PiWatch: A Real-Time Intrusion Detection System

Cloud Computing
SS 2026
Prof. Ur. Christian Baun
Group 7 (PiWatch)

Department of Computer Science and Engineering, Frankfurt University of Applied Sciences, Germany
Muhiba Maheen • Shehroz Khan • Ajay Somaiya • Riya Chauhan • Aliza • Madhusudan • Dawood • Danih

PROJECT OVERVIEW

This project implements a real-time intelligent surveillance system using edge AI across a Raspberry Pi cluster. Camera data is captured and processed locally at the edge, enabling efficient detection and analysis of security and safety-related events, including intruder presence, fire, weapons, people, and containers associated with potential liquid spills. A web-based dashboard provides insights into recent detections, alerts, annotated images, and system health metrics, while Telegram notifications enable rapid delivery of critical event updates. By combining distributed edge processing, AI-based detection, and real-time monitoring capabilities, the system provides a scalable and responsive solution for intelligent surveillance applications.

INTRODUCTION

Traditional surveillance systems rely on cloud processing, which can introduce network delays and increased bandwidth usage. Our project addresses these limitations by utilizing edge computing, where AI-based object detection is performed locally on Raspberry Pi devices. This approach enables faster decision-making, reduces network dependency, and provides a scalable solution for real-time intelligent surveillance.

PROJECT OBJECTIVES

- Build a real-time edge surveillance system using Raspberry Pi devices
- Detect fire, liquid containers, and people
- Enable Pi4-to-Pi5 communication for image transfer and event processing
- Develop a backend and frontend for events, alerts, and monitoring
- Integrate distributed storage for images and metadata
- Assess cluster performance and scalability using HPL / MPI concepts
- Deploy core services using Kubernetes (k3s)

SYSTEM ARCHITECTURE - WORKFLOW

- Pi4 + AI Camera captures images at short intervals
- Images are sent to the backend API on Pi5
- The backend performs detection/inference and event creation
- Raw images, annotated images, and event metadata are stored
- Important detections generate alerts for the monitoring interface
- Monitoring services collect node health, CPU, RAM, disk, and temperature metrics
- Frontend displays events, images, annotated results, and system health

Distributed Storage

Event images and metadata are handled through distributed storage, with Pi5 as the main processing/storage node and Pi4 used for backup replication. This improves reliability and supports future scalability of the cluster.

TECHNOLOGIES USED

Hardware

- Raspberry Pi5
- Raspberry Pi4
- 8 * Raspberry Pi 3
- Raspberry Pi AI Camera
- Local edge network switch
- SSD or external storage for persistent data
- AI Hat+

Software

- Python
- FastAPI
- React / Vite
- YOLOv8
- Docker
- Kubernetes (k3s)
- Prometheus / Grafana
- SeaweedFS for distributed storage
- Git / GitHub

Monitoring and Alerts

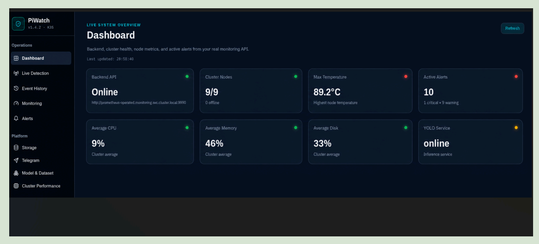
Real-time CPU, memory and storage monitoring using Prometheus and Grafana.

Edge AI - Object Detection

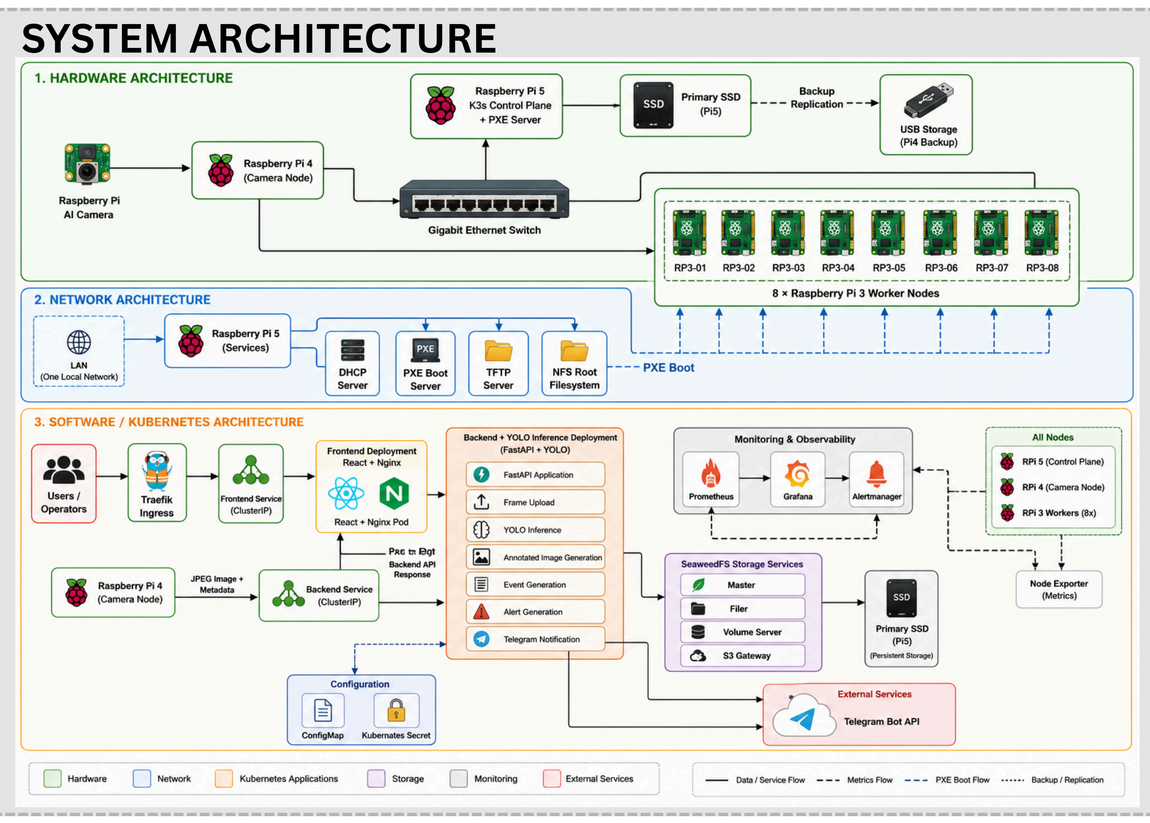
- YOLO analyses images for people, fire, smoke, spills and weapons.
- Generates annotated images with labels, confidence scores and timestamps.
- Event metadata stored with timestamps.
- Results displayed through the PiWatch dashboard.

PIWATCH FRONTEND DASHBOARD

- Presents backend, cluster, YOLO service and node-health status in one interface.
- Displays resource metrics, temperature conditions and active security alerts.
- Provides access to detection history and sends critical event notifications through Telegram.

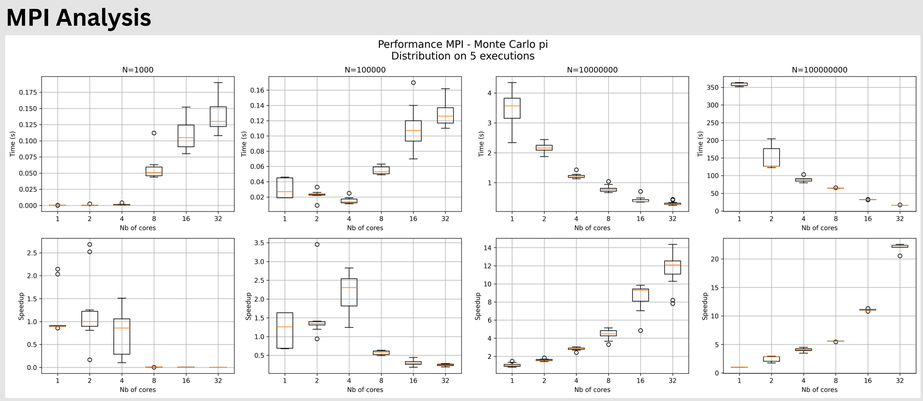


SYSTEM ARCHITECTURE

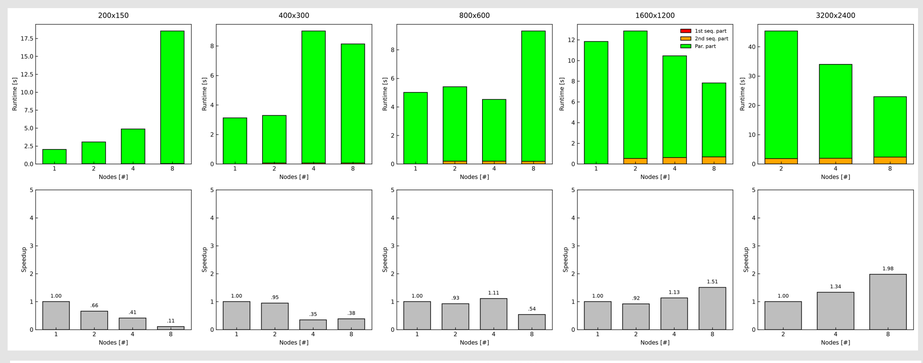


PERFORMANCE GRAPHS

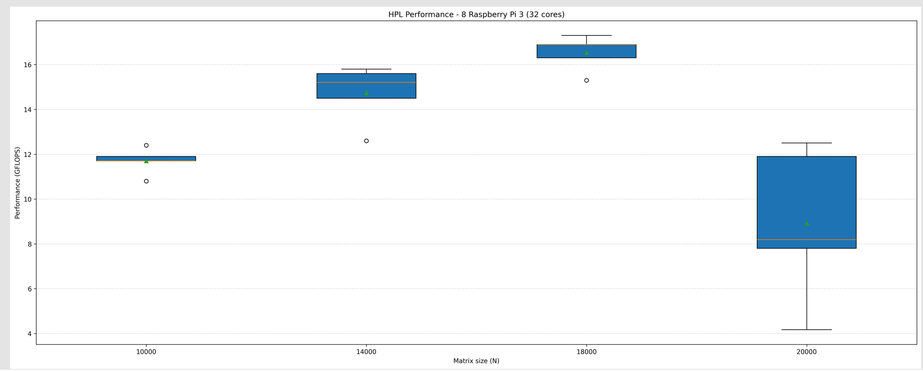
MPI Analysis



- Amdahl's law : fig 1 and 2.
- Gustafson's law: fig 3 and 4.
- Max speed up: 21.5 (using 32 cores)

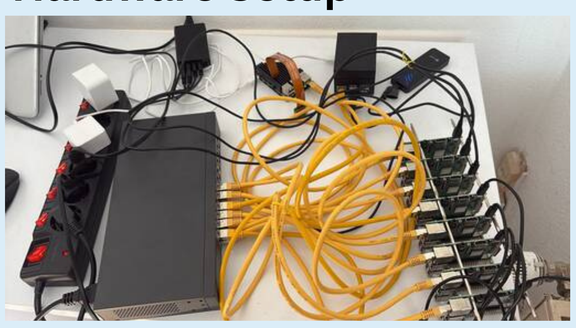


- Amdahl's law : fig 1, 2, 3.
- Gustafson's law: fig 4 and 5.
- Max speed up: 1.98
- Bottleneck: 1 worker for 3200x2400



- Max size: 20000
- Max Gflops : 17 reached using 8 nodes (32 cores) with N = 18000

Hardware Setup



The Raspberry Pi AI Camera is connected via a wired connection to the Raspberry Pi 4 (RP4). The RP4, RP5, and eight RP3 nodes are connected through an Ethernet switch, forming a unified local network.

The RP5 serves as the primary node and is equipped with an AI HAT+ accelerator and an external SSD for additional storage capacity. This setup enables distributed processing across the Raspberry Pi cluster while providing dedicated hardware acceleration and storage resources through the RP5.

and

ML MODEL STATISTICS

Class Name	Precision (How clean detections are)	Recall (How many threats it catches)	mAP50 (General Accuracy)	mAP50-95 (Strict Accuracy)
Container	95.2%	95.5%	97.7%	86.3%
Fire	82.2%	72.3%	79.6%	46.4%
Person	84.4%	77.3%	86.5%	58.1%
Weapon	95.4%	94.7%	96.4%	71.6%
Overall (All Combined)	89.3%	85.0%	90.0%	65.6%

TEAM MEMBERS & RESPONSIBILITIES

Task 1: Infra Setup & PXE Boot - Aliza, Ajay S., Riya C.
Task 2: HPL Benchmarking - Danih A.
Task 3: Cluster Deployment & Scaling - Danih A.
Task 4: Non-MPI Parallelization - Aliza
Task 5: Centralized Monitoring - Riya C.
Task 6: Object Detection & Training - Madhusudan, Danih A.
Task 7: HA Distributed Backend (k3s) - Aliza, Ajay S., Riya C.
Task 8: Presentation Frontend - Shehroz K., Muhiba M.
Task 9: Telegram Notification Gateway - Dawood A.
Task 10: Documentation & Poster - Entire Team