master Go to file Code

ayushpai Update requirements.txt ···		yesterday 15
.idea	Initial commit	3 days ago
frames	Initial commit	3 days ago
LICENSE	Create LICENSE	2 days ago
README.md	Update README.md	yesterday
capture.py	Initial commit	3 days ago
main.py	Update main.py	3 days ago
main_single…	Update and rename singlestor…	yesterday
requirement…	Update requirements.txt	yesterday

README.md

Conversational AI with GPT-4 Vision, OpenAI Whisper, and TTS

Overview

This project integrates GPT-4 Vision, OpenAI Whisper, and OpenAI Text-to-Speech (TTS) to create an interactive AI system for conversations. It combines visual and audio inputs for a seamless user experience.

Demo Video:

https://twitter.com/ayushspai/status/1726222559480557647

About

Conversational AI with GPT-4 Vision, OpenAI Whisper, and TTS

Readme
MIT license
Activity
80 stars
4 watching
11 forks
Report repository

Releases

No releases published

Packages

No packages published

Languages

Python 100.0%

Components

- **GPT-4 Vision**: Analyzes visual input and generates contextual responses.
- **OpenAI Whisper**: Converts spoken language into text.
- **OpenAI TTS**: Transforms text responses into spoken language.

Main Files

- `main.py` : Manages audio processing, image encoding, AI interactions, and text-to-speech output.
- `capture.py` : Captures and processes video frames for visual analysis.

Installation

Prerequisites

- Python 3.x
- An OpenAI API key (set as an environment variable `OPENAI_API_KEY`)

Libraries

Install the necessary libraries with the requirements.txt file.

```
pip install -r requirements.txt
```

Usage

Running the Scripts

- **Start `capture.py`** : Captures video frames and saves them for AI analysis.
    - Reads a video file, displays the video, and saves the current frame as `frame.jpg` .
    - Execute with `python capture.py` .

- **Run `main.py` concurrently**: Orchestrates the conversational AI.
    - Continuously listens for user audio input.
    - Transcribes speech to text, captures the current video frame, and sends both to GPT-4 for analysis.
    - Converts the AI's response to speech and plays it back.
    - Execute with `python main.py`.

Workflow

1. `main.py` listens for audio input and transcribes it using OpenAI Whisper.
2. Meanwhile, `capture.py` captures a video frame.
3. Both the audio transcription and the encoded image are sent to GPT-4 Vision.
4. GPT-4 Vision responds, considering the visual and textual context.
5. The response is vocalized using OpenAI TTS and played to the user.

Notes

- Ensure both `main.py` and `capture.py` are active for the system to function.
- The video file in `capture.py` can be customized.
- Adequate hardware is recommended for smooth audio and video processing.

Conclusion

This project demonstrates a novel approach to combining various AI technologies, creating a dynamic and interactive conversational AI experience. It harnesses the capabilities of GPT-4 Vision, Whisper, and TTS for a comprehensive audio-visual interaction.