1. Introduction

Good morning/afternoon. We are group 7 and we are going to present the work done in the development of ARGOS.

2. Index

3. Problem Statement

This product aims to help identify and report threats resorting to a weapon recognition system. Below is a graph showing the video surveillance market growth from 2018 to what is expected in 2025, indicating that there is a lot of potential for products like ARGOS.

4., 5., 6., 7., 8. System Overview

This Figure depicts the system overview, where one can identify

- Raspberry Pi Board, running an inference engine on footage grabbed from a camera module
- The operational Infrastructure, comprised of cloud and Database services, and the Remote Client

9. Hardware Architecture

The interactions between each sub-system are represented in the hardware architecture diagram

10. Software Architecture

These are the software architecture diagrams for each unit in the system.

11. Hardware Specification

The main hardware components of the computational unit are:

- A raspberry pi board
- A light sensor
- A camera module

12. Software Specification: Task Priority

The tasks in the computational unit are organized by priority as represented in this image.

13., 14., 15. Software Specification: Threads

These flowcharts represent the intended behaviour for each thread in the computational unit

16. Envisioned Storage Framework

The general overview of the interactions between ARGOS, the user, and the environment is depicted in this figure. Following this diagram, one intended to implement a local redundancy framework to assure data and metadata delivery in case of communication failure.

17. Software Specification: Thread Timeline

The task flow and basic interactions between them is plotted in the thread timeline. Note that the grey squares represent python code calls within C++ code.

18. Thread Communication (Condition Variables)

This image depicts the communication between the system threads in terms of the condition variables. To simplify the diagram, mutexes have been left out, even though each condition variable is associated with a mutex.

19. Machine Learning: Overview

Here, on the left side is the representation of the ML model's big picture. On the right side is the edge deployment diagram.

20. Machine Learning: Guidelines

On the left, this slide contains the general workflow for training, deployment and testing of the ML model. One ran tensorflow on the gpu to accelerate the training stage. This is represented in the image on the right.

21. Machine Learning: Dataset Division

To correctly set up the model build one must divide the overall dataset into smaller ones, referring to training validation and testing. The overall image pool ended up containing about 330 images. The training set contains 70% of images and the validation and test sets contain 15% each.

22. Machine Learning: Transfer Learning

To take a previously trained model (Quantized SSD-MobileNet) and adapt it to the application, one recurred to Transfer Learning. This form of learning is based on the idea that the learning of a certain task generates some type of knowledge useful for learning another task, related in some sense to the first one.

23. Machine Learning: Model Training and conversion

In this side you can see the model training and conversion to a tflite model.

24. Machine Learning: Loss Graphs

Regarding the loss graphs, the training took about 12 hours to complete, stopping when the loss was consistently below 2.

25. Database: Stages

These diagrams show the progression of the database from the design stage to its creation in SQL

26. Buildroot packages overview

These are the packages needed for the application

27. Unit Tests: Camera

These were the conducted camera unit tests

28. Unit Tests: Local Cloud Storage

These were the tests relating to the Local Cloud Storage

29. Unit Tests: Machine Learning

These were the tests for the inference tests...

30. Unit Tests: Daemon & Device Drivers

... the daemon and the device driver

31. Integration Tests

Here are the integrated tests

32. Final Storage Framework

As mentioned before, the final prototype can store the inferred frames locally and remotely.

33. Unit Tests Results

These are the unit...

34. Integration Tests Results

... and integration test results

35. Gantt Diagram

...