Video Violence Detection

Industrial Applications Project

Francesco Galardi, Michele Meazzini, Andrea Vagnoli

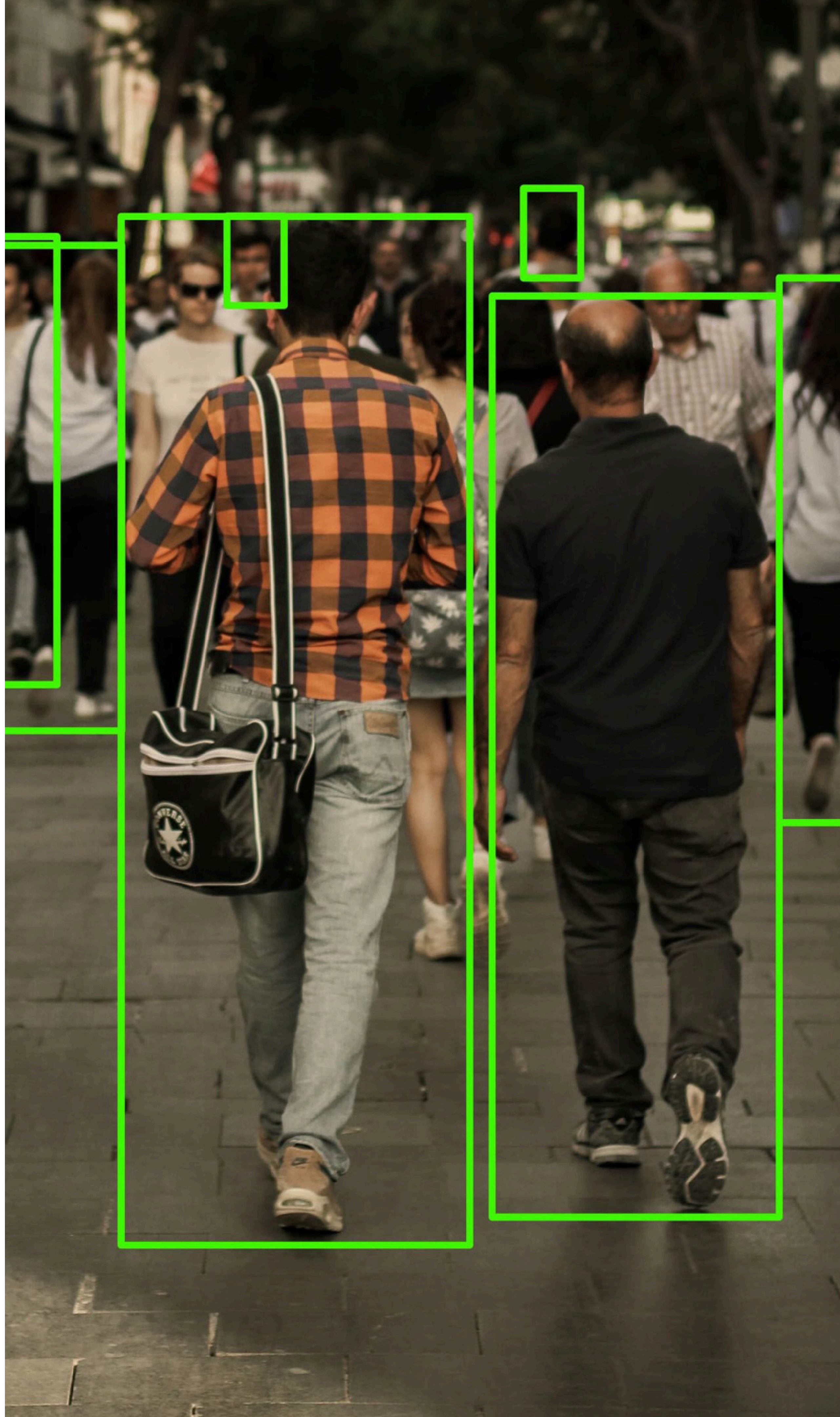
A.A. 2025/2026

Introduction

Ensuring public safety in crowded or poorly supervised environments is an increasingly important challenge in modern urban contexts.

Traditional surveillance systems rely on continuous human monitoring, which is costly and difficult to scale.

This project aims to design a complete automated system for violence detection that balances detection performance with real-world constraints such as latency and computational resources, while analyzing the trade-off between accuracy and latency under different capture settings.



Related Works

Violence detection has been widely studied within the field of Human Activity Recognition and video surveillance, but most existing approaches focus on improving model accuracy in offline settings or fixed-camera scenarios.

In contrast, fewer works address full end-to-end systems that integrate edge devices, real-time data transmission, server-side inference, and human-in-the-loop validation.

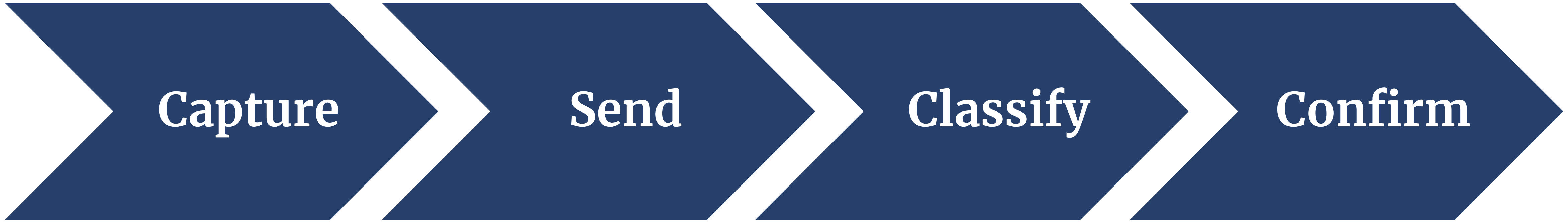
Differently from the studies considered as reference, this project aims to find a practical solution using a low cost edge device that could be used in real contexts.

System Description

The proposed system follows an edge–cloud architecture designed to balance efficiency, scalability, and flexibility.

A lightweight embedded device captures video data from the environment and transmits it to a remote server, which performs inference and exposes the results through a web-based interface that supports user interaction and validation.

The followed pipeline can be summarized with the graph below:



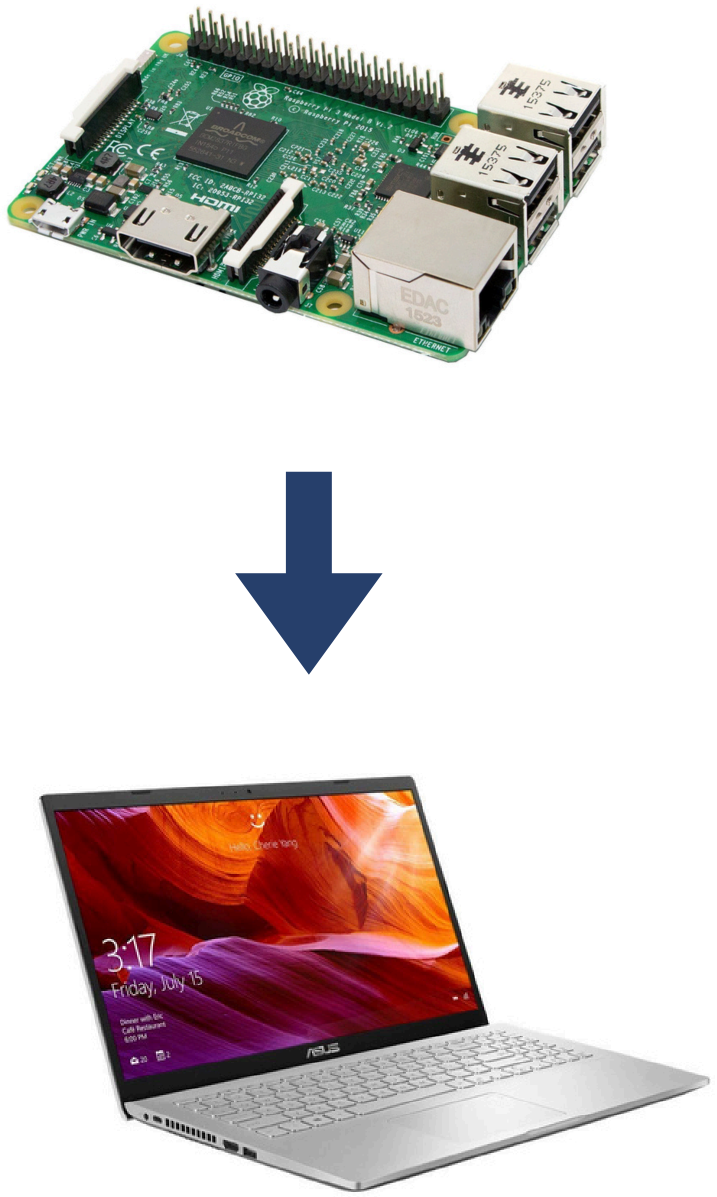
Edge and Server Components

The edge component, implemented on a Raspberry Pi 3, is responsible for video acquisition and data transmission using different operating modes:

- Sending the original video at 10 FPS
- Resampling the video to 1, 2 and 5 FPS

The server manages neural network inference, logging, and system control using GPU acceleration to meet real-time requirements.

This clear separation of responsibilities allows the system to overcome the hardware limitations of embedded devices while remaining scalable.



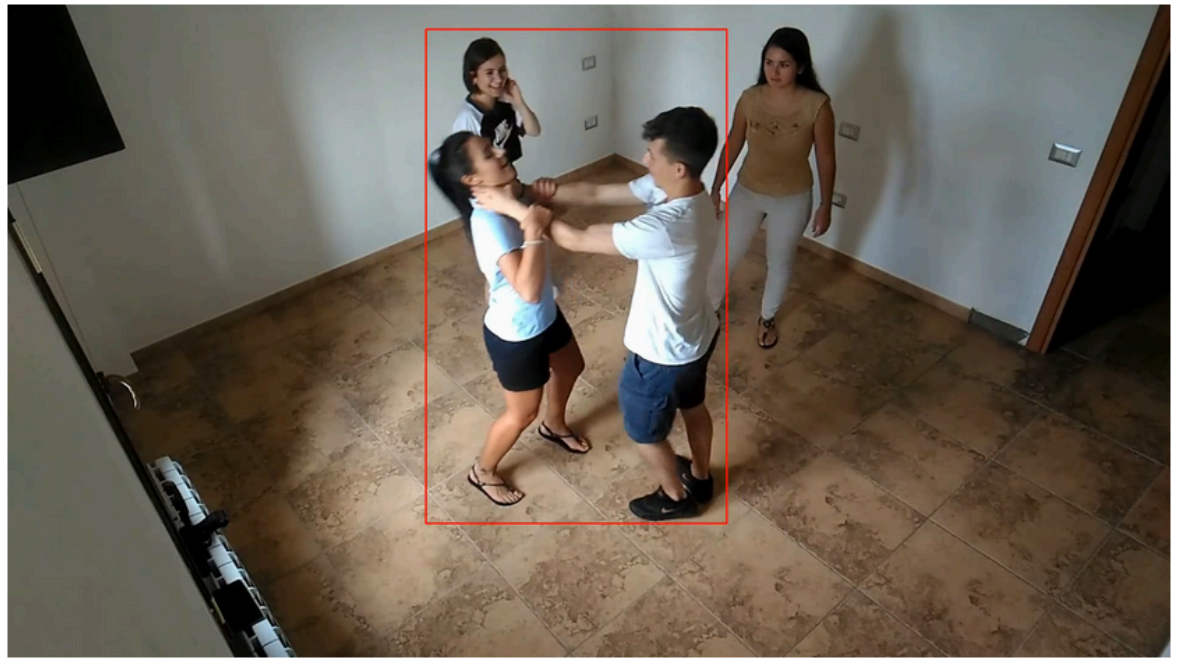
Violence Detection Model

Violence detection is performed using a YOLO-based object detection model executed entirely on the server.

The model processes incoming frames and outputs bounding boxes, confidence scores, and class labels associated with violent behavior.

A video clip is classified as violent if at least one detection exceeds a predefined confidence threshold.

No Violence detected



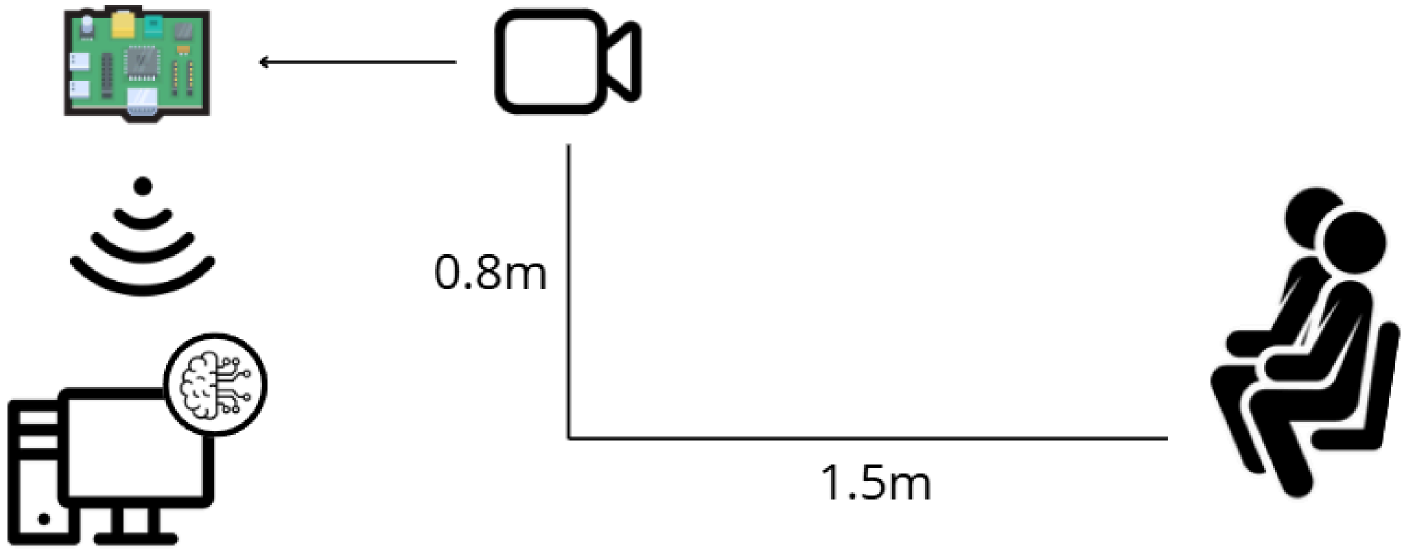
Violence detected

Experimental Setup

Experiments were conducted in a controlled environment designed to simulate the interior of a commercial transport vehicle.

A dedicated dataset of 60 video recordings was collected, evenly divided between violent and non-violent scenarios.

Each recording was evaluated using **multiple frame sampling rates** and **decision thresholds** to analyze system behavior.



Preliminary Considerations

Initial experiments explored the feasibility of performing inference directly on the Raspberry Pi device.

After the trial of many classification models this approach proved impractical due to limited memory, incompatible software dependencies, and excessive inference latency.

As a result, the system architecture was redesigned to perform inference exclusively on the server and the focus of the project became optimizing latency on both clip upload and inference.

The next results analysis will focus on **classification performance** and **end-to-end latency** on different values of frame rate and decision threshold.

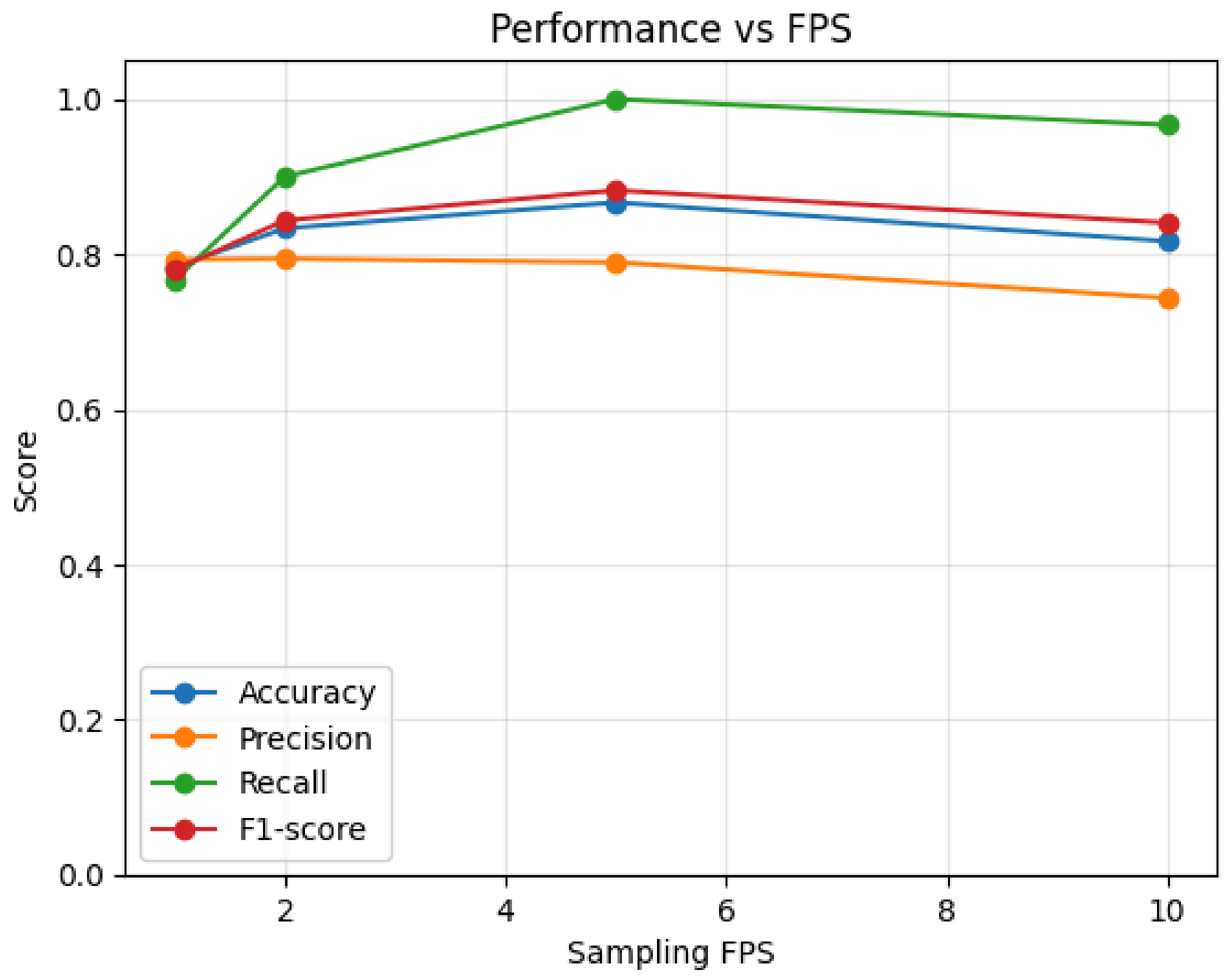
Performance vs Frame Rate

In violence detection, false positives are preferable to false negatives, as the consequences of missed detections can be severe.

With lower frame rates, classification performance improves by increasing the sampling rate.

With higher frame rates this trend is inverted due to the high number of false positives generated.

From this plot it's easy to see that intermediate values of sampling rate are the right choice.

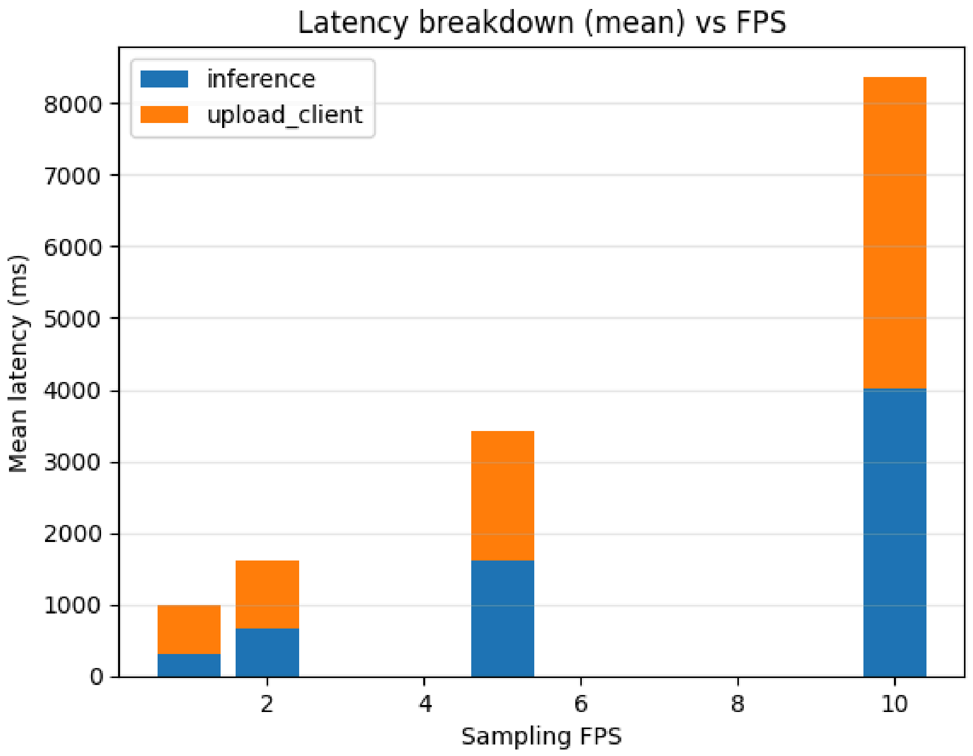


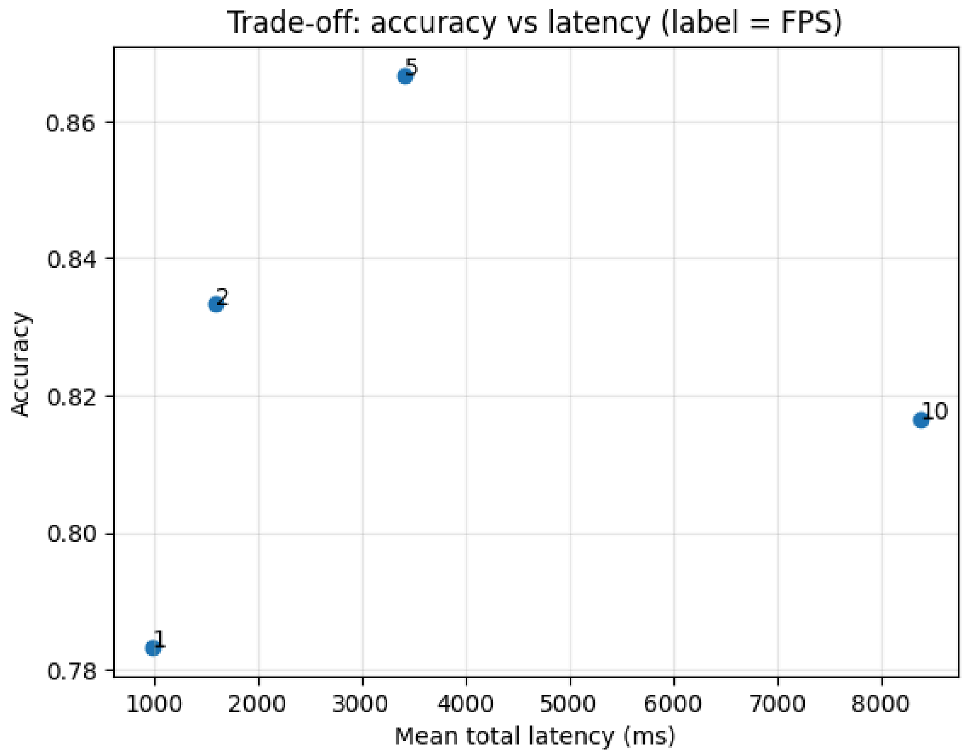
Latency Analysis

The end-to-end latency of the system increases monotonically with the sampling frame rate.

At low frame rates, communication overhead represents a significant portion of the total latency.

At higher frame rates, inference becomes the dominant bottleneck due to increased computational load on the server.



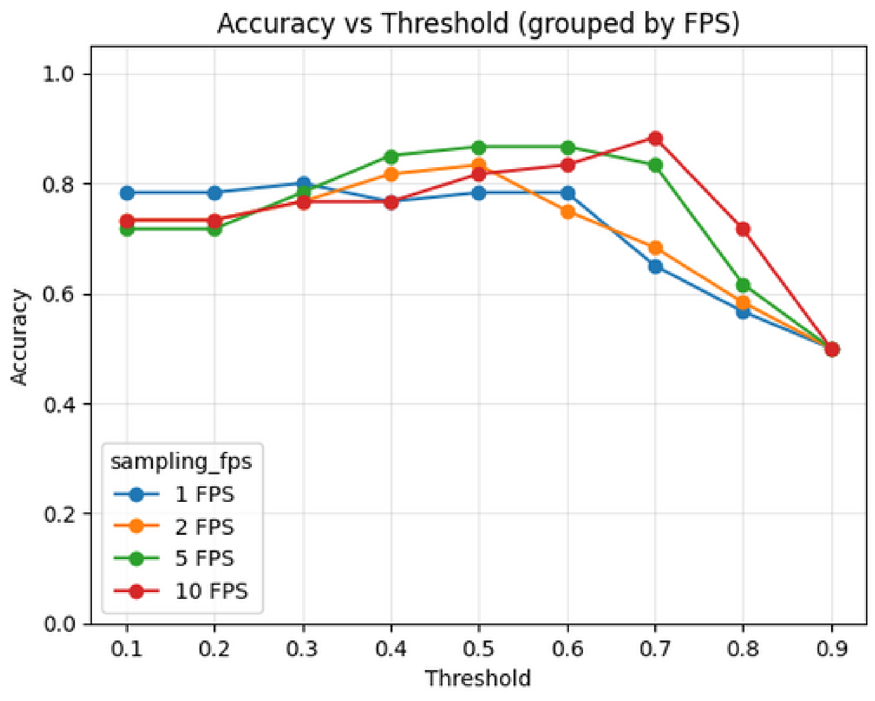


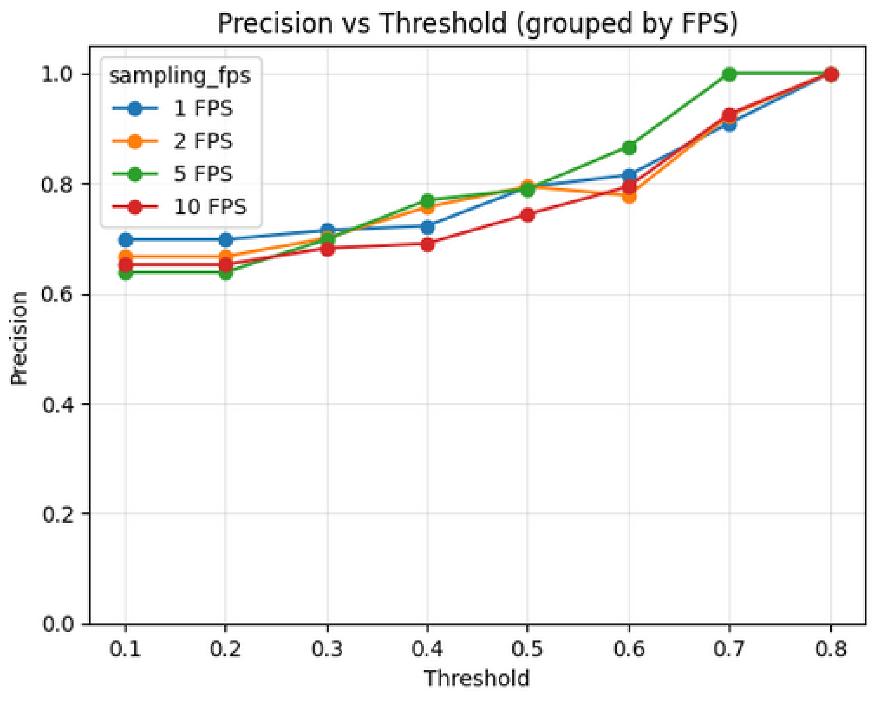
Performance vs Threshold

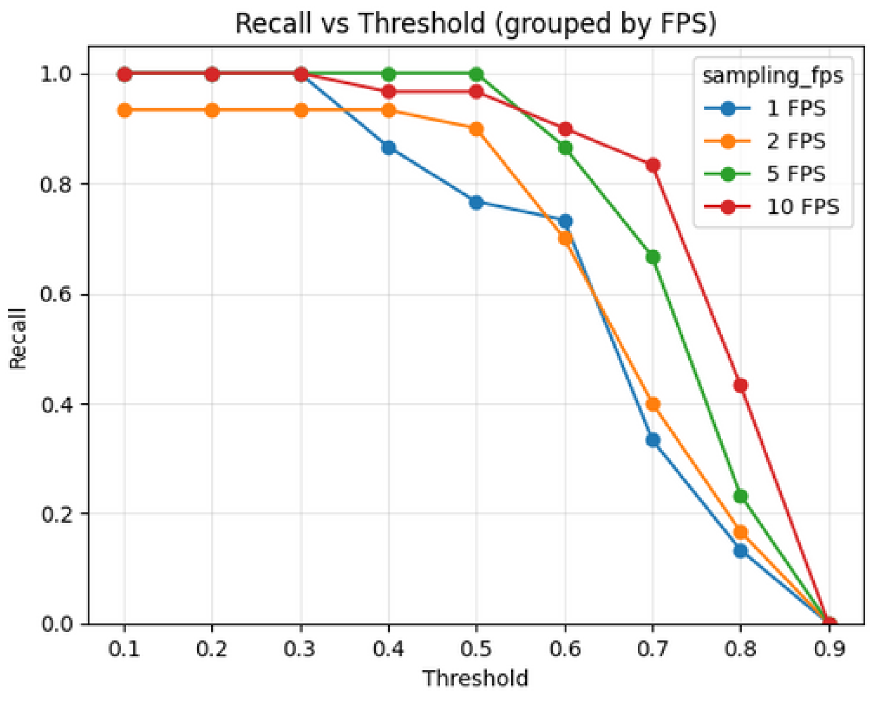
The decision threshold gives a minimum value of confidence above which the clip is classified as Violence.

A threshold that is too low results in a high number of false positives, while a threshold that is too high leads to an excessive number of false negatives.

The classification metrics plots indicate that an intermediate threshold value is preferable.





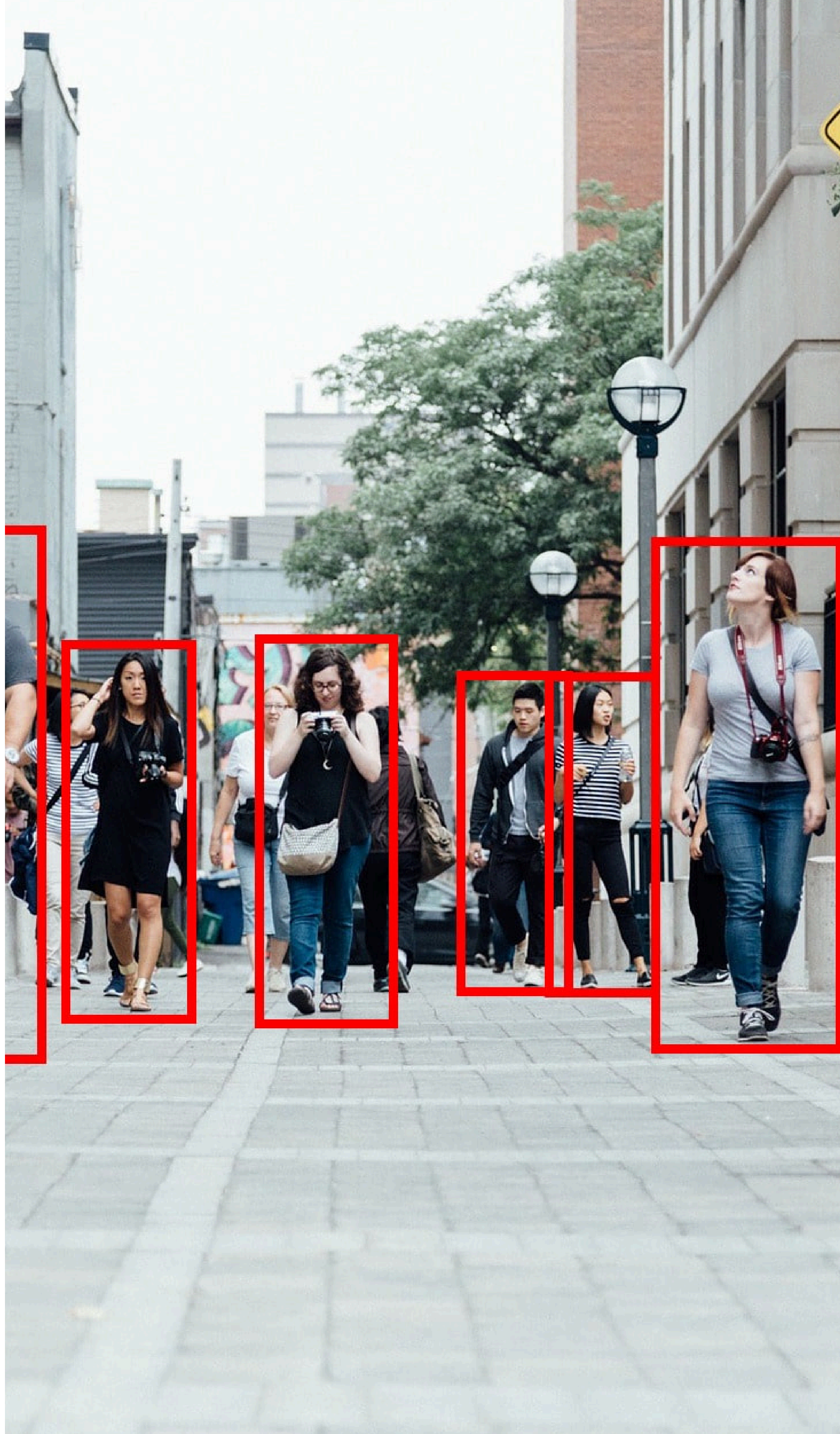


Conclusions and Future Work

This project demonstrates the feasibility of an edge–server architecture for real-time violence detection in realistic scenarios.

The experimental analysis highlights the importance of considering detection metrics as well as system-level constraints such as resources utilization and end-to-end latency.

Future work will focus on larger datasets and on-device inference to further improve robustness and privacy.



Thanks for your attention!