Title: Building AI Powered Solution for Assisting Visually Impaired Individuals

Author: Chandu Geethanjali

Internship: Data Science and Generative AI

Innomatics Research Labs

Abstract

The "AI Vision Care Assist" application leverages cutting-edge Artificial Intelligence to aid visually impaired individuals in understanding and interacting with their environment. This project integrates scene understanding, text extraction, object detection, and task-specific assistance, making it a versatile and practical tool. With multilingual support and text-to-speech capabilities, it provides a highly accessible user experience.

Introduction

- **Problem Statement**:

Visually impaired individuals face significant challenges in navigating their environment and understanding visual information. Existing solutions often lack multi-functionality or regional language support.

- **Objective**:

To develop an AI-powered application that assists visually impaired users by providing scene descriptions, text extraction, object detection, and task-specific guidance in multiple languages.

Features

Scene Description

- **What it does**: Generates a textual description of the uploaded image using Google Generative AI.
- **How it helps**: Provides an overview of the scene to users who cannot see it.

Text Extraction

- **What it does**: Extracts text from images using Tesseract OCR.
- **How it helps**: Allows users to read printed text from documents, labels, and signage.

Object Detection

- **What it does**: Detects objects in images using a pre-trained Faster R-CNN model.
- **How it helps**: Assists users in identifying obstacles or items in their environment.

Personalized Assistance

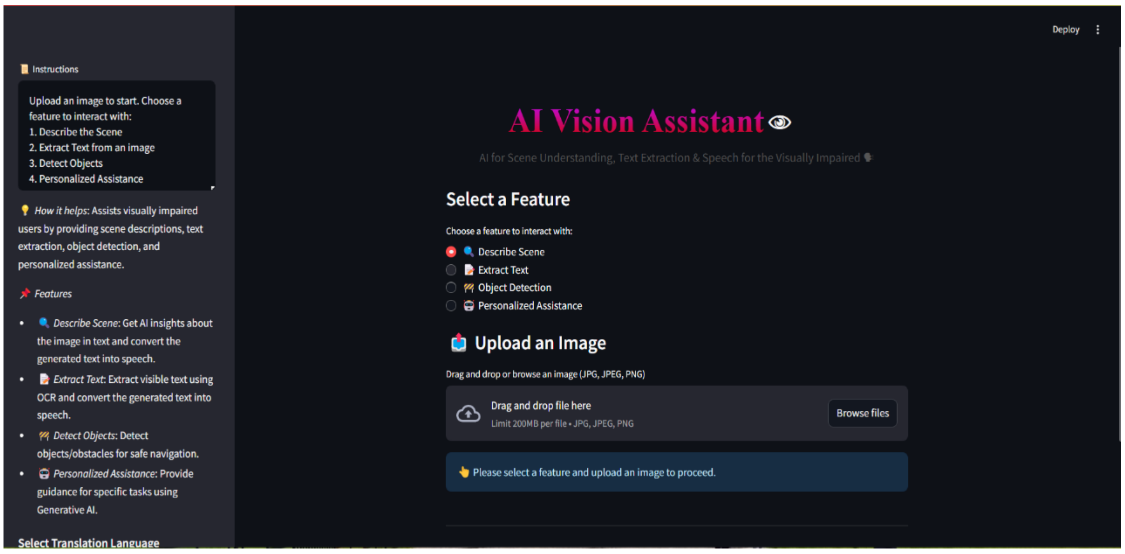
- **What it does**: Uses Generative AI to provide task-specific guidance based on the image content.
- **How it helps**: Offers detailed assistance, such as reading labels or recognizing items.

Text-to-Speech

- **What it does**: Converts the generated text into speech using pyttsx3.
- **How it helps**: Enables users to listen to the content in English.

Multilingual Support

- **What it does**: Translates text into regional languages (Telugu, Hindi, Kannada, and Malayalam) using Google Translate API.
- **How it helps**: Improves accessibility for non-English-speaking users.



Instructions

Upload an image to start. Choose a feature to interact with:
1. Describe the Scene
2. Extract Text from an image
3. Detect Objects
4. Personalized Assistance

💡 *How it helps*: Assists visually impaired users by providing scene descriptions, text extraction, object detection, and personalized assistance.

📌 *Features*

- 🔍 *Describe Scene*: Get AI insights about the image in text and convert the generated text into speech.
- 📝 *Extract Text*: Extract visible text using OCR and convert the generated text into speech.
- 🚧 *Detect Objects*: Detect objects/obstacles for safe navigation.
- 🤖 *Personalized Assistance*: Provide guidance for specific tasks using Generative AI.

Select Translation Language

AI Vision Assistant 👁

AI for Scene Understanding, Text Extraction & Speech for the Visually Impaired 🗣

Select a Feature

Choose a feature to interact with:

[✓ 🔍 Describe Scene]
📝 Extract Text
🚧 Object Detection
🤖 Personalized Assistance

📤 Upload an Image

Drag and drop or browse an image (JPG, JPEG, PNG)

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

👆 Please select a feature and upload an image to proceed.

Technologies Used

- **Streamlit**: For building the user interface.
- **Google Generative AI (Gemini API)**: For generating scene descriptions and personalized assistance.
- **Tesseract OCR**: For extracting text from images.
- **PyTorch (Faster R-CNN)**: For object detection.
- **Google Translate API**: For multilingual text translation.
- **pyttsx3**: For text-to-speech functionality.
- **Python Libraries**: PIL, torchvision, dotenv, and others.

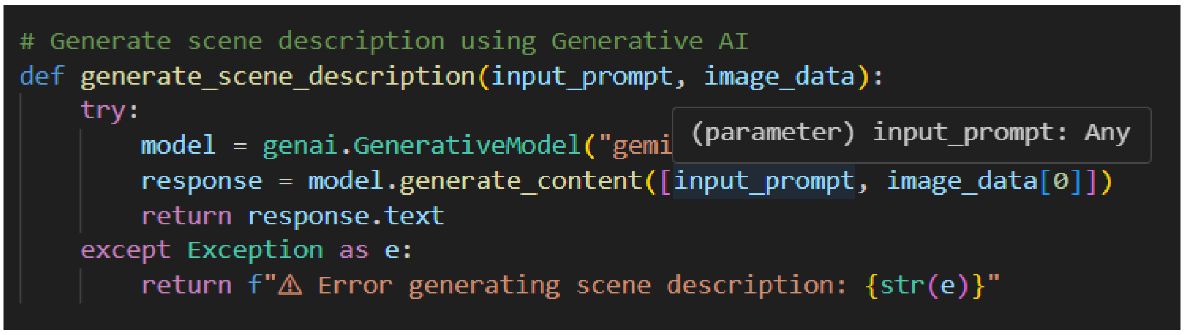
Implementation :

Scene Description

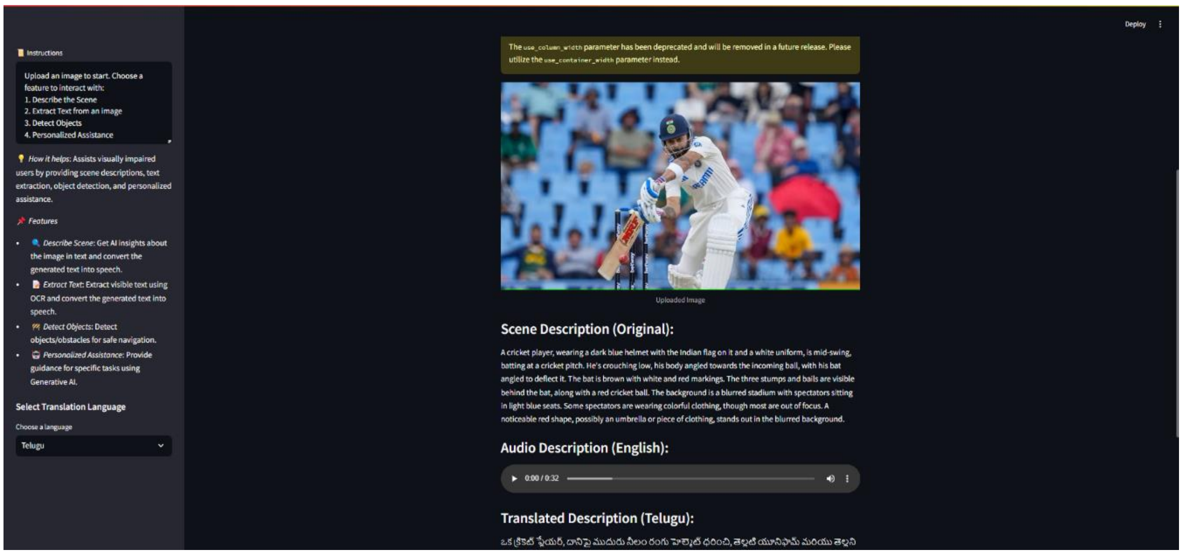
Provides an overview of the scene to users who cannot see it.

Google Generative AI (Gemini API): For generating scene descriptions and personalized assistance.

pyttsx3: For text-to-speech functionality.



```python
# Generate scene description using Generative AI
def generate_scene_description(input_prompt, image_data):
    try:
        model = genai.GenerativeModel("gemi (parameter) input_prompt: Any
        response = model.generate_content([input_prompt, image_data[0]])
        return response.text
    except Exception as e:
        return f"⚠ Error generating scene description: {str(e)}"
```



📒 Instructions

Upload an image to start. Choose a feature to interact with:
1. Describe the Scene
2. Extract Text from an image
3. Detect Objects
4. Personalized Assistance

💡 *How it helps*: Assists visually impaired users by providing scene descriptions, text extraction, object detection, and personalized assistance.

📌 *Features*

- 🔍 *Describe Scene*: Get AI insights about the image in text and convert the generated text into speech.
- 📝 *Extract Text*: Extract visible text using OCR and convert the generated text into speech.
- 🚧 *Detect Objects*: Detect objects/obstacles for safe navigation.
- 🤖 *Personalized Assistance*: Provide guidance for specific tasks using Generative AI.

Select Translation Language

Choose a language

Telugu

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

Uploaded Image

Scene Description (Original):

A cricket player, wearing a dark blue helmet with the Indian flag on it and a white uniform, is mid-swing, batting at a cricket pitch. He's crouching low, his body angled towards the incoming ball, with his bat angled to deflect it. The bat is brown with white and red markings. The three stumps and bails are visible behind the bat, along with a red cricket ball. The background is a blurred stadium with spectators sitting in light blue seats. Some spectators are wearing colorful clothing, though most are out of focus. A noticeable red shape, possibly an umbrella or piece of clothing, stands out in the blurred background.

Audio Description (English):

0:00 / 0:32

Translated Description (Telugu):

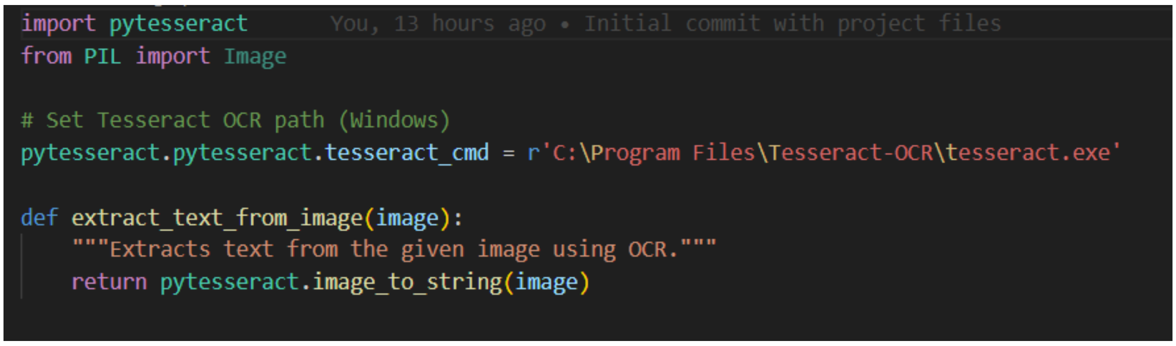
ఒక క్రికెట్ ప్లేయర్, దానిపై ముదురు నీలం రంగు హెల్మెట్ ధరించి, తెల్లటి యూనిఫామ్ మరియు తెల్లని

Text Extraction :

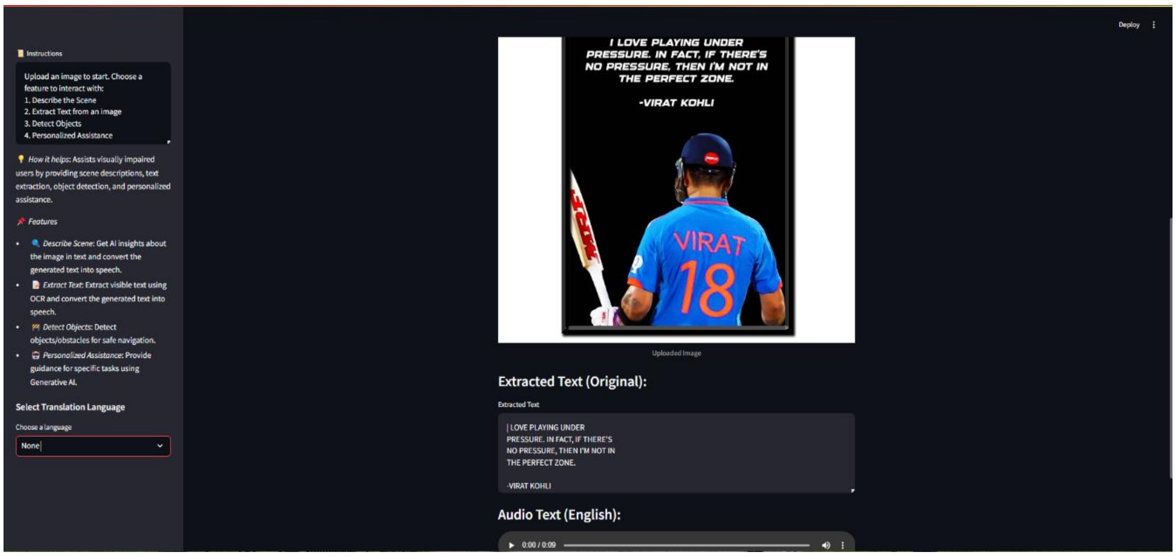
Allows users to read printed text from documents, labels, and signage.

Tesseract OCR: For extracting text from images.

pyttsx3: For text-to-speech functionality.



```python
import pytesseract        You, 13 hours ago • Initial commit with project files
from PIL import Image

# Set Tesseract OCR path (Windows)
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

def extract_text_from_image(image):
    """Extracts text from the given image using OCR."""
    return pytesseract.image_to_string(image)
```



Instructions

Upload an image to start. Choose a feature to interact with:
1. Describe the Scene
2. Extract Text from an image
3. Detect Objects
4. Personalized Assistance

How it helps: Assists visually impaired users by providing scene descriptions, text extraction, object detection, and personalized assistance.

Features

- *Describe Scene*: Get AI insights about the image in text and convert the generated text into speech.
- *Extract Text*: Extract visible text using OCR and convert the generated text into speech.
- *Detect Objects*: Detect objects/obstacles for safe navigation.
- *Personalized Assistance*: Provide guidance for specific tasks using Generative AI.

Select Translation Language

Choose a language: None

Uploaded Image

Extracted Text (Original):

Extracted Text

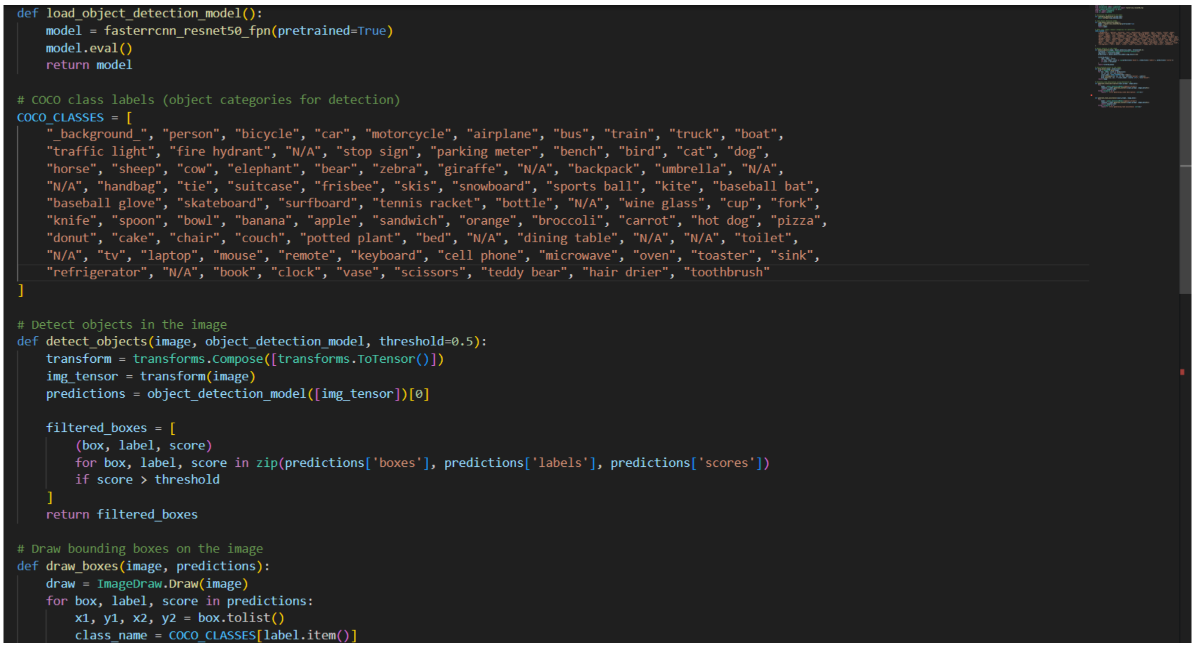
```
| LOVE PLAYING UNDER
PRESSURE. IN FACT, IF THERE'S
NO PRESSURE, THEN I'M NOT IN
THE PERFECT ZONE.

-VIRAT KOHLI
```

Audio Text (English):

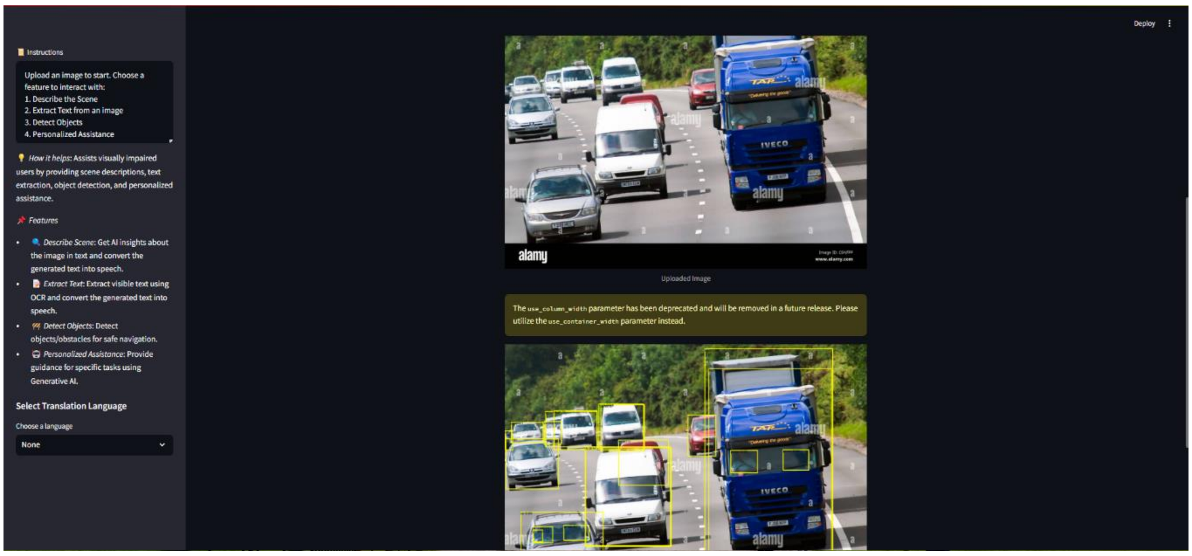
Object Detection

Assists users in identifying obstacles or items in their environment.

PyTorch (Faster R-CNN): For object detection.



```python
def load_object_detection_model():
    model = fasterrcnn_resnet50_fpn(pretrained=True)
    model.eval()
    return model

# COCO class labels (object categories for detection)
COCO_CLASSES = [
    "_background_", "person", "bicycle", "car", "motorcycle", "airplane", "bus", "train", "truck", "boat",
    "traffic light", "fire hydrant", "N/A", "stop sign", "parking meter", "bench", "bird", "cat", "dog",
    "horse", "sheep", "cow", "elephant", "bear", "zebra", "giraffe", "N/A", "backpack", "umbrella", "N/A",
    "N/A", "handbag", "tie", "suitcase", "frisbee", "skis", "snowboard", "sports ball", "kite", "baseball bat",
    "baseball glove", "skateboard", "surfboard", "tennis racket", "bottle", "N/A", "wine glass", "cup", "fork",
    "knife", "spoon", "bowl", "banana", "apple", "sandwich", "orange", "broccoli", "carrot", "hot dog", "pizza",
    "donut", "cake", "chair", "couch", "potted plant", "bed", "N/A", "dining table", "N/A", "N/A", "toilet",
    "N/A", "tv", "laptop", "mouse", "remote", "keyboard", "cell phone", "microwave", "oven", "toaster", "sink",
    "refrigerator", "N/A", "book", "clock", "vase", "scissors", "teddy bear", "hair drier", "toothbrush"
]

# Detect objects in the image
def detect_objects(image, object_detection_model, threshold=0.5):
    transform = transforms.Compose([transforms.ToTensor()])
    img_tensor = transform(image)
    predictions = object_detection_model([img_tensor])[0]

    filtered_boxes = [
        (box, label, score)
        for box, label, score in zip(predictions['boxes'], predictions['labels'], predictions['scores'])
        if score > threshold
    ]
    return filtered_boxes

# Draw bounding boxes on the image
def draw_boxes(image, predictions):
    draw = ImageDraw.Draw(image)
    for box, label, score in predictions:
        x1, y1, x2, y2 = box.tolist()
        class_name = COCO_CLASSES[label.item()]
```



Instructions

Upload an image to start. Choose a feature to interact with:
1. Describe the Scene
2. Extract Text from an image
3. Detect Objects
4. Personalized Assistance

💡 *How it helps*: Assists visually impaired users by providing scene descriptions, text extraction, object detection, and personalized assistance.

📌 *Features*

- 🔍 *Describe Scene*: Get AI insights about the image in text and convert the generated text into speech.
- 📄 *Extract Text*: Extract visible text using OCR and convert the generated text into speech.
- 🚧 *Detect Objects*: Detect objects/obstacles for safe navigation.
- 🤖 *Personalized Assistance*: Provide guidance for specific tasks using Generative AI.

Select Translation Language

Choose a language: None

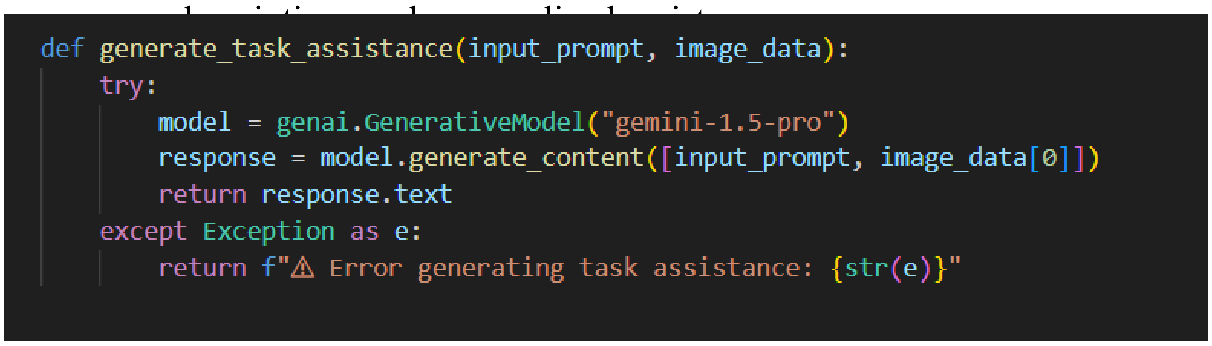
Uploaded Image

The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.

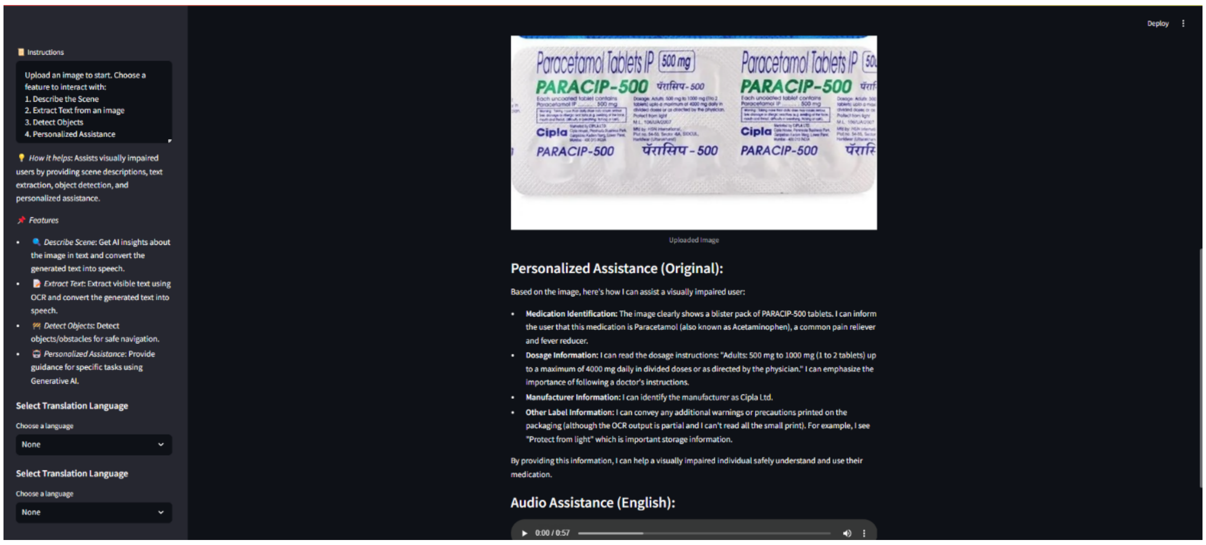
Personalized Assistance :

Offers detailed assistance, such as reading labels or recognizing items.

Google Generative AI (Gemini API): For generating scene



```python
def generate_task_assistance(input_prompt, image_data):
    try:
        model = genai.GenerativeModel("gemini-1.5-pro")
        response = model.generate_content([input_prompt, image_data[0]])
        return response.text
    except Exception as e:
        return f"⚠ Error generating task assistance: {str(e)}"
```



Instructions

Upload an image to start. Choose a feature to interact with:
1. Describe the Scene
2. Extract Text from an image
3. Detect Objects
4. Personalized Assistance

💡 *How it helps*: Assists visually impaired users by providing scene descriptions, text extraction, object detection, and personalized assistance.

📌 *Features*

- 🔍 *Describe Scene*: Get AI insights about the image in text and convert the generated text into speech.
- 📝 *Extract Text*: Extract visible text using OCR and convert the generated text into speech.
- 🚧 *Detect Objects*: Detect objects/obstacles for safe navigation.
- 🤖 *Personalized Assistance*: Provide guidance for specific tasks using Generative AI.

Select Translation Language

Choose a language: None

Select Translation Language

Choose a language: None

Personalized Assistance (Original):

Based on the image, here's how I can assist a visually impaired user:

- **Medication Identification:** The image clearly shows a blister pack of PARACIP-500 tablets. I can inform the user that this medication is Paracetamol (also known as Acetaminophen), a common pain reliever and fever reducer.
- **Dosage Information:** I can read the dosage instructions: "Adults: 500 mg to 1000 mg (1 to 2 tablets) up to a maximum of 4000 mg daily in divided doses or as directed by the physician." I can emphasize the importance of following a doctor's instructions.
- **Manufacturer Information:** I can identify the manufacturer as Cipla Ltd.
- **Other Label Information:** I can convey any additional warnings or precautions printed on the packaging (although the OCR output is partial and I can't read all the small print). For example, I see "Protect from light" which is important storage information.

By providing this information, I can help a visually impaired individual safely understand and use their medication.

Audio Assistance (English):

0:00 / 0:57

Conclusion :

- The project provides a versatile tool to improve the independence and quality of life for visually impaired individuals.
- It combines state-of-the-art AI technologies with practical features like translation and audio output to enhance accessibility.