:

help me to cross the road

Fig 8.4 Frontend of the model taking the input command



```
[INFO] loading YOLO from disk...
Start...
Found mic
cross the road
Session path: projects/vpi-model/agent/sessions/0

Start...
Found mic
help me to cross the road
Session path: projects/vpi-model/agent/sessions/0
```

Fig 8.5 Backend of the model processing the input command

CHAPTER 9

CONCLUSION

9. CONCLUSION

The "Virtual Assistant for Visually Impaired" project is a valuable and innovative application aimed at enhancing the daily lives of visually impaired individuals. This project successfully combines a variety of technologies and modules to create a comprehensive virtual assistant that provides various functionalities to assist visually impaired users.

Key highlights and conclusions of the project include:

Enhanced Accessibility: The project offers enhanced accessibility to visually impaired users by providing them with a virtual assistant capable of understanding their voice commands and offering a range of services.

Voice Recognition: The integration of Google Cloud's speech-to-text service enables the assistant to recognize voice commands and respond accordingly. This empowers users to interact with the assistant through speech, making it a user-friendly tool for those with visual impairments.

Object Recognition: The project utilizes a YOLO (You Only Look Once) model to detect and describe objects in real-time. This feature is especially useful for visually impaired users, as it can help them navigate their environment more effectively.

Time and Date Information: Users can easily retrieve time and date information through voice commands, which is essential for scheduling daily activities and staying organized.

User-Friendly Interface: The Streamlit-based user interface is designed with the needs of visually impaired users in mind. It incorporates high-contrast colors, larger text, and input fields, ensuring a user-friendly experience.

In conclusion, the "Virtual Assistant for Visually Impaired" project showcases the potential of technology to improve the lives of those with visual impairments. It offers a wide range of functionalities, including voice recognition, object detection, form filling, and more. The professional and user-friendly interface further enhances its appeal. This project has the potential to make a meaningful difference in the lives of visually impaired individuals and provides a strong starting point for further development and innovation in this important area.

10. REFERENCES

- https://chat.openai.com/
- https://github.com/
- https://www.youtube.com/
- Kirsty Williamson, Steve Wright, Don Schauder and Amanda Bow, "The internet for the blind and visually impaired", Journal of Computer-Mediated Communication, vol. 7, no. 1, October 2001.
- "Cloud Text To Speech Basics", Google, July 2019, [online] Available: https://cloud.google.com/speech- to-text/ docs/basics. [14].
- Google Cloud, Cloud Text-to-Speech basics, July 2019, [online] Available: https://cloud.google.com/text-to-speech/ docs/basics
- "SSD: Single Shot Detector for object detection using MultiBox", 30 Nov. 2019, [Online] Available: https://towardsdatascience.com/ .