Robotic Manipulation Final Proj - Emotion Recognition on NXROBO Spark-T

Group 16: Dong Ding 224040226 Haosen Lou 224040328 Tingwei Wang 223040049

May 2025

Abstract

With the widespread adoption of robotics in modern society, robots have evolved beyond providing mere automation and intelligent services to also offer emotional support, such as companionship for children or the elderly. To better fulfill these needs, robots must adapt their behavior based on users' emotional states. In this project, we employ the NXROBO Spark-T robot as a platform to develop an emotional analysis and fixed-decision system, which comprises a voice module, a vision module, and an action module. The voice module leverages large language models to recognize spoken input and assess emotional probabilities; the vision module utilizes DeepFace to analyze facial emotions from camera input; finally, the action module fuses the emotional results from both modules and executes a fixed decision, enabling the robot to respond to user emotions in real time and adjust its behavior accordingly.

Keywords: robotics, control, motion planning, emotion recognization

1 Introduction

With the continuous advancement and integration of technology, robots have found widespread applications in various fields such as industrial manufacturing, healthcare, and service industries, demonstrating significant potential in production efficiency and intelligence. In recent years, increasing attention has been paid to the emotional value that robots can provide. For instance, companion robots designed for children and the elderly not only offer daily assistance and care but also deliver psychological support through emotional interaction, thereby enhancing the overall quality of users' lives.

The development of emotion recognition technology has provided the foundation and support for such emotional companionship needs. In early stages, emotion recognition mainly relied on traditional pattern recognition and feature extraction from voice and images (especially facial expressions). With the introduction of deep learning and large language models, researchers can utilize more powerful feature representations and multimodal fusion strategies to improve recognition accuracy. Common approaches now include voice-based emotion analysis, facial expression recognition from images or video sequences, and integrated analysis of multimodal data. The emergence of large-scale models brings higher-level comprehension and reasoning, enabling robots to more accurately perceive and analyze users' emotional states.

In this project, we divide the overall system into three parts: a voice module, a vision module, and an action module. The voice module uses PyAudio to capture microphone input and uses Zhipu Qingyan's GLM-ASR and GLM-4-PLUS to perform voice recognition and emotion analysis on microphone input. GLM-ASR transcripts audio into text that conforms to language habits based on context understanding, significantly improving the fluency and readability of the output results. Meanwhile, the model performs significantly better than the current model in the noisy environment and is not disturbed by non-verbal noise. The model supports Chinese, English and various dialects from different regions (Northeastern Mandarin, Jiaoliao Mandarin, Beijing Mandarin, Jilu Mandarin, Central Plains Mandarin, Jianghuai Mandarin, Lanyin Mandarin and Southwest Mandarin). GLM-4-Plus uses a large number of models to assist in constructing high-quality synthetic data to enhance model performance. It utilizes PPO to effectively improve the performance of model reasoning (such as mathematics, code algorithm problems, etc.) and better reflect human preferences. In terms of various performance indicators, the GLM-4-Plus has reached the same level as first-tier models such as the GPT-4o. The vision module obtains images and conducts emotion recognition via Deepface to acquire users' emotional information. DeepFace is a lightweight face recognition and facial attribute analysis (age, gender, emotion and race) framework for python. It is a hybrid face recognition framework wrapping state-of-the-art models: VGG-Face, FaceNet, OpenFace, DeepFace, DeepID, ArcFace, Dlib, SFace, GhostFaceNet, Buffalo_L. The action module, inspired by the SwiftPro project, handles the robot's control logic: it first fuses the emotional results from the voice and vision modules, and then makes fixed decisions based on the final emotional outcome to determine the robot's actions. Swift and swiftpro ROS package designed by Roger Cui(roger@ufactory.cc) and David Long (xiaokun.long@ufactory.cc). These packages support Moveit!, RViz and serial communication with swift and swiftpro. All hardware interactions with the robot are implemented through ROS and NXROBO's Spark-T sample project (https://github.com/NXROBO/spark), ensuring stable operation in an embedded environment.

2 Experiment

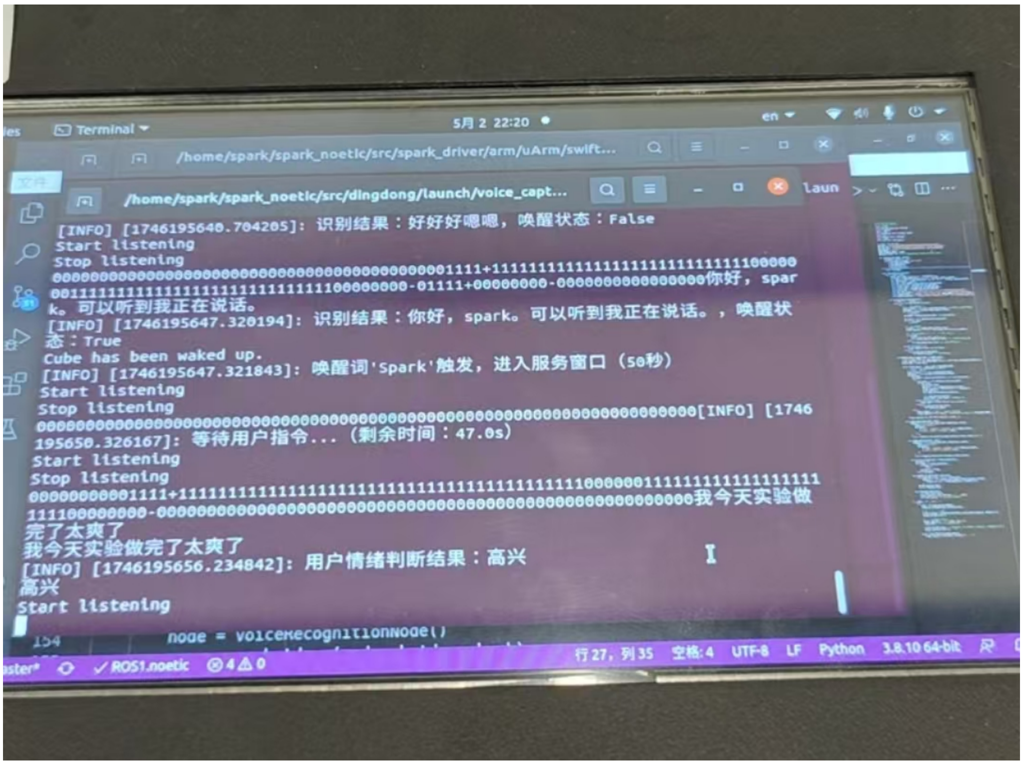
In this section, the detail of each module will be shown.

2.1 Voice Module

The voice Module is responsible for recognizing and capturing external user voice and determining its emotional content. The basic approach for capturing microphone input is adapted from the same Spark-T project test examples. In those test examples, there is a "lib" library containing a pre-written Microphone class, which retrieves microphone input frame by frame. This class uses PyAudio to open an audio input stream at a fixed sample rate of 16 kHz, 16-bit depth per frame, and single-channel input, reading audio data from the microphone.

After reading the voice, by stitching together each audio frame according to the sampling rate and bit width, the voice can be reconstructed and saved to a WAV file for subsequent analysis.

Because of hardware memory constraints, voice analysis is carried out by invoking an external API of model GLM-ASR from Zhipu Qingyan—to recognize the voice content. Once the text is recognized, GLM-4-PLUS is then called to identify the emotional state of the text. Based on this analysis, it returns probabilities for four emotions: happiness, sadness, anger, and calmness.



Terminal
5月 2 22:20
en
/home/spark/spark_noetic/src/spark_driver/arm/uArm/swift...
/home/spark/spark_noetic/src/dingdong/launch/voice_capt...
laun
行 27，列 35
空格:4
UTF-8
LF
Python
3.8.10 64-bit
✓ROS1.noetic
⊗4⚠0

```
[INFO] [1746195640.704205]: 识别结果：好好好嗯嗯，唤醒状态：False
Start listening
Stop listening
000000000000000000000000000000000000000000000001111+111111111111111111111111111100000
0011111111111111111111111111111100000000-01111+00000000-0000000000000000你好，spar
k。可以听到我正在说话。
[INFO] [1746195647.320194]: 识别结果：你好，spark。可以听到我正在说话。，唤醒状
态：True
Cube has been waked up.
[INFO] [1746195647.321843]: 唤醒词'Spark'触发，进入服务窗口（50秒）
Start listening
Stop listening
000000000000000000000000000000000000000000000000000000000000000000000000[INFO] [1746
195650.326167]: 等待用户指令...（剩余时间：47.0s）
Start listening
Stop listening
00000000001111+11111111111111111111111111111111111111111111000000111111111111111111
11110000000000-00000000000000000000000000000000000000000000000000000000我今天实验做
完了太爽了
我今天实验做完了太爽了
[INFO] [1746195656.234842]: 用户情绪判断结果：高兴
高兴
Start listening
```

Figure 1: ASR Example

To prevent the robot from missing important calls or valuable information due to receiving meaningless data, we choose "spark" as the voice wake word. Only when the recognized voice contains the word "spark" will the robot enter the listening mode to analyze the emotion of the subsequent voice input.

2.2 Vision Module

The Vision Module is responsible for detecting the human face in the images captured by the camera and identifying the emotion in the images. The basic approach of acquiring camera input follows the Spark-T project test examples. The provided astra_camera project supports the astra_pro camera type. By running the astra_camera node and using nodelets to replicate multiple camera nodes (enabling zero-copy), you can separately obtain grayscale images, RGB images, and depth images. The project publishes a topic /camera/rgb/image_raw. By subscribing to this topic and running a callback, you can retrieve the images and save them locally.

After obtaining the images, we use DeepFace in the robot' s local microservice to perform facial emotion recognition. The camera feed is encoded to Base64 and sent via HTTP request to a Flask service running in the same container. There, OpenCV decodes the Base64 data to a BGR image, which is then passed directly to DeepFace' s analyze method. Internally, DeepFace first uses MTCNN to locate the face and then applies various pre-trained models to estimate the confidence for four emotions. It then returns a JSON response back to the robot side. This end-to-end, locally deployed flow greatly simplifies the script logic while ensuring millisecond-level response times and high robustness.

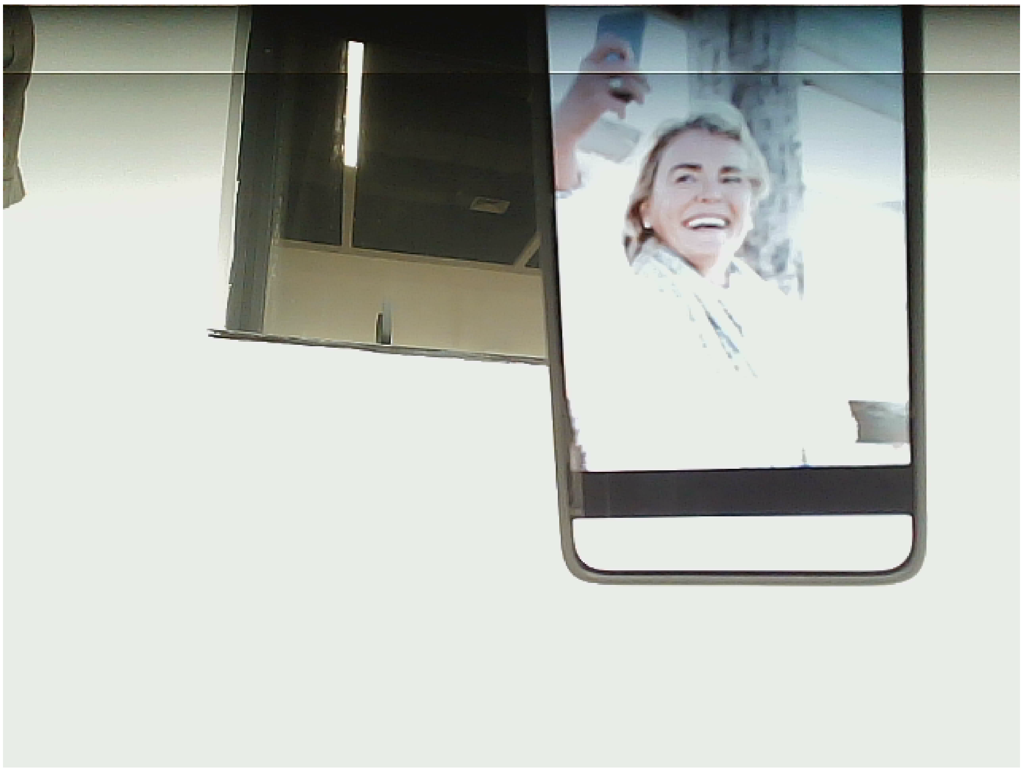
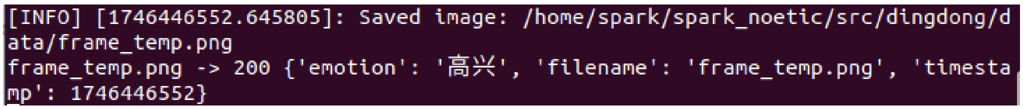


Figure 2: Camera Capture Example



```
[INFO] [1746446552.645805]: Saved image: /home/spark/spark_noetic/src/dingdong/d
ata/frame_temp.png
frame_temp.png -> 200 {'emotion': '高兴', 'filename': 'frame_temp.png', 'timesta
mp': 1746446552}
```

Figure 3: Emotion Recognization Example

2.3 Motion Module

The Action Module is responsible for directly controlling the movements of the robotic arm. Basic robotic arm control is adapted from related code in the Spark-T test examples. The programs for reading signals from the robotic arm are based on the RosForSwiftAndSwiftPro project, developed by Roger Cui (`roger@ufactory.cc`) and David Long (`xiaokun.long@ufactory.cc`).

This project acquires real-time information from the physical device via the device port and provides four primary node processes:

- `swiftpro_moveit_node` and `swiftpro_rviz_node`, which enable MoveIt and RViz visualization for the robotic arm.

- `swiftpro_read_node` and `swiftpro_write_node`, which respectively offer ROS topics to read and write terminal position parameters of the robotic arm.

In this project, we mainly utilize the two nodes for reading and writing the terminal position. These two nodes offer two topics, `SwiftproState_topic` and `position_write_callback`, which are used respectively to read and write the robotic arm' s end-effector (xyz) coordinates and gripper states. By manually adjusting the arm' s position and listening to the `SwiftproState_topic`, we can obtain the path coordinates of the desired motions. Then, by publishing these path coordinates to the `position_write_callback` topic in an actual application, we can achieve specific robotic arm movements.

We designed four distinct actions in our project, corresponding to the following emotional states: happiness, sadness, anger, and calmness:

- **Happiness:** The robot mimics a bee' s "figure-eight dance."
- **Sadness:** The arm tilts aside and curls up.
- **Anger:** The robot simulates a hammering motion.
- **Calmness:** The robot draws a wave-like motion (implemented with a sine function).

2.4 Inter-module Communication

These modules communicate by subscribing and publishing to ROS topics. A centralized "emotion-processing module" receives confirmed emotional speech, retrieves images for analysis, then combines the results from both recognition processes. After taking a weighted average of the probabilities, it derives the final emotional result and publishes it on an emotion_result topic. The Action Module subscribes to this topic and makes fixed decisions based on the incoming results, executing specific movements accordingly.

3 Summary

In this project, we used ROS and large-scale models (LLMs) to build a robot with speech and image recognition and analysis capabilities, producing emotion-driven, fixed-decision movements. The complete code is placed in our github repo(https://github.com/Fa11enDeity/AIR5021-Team16-FinalProject). Through this process, we gained deeper insights into using ROS and furthered our understanding of both theoretical and practical approaches to robotic control. With more optimization and extensions in the future, integrating large models with robot control could become a solution applicable to areas such as elderly or childcare assistance, offering meaningful real-world impact.

This project still has considerable development potential. Currently, we use fixed decision logic for actions. By deploying a relatively lightweight reinforcement learning model, the robot might learn and execute new action logic on its own. Additionally, since we use a fixed camera, the image quality may be limited. It is worth exploring whether, after receiving speech input, the robot can combine a LiDAR scan and visual input to locate the user in its surroundings and capture clearer, more appropriate images. Furthermore, this project combines a single speech model and a single image model, to some extent disconnecting speech from vision. If a multimodal model could simultaneously analyze speech, images, and possibly textual prompts, the accuracy of the emotion analysis could be significantly improved.

4 Member Distribution

In the experiment, Dong Ding (224040226) was responsible for the design of the motion module and clerical work, Haosen Lou (224040328) was responsible for the design of the voice module and on-machine test, and Tingwei Wang (223040049) was responsible for the design of the vision module.

Appendix A GitHub Repo

(https://github.com/Fa11enDeity/AIR5021-Team16-FinalProject)