Indic Image Generation: Speech-to-Text and Image Series Generation

Team: MLTrio

Nishant Verma (ID: 12241170)
Rajeev Goel (ID: 12241460)
Soni Kumari (ID: 12241780)

2024

1 Introduction

This project focuses on building a system for **Indic Image Generation**, which converts spoken input into text and then generates a series of evolving images based on the translated text. The system integrates Automatic Speech Recognition (ASR), Machine Translation, and Text-to-Image Generation, allowing users to create a sequence of images progressing through stages of refinement. This enables interactive storytelling and creative content generation based on voice prompts.

The project also uses the **OpenJourney model**, a variation of Stable Diffusion, to produce high-quality images from text inputs, progressively refining them based on additional user inputs.

2 Speech-to-Text and Image Series Generation

The core components of our project are:

1. **Speech-to-Text Conversion:** Using Google's Speech Recognition API, the system transcribes spoken Hindi inputs into text.

2. **Text Translation:** The transcribed Hindi text is translated into English using Google Translate API.

3. **Image Generation in Series:** The OpenJourney model generates images from text inputs, progressively refining them based on additional user prompts.

3 Model and APIs Used

3.1 Google Speech Recognition API

The ASR functionality is implemented using Google's Speech Recognition API. This API converts the user's spoken input into text, supporting Hindi and other Indic languages.

3.2 Google Translate API

The translated text is generated using Google Translate, allowing Hindi (or other Indic languages) to be transformed into English for use by the image generation model.

3.3 OpenJourney Model

The image generation pipeline is powered by the **OpenJourney model**, which is based on Stable Diffusion. This model creates and refines images based on text prompts, providing a more artistic rendering of the user's inputs. The model is accessed using the `diffusers` library, allowing for a highly customizable image creation process.

3.4 ffmpeg for Audio Processing

Audio inputs are recorded and processed using `ffmpeg`, which converts raw audio into a format usable by the speech recognition module.

4 Use Cases

4.1 Visual Storytelling

A storyteller narrates a traditional Indian folk tale in Hindi, and the system generates evolving images reflecting each stage of the story, such as changes in the landscape or new characters being introduced.

4.2 Education

Teachers describe scientific processes in Hindi, such as the stages of the water cycle, and the system generates corresponding images that visually illustrate each concept.

4.3 Cultural Preservation

Historians describe ancient cultural events or sites in Hindi, and the system generates images that preserve and document the historical significance of these sites through a visual medium.

5 Example Outputs

5.1 Speech Input:

"Ek pahad ke upar suryast ho raha hai." (The sun is setting over the mountain.)

- **Transcribed Text:** "Ek pahad ke upar suryast ho raha hai."
- **Translated Text:** "The sun is setting over the mountain."
- **Generated Image:** The system creates an image showing a sunset over a mountain.

5.2 Refinement:

"Ab pahad ke paas ek jheel hai." (Now there is a lake near the mountain.)

- **Transcribed Text:** "Ab pahad ke paas ek jheel hai."
- **Translated Text:** "Now there is a lake near the mountain."
- **Refined Image:** The system refines the existing image by adding a lake near the mountain.

This iterative process continues as users provide new inputs, refining and evolving the images.

5.3 Prompts and Images as Output

Example 1:

Prompt: A moment in the golden light of the sunset.

Generated Images:

The image is generated in three parts:

- A moment in
- A moment in the golden light
- A moment in the golden light of sunset.

Example 2:

Prompt: A plane is flying over the mountain.

Generated Images:

The image is generated in three parts:

- A plane is
- A plane is flying over the

- A plane is flying over the mountain.

Example 3: Image is generated in a loop

Prompt 1: sunset .

Generated Images:

Prompt 2: a man is standing.

Generated Images:

The image is generated in two parts:

- a man is
- a man is standing

Prompt 3: bombs falling from above.

Generated Images:

The image is generated in two parts:

- bombs falling from
- bombs falling from above

6 Contribution of Each Team Member

Our team divided the project work into three primary tasks: Speech-to-Text transcription, Transcription to Translation, and Image Generation. The integration of all components was a collaborative effort.

6.1 Speech-to-Text Transcription (Nishant Verma)

Nishant implemented the **Speech-to-Text** module using Google's Speech Recognition API. This module converts Hindi speech into text, enabling the system to handle voice inputs from users.

- Nishant used the `SpeechRecognition` library to interface with Google's API.
- The system processes speech inputs in Hindi, allowing users to provide instructions in their native language.
- Example of input: *Ek pahad ke upar suryast ho raha hai.* (The sun is setting over the mountain.)

6.2 Transcription to Translation (Soni Kumari)

Soni was responsible for translating the transcribed Hindi text into English using Google Translate API. Since the OpenJourney model is optimized for English inputs, this step ensures compatibility with the image generation model.

- Soni implemented the translation using the `googletrans` library.
- The translation module ensures that the Hindi text transcribed from the speech recognition step is accurately translated into English for the next stage.
- Example translation: *Ek pahad ke upar suryast ho raha hai* is translated to *The sun is setting over the mountain.*

6.3 Image Generation (Rajeev Goel)

Rajeev handled the **Image Generation** module using the **OpenJourney model**, a variation of Stable Diffusion designed for artistic image generation. This model creates a series of images based on the translated text and allows for progressive refinement as additional inputs are provided.

- Rajeev integrated the `StableDiffusionPipeline` from the `diffusers` library, with the OpenJourney model as the base.
- Each image is generated from a textual description and is refined with each new input.
- Example: Based on the translated text "The sun is setting over the mountain", the system generates an image of a mountain with a sunset in the background. If a new input is provided, such as *Now there is a lake near the mountain*, the system refines the image by adding a lake while maintaining visual continuity.

6.4 Collaboration on Integration

All members collaborated to integrate the modules. The Speech-to-Text, Translation, and Image Generation modules were combined to create a seamless pipeline where user inputs evolve into a series of related images.

- Each member worked on debugging and optimizing the workflow for smooth transitions between the stages.
- The collaborative effort ensured the final output was coherent and responsive to sequential user inputs.

7 Future Work

While the current system focuses on Hindi-to-English transcription and translation for generating images, there are several key areas where this project can be expanded to improve its versatility and user experience. The primary future enhancements include:

7.1 Multilingual Transcription and Translation

One of the most promising extensions of this project is to support multiple languages for transcription and translation. The current system is optimized

for Hindi inputs, but the model can be extended to include other Indic languages as well as global languages like Spanish, French, and Mandarin. This will involve:

- **Multilingual Speech-to-Text:** Expanding the Speech Recognition API to handle inputs in multiple languages, allowing users to provide prompts in any supported language.
- **Multilingual Translation:** Enhancing the translation module to detect and translate text from a wider range of languages. This can be achieved by integrating language detection alongside the translation process, ensuring seamless switching between languages.
- **Contextual Translation Improvements:** Incorporating advanced translation APIs or models that better handle the nuances of each language, providing more accurate prompts for the image generation model.

This will allow the system to cater to a broader audience and support diverse use cases in education, storytelling, and creative industries.

7.2 User Interface (UI) Development

To improve accessibility and usability, the development of a dedicated User Interface (UI) is crucial. A web-based or desktop UI will allow users to interact with the system more intuitively. Some of the key features of the proposed UI would include:

- **Voice Input Integration:** A built-in voice input feature that allows users to record their speech directly through the interface, eliminating the need for separate audio recording tools.
- **Multilingual Support in UI:** Dropdown options for selecting input and output languages, simplifying the process for users who want to generate images from various languages.
- **Real-time Image Generation:** Displaying the generated images and their refinements in real-time as users provide new prompts, giving immediate feedback and enhancing interactivity.

Developing a UI will significantly enhance the user experience by making the system more accessible and interactive, enabling even non-technical users to benefit from the technology.

8 Conclusion

The Indic Image Generation system integrates speech recognition, translation, and text-to-image generation to create evolving images from spoken inputs. Using the OpenJourney model, the system offers an interactive and dynamic approach to storytelling, education, and cultural preservation by generating image series based on sequential voice commands. The progressive refinement ensures that images evolve smoothly, making this project an innovative tool for interactive content creation.

9 GitHub Repository Links

- **GitHub Repository Link:** `https://github.com/RajeevG187/Speech2Image`
- **GitHub Cloning Link:** `https://github.com/RajeevG187/Speech2Image.git`

Repository Details:
The repository consists of four branches: main ,rajeev, nishant ,soni