Distributed Systems - Surveillance System Project

1st Cedric Sillaber 2nd Alan Gallo 3rd Frantisek Sova

I. Introduction

In our surveillance system, cameras are simulated using the WiseNET dataset. The system is designed to capture video streams and process them efficiently for real-time object and facial recognition. The video data undergoes preprocessing at the edge, minimizing the workload on the cloud, reducing latency, and optimizing the overall performance of the surveillance system. The system leverages edge devices and cloud services to ensure scalability, fast response times, and effective monitoring.

II. System Architecture

The video streams are stored in an SQLite database on Amazon S3 1. Camera containers running on EC2 instances fetch the data from the database and split it into image sequences using OpenCV. These sequences are then sent to edge devices at regular intervals, simulating a real-time video stream. Communication between components is handled through a REST API, which is capable of managing large payloads and scaling efficiently. Additionally, the EC2 instance houses an alarm container, completing the IoT layer of the system.

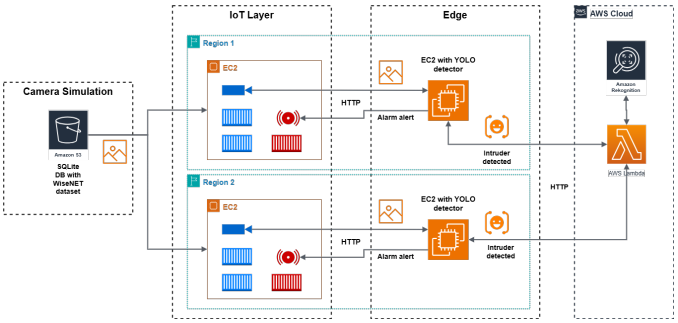

Fig. 1. Example of your architectural diagram.

The edge layer consists of two EC2 instances simulating edge devices in different locations. At the edge, the video data is processed using the YOLO (You Only Look Once) algorithm to detect objects and people. When a person-like object is detected, the relevant data is forwarded to AWS Lambda in the cloud, ensuring the system can scale dynamically based on demand.

The cloud layer integrates with Amazon Rekognition to perform facial recognition, comparing the detected face against a collection of known individuals. If an unknown person is identified, Lambda triggers an alert, which is sent back to the edge device and activates the alarm in the IoT layer. This architecture reduces the workload on the cloud, minimizes latency, and ensures the system can handle large volumes of video data in a responsive and efficient manner, making it a robust solution for surveillance applications.

III. Implementation Details

Provide details about the frameworks and resources used. Justify your decisions carefully.

IV. Evaluation

Evaluation of the response time and scalability (number of devices and traffic) to prove the correctness of your implementation. The more detailed the better.