703088 (2024WS) - PS Distributed Systems
Distributed and Parallel Systems group, University of Innsbruck
Instructors: Juan Aznar Poveda, Stefan Pedratscher, Marlon Etheredge

Final project description

During the execution of this project, you will apply the knowledge acquired during the previous weekly exercises. The outcome of this project should be:

- The implementation of the project according to the specifications presented in this description.
- A presentation of the project, including its implementation and evaluation.
- A technical report that explains your solution.

Work in groups: The project should be done in groups of 2-4 students.

Geo-distributed Surveillance System

In this project, you will develop a distributed Cloud-Edge-IoT surveillance system that triggers alerts upon detecting *unknown* individuals in images captured over time.

Detecting unknown people

For the input stream of images, the system should keep track of known people.

Computing continuum layers

IoT Layer

- Incorporates cameras for image capture.
- Captures images at a default interval (the lower the better).

Edge Layer

- Pre-processes captured images to filter out irrelevant ones (i.e., those that do not contain persons).
- Conducts lightweight pre-processing to minimize unnecessary analysis.
- Forwards only relevant images to the Cloud layer for further analysis.

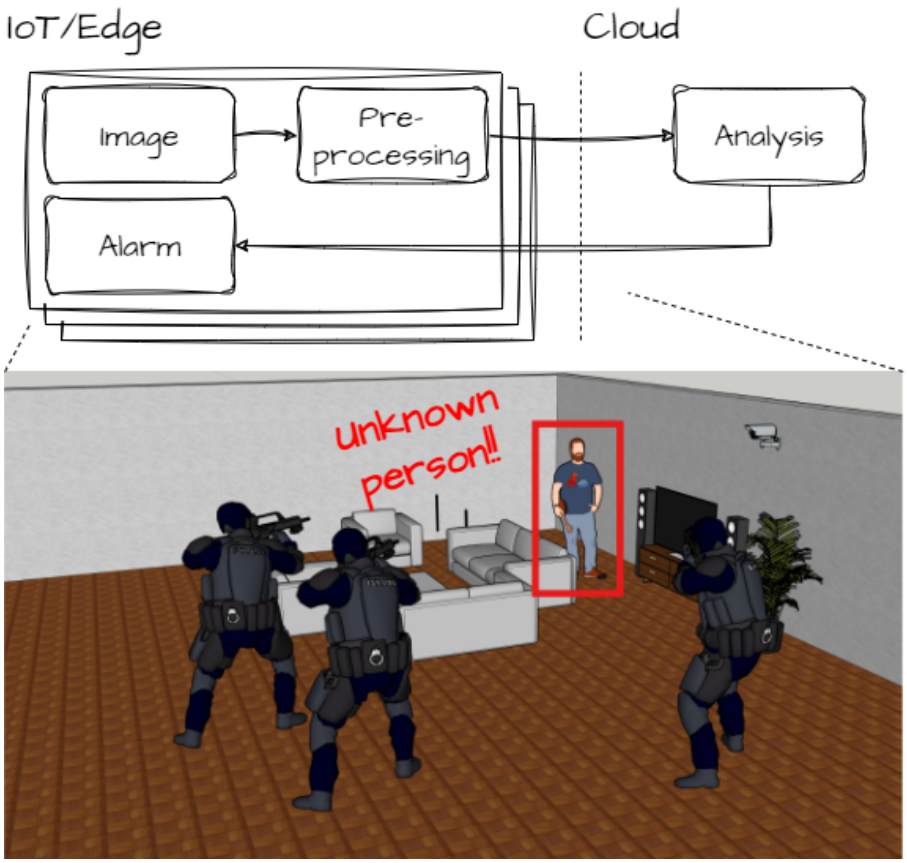
Cloud Layer

- Performs more computationally intensive analysis on the forwarded relevant images.
- Triggers alarms in the IoT layer based on the analysis results.

System description

The figure below shows the basic operation of the proposed system, capturing the basic functionalities required (you can add more!):

- an image is captured from an IoT device (you need to emulate an IoT device using e.g., WiseNET);
- we suggest to implement a service that for example uses the WiseNET videos and can be invoked (e.g., via HTTP) to get the current surveillance image, make sure to include multiple videos to emulate different cameras, and also make sure that time is sufficiently simulated (i.e., suppose 1 minute passes between subsequent invocations, then the video frames returned by the service should be 1 minute in the video apart). For the purpose of reading the video file, you should use an existing library/framework such as OpenCV;
- during Edge pre-processing, lightweight person detection occurs;
- for this purpose, we suggest to use YOLO object detector trained to detect persons, implementations are readily available as open-source projects;
- iff persons are detected, Cloud analysis is initiated;
- a suggestion is to use AWS Rekognition to detect known and unknown people;
- upon detecting an unknown person, alarm devices near involved cameras are triggered (for this you need to implement an alarm service);
- alarms can be disarmed.



Simple example diagram of the targeted surveillance system

Requirements

The implementation should be as realistic as possible in terms data, distribution, resources, and tools employed.

- **Quality of datasets**. Find high quality and realistic datasets, e.g. WiseNET.
- **High distribution**. The application should be highly and appropriately distributed using at least 5 IoT cameras and alarms, 2 Edge devices and 1 Cloud resource.
- **Tools and frameworks**. The way you implement the surveillance system is totally up to you. Be aware that the tools and frameworks will play an important role during the assessment of the project, so try to think well what is a good fit for your application. Maybe it is useful to keep in mind the tools used during the PS exercises.
- **Code quality**.
- **Technical report**. the technical report (5 pages) should be self-contained and justify your implementation rigorously, as well as provide the obtained evaluation results. **Use the template provided in OLAT**. Outline for your report:
  - Short introduction to the problem
  - System architecture (detailed diagram and explanation)
  - Implementation details (e.g., frameworks, resources, etc.)
  - Evaluation of the response time and scalability (number of devices and traffic) to prove the correctness of your implementation.
- **Presentation**. The presentation (10 + 5 minutes QA) should clearly summarize your contributions and use the following structure:
  - Cover slide (Title and the name of the members of the group)
  - System architecture
  - Slide summarizing the datasets, tools, frameworks and resources used (justify why)
  - Live demo
  - Evaluation results

Milestones and deadlines

The project will be tracked through weekly milestones in which you should submit your current progress of the project. You should use the final report template and incrementally improve and complete it based on the following weekly milestones:

- MS1: Architectural diagram
  - Deadline: 28.11.2024, 23:59h
- MS2: IoT implementation
  - Deadline: 05.12.2024, 23:59h
- MS3: Edge implementation
  - Deadline: 12.12.2024, 23:59h
- MS4: Cloud implementation

    - Deadline: 09.01.2025, 23:59h
- MS5: System deployment and evaluation
    - Deadline: 16.01.2025, 23:59h

Final submission

- **Final submission: including code, technical report, and presentation**
    - **Deadline: 22.01.2025, 23:59h**

  The final project will be presented in the last two lab sessions.

Grading

The project will be graded in three parts:

- Advance in weekly milestones (20%)
- Mandatory presentation (40%), graded based on:
    - Content
    - Live demo
    - Explanation of your solution
    - Clarity and correctness
- Technical report (40%)

Recall that the project represents the 50% of the total grade of the PS course. At least 30% of each individual part (i.e., exercises and project) is required to average the grade of both parts and be able to pass. Note that the content of the rest of exercises is needed for the project itself, so we highly recommend you to equally effort in all tasks and do not put some aside.