Scenario Description

Scenario Description

- Experiment Objective
- Experiment Steps
- Experimental Results
- Main Source Code Analysis
- Overall Flowchart

Experiment Objective

Understand and master a basic function of a large AI model, interact with a smart car through language dialogue, and enable the smart car to describe the current real-time scene.

Experiment Steps

1. Observe the IP address of the OLED screen and log in to the remote desktop via VNC.
2. Follow the prerequisite configuration tutorial to complete the Tongyi Qianwen key for the Chinese version and the OpenRouter key for the English version.
3. Open a new terminal and run the command:

```
cd /home/pi/project_demo/09.AI_Big_Model/
```

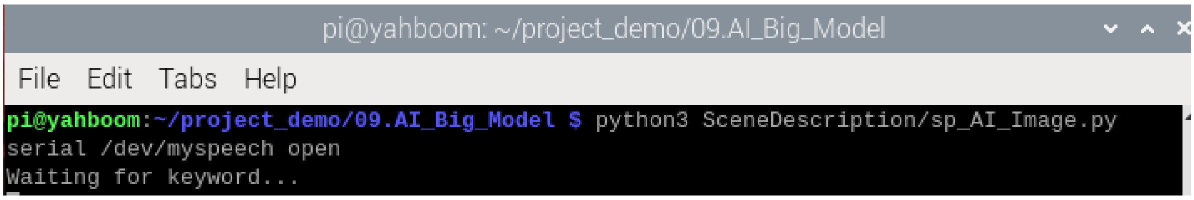
```
#Startup command for the Chinese version
python3 SceneDescription/sp_AI_Image.py
```

```
#Startup command for the English version
python3 SceneDescription/sp_AI_Image_en.py
```

4. Wake up the car using the wake-up phrase "Hi, Yahboom" (for international users).
5. After successfully waking up, the car will respond with a honking sound and wait for about half a second. You can then ask the car a question related to the scene.
6. After the robot recognizes your voice, wait a few seconds for relevant information to be displayed on the terminal interface and speaker. You can also terminate the current conversation by using the wake-up word.
7. This concludes the conversation. To start a new one, repeat steps 4-6.

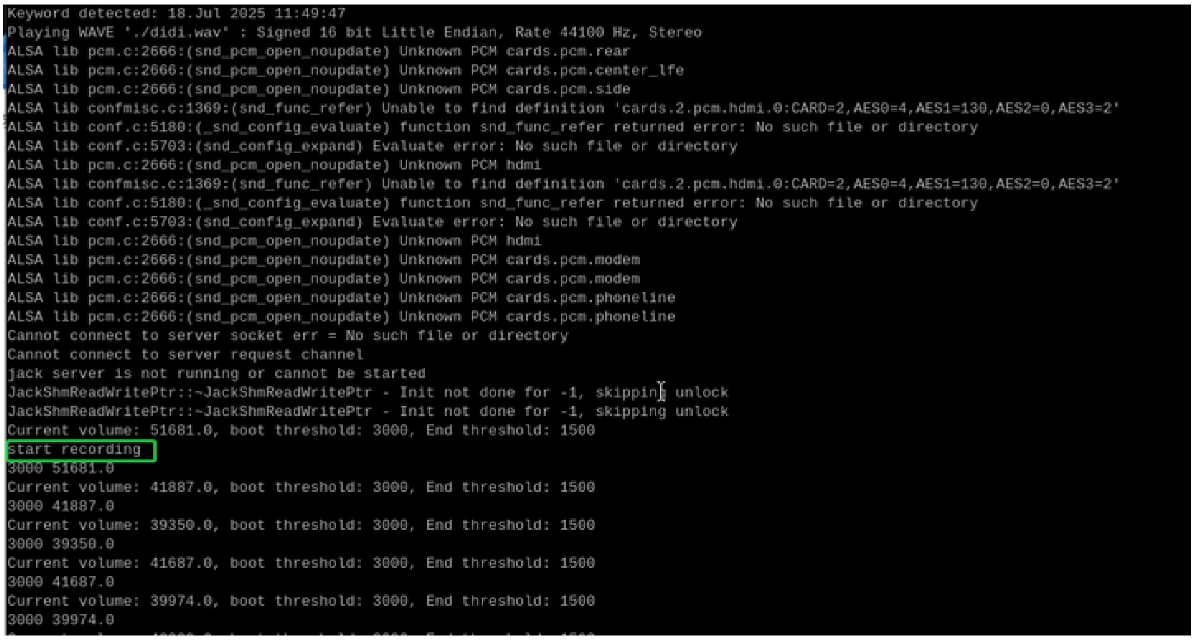
Experimental Results

1. Waiting for wake-up



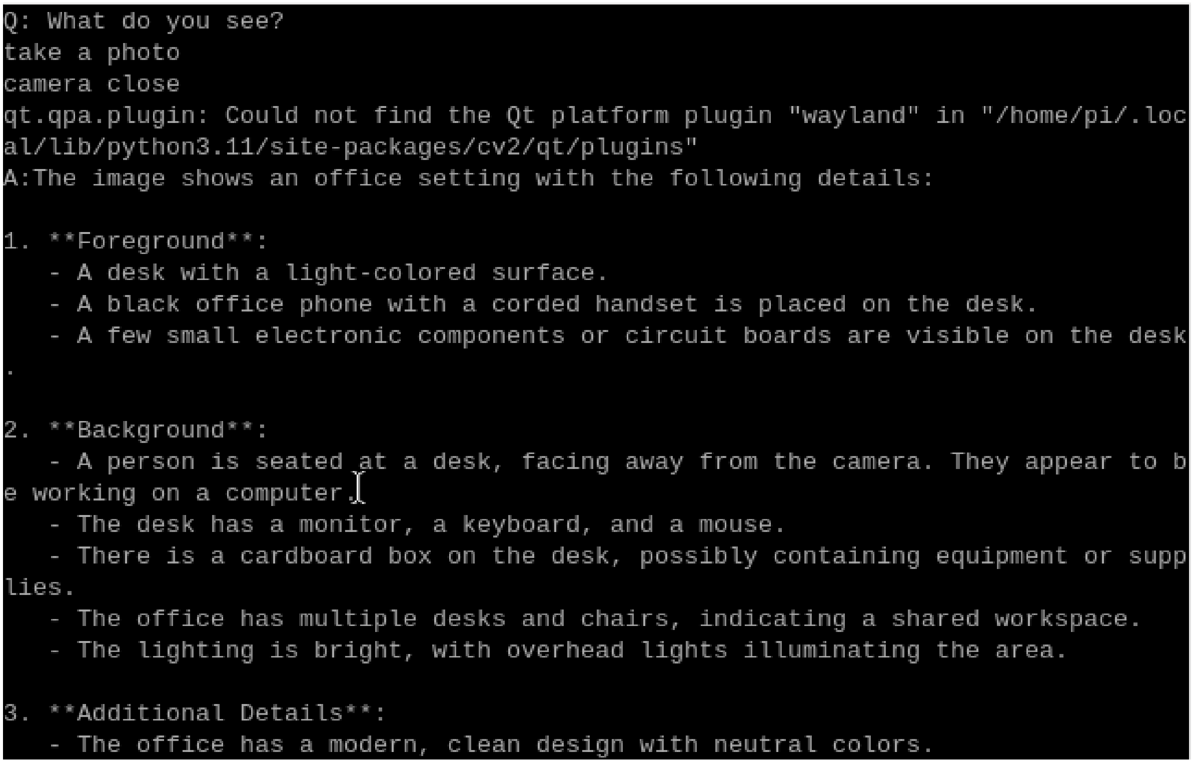
```
pi@yahboom:~/project_demo/09.AI_Big_Model $ python3 SceneDescription/sp_AI_Image.py
serial /dev/myspeech open
Waiting for keyword...
```

2. After successful wake-up, the green-boxed message in the image appears. You can then ask your question.



```
Keyword detected: 18.Jul 2025 11:49:47
Playing WAVE './didi.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib confmisc.c:1369:(snd_func_refer) Unable to find definition 'cards.2.pcm.hdmi.0:CARD=2,AES0=4,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:5180:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory
ALSA lib conf.c:5703:(snd_config_expand) Evaluate error: No such file or directory
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM hdmi
ALSA lib confmisc.c:1369:(snd_func_refer) Unable to find definition 'cards.2.pcm.hdmi.0:CARD=2,AES0=4,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:5180:(_snd_config_evaluate) function snd_func_refer returned error: No such file or directory
ALSA lib conf.c:5703:(snd_config_expand) Evaluate error: No such file or directory
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM hdmi
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
Current volume: 51681.0, boot threshold: 3000, End threshold: 1500
start recording
3000 51681.0
Current volume: 41887.0, boot threshold: 3000, End threshold: 1500
3000 41887.0
Current volume: 39350.0, boot threshold: 3000, End threshold: 1500
3000 39350.0
Current volume: 41687.0, boot threshold: 3000, End threshold: 1500
3000 41687.0
Current volume: 39974.0, boot threshold: 3000, End threshold: 1500
3000 39974.0
```

3. After a while, the terminal will print out the question and answer.



```
Q: What do you see?
take a photo
camera close
qt.qpa.plugin: Could not find the Qt platform plugin "wayland" in "/home/pi/.loc
al/lib/python3.11/site-packages/cv2/qt/plugins"
A:The image shows an office setting with the following details:

1. **Foreground**:
   - A desk with a light-colored surface.
   - A black office phone with a corded handset is placed on the desk.
   - A few small electronic components or circuit boards are visible on the desk
.

2. **Background**:
   - A person is seated at a desk, facing away from the camera. They appear to b
e working on a computer.
   - The desk has a monitor, a keyboard, and a mouse.
   - There is a cardboard box on the desk, possibly containing equipment or supp
lies.
   - The office has multiple desks and chairs, indicating a shared workspace.
   - The lighting is bright, with overhead lights illuminating the area.

3. **Additional Details**:
   - The office has a modern, clean design with neutral colors.
```

Main Source Code Analysis

```
def Speak_Vioce():
    global response
    if TTS_IAT_Tongyi:
        tonyi_tts(response)
    else:
        Xinghou_speaktts(response)

def main():
    while True:
        if detect_keyword():
            cv2.destroyAllWindows()
            os.system("pkill mplayer")
            time.sleep(.2)

            start_recording()

            if TTS_IAT_Tongyi:
                content = rec_wav_music_Tongyi()
            else:
                content = rec_wav_music()

            if content != "":
                print("Q:"+content)

                take_photo()
                time.sleep(1)
                mymytext = tongyi_Image(content)

                time.sleep(1)

                print("A:"+mymytext)


                try:
                    response = mymytext
                    tts_thread = threading.Thread(target=Speak_Vioce)
                    tts_thread.daemon = True
                    tts_thread.start()

                except:
                    pass
            if content == 0:
               break

        time.sleep(0.1)
```

take_photo: Records a photo on the spot
detect_keyword: Wake-up function handler
start_recording: Recording function handler
tongyi_Image: Visual large model analysis interface

Chinese version-specific options:
rec_wav_music_Tongyi: Tongyi Qianwen voice recognition
rec_wav_music: iFlytek Spark voice recognition
You can choose either voice recognition mode. You can enable or disable it in the **API_KEY.py** file. When TTS_IAT_Tongyi=True, either Tongyi Qianwen voice recognition or iFlytek Spark voice recognition is enabled.
tonyi_tts: Tongyi Qianwen speech synthesis
Xinghou_speaktts: iFlytek Xinghuo speech synthesis
You can choose between two speech synthesis options. You can enable or disable them in the **API_KEY.py** file. When TTS_IAT_Tongyi=True, either Tongyi Qianwen speech synthesis or iFlytek Xinghuo speech synthesis is enabled.

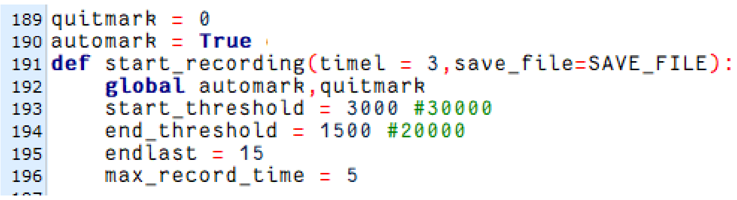
English Version
Speech recognition and synthesis use the iFlytek Spark API by default. You don't need to specify the iFlytek Spark API in API_KEY.py; simply fill in the **openAI_KEY** key.

Modifying the recording duration, start threshold, and end threshold

1. In the terminal, enter:

```
cd /home/pi/project_demo/09.AI_Big_Model/
nano audio.py
```

2. Find the source code shown below.



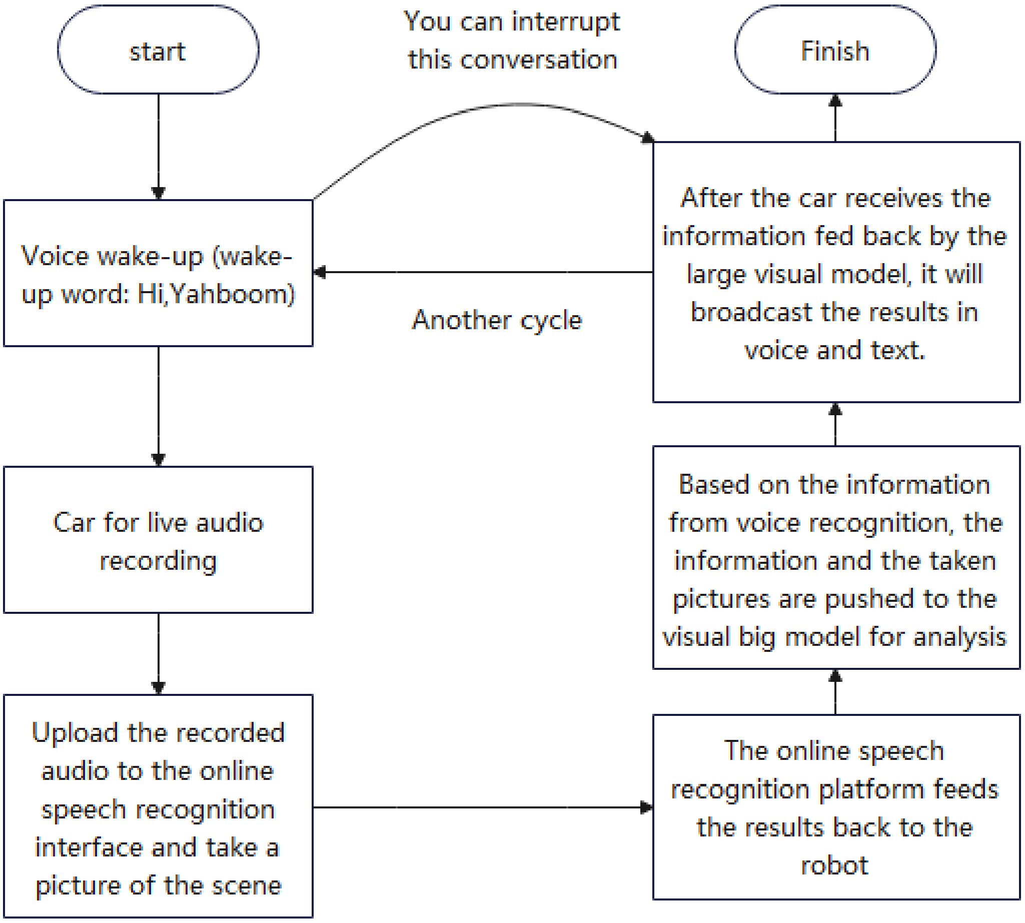
```
quitmark = 0
automark = True
def start_recording(timel = 3,save_file=SAVE_FILE):
    global automark,quitmark
    start_threshold = 3000 #30000
    end_threshold = 1500 #20000
    endlast = 15
    max_record_time = 5
```

- start_threshold: The threshold for starting recording when sound is detected (reduced to 5000 in quiet environments and increased to 150000+ in noisy environments).
- end_threshold: The threshold for stopping recording when sound is detected. A recommended value is 30-50% of start_threshold.
- endlast: The number of times to stop recording. Here, 15 is used. For example, Recording will automatically terminate if 15 consecutive sound values meet the stop threshold.
- max_record_time: Recording duration, set to 5 here.
  Note: start_threshold > end_threshold. This is a required rule, and its value can be determined based on the environment.

3. Directory Structure of the Main Files for this Experiment

```
├── sp_AI_Image_en.py #Main Entry Point for the English Version
├── sp_AI_Image.py #Main Entry Point for the Chinese Version
├── tongyi_api_image.py #Tongyi Qianwen Visual Model (This file is required only for model interface modifications)
├── tongyi_speak_iat.py #Tongyi Qianwen Speech Recognition
├── tongyi_tts.py #Tongyi Qianwen Speech Synthesis
├── xinghou_speak_iat.py #iFlytek Spark Speech Recognition
└── xinghou_tts.py #iFlytek Spark Speech Synthesis
```

Overall Flowchart


start
You can interrupt this conversation
Finish
Voice wake-up (wake-up word: Hi,Yahboom)
Another cycle
After the car receives the information fed back by the large visual model, it will broadcast the results in voice and text.
Car for live audio recording
Based on the information from voice recognition, the information and the taken pictures are pushed to the visual big model for analysis
Upload the recorded audio to the online speech recognition interface and take a picture of the scene
The online speech recognition platform feeds the results back to the robot