



INNOMATICS  
RESEARCH LABS

## ***PROJECT REPORT ON***

AI-Powered Solution for Assisting Visually Impaired Individuals

*Submitted by:*

Payal Kumari

**INNOMATICS RESEARCH LABS**

# **AI-Powered Solution for Assisting Visually Impaired Individuals**

## **Problem Statement:**

- Visually impaired individuals often face challenges in understanding their environment, reading visual content, and performing tasks that rely on sight.
- This project leverages Generative AI to assist visually impaired individuals by providing functionalities like real-time scene understanding, text-to-speech conversion, and object detection.

## **Objectives:**

- Provide real-time scene understanding.
- Extract and convert text from images to speech.
- Identify objects and obstacles for safe navigation.
- Offer personalized assistance for daily tasks.

## **Implementation Details**

### **Technologies Used**

- 1. *LangChain*:** Conversational AI capabilities.
- 2. *Streamlit*:** User-friendly web application interface.

**3. Google Generative AI (Gemini API):** Generates detailed scene descriptions.

**4. Pytesseract:** Optical Character Recognition (OCR) to extract text from images.

**5. pyttsx3:** Text-to-Speech engine for audio output.

## Functionalities Implemented:

### 1. Real-Time Scene Understanding

- Generates detailed scene descriptions using Google Generative AI.
- Output includes objects, overall scene description, and safety suggestions.

### 2. Text-to-Speech Conversion

- Extracts text from images using OCR.
- Converts the extracted text to audible speech for accessibility.

## User Interaction Flow:

**1. Image Upload:** Users upload an image through the Streamlit app.

**2. Feature Selection:** Choose between scene description, text extraction, or text-to-speech.

### 3. Processing and Output:

- Scene Understanding: Detailed descriptions generated.
- OCR and Speech: Text extracted and read aloud.

## **Outputs:**

### **Example 1: Scene Understanding**

- **Uploaded Image:** A street with pedestrians and vehicles.
- **Generated Description:**
  - "The image shows a busy street with pedestrians crossing at a crosswalk and vehicles waiting at a traffic light."
  - Suggestions: "Ensure to cross using designated crosswalks with assistance if needed."

### **Example 2: Text Extraction and Speech**

- **Uploaded Image:** A signboard with text "Welcome to Vision Assist."
- **Extracted Text:** "Welcome to Vision Assist."
- **Speech Output:** Text is read aloud as audio.

## **Implementation Code:**

```
```python
import streamlit as st
from PIL import Image
import pytesseract
import pyttsx3
import google.generativeai as genai
```

## Initialize Google Generative AI

```
GEMINI_API_KEY      =      "your_api_key"

genai.setup(api_key=GEMINI_API_KEY)

engine = pyttsx3.init()

def extract_text_from_image(image):

    text = pytesseract.image_to_string(image)

    return text

def text_to_speech(text):

    engine.say(text)

    engine.runAndWait()

def generate_scene_description(prompt, image_data):

    model = genai.GenerativeModel('gemini-1.5-pro')

    response = model.generate_content([prompt, image_data[0]])

    return response.text
```

## Streamlit Application

```
st.title("SightAssist: AI for Visually Impaired")

uploaded_file = st.file_uploader("Upload an

image") if uploaded_file:

    image = Image.open(uploaded_file)
```

**INNOMATICS RESEARCH LABS**

```
st.image(image, caption="Uploaded Image")

text = extract_text_from_image(image)

st.write("Extracted Text: ", text)

text_to_speech(text)

...
```

## IMPLEMENTATION:

```
# Buttons for functionalities

col1, col2, col3 = st.columns(3)

scene_button = col1.button("🔍 Describe
Scene")

ocr_button = col2.button("📝 Extract Text")

tts_button = col3.button("🔊 Text-to-
Speech")
```

```
# Input Prompt for AI Scene Understanding
```

```
input_prompt = """"
```

You are an AI assistant helping visually impaired individuals by describing the scene in the image. Provide:

1. List of items detected in the image with

their purpose.

2. Overall description of the image.
3. Suggestions for actions or precautions for the visually impaired.

"""

```
# Process based on user interaction

if uploaded_file:

    image_data =

    input_image_setup(uploaded_file)

    if scene_button:

        with st.spinner("Generating scene
description..."):

            response =

            generate_scene_description(input_prompt,
image_data)

            st.subheader("Scene Description")

            st.write(response)

    if ocr_button:

        with st.spinner("Extracting text from
image..."):

            text =
```

```
extract_text_from_image(image)

    st.subheader("Extracted Text")

    st.write(text)

if tts_button:

    with st.spinner("Converting text to
speech..."):

        text =

extract_text_from_image(image)

if text.strip():

    text_to_speech(text)

    st.success("Text-to-Speech
Conversion Completed!")

else:

    st.warning("No text found in the
image.")
```

## Evaluation Criteria

- ❖ Uniqueness of Implementation: The project integrates cutting-edge Generative AI and OCR with a simple UI.
- ❖ Successful Features: Implements at least two core functionalities.
- ❖ Technical Accuracy: Uses Google Generative AI, OCR, and TTS effectively.
- ❖ Documentation: Provides clear code and explanation.

## Conclusion

- This project demonstrates an effective application of AI to assist visually impaired individuals. It combines real-time scene understanding with OCR and text-to-speech capabilities, enabling enhanced accessibility and independence.