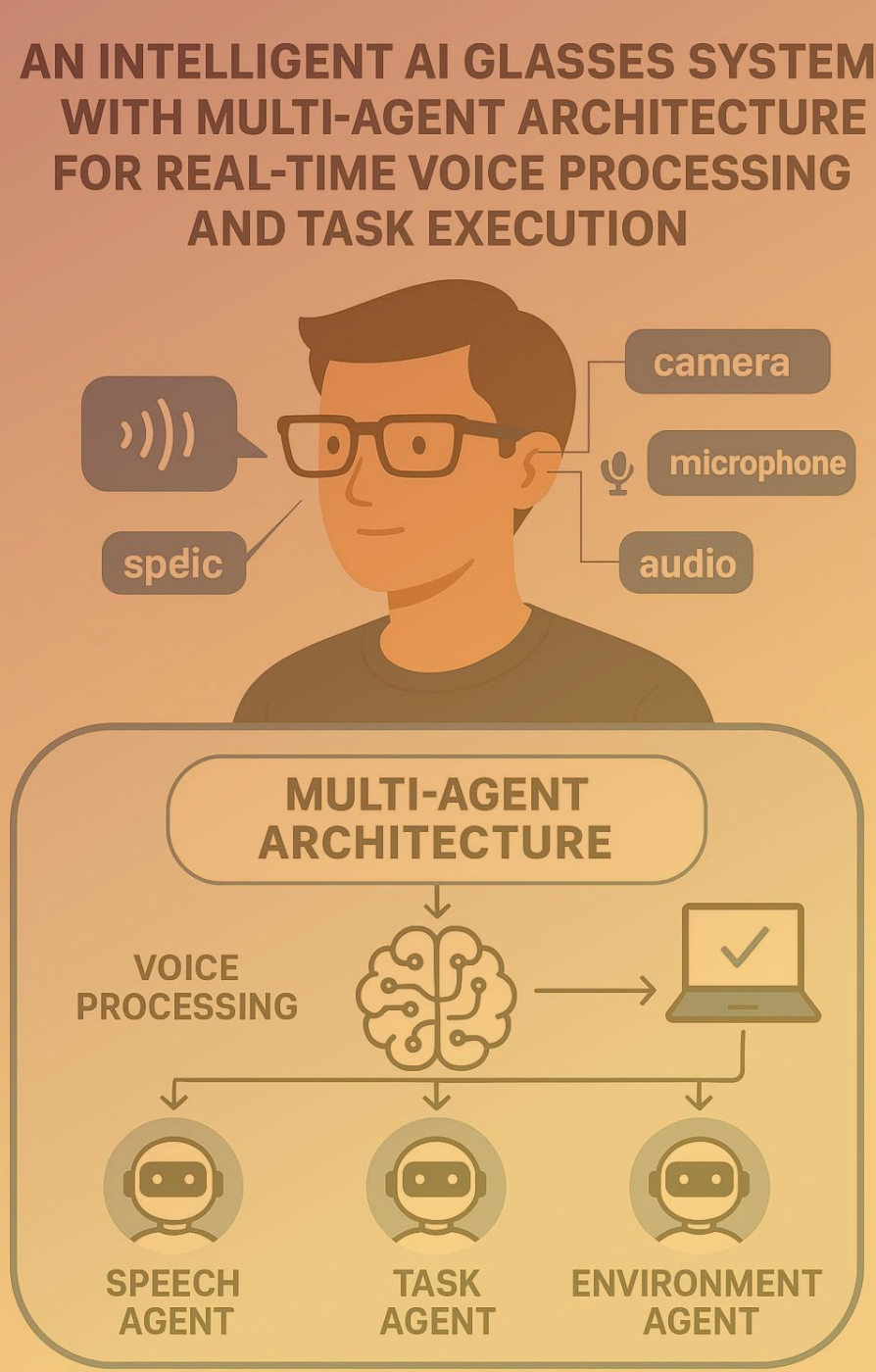
NARLabs 財團法人國家實驗研究院
國家高速網路與計算中心
National Center for High-performance Computing

NCS 2025

An Intelligent AI glasses System with Multi-Agent Architecture for Real-Time Voice Processing and Task Execution

October 2025

Presenter: Jyu-Hong Wu
Authors: Sheng-Kai Chen, Jyh-Horng Wu, Ching-Yao Lin and Yen-Ting Lin



Contents

Background and Motivation

AR or AI Glasses are now popular but still occur some problems of application.

Research Background

- **Augmented Reality**

  Currently, AR technology can be applied in many daily lives, such as using cell phones or glasses to virtually wear clothes, placing virtual characters in games into real environments, etc.

- **Large Language Model**

  In recent years, with the maturity of large language models, generative AI has gradually been integrated into our daily lives, such as ChatGPT.

Pain Point Analysis

- **Models are not fully functional**

  Large language models can answer many of our questions, but if the model is set up locally, it will not be possible to find data in real time or query specific information.

- **Applications cannot build a complete system**

  In the application of AR glasses, it is still impossible to give users the most accurate advice on how to proceed to the next step.

Goals

- **Collect data from glasses**

  Collect data from AI Glasses, such as eye detection etc.

- **Building AI agent**

  Build an AI agent that provides more practical advice.

- **Real time streaming**

  Stream the remote screen to the glasses.

Methodology

Showing full system architecture.

AI

- **MCP (Model Context Protocol) + Ollama**
  Use MCP to connect language models with various APIs to make responses more accurate and use the Ollama package to load various local models.
- **ASR(Automatic Speech Recognition)**
  Use the Whisper.cpp package to convert real-time voice data into text and turn it into prompt input into the model.

AR

- **Real-Time Streaming**
  Stream the remote host's computer screen to the AR glasses in real time for users to view.
- **Eye Detection**
  Through eye detection technology, the glasses detect where the eyes are looking and transmit it back to the system.

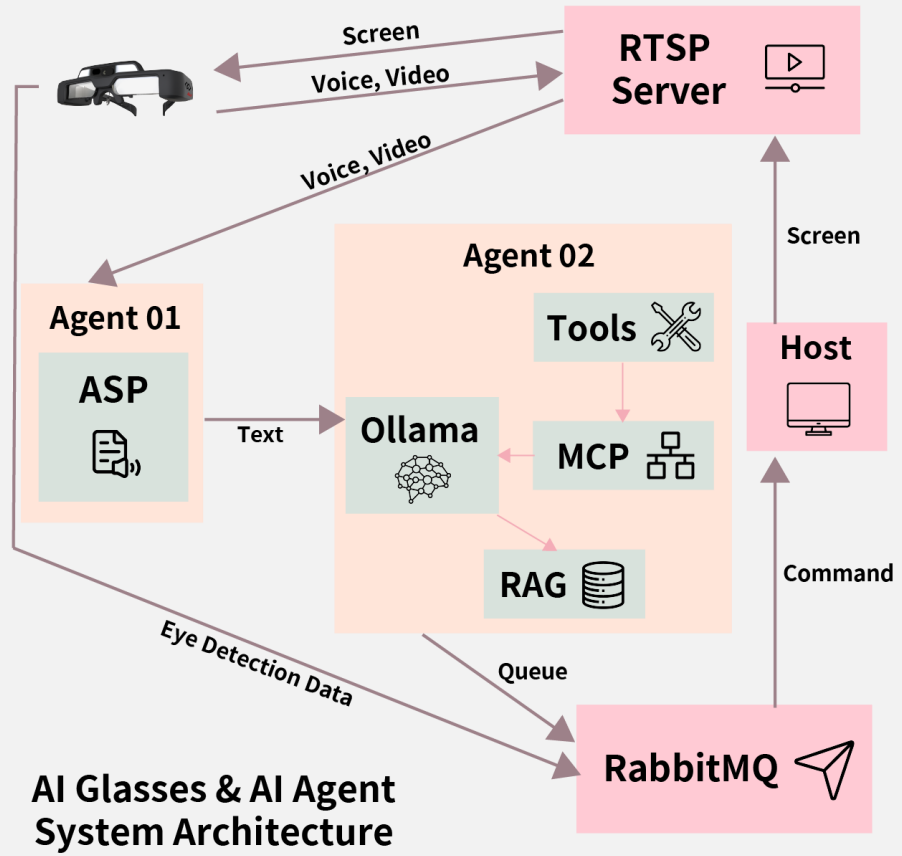


Fig 1. System Architecture

Methodology

Showing the design of AI agents.

Speech Recognitions

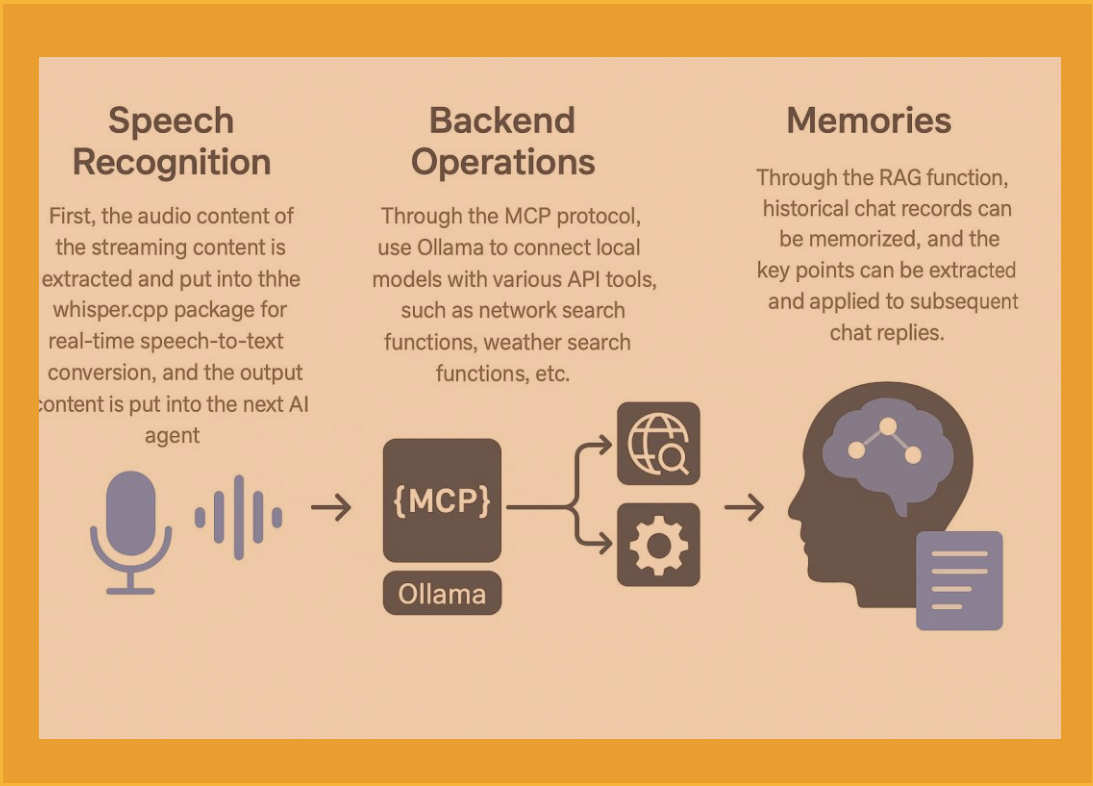
First, the audio content of the streaming content is extracted and put into the whisper.cpp package for real-time speech-to-text conversion, and the output content is put into the next AI agent.

Backend Operations

Through the MCP protocol, use Ollama to connect local models with various API tools, such as network search functions, weather search functions, etc.

Memories

Through the RAG function, historical chat records can be memorized, and the key points can be extracted and applied to subsequent chat replies.



Methodology

Showing the details of eye tracking procedure.

Eye Tracking

- **APIs**

  Through Ganzin hardware sensors via Android APIs.
- **Data**

  Through Develop APIs, data that can be extracted are left eye, right eye locations and each contains its x, y and z.

RabbitMQ

- **Format**

  Data are represented in JSON form.
- **AI agent**

  Data will contain the command like open the browser and content like web address.
- **Eye data**

  Through eye tracking, all locations data upload to server.

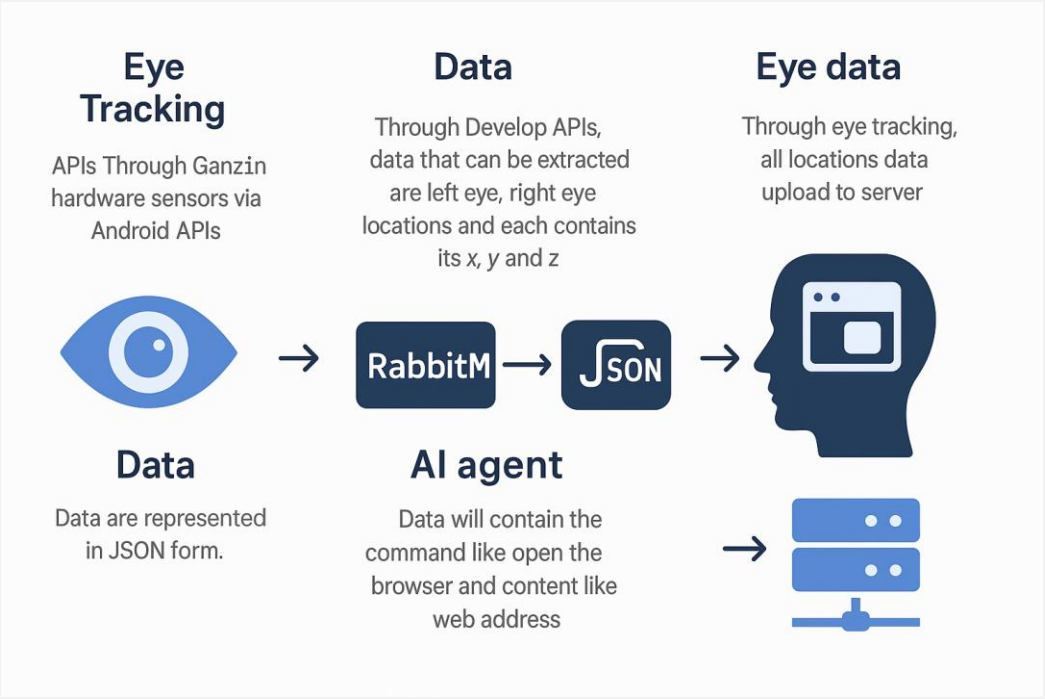
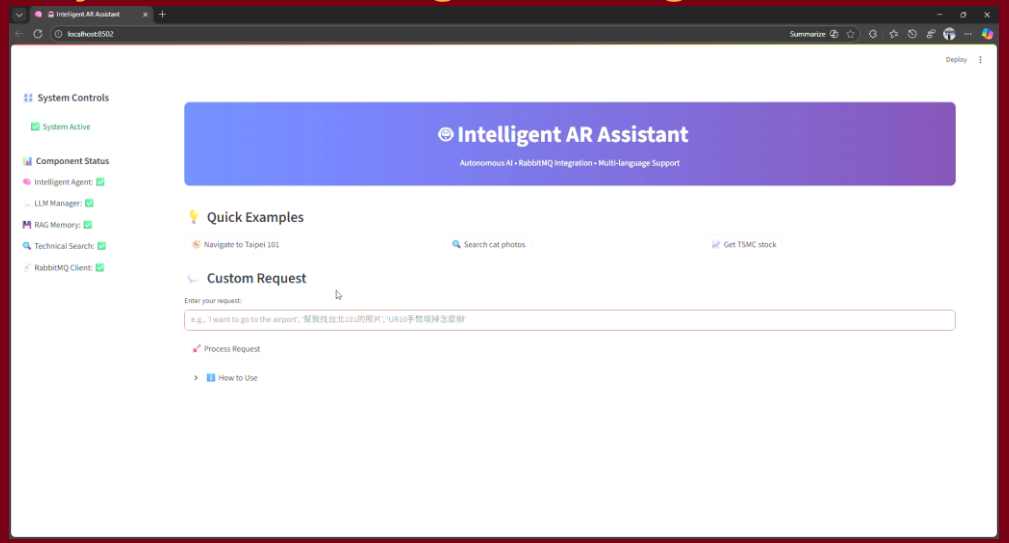


Fig 2. Server and eye tracking procedure

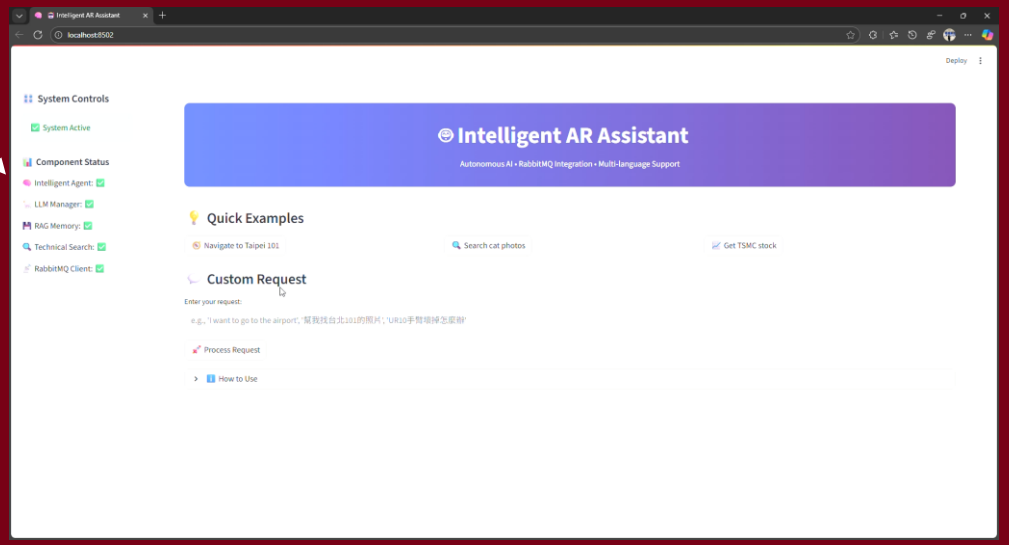
Results

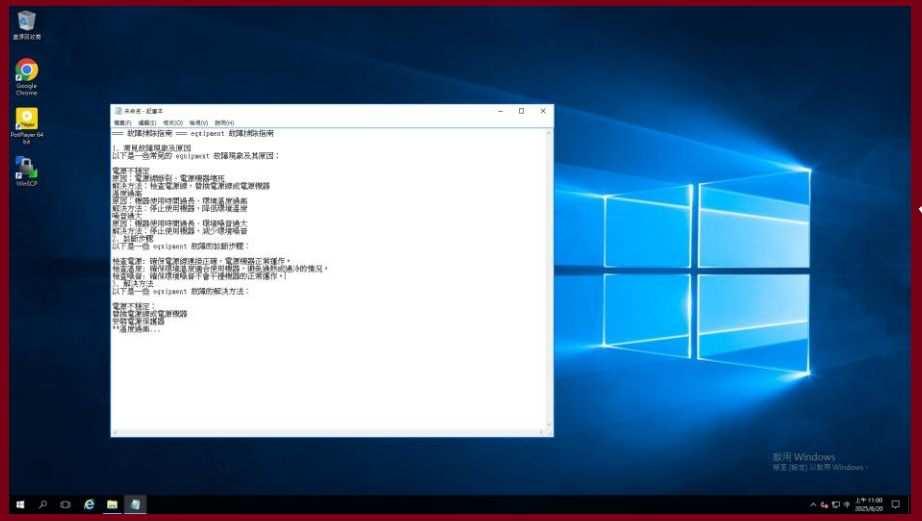
Showing the system streaming and AI agent functioning.

Voice

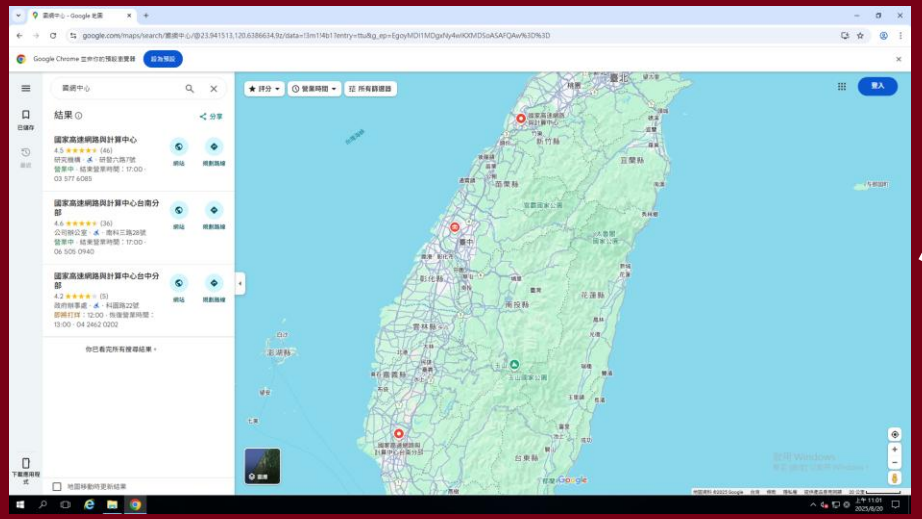


Voice





Screen



Screen

Future Work

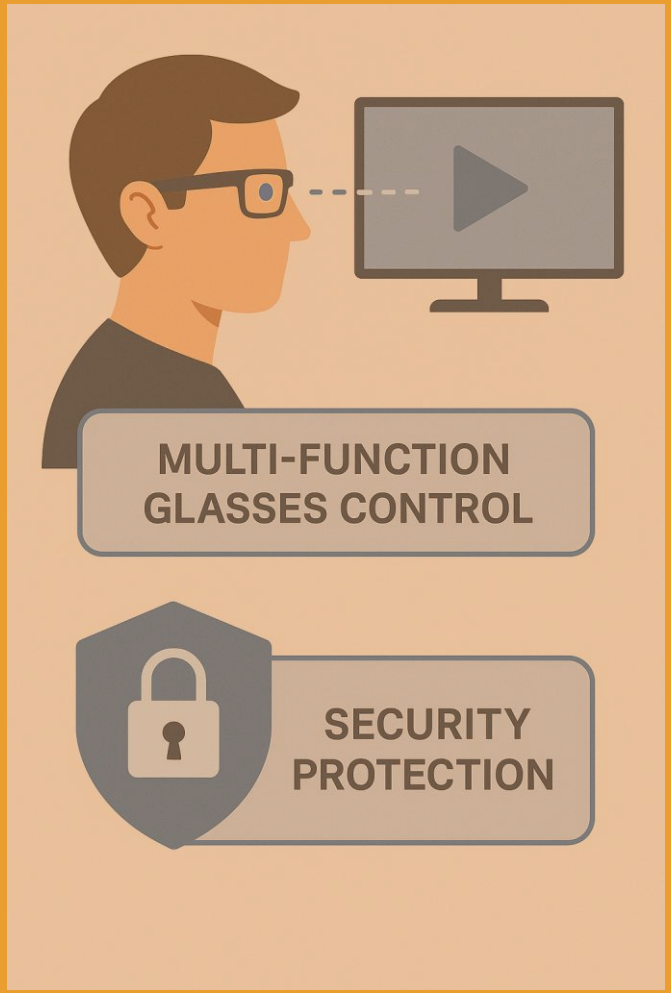
Where it can be applied in future projects.

Multi-function glasses control

Through eye tracking and the actual use of the hardware on the glasses, remote control can be performed.

Security protection

When using AI Agent and RTSP streaming, data protection measures must be added.



Thank You

Sheng-Kai Chen
s1134807@mail.yzu.edu.tw

Jyh-Horng Wu
jhwu@niar.org.tw

Ching-Yao Lin
chingyao@niar.org.tw

Yen-Ting Lin
2303131@niar.org.tw