Assistive Image Processing Tool

The **Assistive Image Processing Tool** is a Streamlit-based application designed to aid visually impaired individuals in understanding and interacting with images. This tool leverages Google Generative AI models to describe images in detail, extract text from images, and convert the extracted text to speech.

Key Features

1. **Image Description:** Provides a comprehensive, contextual, sensory, and actionable description of the uploaded image.
2. **Text Extraction:** Extracts visible text from the image.
3. **Text-to-Speech Conversion:** Converts the extracted text into an audio file.

Technology Stack

- **Streamlit:** For creating the web application interface.
- **Google Generative AI:** For image understanding and text extraction.
- **Pillow (PIL):** For handling image files.
- **NumPy:** For processing image data.
- **gTTS (Google Text-to-Speech):** For converting extracted text into speech.
- **JSON:** For handling API responses and data parsing.

Prerequisites

1. Install the required Python libraries:

pip install streamlit pillow numpy google-generativeai gtts

2. Obtain an API key for Google Generative AI and save it in a file named key2.txt in the project directory.

How It Works

1. Configuring Generative AI API

The API key is read from the key2.txt file and used to configure the Google Generative AI models.

2. Defining Generative Models

- **Model 1:** Provides detailed descriptions of images.
- **Model 2:** Extracts text from images.

3. Functions

- **describe_image(img):**
    - Takes an image as input.
    - Uses Model 1 to generate a detailed description.
    - Returns a structured JSON response with a summary and detailed description.

- **img_to_txt(img):**
    - Takes an image as input.
    - Uses Model 2 to extract visible text from the image.
    - Returns the text in JSON format.
- **img_to_speech(img):**
    - Extracts text from the image using img_to_txt.
    - Converts the text into speech using gTTS and saves it as an audio file.
    - Returns the audio file path.

4. Streamlit Application

- **Upload Image:** Users can upload an image in PNG, JPG, or JPEG formats.
- **Select Action:** Users choose one of the following actions:
    - *Describe the image*
    - *Convert to Speech*
- **Display Results:**
    - Displays the generated description or plays the audio file.

Application Workflow

1. **Start the Application:**

streamlit run app.py

2. **Upload an Image:** Select an image file to process.
3. **Choose an Action:**
    - *Describe the image*: The tool generates and displays a structured description of the image.
    - *Convert to Speech*: The tool extracts text from the image, converts it to speech, and provides an audio file for playback.
4. **Output:** The selected action's result (description or audio) is displayed in the application.

Example Use Case

Input:

- An image of a park.

Output:

- **Image Description:**
    - *Summary*: The image shows a bustling park on a sunny afternoon, filled with people enjoying outdoor activities.

- *Detailed Description*: The park is vast and green, with well-maintained grass and tall trees providing shade. Children are playing on a colorful playground set that includes swings, slides, and monkey bars. Nearby, families are seated on blankets, having picnics with baskets of food and drinks. Joggers and cyclists are using a paved path that winds through the park. A small pond is visible in the background, with ducks swimming and a couple sitting on a bench, feeding them. The sky is clear blue, with a few fluffy clouds, and the sunlight creates a warm and cheerful atmosphere.

Output for Speech:

- A playable audio file describing the text extracted from the image.

Error Handling

- If no text is extracted from the image for speech conversion, an error message is displayed: "No text was extracted to convert to speech."

Limitations

- Requires an active internet connection to use Google Generative AI.
- Limited by the quality and resolution of the uploaded images.
- Text-to-speech is currently supported only in English.

Future Enhancements

- Support for multiple languages in text-to-speech conversion.
- Improved error handling and user feedback.
- Integration with OCR for enhanced text extraction capabilities.

Conclusion

The Assistive Image Processing Tool provides significant support to visually impaired individuals by leveraging cutting-edge AI technologies to interpret and interact with visual content effectively.