About

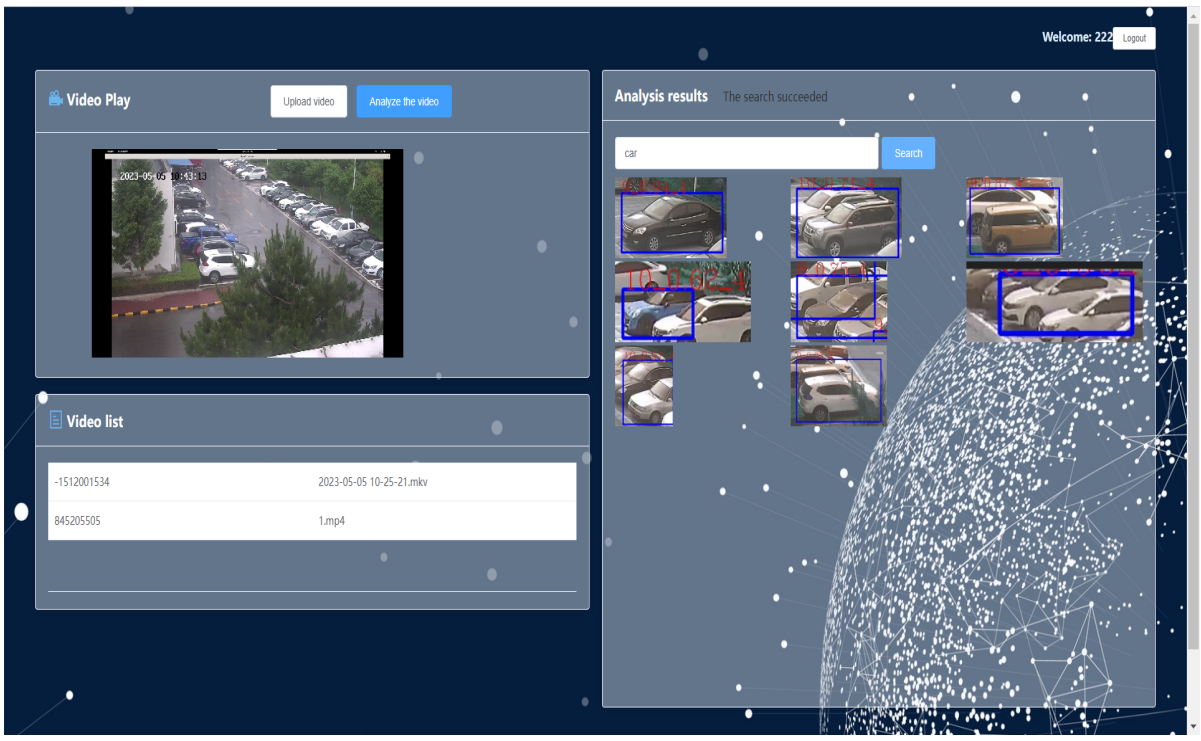
Project Background

As more and more cameras are deployed in public places and even in individual homes, the amount of frequency data generated is undoubtedly increasing rapidly. The data scale has increased from gigabytes to terabytes to petabytes, and the video surveillance data of a medium-sized city can reach more than petabytes in just one day.

The information contained in these videos can effectively solve the challenges in many fields of our life,and has been widely used in traffic management, social security, production safety and other fields.

PROJECT ARTIFACTS

Models



Important Components

Spring Boot is designed to simplify the creation of production-level Spring applications and services, simplifying configuration files.

Spring Cloud is a microservice framework developed based on the spring Boot framework, providing a set of microservice solutions.

MongoDB is used to save data, which can meet the application of small-scale systems.

MQ can accumulate data in a short period of time to prevent a large amount of information from flooding into the database and causing system crashes during peak requests.

About Gstreamer

1.A framework for creating streaming media applications.

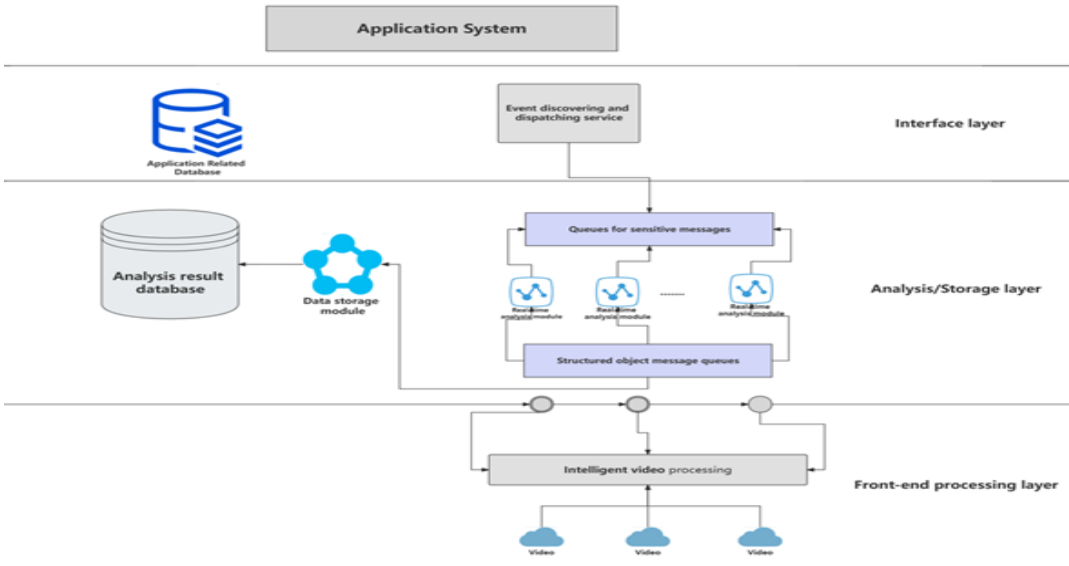
2.Designed to simplify the process of writing applications that handle audio or video.

3.Based on plug-ins that will provide various codecs and other capabilities.

Architecture

System logics

Three-tier architecture

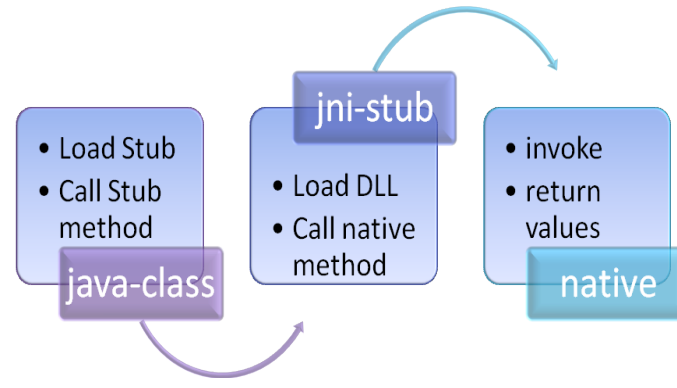


Front-end processing layer: the live video streams are first received through the video analysis service.

Storage layer: be converted into structured data through the message queue of the analysis and storage layer.

Interface layer: applications can access these data to meet different requirements.

Microservice Architecture



First, the real-time data stream information will enter the video analysis system for analysis. The system sends the results to the message queue.

The application system will obtain the required analysis data through subscription service. Both the application system and the message queue have access to the data warehouse.

Finally, the application system will provide services for the applications that people use.

Challenges

Technical

1. Not familiar with jni interface and how to implement jni callback

After completing the video parsing algorithm, we need to implement the jni function in c++, and call the callback interface of the java web service after detecting the target. However, we do not have any relevant experience in this part.

2. Data Overflow

In the code for the appsink data output, the code always overflows data in the buffer we use.

3. The recognition result is inaccurate

When identifying targets, it is impossible to circle all the targets, and sometimes there are multiple returned results, but not all targets can be detected in every frame.

Non-Technical

1. Communication Restrictions

Since some students have internships at the end of the semester and do not live on campus, it is difficult to communicate offline and connect with tutors.

2. Drawing restrictions

Since we do not have professional designers, we have encountered difficulties in drawing architecture diagrams, system logic diagrams and related diagram beautification convenience, and it takes a certain amount of time to familiarize ourselves with the

Review

What we worked well

Communication with mentors and clients was relatively smooth. Our doubts during the project process were basically given timely feedback, and it also helped us a lot to complete the documents. And in the web and video analysis part of the project, we implemented the vision very well.

What Didn't Worked Well

Some members are not familiar with the interface, and we spend more time implementing the call logic of our system. At the same time, during the completion of the project, we sometimes did not make timely backups and debug the code modularly, which cost a certain amount of time.

Target

This project's target is going to create a video analysis framework based on Gstreamer.

Complete the video stream decoding processing.

Analyze the video stream parsing code writing, and analyze the real-time video stream.

Finally, it should realize the function of parsing multiple video streams, which can convey the analysis results to the web page for display.

It also should provide the function of visual rendering of video analysis results. into the video stream with playback in the web portal.

Key Milestones

1) Construct hardware and build development environment
2) Build architecture of the project
3) Core system code design
4) Test and optimization
5) Reports writing

Design of video analysis framework based on Gstreamer

Qinshi Zhang, Yiru Zhu, Yingying Huang, Weiyi Ren and Shimei Meng
Auckland University of Technology

Research & Development Project
Supervisor:
Tao Haijun, Xiang Lin
Client: Tao Haijun Xiang lin
Date: S2 2023

About

Project Background

As more and more cameras are deployed in public places and even in individual homes, the amount of frequency data generated is undoubtedly increasing rapidly. The data scale has increased from gigabytes to terabytes to petabytes, and the video surveillance data of a medium-sized city can reach more than petabytes in just one day.

The information contained in these videos can effectively solve the challenges in many fields of our life,and has been widely used in traffic management, social security, production safety and other fields.

Target

This project's target is going to create a video analysis framework based on Gstreamer.

Complete the video stream decoding processing.

Analyze the video stream parsing code writing, and analyze the real-time video stream.

Finally, it should realize the function of parsing multiple video streams, which can convey the analysis results to the web page for display.

It also should provide the function of visual rendering of video analysis results. into the video stream with playback in the web portal.

Key Milestones

1) Construct hardware and build development environment
2) Build architecture of the project
3) Core system code design
4) Test and optimization
5) Reports writing

PROJECT ARTIFACTS

Models

Important Components

Spring Boot is designed to simplify the creation of production-level Spring applications and services, simplifying configuration files.

Spring Cloud is a microservice framework developed based on the spring Boot framework, providing a set of microservice solutions.

MongoDB is used to save data, which can meet the application of small-scale systems.

MQ can accumulate data in a short period of time to prevent a large amount of information from flooding into the database and causing system crashes during peak requests.

About Gstreamer

1.A framework for creating streaming media applications.

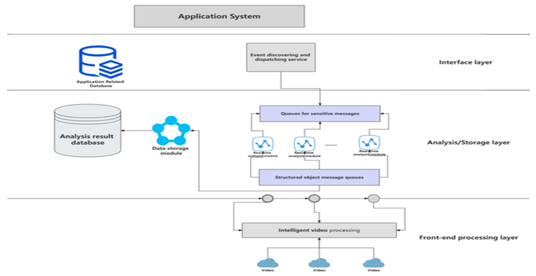
2.Designed to simplify the process of writing applications that handle audio or video.

3.Based on plug-ins that will provide various codecs and other capabilities.

Architecture

System logics

Three-tier architecture

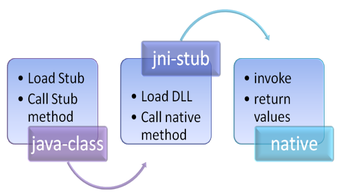


Front-end processing layer: the live video streams are first received through the video analysis service.

Storage layer: be converted into structured data through the message queue of the analysis and storage layer.

Interface layer: applications can access these data to meet different requirements.

Microservice Architecture



First, the real-time data stream information will enter the video analysis system for analysis. The system sends the results to the message queue.

The application system will obtain the required analysis data through subscription service. Both the application system and the message queue have access to the data warehouse.

Finally, the application system will provide services for the applications that people use.

Challenges

Technical

1. Not familiar with jni interface and how to implement jni callback

After completing the video parsing algorithm, we need to implement the jni function in c++, and call the callback interface of the java web service after detecting the target. However, we do not have any relevant experience in this part.

2. Data Overflow

In the code for the appsink data output, the code always overflows data in the buffer we use.

3. The recognition result is inaccurate

When identifying targets, it is impossible to circle all the targets, and sometimes there are multiple returned results, but not all targets can be detected in every frame.

Non-Technical

1. Communication Restrictions

Since some students have internships at the end of the semester and do not live on campus, it is difficult to communicate offline and connect with tutors.

2. Drawing restrictions

Since we do not have professional designers, we have encountered difficulties in drawing architecture diagrams, system logic diagrams and related diagram beautification convenience, and it takes a certain amount of time to familiarize ourselves with the

Review

What we worked well

Communication with mentors and clients was relatively smooth. Our doubts during the project process were basically given timely feedback, and it also helped us a lot to complete the documents. And in the web and video analysis part of the project, we implemented the vision very well.

What Didn't Worked Well

Some members are not familiar with the interface, and we spend more time implementing the call logic of our system. At the same time, during the completion of the project, we sometimes did not make timely backups and debug the code modularly, which cost a certain amount of time.