Real-Time Image and Video Indexing

Lakshya Goyal, Roopasree Naidu, Youngsik Yoon

Continuous Analysis of Many Cameras, School of Computer Science, School of Electrical & Computer Engineering

Introduction

The goal of the project is to build a storage system for real-time indexing of incoming surveillance streams and make it searchable. We want to design, implement and evaluate a database that can retrieve image or video data, build index-by-object detection and store the respective data and index in an easy to access manner.

Due to lack of image feature information, indexing and accessing data is cumbersome. However with the image database, students and researchers can further their work by having access to specialized datasets.

Objective

To develop a storage system for indexing incoming image and video from networked cameras in real-time and make the data searchable by features.

The system should be scalable for multiple geolocations around the world to increase speed of data retrieval.

Data

Use Cases

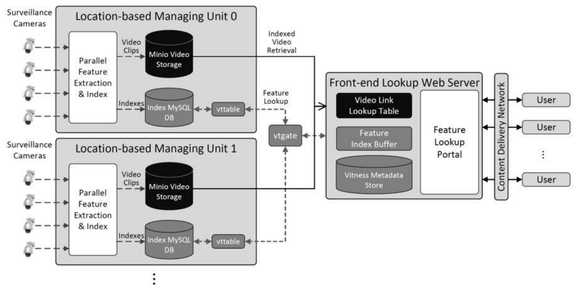
To develop the system by slices, the team collected use cases. This allowed the team to get a minimum viable product to solve subset of all user stories. Use cases were collected by interviewing and surveying other CAM^2 teams.

Videos and Images

To initially populate the image database, the local historical camera data can be used. Real-time indexing will allow networked cameras to fill in the image database. Additionally, users will be able to populate the image database with their own images. For labeling data, users will submit their own data labels which will be inspected before being inserted into the image database.

Design

Network Architecture

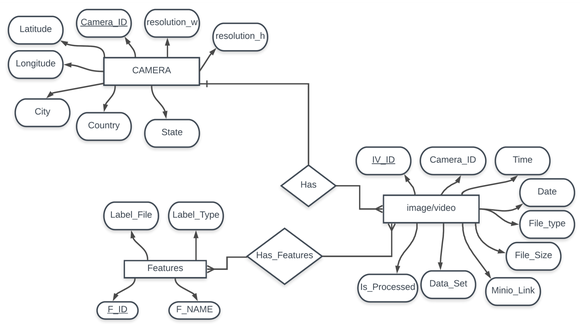


The purpose of this database is to enable networked cameras all around the world to be searched for. And so, we needed an easily scalable and distributed architecture. Currently, we are working on just local images and videos which are uploaded manually by the user. This represents the inputs form the networked cameras.

Within the distributed server, we will perform feature extraction of the incoming images. These images will be stored in the Minio Object Storage Database and their features will be stored in Vitess, which will run the front end of the user interaction. We chose Minio because of it stores unstructured data; useful for images and videos.

Vitess ER Diagram

Vitess is a database clustering system for horizontal scaling of MySQL. It is a database with native ACID support for structural data. The Vitess database contains tables that stores relevant information related to the images and the camera. It will be used for processing indexed features and serving real-time search requests.



Implications

A few of the use cases for this database are as follows:

1. Researchers and other users will be able to use data from this database and conduct their research more efficiently.
2. This database is not restricted to a specific application. Any individual in need of image data sets can use this database.

Stretched Implications

- Video cameras are everything and many of them can be accessed by public
- Surveillance footage can be used to monitor emergencies
- In the case of emergency, surveillance footage can be used to assess the damage
- Surveillance footage can be used for environmental studies

Therefore, being able to index real-time surveillance streams is crucial to above applications.

Conclusions

The image database hopes to make it faster and easier for users to find specific cameras that contain the features that are related to their interest areas.

The minimum viable product for the project will integrate the network architecture as well as the storage for the features and images. The image database will solve the subset of all user stories collected. For later functions, we will need to collect more use cases.

The current image database will run on a command line interface where users will input commands for updating, inserting, deleting, or retrieving video and image features.

Future Goals

Future goals for the project include incorporating other CAM2 teams such as Parallel Computing for Vision. The increase of confidence in object detection in real time analysis can help improve real-time indexing of incoming surveillance streams.

Integrate real time object detection systems like YOLO into our workflow to automatically run on our data and store the resulting bounding boxes in the database.

The system should geo-distributed so that each node stores data from the cameras closest to it, to reduce latency in image storage and improve data availability

Developing a web interface to provide users with a more friendly experience, clear view of real time data, overview of various operation options and other functionalities.

References

Chen, Shuo-Han. “Real - Time Surveillance Video Indexing Storage.” 2018.

“Distributed Minio Quickstart Guide.” Minio Docs, docs.minio.io/docs/distributed-minio-quickstart-guide

“What Is Vitess.” Vitess, vitess.io/docs/overview/whatisvitess/.

Acknowledgements

1. Dr. Yung-Hsiang Lu's and Shuo-Han Chen's Continuous Analysis of Many Cameras Team
2. Vertically Integrated Projects – Electrical & Computer Engineering
3. Image Database Team - Ling Zhang, Haoran Wang, Shunqiao Huang, Sripath Misra, Ayushi Gupta