

# ENVISION: AI VISUAL ASSISTANT

## 1. PROJECT OVERVIEW

Over 2.2 billion people worldwide live with some form of vision impairment or blindness, according to the World Health Organization (2023). Many individuals face significant barriers in navigating the world around them, reading text, or identifying objects and hazards. These challenges can hinder their independence and access to critical information, creating a gap between them and the opportunities available to sighted individuals.

**Envision** is an AI-powered solution designed to bridge this gap. By combining cutting-edge technologies like Langchain, Generative AI and Streamlit, *Envision* provides real-time assistance to visually impaired individuals. The application interprets images, and offers navigation assistance, all through detailed, voice-assisted descriptions. Users can upload images and receive verbal or written descriptions of their surroundings, making it easier to navigate spaces and perform everyday tasks like reading labels or identifying obstacles.

### 1.1 Objectives

The primary goal of *Envision* is to empower visually impaired individuals by providing them with tools to engage more independently with their environment. Key objectives include:

- **Scene Understanding:** Offering detailed image descriptions to help users understand their surroundings.
- **Text-to-Speech Conversion:** Enabling access to printed or digital text via voice output.
- **Task Assistance:** Providing personalized help based on images, including recognizing objects and offering safety guidance for daily tasks.

## 2. KEY FEATURES

- **Real-Time Scene Understanding:** Generate detailed textual descriptions of images to help users understand the scene effectively.
- **Text-to-Speech Conversion:** Convert generated description into audible speech for seamless content accessibility.

- **Personalized Assistance:** Provide guidance for specific tasks such as recognizing items, reading labels, or identifying objects in the environment.

### 3. TECHNOLOGY STACK

- **Streamlit:** Used to develop the interactive user interface for the application.
- **Langchain:** Integrates conversational AI features to enable interactive communication.
- **Google Generative AI (Gemini API):** Powers the generation of scene descriptions and offers personalized support.
- **gtts:** Implements text-to-speech functionality, converting text into audible speech.
- **PIL:** Handles image processing tasks, ensuring proper image manipulation and preparation.
- **Programming Language:** Python

### 4. CODE IMPLEMENTATION

#### 4.1 Directory Structure

The directory structure of the AI\_Envision project is organized as follows:

- `app.py`: Main entry point for the application.
- `views/`: Contains individual pages and features of the app.
  - `features/`: Includes the pages for specific features like describing images, assisting with tasks, and reading text.
  - `home.py`: The landing page of the application.
- `helpers.py`: Contains helper functions for tasks like image processing and data handling.
- `assets/`: Stores static files, such as the project logo.
- `.streamlit/`: Configuration files for Streamlit, including `.config.toml` for settings and `secrets.toml` for securely managing API keys.

```
AI_Envision/  
├─ app.py  
├─ helpers.py  
├─ assets/  
|   └─ logo.png  
├─ views/  
|   └─ features/  
|       └─ describe.py  
|       └─ task.py  
|       └─ read.py  
|   └─ home.py  
├─ .streamlit/  
|   └─ config.toml  
|   └─ secrets.toml
```

## 4.2 Describe Image (Feature)

Generates detailed textual descriptions of images to help users understand the scene effectively.

### Prompt:

You are an AI assistant specifically designed for visually impaired individuals. Your task is as follows:

- Analyse the uploaded image and provide clear, simple language and detailed description of it in paragraph form.
- The description should include information on Scene Overview, Key Objects, Human Activities, and Setting (Colours and Lighting).

```
# Function to "DESCRIBE THE IMAGE"
def describe_image(uploaded_image):

    # AI Prompts
    system_prompt = (
        "system",
        """
        You are an AI assistant specifically designed for visually impaired individuals. Your task is as follows:
        Analyze the uploaded image and provide clear, simple language and detailed description of it in paragraph form.
        The description should include information on Scene Overview, Key Objects, Human Activities, and Setting (Colors and Lighting).
        """)
    )
    human_prompt = (
        "human",
        [
            {"type": "text", "text": "Please provide a detailed description of the uploaded image."},
            [{"type": "image_url", "image_url": f"data:image/png;base64,{uploaded_image}"}]
        ]
    )

    # Create LangChain Pipeline
    chat_prompt = ChatPromptTemplate.from_messages([system_prompt, human_prompt])
    output_parser = StrOutputParser()
    chat_model = ChatGoogleGenerativeAI(google_api_key=GEMINI_API_KEY, model=MODEL_NAME)
    pipeline = chat_prompt | chat_model | output_parser

    try:
        description = pipeline.invoke({"input": "Analyze the scene.", "image_data": uploaded_image})
        if description:
            st.session_state["generated_text"] = description # Save to session state
            return description
    except Exception as e:
        st.error(f"Error analyzing image: {str(e)}")
        return None
```

## Output:

Dog.jpg 68.24KB



Uploaded Image

Analyze Image

### Description

The image depicts a heartwarming scene of a young woman embracing a golden retriever dog. The overall scene is one of affection and tranquility.

The key objects are the woman and the dog. The dog, a golden retriever with long, fluffy fur, appears calm and content. The woman has long, light brown hair, some strands of which are gently blowing in a slight breeze. She's wearing a dark reddish-brown, textured sweater, possibly knit or ribbed.

The main human activity is the embrace. The woman has her eyes closed and is leaning in, gently pressing her cheek against the dog's head. Her hands are placed on either side of the dog's face, further emphasizing the affectionate connection. The dog appears receptive to the affection.

The setting appears to be outdoors, possibly a park or a wooded area. The background is blurred, suggesting a shallow depth of field, with warm, out-of-focus colors hinting at autumn foliage. The lighting is soft and golden, likely late afternoon or early evening sunlight, which creates a warm and intimate atmosphere. The light catches highlights in the woman's hair and the dog's fur, adding to the sense of peace and warmth.

## 4.3 Task Assistance (Feature)

Provide guidance for specific tasks such as recognizing items, reading labels, or identifying objects in the environment.

## Prompt:

You are an AI assistant specifically designed to help visually impaired individuals by analyzing uploaded images and providing them with actionable guidance. Ensure responses are detailed, easy to understand, and tailored to the user's needs; and should include:

1. Item Identification (Names, Types, Key Features; Relative Positions)
2. Label Reading and Text Recognition (Warnings, Instructions, or Important Markings)
3. Navigation Safety (Guide through space, Highlight key landmarks/obstacles/hazards)
4. Guidance for Daily Tasks (Step-by-Step Instructions, Suggest Safety Precautions, Highlight Important Object Location)

```
# Function to "OFFER PERSONALISED TASK ASSISTANCE"
def task_assistance(uploaded_image):

    # AI Prompts
    system_prompt = (
        "system",
        """
        You are an AI assistant specifically designed to help visually impaired individuals by analyzing uploaded images and providing them with
        1. Item Identification (Names, Types, Key Features; Relative Positions)
        2. Label Reading and Text Recognition (Warnings, Instructions, or Important Markings)
        3. Navigation Safety (Guide through space, Highlight key landmarks/obstacles/hazards)
        4. Guidance for Daily Tasks (Step-by-Step Instructions, Suggest Safety Precautions, Highlight Important Object Location)
        """
    )

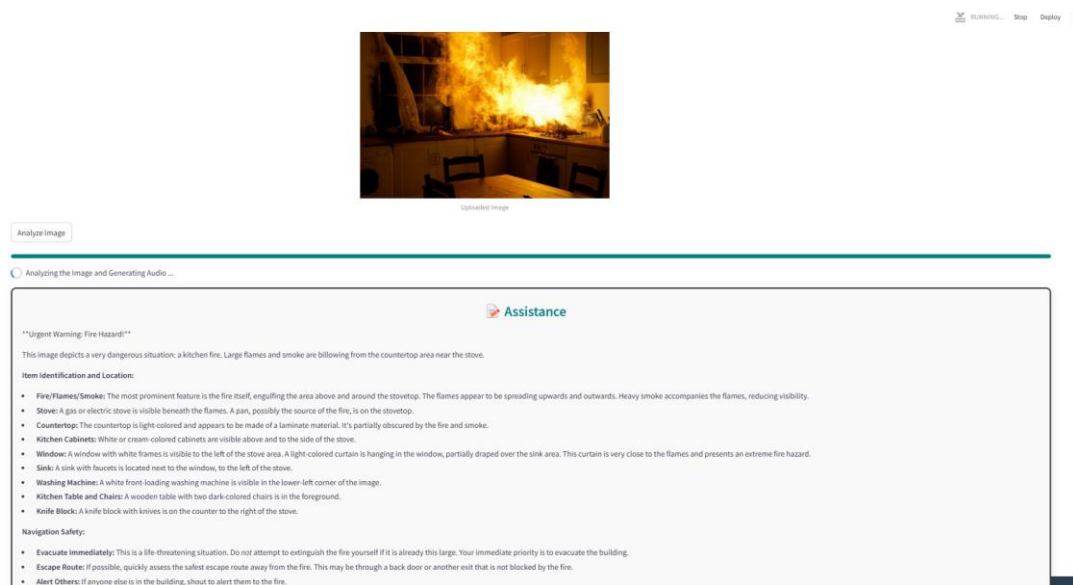
    human_prompt = (
        "human",
        [
            {"type": "text", "text": "Please provide a detailed description of the uploaded image."},
            {"type": "image_url", "image_url": f"data:image/png;base64,{uploaded_image}"}
        ]
    )

    # Create LangChain Pipeline
    chat_prompt = ChatPromptTemplate.from_messages([system_prompt, human_prompt])
    output_parser = StrOutputParser()
    chat_model = ChatGoogleGenerativeAI(google_api_key=GEMINI_API_KEY, model=MODEL_NAME)
    pipeline = chat_prompt | chat_model | output_parser
```

```
# Create LangChain Pipeline
chat_prompt = ChatPromptTemplate.from_messages([system_prompt, human_prompt])
output_parser = StrOutputParser()
chat_model = ChatGoogleGenerativeAI(google_api_key=GEMINI_API_KEY, model=MODEL_NAME)
pipeline = chat_prompt | chat_model | output_parser

try:
    description = pipeline.invoke({"input": "Analyze the scene.", "image_data": uploaded_image})
    if description:
        st.session_state["generated_text"] = description # Save to session state
        return description
except Exception as e:
    st.error(f"Error analyzing image: {str(e)}")
    return None
```

## Output:



The screenshot displays a web interface for image analysis. At the top, there's a button labeled "Analyze Image". Below it, a progress indicator shows "Analyzing the Image and Generating Audio...". The main content area, titled "Assistance", contains a warning: "\*\*Urgent Warning: Fire Hazard!\*\*". It follows with a summary: "This image depicts a very dangerous situation: a kitchen fire. Large flames and smoke are billowing from the countertop area near the stove." Below this is a section "Item Identification and Location:" with a bulleted list of items and their locations. At the bottom, a "Navigation Safety:" section provides instructions on what to do in case of a fire.

Assistance

**\*\*Urgent Warning: Fire Hazard!\*\***

This image depicts a very dangerous situation: a kitchen fire. Large flames and smoke are billowing from the countertop area near the stove.

**Item Identification and Location:**

- **Fire/Flames/Smoke:** The most prominent feature is the fire itself, engulfing the area above and around the stovetop. The flames appear to be spreading upwards and outwards. Heavy smoke accompanies the flames, reducing visibility.
- **Stove:** A gas or electric stove is visible beneath the flames. A pan, possibly the source of the fire, is on the stovetop.
- **Countertop:** The countertop is light-colored and appears to be made of a laminate material. It's partially obscured by the fire and smoke.
- **Kitchen Cabinets:** White or cream-colored cabinets are visible above and to the side of the stove.
- **Window:** A window with white frames is visible to the left of the stove area. A light-colored curtain is hanging in the window, partially draped over the sink area. This curtain is very close to the flames and presents an extreme fire hazard.
- **Sink:** A sink with faucets is located next to the window, to the left of the stove.
- **Washing Machine:** A white front-loading washing machine is visible in the lower-left corner of the image.
- **Kitchen Table and Chairs:** A wooden table with two dark-colored chairs is in the foreground.
- **Knife Block:** A knife block with knives is on the counter to the right of the stove.

**Navigation Safety:**

- **Evacuate Immediately:** This is a life-threatening situation. Do not attempt to extinguish the fire yourself if it is already this large. Your immediate priority is to evacuate the building.
- **Escape Route:** If possible, quickly assess the safest escape route away from the fire. This may be through a back door or another exit that is not blocked by the fire.
- **Alert Others:** If anyone else is in the building, shout to alert them to the fire.

## 4.4 Convert Text to Speech (Feature)

Convert generated description into audible speech for seamless content accessibility.

```
# Function to "CONVERT TEXT TO SPEECH"
def read_text():
    try:
        # Check if Text is Available in Session State
        if "generated_text" not in st.session_state or not st.session_state["generated_text"].strip():
            st.warning("No text available to read. Please analyze an image first.")
            return

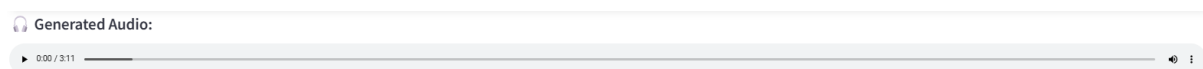
        text_to_read = st.session_state["generated_text"]
        text_to_read = clean_text(text_to_read)

        # Convert Text to Speech using gTTS
        tts = gTTS(text=text_to_read, lang='en-uk')
        audio_output = io.BytesIO()
        tts.write_to_fp(audio_output) # Write audio data to buffer
        audio_output.seek(0)

        # Display the Generated Audio
        st.markdown("### 🎧 Generated Audio:")
        st.audio(audio_output, format="audio/mp3")

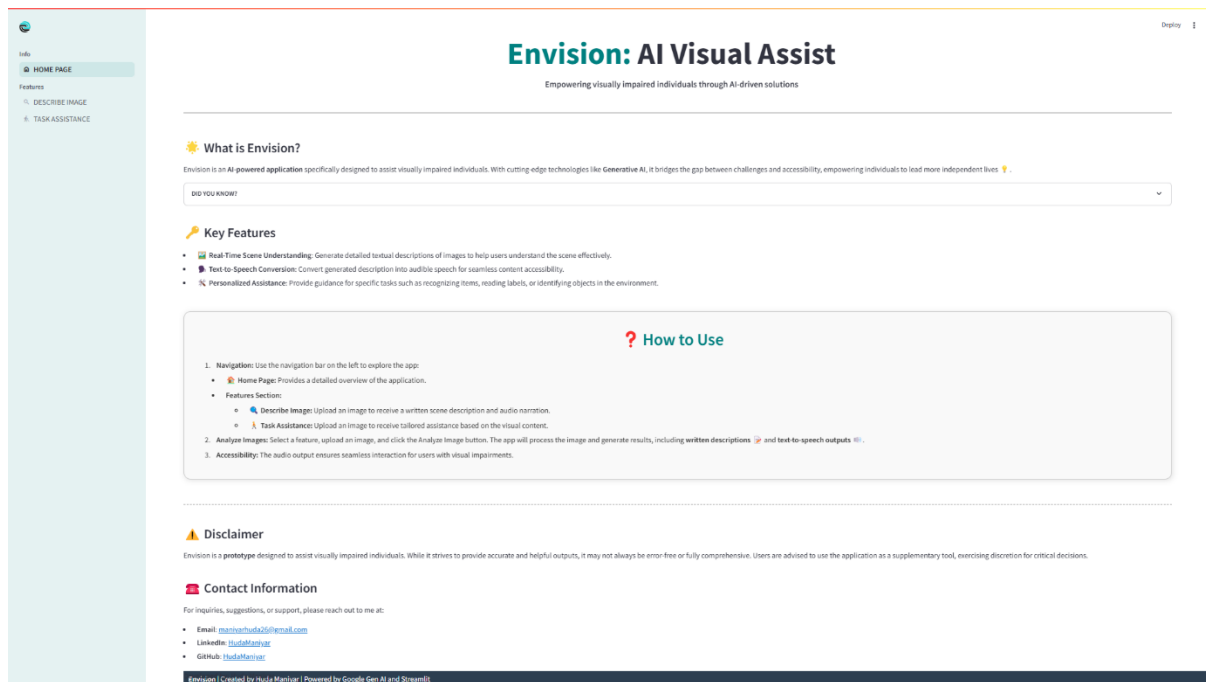
    except Exception as e:
        st.error(f"Error generating audio: {str(e)}")
```

## Output:



The screenshot shows an audio player interface. At the top, it says "Generated Audio:". Below this is a progress bar with a play button on the left, a time indicator "0:00 / 3:11", and a volume icon on the right.

## 4.5 Home Page (UI)



## 5. USER MANUAL – HOW TO USE

The application allows users to upload images and choose from two main features:

1. **Describe Image:** Generates a detailed scene description and audio narration.
2. **Task Assistance:** Provides guidance for specific tasks based on the visual content.

### 5.1 Steps to Use

1. **Navigate:** Use the sidebar to choose between the Home or Features sections.
2. **Upload Image:** Choose an image to upload for processing.
3. **Click "Analyse Image":** The system will process the image and provide both written and spoken descriptions.

## 6. FUTURE ENHANCEMENT

- **Support for Multiple Languages:** Introduce multilingual text-to-speech capabilities, making the app more accessible to users worldwide.

- **Wearable Device Integration:** Enable users to interact with the app via wearable devices or smartphones, providing real-time image analysis and feedback through camera features.
- **Voice-Controlled Interaction:** Develop a fully voice-operated interface for hands-free navigation, catering to the needs of visually impaired users.
- **Customizable UI:** Allow users to adjust themes and font sizes to personalize their experience, optimizing the interface for better accessibility, especially for visually impaired individuals.

## 7. CONCLUSION

Envision is an AI-powered tool designed to assist visually impaired individuals by providing real-time scene descriptions, text-to-speech, and task guidance. It enhances accessibility and independence. Future updates, including multilingual support and voice-controlled navigation, will further improve its functionality, making it a more inclusive solution for users worldwide.