Distributed Fog Computing for Real-Time Surveillance

Joaquín Olivares, Héctor Martínez, Fernando León, José M. Palomares
Dept of Electronic and Computer Engineering
Universidad de Córdoba
Córdoba, Spain
{olivares, el2mapeh, fernando.leon, jmpalomares}@uco.es

***Abstract*— The increasing need for automated surveillance systems has led to the development of intelligent solutions that integrate artificial intelligence and distributed computing. This paper presents a Fog Computing-based surveillance system for real-time weapon detection and face recognition. Unlike traditional cloud-centric systems, the proposed model leverages Edge, Fog, and Cloud layers to optimize processing efficiency, reduce network congestion, and minimize power consumption. Distributing computing tasks across multiple layers significantly reduces transmitted data volume while ensuring fast and reliable threat identification. Experimental results suggest the proposed system can substantially reduce data transmission, preserving high detection accuracy. Adaptive and on-demand activation of the Edge, Fog, and Cloud processing layers contributes to improved efficiency and responsiveness, making the system a promising option for scalable and large-scale surveillance applications.**

Keywords*— *Weapon detection, Face recognition, Distributed computing, Fog computing, Deep Learning.

I. Introduction

Surveillance systems play a crucial role in security and law enforcement. However, conventional monitoring methods require constant human supervision, which is inefficient and prone to errors. Automated surveillance has become feasible with the advent of deep learning and the Internet of Things (IoT) [1]. However, cloud-based solutions suffer from high latency and bandwidth limitations, making them unsuitable for real-time applications. So, it is imperative to optimize data transmissions, reducing unuseful information [2].

In the Internet of Things (IoT) ecosystem, data processing is distributed across three primary computing layers: Edge, Fog, and Cloud [3]. Each of these layers plays a crucial role in managing data efficiently while addressing challenges such as latency, bandwidth usage, and power consumption [4]. Edge computing processes data directly on IoT devices or nearby local servers, significantly reducing latency by handling data close to its source [5]. In Wireless Sensor Networks (WSNs), edge computing is particularly beneficial as it minimizes energy consumption by processing data locally, thereby reducing the need for energy-intensive data transmission [6]. Fog computing is an intermediary between the edge and the cloud, extending cloud capabilities to the local network. It enables efficient data processing and storage closer to the data source, reducing latency and bandwidth usage [7]. This architecture is particularly beneficial for applications requiring real-time analytics and quick decision-making, as it distributes computational tasks across multiple nodes rather than relying solely on distant cloud servers. Cloud computing provides centralized, large-scale data storage and processing capabilities. While offering vast computational resources, it introduces higher latency due to the physical distance from data sources, making it less suitable for time-sensitive applications. However, it remains essential for tasks requiring extensive data analysis, long-term storage, and large-scale machine learning model training.

Computer Vision approaches rely on understanding image structures through techniques like color segmentation, interest points, shape detection, and edge detection [8]. However, their effectiveness highly depends on image quality and angle, making occlusion and noise significant challenges [9]. In contrast, Deep Learning algorithms, based on Neural Networks [10], enable computers to infer features autonomously, allowing them to detect occluded objects more effectively when trained with large datasets.

Hardware accelerators, such as GPUs [11], TPUs [12], PIM [13], and FPGAs [14], are essential for deep learning and image processing due to the high computational demands of these tasks. Neural networks require extensive matrix operations and parallel processing, which traditional CPUs handle inefficiently. Accelerators significantly improve performance by enabling faster training and inference, reducing power consumption, and supporting real-time applications. Their use is critical in tasks like object detection, image recognition, and video processing, where large datasets and complex models demand high-speed computation.

This research introduces a distributed computing approach that integrates Edge, Fog, and Cloud layers to enhance the efficiency of surveillance systems, particularly in detecting weapons and identifying individuals [15]. In the future, Fog surveillance could also be helpful in robot surveillance applications [16], allowing lightweight robots to process intensive deep-learning tasks in the building network infrastructure.

The key contributions of this work include: 1) A multi-layered processing framework optimizing real-time surveillance. 2) Reduction of network congestion through efficient data filtering at Edge and Fog layers. 3) Low-power consumption and scalability for deployment in smart city environments.

ISBN 978-3-903176-72-0

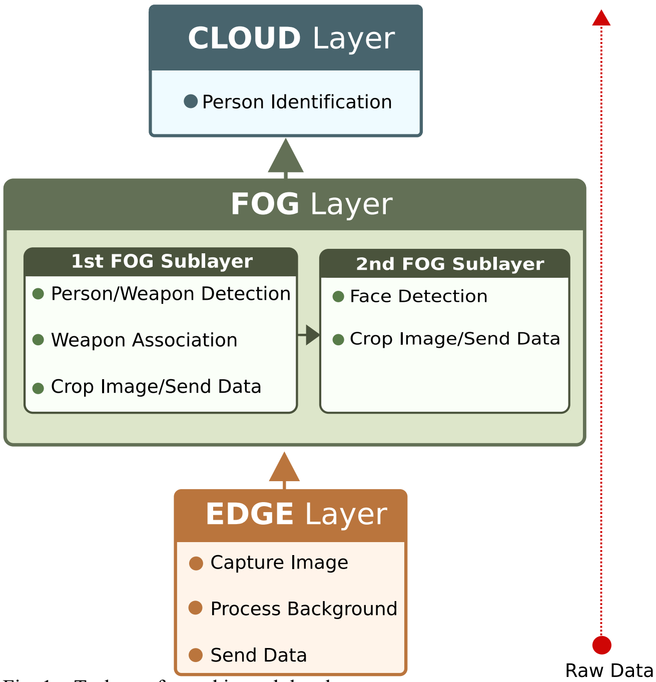

Fig. 1. Tasks performed in each level.

II. Proposed System Architecture

The proposed system consists of three computational layers with distributed computing in Edge, Fog, and Cloud. The information sent by a surveillance system is very sensitive [17], because of this, the lightweight GS3 cybersecure protocol is implemented for communications [18]. Each layer, shown in Fig. 1, is responsible for the following specific tasks to maximize efficiency:

A. Edge layer

The Edge layer captures images using camera sensors and performs preliminary Background Subtraction. BS works by subtracting the current frame from a background, isolating dynamic objects based on scene characteristics. Filters out frames without significant changes to reduce unnecessary processing in the following Fog layer.

B. Fog layer

The Fog layer is structured in two sublayers. The first one implements deep learning-based detection algorithms to identify people and weapons. It extracts the relevant regions of interest (ROIs) linking detected weapons to individuals. Finally, it sends only the relevant ROIs to the second layer to minimize data transmission. Every device in the first Fog sublayer can service several Edge camera devices. The second layer receives the armed person's ROI and extracts its face, sending it to the Cloud layer for people identification. The Cloud contains a large biometrics database, for people's identification. Every device in the second Fog sublayer can service several first Fog sublayer devices.

YOLO v5 [19] was chosen for detection because it offers a good balance between embedded GPU performance requirements and detection quality. The people detection was performed using the pre-trained YOLOv5l6 set, and the weapon detection was trained with 15600 images.

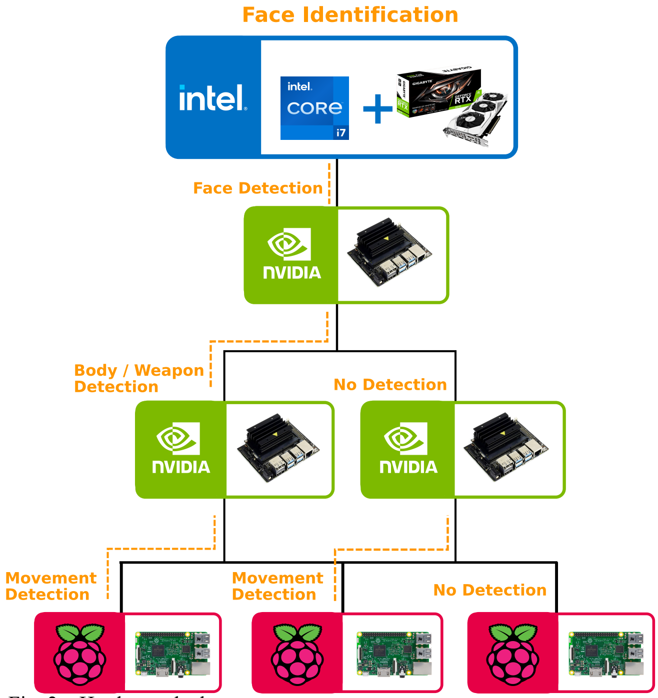

Fig. 2. Hardware deployment.

C. Cloud layer

It performs facial recognition to verify the identity of detected individuals, determining whether they are authorized to carry a weapon, and sends alerts only when unauthorized armed persons are detected. The face recognition is performed in the Cloud using the Geitgey library [20].

III. Experimentation

The system was tested under real-world scenarios using a distributed hardware setup, shown in Fig. 2:

- Edge Layer: Raspberry Pi 3 (RPi3) for initial image processing of Background Subtraction (BS).
- Fog Layer: NVIDIA Jetson Nano devices for weapon, person, and face detection.
- Cloud Layer: High-performance server for identity verification.

We developed five experimental test scenarios:

- Case 1: Background Detection: No persons in the frame, ensuring no unnecessary data transmission.
- Case 2: Single Person Without Weapon: Detection of individuals without alert.
- Case 3: Single Person With Weapon: Processing and transmitting of his face to the Cloud layer.
- Case 4: Multiple Persons Without Weapons: Detection of multiple individuals without weapon alerts.
- Case 5: Multiple Persons With Weapons: Processing and transmitting of multiple faces to the Cloud layer.

Fig. 3. Images and sub-images sent.

In Fig. 3, we present the image transferred in each processing level: At left, the image captured by the Edge camera is sent to the first Fog level, which detects an armed person, extracts his bounding box, and sends it to the second Fog level; in the middle, the second Fog level detects and extracts the face, sending it to the Cloud. Results depend strongly on the scene, so we proposed those five scenarios. Data transaction percentages are shown in Table I. When there are no changes in the background, no data is sent. When people are in the scene without movement, there is a notable reduction of data sent to Fog. There is a very low probability of appearing weapons in real life, so transfers are very sparse. Cloud only receives the bounding box with the faces of armed people, representing 2% of the initial data.

TABLE I. DATA TRANSACTIONS (%)

	Rpi » 1st Nano	1st » 2nd Nano	2nd Nano » Cloud
Case1	0.00	0.00	0.00
Case2	7.23	0.001	0.00
Case3	16.13	0.77	0.02
Case4	24.31	0.01	0.00
Case5	17.10	0.58	0.02

IV. CONCLUSION AND FUTURE WORK

This research presents a useful and easily scalable Fog surveillance system designed for real-time weapon detection and face recognition. It demonstrated high efficiency in filtering unnecessary data, achieving significant reductions in network load: In cases where no weapons were detected, data transmission was reduced to nearly zero, whereas in threat scenarios, only 2% of the original data was transmitted to the Cloud for identity verification. Since transmissions are drastically reduced, the distributed computing model significantly reduces power usage. Edge and Fog layers are dynamically activated only when needed, conserving energy. This multi-layered approach ensures that data is processed at the most appropriate level, balancing computational efficiency with energy consumption and network performance. Experimental validation confirms the system's efficiency in reducing data transmission while maintaining detection accuracy. Future work will focus on integrating advanced AI models for improved recognition and deploying the system in large-scale smart city environments.

ACKNOWLEDGMENT

This work has been partly supported by the grants RTI2018-098371-B-I00 and 1380974-F FEDER.

REFERENCES

[1] D. Mohanapriya and K. Mahesh, "An efficient framework for object tracking in video surveillance," in The Cognitive Approach in Cloud Computing and Internet of Things Technologies for Surveillance Tracking Systems. Elsevier, pp. 65–74, 2020.

[2] F. León-García, et al. "Data communication optimization for the evaluation of multivariate conditions in distributed scenarios," IEEE Access, vol. 7, pp.123473–123489, 2019.

[3] Z. Hao, E. Novak, S. Yi, and Q. Li, "Challenges and software architecture for fog computing," IEEE Internet Computing, vol. 21, no. 2, pp. 44–53, 2017.

[4] J. M. Castillo-Secilla, et al. "Experimental procedure for the characterization and optimization of the power consumption and reliability in zigbee mesh networks," Int. Conf. on Advances in Mesh Networks, pp. 13–16, 2010.

[5] Shreshth Tuli, et al. "AI augmented Edge and Fog computing: Trends and challenges," Journal of Network and Computer Applications, vol. 216, 103648, 2023.

[6] F. León-García, J. M. Palomares, and J. Olivares, "D2r-ted: Data—domain reduction model for threshold-based event detection in sensor networks," Sensors, vol. 18, no. 11, p. 3806, 2018.

[7] Zolghadri, M., Asghari, P., Dashti, S.E., Hedayati, A.. "Resource allocation in Fog–Cloud Environments: State of the ar,." J. Netw. Comput. Appl. 227, 103891, 2024.

[8] R. Palomar, et al. "Parallelizing and Optimizing LIP-Canny Using NVIDIA CUDA." LNAI 6098. Springer, pp 389–398, 2010.

[9] J. Sánchez, N. Monzón, and A. Salgado, "An Analysis and Implementation of the Harris Corner Detector," Image Processing On Line, vol. 8, pp. 305–328, 2018.

[10] Zhao, Z.Q., et al. "Object detection with deep learning: A review". IEEE Trans. Neural Netw. Learn. Syst. 30 (11), 3212–3232. 2019

[11] R. Palomar, et al. "High-performance computation of bézier surfaces on parallel and heterogeneous platforms", Int. J. Parallel Program. 46 (6) , pp. 1035–1062. 2018

[12] K.-C. Hsu and H.-W. Tseng, "Accelerating Applications using Edge Tensor Processing Units," SC21: International Conference for High Performance Computing, Networking, Storage and Analysis, St. Louis, MO, USA, pp. 1–16, 2021.

[13] G. F. Oliveira, et al. "Proteus: Achieving High-Performance Processing-Using-DRAM via Dynamic Precision Bit-Serial Arithmetic". arXiv preprint arXiv:2501.17466. 2025.

[14] J. Olivares, et al. "Minimum Sum of Absolute Differences Implementation in a Single FPGA Device". Field Programmable Logic and Application. LNCS vol 3203, pp. 986–990, 2004.

[15] H Martinez, et al. "Distributed Fog computing system for weapon detection and face recognition," Journal of Network and Computer Applications, vol 232, 104026, 2024.

[16] F. Espinosa, et al. "ALCOR Project: Contributions to Optimizing Remote Robot Guidance in Intelligent Spaces," Revista Iberoamericana de Automatica e Informatica Industrial, vol 15, 4, pp. 416–426, 2018.

[17] F. Alcaraz, et al. "Lightweight method of shuffling overlapped data-blocks for data integrity and security in WSNs," Computer Networks, vol 199, pp. 108470, 2021.

[18] F. Alcaraz, et al. "GS3: A Lightweight Method of Generating Data Blocks With Shuffling, Scrambling, and Substituting Data for Constrained IoT Devices," in IEEE Internet of Things Journal, vol. 11, no. 15, pp. 25782-25801, 2024.

[19] Jocher, G., 2021. YOLOv5. GitHub https://github.com/ultralytics/yolov5.

[20] Geitgey, A., 2020. face-recognition 1.3.0: Recognize faces from Python or from the command line. pypi.org, 20-Feb-2020. [Online]. Available: https://pypi.org/project/face-recognition/