

Article

Pedestrian Abnormal Behavior Detection System Using Edge–Server Architecture for Large–Scale CCTV Environments

Jinha Song  and Jongho Nang *

Department of Computer Science and Engineering, Sogang University, Seoul 04107, Republic of Korea; jinhasong@sogang.ac.kr

* Correspondence: jhnang@sogang.ac.kr

Abstract: As the deployment of CCTV cameras for safety continues to increase, the monitoring workload has significantly exceeded the capacity of the current workforce. To overcome this problem, intelligent CCTV technologies and server-efficient deep learning analysis models are being developed. However, real-world applications exhibit performance degradation due to environmental changes and limited server processing capacity for multiple CCTVs. This study proposes a real-time pedestrian anomaly detection system with an edge–server structure that ensures efficiency and scalability. In the proposed system, the pedestrian abnormal behavior detection model analyzed by the edge uses a rule-based mechanism that can detect anomalies frequently, albeit less accurately, with high recall. The server uses a deep learning-based model with high precision because it analyzes only the sections detected by the edge. The proposed system was applied to an experimental environment using 20 video streams, 18 edge devices, and 3 servers equipped with 2 GPUs as a substitute for real CCTV. Pedestrian abnormal behavior was included in each video stream to conduct experiments in real-time processing and compare the abnormal behavior detection performance between the case with the edge and server alone and that with the edge and server in combination. Through these experiments, we verified that 20 video streams can be processed with 18 edges and 3 GPU servers, which confirms the scalability of the proposed system according to the number of events per hour and the event duration. We also demonstrate that the pedestrian anomaly detection model with the edge and server is more efficient and scalable than the models with these components alone. The linkage of the edge and server can reduce the false detection rate and provide a more accurate analysis. This research contributes to the development of control systems in urban safety and public security by proposing an efficient and scalable analysis system for large-scale CCTV environments.

Keywords: pedestrian abnormal behavior analysis; edge–server architecture; edge computing; CNN; 3D CNN



Citation: Song, J.; Nang, J. Pedestrian Abnormal Behavior Detection System Using Edge–Server Architecture for Large–Scale CCTV Environments. *Appl. Sci.* **2024**, *14*, 4615. <https://doi.org/10.3390/app14114615>

Academic Editor: João M. F. Rodrigues

Received: 3 April 2024
Revised: 6 May 2024
Accepted: 23 May 2024
Published: 27 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the number of closed-circuit television (CCTV) cameras installed to ensure pedestrian safety has increased dramatically, so has the monitoring workload for personnel. Initially, this challenge was addressed by increasing staff numbers. However, as the number of CCTVs continues to grow, it is becoming increasingly difficult to accurately identify and respond quickly to pedestrian safety issues in fast-paced urban environments by simply increasing the number of personnel. To address these challenges, recent research has focused on developing intelligent CCTV services that leverage advanced computing technologies and deep learning algorithms to detect and analyze real-time pedestrian anomalies in CCTV footage. The current research trends are primarily directed toward leveraging edge computing technologies to reduce the server’s data processing load and implement systems that can react in real time. This approach minimizes the latency across the network, enables more efficient data processing, and improves the overall performance of the CCTV system. However, these systems are primarily rule-based, which poses the limitation

that their performance can degrade due to environmental factors such as changes in the shooting angle or lighting conditions. In addition, despite advancements in intelligent CCTV, existing analytics systems cannot adequately address the problem of limited server processing capacity as the number of connected CCTV cameras increases.

To address these issues, this study proposes a real-time pedestrian abnormal behavior detection system based on an edge–server structure that ensures efficiency and scalability. The proposed system detects pedestrian abnormal behavior by linking edge devices and GPU servers. For the linkage of edge devices and GPU servers, the integration control server (ICS) is adopted; this is a two-stage analysis method that allows the GPU server to analyze the temporal section of the CCTV when pedestrian abnormal behavior occurs. The system is designed to balance the load on the server and minimize the network transmission data. This approach is intended to make the system scalable and easily adaptable to different environments and requirements. A similar approach is adopted in existing research [1,2]. In [1], the authors describe a method that analyzes the edge device and then integrates the results into the cloud. However, these methods can be less accurate because they rely on analytics results from the edge and do not utilize high-performance analytics models. In [2], the researchers also layer multiple machine learning models to improve the prediction performance and reliability. However, this method may be limited by the computing environment because it requires the hierarchical division of the analytical models depending on the computing environment. To overcome these problems, the system proposed in this paper proposes an efficient two-stage analysis method that first detects abnormal behavior at the edge and then analyzes the detected data in detail on the server. At the edge device, an analysis model with low computing power but a high recall rate is utilized to analyze pedestrian abnormal behavior events as much as possible, and at the GPU server, an analysis model with high precision is utilized because it analyzes only the sections where pedestrian abnormal behavior is detected at the edge device. The proposed system first analyzes pedestrian abnormal behavior on the edge device. When pedestrian abnormal behavior is detected on the edge device, the information about the detected camera and detection section is sent to the GPU server through the ICS for further analysis. The edge device uses DNN-based object detection results to analyze pedestrian abnormal behavior in the rule base. The GPU server analyzes only the temporal section of the CCTV when pedestrian abnormal behavior is detected on the edge device, and it uses a deep learning-based mechanism to detect pedestrian abnormal behavior via DNN-based object detection and pose estimation model analysis. The initial analysis at the edge stage optimizes the amount of data sent to the server to reduce the network load and maximize the analysis efficiency of the entire system. Through experiments, we demonstrate that the system with the proposed edge–server architecture is more efficient and scalable than the use of the edge and server alone and that it can reduce the false positive rate of the pedestrian anomaly detection model. The proposed edge–server architecture-based pedestrian abnormal behavior detection system is innovative in that it reduces the amount of data transmission, shortens the reaction time of the system, and secures scalability by analyzing only the necessary segments on the server, unlike existing server analysis systems that allocate the resources of the analysis server even at unnecessary times by analyzing the connected CCTVs at all times. In addition, the contribution of such a system is that it can improve the accuracy of real-time data processing in large networks and reduce operating costs. These innovations and contributions of the proposed system are expected to secure the safety of citizens in future cities, reduce the workload of monitoring personnel in large-scale CCTV environments, and contribute to securing the safety of pedestrians.

Section 2 reviews the current research on CCTV video analysis systems and pedestrian abnormal behavior detection. Section 3 outlines the application scenario and describes the proposed system. Section 4 evaluates the proposed system through a comparative analysis in terms of efficiency and scalability by comparing the experimental environment, dataset, evaluation metrics, performance for the edge and server, and performance after edge–server linkage. Section 5 discusses the challenges encountered, potential vulnerabilities, and areas

for improvement in the models. Finally, Section 6 concludes the paper by summarizing the research findings and highlighting the contributions of this work.

2. Related Work

Section 2.1 describes the latest research trends in the application of deep learning, Section 2.2 describes the CCTV video analysis system related to this paper, and Section 2.3 introduces the pedestrian abnormal behavior detection method in this paper.

2.1. Recent Trends in Applied Deep Learning Research

In recent years, deep learning techniques have made remarkable progress in various fields, contributing significantly to extracting meaningful information from complex data and making predictions. This section explores how these techniques are applied to solve problems in specific domains, most notably in image recognition, natural language processing, and time series data analysis. Reference [3] uses a neural network based on the Levenberg–Marquardt algorithm to maximize the efficiency of DC–DC converters and extend their lifetime. The technique estimates and manages the cumulative damage in each cell of the converter, reducing maintenance costs and optimizing the entire system’s performance. Reference [4] proposes a novel method for predicting remaining useful life (RUL) across different domains, specifically using a federated learning framework that enables highly efficient learning from distributed data sources while maintaining privacy. This approach helps to build more accurate prediction models by utilizing data collected from different environments. In reference [5], a feature decouple and gated recalibration network (FDGR–Net) is proposed for landmark detection in medical images. This method independently processes the characteristics of landmarks through the feature decouple module (FDM), enhances the information of target landmarks through the gated feature recalibration module (GFRM), and minimizes the disturbance of non-target landmarks. This enables landmark detection in medical images with higher accuracy than existing methods. Meanwhile, research is also emerging on efficiently applying deep learning models to systems when applying them. The study [2] also layers multiple machine learning models to improve prediction performance and reliability. This provides a new solution to the multi-layer response regression problem and suggests adding a new level of accuracy and efficiency to the reliability assessment of complex systems. In [1], we propose a method to leverage LSTM models to predict object appearance patterns in a hierarchical edge computing environment for analyzing surveillance video in systems that analyze surveillance video, such as CCTV. The proposed method is proper when object occurrences are infrequent because it saves GPU memory by loading the model only when needed and releasing the model when no objects are detected. Research studies utilizing deep learning models in these research fields have made remarkable progress and play an important role in designing complex system structures and data processing. The system structure proposed in this paper can serve as the basis for applying these latest technologies, enabling faster, more accurate, and more energy-efficient video analysis. It is important to continue integrating and advancing the latest research trends in this field, which will play a major role in meeting future challenges and improving existing methods.

2.2. Surveillance Video Analysis System

In recent years, CCTV video analytics systems have been widely used for security and surveillance, leveraging advanced computing technologies and deep learning algorithms. In particular, edge-based processing technologies are contributing to enhancing the safety and security of cities through their advanced computing capabilities; they overcome the limitations of centralized data processing and enable real-time data analysis. Recent research has focused on utilizing edge devices and applying artificial intelligence models to video analytics systems. For example, reference [6] proposes an anomaly recognition system for edge computing using Raspberry Pi-based edge devices. By preprocessing and analyzing video data at the network’s edge, the system improves real-time responsiveness and re-

duces the burden of data transmission on the central server. Similarly, ref. [7] introduces NAVIBox, a real-time vehicle-pedestrian hazard prediction system. The system focuses on detecting and predicting potential collisions between pedestrians and vehicles in a real-time edge vision environment. In reference [1], a long short-term memory (LSTM) [8] model on an edge computing system is used to recognize and predict the appearance patterns of objects in surveillance videos. In particular, it applies optimization to improve the system's efficiency and real-time processing power. It proposes an efficient analysis method using edge devices, reducing the bandwidth use and improving the response time by processing the videos on edge devices and sending only the results to the cloud for final analysis. The aim of the research in [6,7] is to embed advanced computing functions in edge devices to overcome the limitations of centralized data processing and enable real-time data analysis. The system proposed in this research uses edge devices capable of real-time data analysis to distribute the load concentrated on the analysis server by connecting edge devices and GPU servers. Another study that utilizes AI models in a centralized video analytics system is [9], which integrates AI into CCTV systems to comprehensively evaluate the effectiveness of intelligent video surveillance in community spaces. The study discusses the various benefits of integrating AI into CCTV systems, including a methodology for the development and integration of AI-based video analytics algorithms, case studies of real-world applications of AI-integrated CCTV systems in community spaces, a performance evaluation and analysis of AI-enhanced surveillance systems, and ethical and legal considerations related to AI-integrated surveillance, such as privacy and data security. Reference [10] explores the application of deep learning techniques to improve the performance of video surveillance systems in financial institutions such as banks. The work focuses on how deep learning technology can reduce security vulnerabilities and improve crime prevention and early detection. It introduces experimental studies to verify the performance of deep learning-based video surveillance systems in natural banking environments, as well as optimization methodologies for deep learning models to improve the performance of video surveillance systems. It highlights the potential for deep learning-based video surveillance systems to become the new standard for bank security and highlights the importance of continued research and technology development in this area. The system proposed in this paper performs the final analysis on the GPU server only on the CCTV videos that have been filtered through pre-analysis on the edge device, similar to the research in [9,10], which applies AI models to a system that analyzes CCTV videos. However, recent research on CCTV video analytics systems has focused on real-time pedestrian anomaly detection using advanced computing technologies and deep learning algorithms. In particular, studies utilizing edge computing based on edge devices reduce the burden of data transmission on the central servers by reducing the load on the network and improving the processing speed. The system proposed in this research extends these existing works even further. It maximizes the efficiency and scalability of real-time analysis by linking the edge devices with the GPU servers. The system performs an initial analysis on the edge devices. It sends only filtered data to the GPU server for high-precision analysis, which increases the data processing efficiency and provides a scalable network structure. This approach revolutionizes the management of large-scale CCTV networks and can potentially improve pedestrian safety while significantly reduce the burden on monitoring personnel.

2.3. Pedestrian Abnormal Behavior Detection in Surveillance Videos

Pedestrian anomaly analysis plays an important role in urban safety and public security and is constantly evolving with advances in real-time detection and response techniques. This research uses advanced computing technologies and deep learning algorithms to detect and analyze real-time pedestrian anomalies. This research proposes a novel system that maximizes the efficiency and scalability of real-time analysis by linking edge devices and GPU servers. The system performs initial data processing at the edge and sends only filtered data to the GPU server, increasing the efficiency of data processing and minimizing the network load. In the proposed system, the pedestrian anomaly analysis

is initially performed on the edge device and finally on the GPU-equipped server, so a pedestrian anomaly analysis method is required for application to the edge device, and a pedestrian anomaly analysis method must be applied to the GPU-equipped server. Therefore, in this section, we introduce related works that are relevant to the edge devices and GPU-equipped servers in the analysis system proposed in this study. First, there are three studies on pedestrian anomaly analysis methods that can be used on edge devices, i.e., refs. [11–13], both of which use the object detection results of the YOLOv7 [13] model and propose a method for the analysis of pedestrian anomalies in the rule base. Reference [11] proposes a real-time assault event detection method based on object tracking to detect assaults in CCTV images. Reference [12] focuses on the real-time detection of fall events. These two studies propose efficient algorithms and system structures for the real-time processing of CCTV videos based on object detection. Their efficiency and accuracy are verified through specific case studies that identify assaults and falls. Meanwhile, studies on pedestrian abnormal behavior analysis methods applicable to GPU-equipped servers are described in [14–17]. Reference [14] proposes a lingering behavior detection method based on pedestrian activity area classification, which dynamically calculates pedestrian activity areas to classify lingering behavior into three categories and uses an algorithm that can accurately and robustly detect different types of lingering behavior without complex trajectory computation. The authors in [15] propose a 3D convolutional neural network-based method for the more accurate detection of anomalous behavior in sparsely populated environments. The study develops a novel analysis framework that can effectively identify anomalous behavior, even in relatively sparsely populated environments, by considering temporal and spatial features. Reference [16] focuses on violence detection based on spatial and temporal features. By utilizing 3D convolutional neural networks, we develop a methodology to detect and predict violent behavior in real time, especially in public spaces, which can contribute to the early detection of and response to violent incidents. Reference [17] explores the use of deep neural networks based on pose estimation to detect falls in human objects. The research focuses on developing high-performance algorithms to analyze human posture in real time and accurately identify abnormal movements such as falls. While these methods may perform well when processing a single video stream, without considering efficiency and scalability, they may not be appropriate for the processing of multiple video streams. Pedestrian abnormal behavior analysis methods applicable to edge devices can process one or more CCTV video streams, depending on the number of frames per second required for pedestrian abnormal behavior analysis and the number of frames per second that the edge device can process. Even centralized pedestrian abnormal behavior analysis, using a GPU-equipped server, can process one or more CCTV video streams, depending on the number of frames per second required for pedestrian abnormal behavior analysis and the number of frames per second that the GPU server can process. Edge devices are cheaper than GPU servers. However, due to their lower computing power, they have poorer analysis performance because they must use pedestrian anomaly analysis methods that can be executed with less computing power. GPU servers provide high computing power, which enables complex pedestrian anomaly analysis, but they are expensive and have very low efficiency when processing a small number of video streams. To solve these problems, this paper proposes a system that can secure the efficiency and scalability of real-time analysis by linking edge devices and GPU servers and explores its applicability in urban environments with large-scale CCTV environments. This work aims to contribute to the technological advances in pedestrian anomaly analysis and offer practical contributions to public safety and urban security.

3. Proposed System

This section proposes an edge–server architecture pedestrian abnormal behavior analysis system using a CNN. Section 3.1 describes the system structure proposed in this paper. Then, Section 3.2 describes the application scenarios in which the system proposed

in this paper can be applied. The performance of the proposed system in terms of efficiency and scalability is discussed in Section 4.

3.1. Edge–Server Architecture-Based Pedestrian Abnormal Behavior Detection System

This section describes the structure of the proposed system in detail. Figure 1 shows the structure of the proposed system. The system is divided into the CCTV, edge, integration control server (ICS), and GPU server. The CCTV transmits video streams to the edge using a real-time streaming protocol. Figure 2 provides a detailed description of the analytics engine that constitutes the analytics module of the edge. The structure shown in the figure illustrates the analytics engine configured for the experiments described in this paper. The number of analytics engines may vary depending on the pedestrian anomaly events to be analyzed, and object tracking or other preprocessing engines may be added depending on the preprocessing modules required by the pedestrian anomaly engine. In this paper, to experiment with the structure of the proposed system, the edge analysis module is configured to perform pedestrian abnormal behavior analysis through object detection, object tracking, and pedestrian abnormal behavior detection engines. In addition, more than one CCTV camera can be connected to the edge, and the number of CCTV connections is determined by the frame throughput that the object detection engine and pedestrian abnormal behavior detection engine of the edge can handle. In Section 4, we discuss this in more detail.

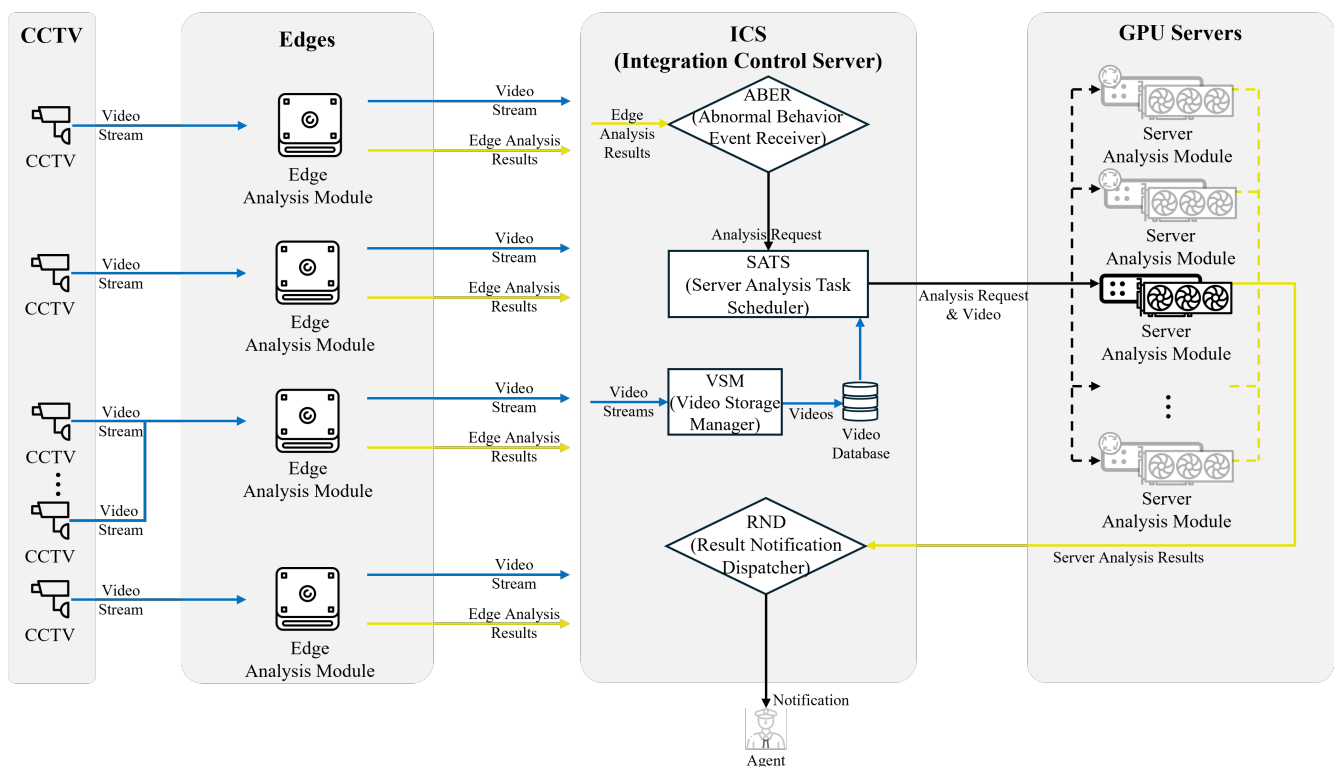


Figure 1. Overview of edge–server architecture-based pedestrian abnormal behavior detection system.

The edge’s pedestrian anomaly detection engine is rule-based, allowing real-time analysis in environments with low computing power. The edge receives CCTV video streams in real time and delivers the original video to the ICS; extracts frames from the received video stream; performs object detection, object tracking, and pedestrian abnormal behavior detection analysis based on these frames; and delivers the analysis results to the ICS’s abnormal behavior event receiver (ABER).

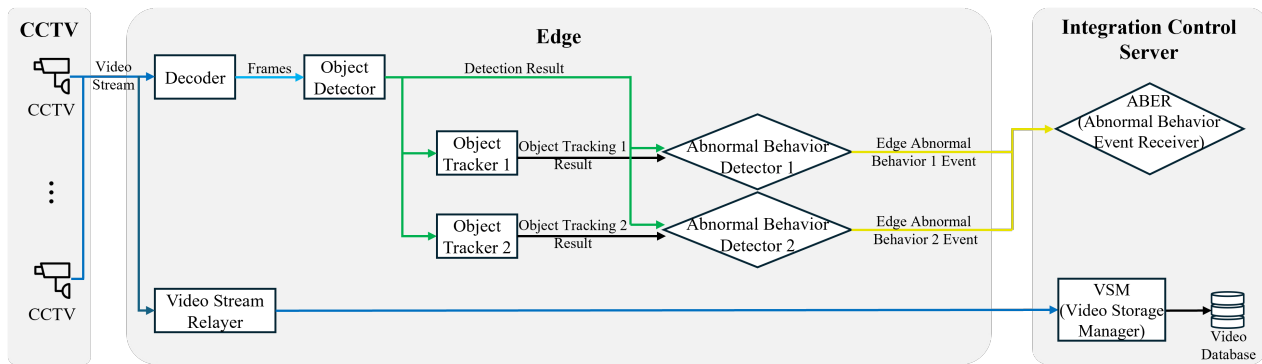


Figure 2. Analysis module structure of the edge device. The video stream from the CCTV is analyzed by the edge device to detect the abnormal behavior of a pedestrian. After frame separation, object detection, and tracking, the event information is sent to the ABER in the ICS if abnormal behavior is detected.

The ICS is composed of four main components: the abnormal behavior event receiver (ABER), the server analysis task scheduler (SATS), the video storage manager (VSM), and the result notification dispatcher (RND). The ABER collects analytics results from the edge's pedestrian anomaly detection engine. When a pedestrian anomaly event occurs in the edge's analytics engine, the ABER sends an analysis request to the SATS, which sends the corresponding CCTV and event time to the GPU server's analytics module for analysis. If no idle GPU exists, the analytics request is assigned to the server analytics module with the most analytics requests. The VSM stores and manages the video streams from the edge connected to the ICS. The pedestrian anomaly analysis results analyzed by the edge device send an analysis start event to the ABER at the beginning of the anomaly detection process and an analysis end event to the ABER at the end of the anomaly detection process. When the ABER receives a pedestrian anomaly event from the edge, it forwards an analysis request to the SATS. The algorithm enabling the SATS to assign analysis requests to the GPU server's analysis module is described in Algorithm 1. The SATS sends the CCTV section to be analyzed from the video database connected to the VSM and the GPU server's analysis module through the HLS protocol. If the type of analysis request received from the ABER starts, it is assigned to the analysis module currently connected to the SATS, which has no analysis requests in process or the smallest number of analysis requests. If all analytics modules are processing the same number of analytics requests, the analytics request is assigned to the analytics module with the fastest analysis speed, considering the interval and start time of the analytics request. The analysis module on the GPU server selected by the SATS receives the video address from the video database and analyzes the pedestrian anomaly. Meanwhile, when the ABER receives a pedestrian anomaly end event from the edge, the SATS passes the analysis end time to the analysis module on the GPU server to which it assigned the analysis request, and the analysis module analyzes the video only until the analysis end time. The SATS has information about the connected analytics module and maintains the number of assigned analytics requests and the section of the analytics request currently being processed. Since the analysis module installed on the GPU server is docker-based, it can be installed as long as CUDA and docker are installed. SATS can simply connect the ICS and GPU using IP and port information, contributing to the system's scalability.

Figure 3 shows the analysis engines comprising the GPU server's analysis modules. The GPU server's pedestrian anomaly detection engine performs DNN-based pedestrian anomaly analysis based on the results of the object detection, object tracking, and pose estimation engines. For more information about the pedestrian anomaly detection engine used in the GPU server, see the experiments described in Section 4. Since most pedestrian anomaly analyses are performed on GPUs, we can install as many analysis engines as there are GPUs on the server. The GPU server analysis module then passes the results to the

RND. If the analysis engine on the GPU server detects pedestrian anomalies, the RND notifies the control center operator of the CCTV section in which the anomaly occurred.

Algorithm 1 Assign and manage analysis requests in analysis modules.

```

1: Input: analysisRequest—a request with a type (start/end) and associated data.
2: Input: analysisModules—list of available analysis modules, each with a current task list.
3: Output: assignment—description of how the request was handled.
4: procedure MANAGEREQUEST(analysisRequest, analysisModules)
5:   bestModule  $\leftarrow$  None
6:   if analysisRequest.type = "start" then
7:     minTasks  $\leftarrow$   $\infty$ 
8:     earliestCompletionTime  $\leftarrow$   $\infty$ 
9:     for each module in analysisModules do
10:      if module.currentTasks < minTasks then
11:        minTasks  $\leftarrow$  module.currentTasks
12:        earliestCompletionTime  $\leftarrow$  module.estimatedCompletionTime
13:        bestModule  $\leftarrow$  module
14:      else if module.currentTasks = minTasks then
15:        if module.estimatedCompletionTime < earliestCompletionTime then
16:          earliestCompletionTime  $\leftarrow$  module.estimatedCompletionTime
17:          bestModule  $\leftarrow$  module
18:      SATS informs bestModule to prepare for new task
19:      Append analysisRequest to bestModule's queue
20:      bestModule.currentTasks  $\leftarrow$  bestModule.currentTasks + 1
21:      assignment  $\leftarrow$  "Request assigned to module "&bestModule.id
22:   else if analysisRequest.type = "end" then
23:     requestModule  $\leftarrow$  None
24:     for each module in analysisModules do
25:       if module is currently processing analysisRequest then
26:         requestModule  $\leftarrow$  module
27:         break
28:     if requestModule  $\neq$  None then
29:       SATS manages requestModule to end task
30:       Notify requestModule of analysisRequest's end
31:       requestModule.currentTasks  $\leftarrow$  requestModule.currentTasks − 1
32:       assignment  $\leftarrow$  "Request end notified to module "&requestModule.id
33:   else
34:     assignment  $\leftarrow$  "Request end not found in any module"
35:   return assignment

```

The proposed system first processes CCTV videos on the edge device, and the number of CCTVs that can be processed on the edge device ($N_{\text{edge_cap}}$) according to the frames that can be processed on the edge device ($f_{\text{edge_need}}$) and ($f_{\text{edge_cctv}}$) can be defined as follows: Formula (1), which gives the number of CCTVs that an edge device can handle. If only existing edge devices are utilized, the number of CCTVs that can be accommodated in one edge device by the Formula (1) can be used to check the capacity of the system. However, since the system proposed in this study is a structure in which edge devices and GPU servers are linked, the number of events and event duration must be considered to evaluate the scalability of the system.

$$N_{\text{edge_cctv}} = \frac{f_{\text{edge_need}}}{f_{\text{edge_cap}}} \quad (1)$$

It analyzes it on the GPU-equipped server when a pedestrian anomaly event is detected on the edge device, so the system's scalability can be evaluated according to the number of

pedestrian anomalies and the duration of the anomaly in the area in which the system is installed. Formula (2) denotes the average number of events (λ) during the experimental time in the region in which the system is installed, the time to run the experiment (T), and the duration of each event (d). The number of analytics modules on the GPU server (N_{server}) is based on the number of frames per second that can be processed by the analytics modules on the GPU server ($f_{server_{cap}}$) and the number of frames per second needed for analysis ($f_{server_{need}}$). For a centralized server-based analysis system, the number of servers should be specified according to the number of frames that can be processed and the number of frames required for analysis. However, for the system proposed in this study, the number of analysis modules on the GPU server will vary depending on the number of events and the duration of each event, as the system only analyzes the time bins of the CCTV devices that detect pedestrian anomalies on the edge devices.

$$N_{server} = \left\lceil \frac{f_{server_{need}} \times \frac{\lambda}{T} \times d}{f_{server_{cap}}} \right\rceil \quad (2)$$

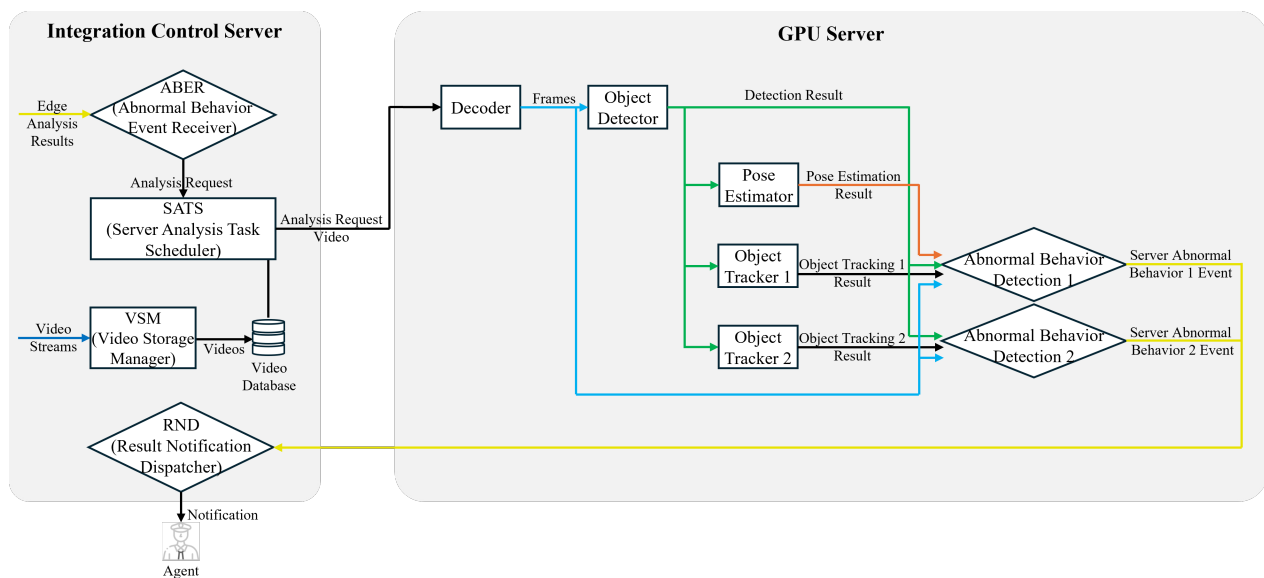


Figure 3. Analysis module structure of the GPU server. This figure shows how the results of the pedestrian abnormal behavior analysis performed on the edge device are sent to the GPU server for processing. The server performs analysis based on the requirements of the pedestrian abnormal behavior analysis engine, which requires additional preprocessing engines such as pose estimation. Based on the analysis results sent from the edge, the potent processing power inside the GPU server is used to perform more sophisticated analyses, including pose estimation, to analyze the pedestrian abnormal behavior further. The events detected by the server are sent back to the ICS, which ultimately notifies the controllers of the pedestrian’s abnormal behavior.

In particular, the system proposed in this research greatly improves the efficiency of real-time data processing by reducing unnecessary data transmission through initial analysis at the edge and processing only more complex data on the server, thereby improving the responsiveness and data processing capabilities of the entire system. Compared to the existing centralized server system that utilizes a deep learning model with high complexity, the proposed system is innovative in terms of data processing capability and scalability of the entire system in that the existing system requires the server to continuously analyze the video stream when the video stream to be analyzed is recorded and processed in real-time. In contrast, the proposed system analyzes only the sections that need to be analyzed in the server, which requires the server to always analyze all CCTVs connected to the system. In the proposed system, the edge stage detects pedestrian abnormalities based on rules based

on object recognition and tracking, and when pedestrian abnormalities are detected, the CCTVs with detected pedestrian abnormalities and the detection time are sent to the server for in-depth analysis. This distributed processing method minimizes the network load and maximizes the entire system's performance by using optimized algorithms at each stage. In addition, the server can provide more accurate analysis using high-performance deep learning algorithms, providing real-time processing speed and high analysis accuracy.

The proposed system adopts a dual-stage approach where initial analysis is performed on the edge devices, and only critical events are deeply analyzed on the GPU server. This structure allows each edge device to process initial data independently and only send the information to the GPU server when abnormal behavior is detected, significantly reducing the throughput and improving the response speed of the entire system. In particular, the integration control server (ICS) coordinates these two stages of processing and effectively manages the load on the server. The installation of the analysis module is based on docker, which is easy to install. The connection between the GPU server where the analysis module is installed and the ICS is simple, with only IP and port, which makes it easy to expand the entire system. This approach allows data from large-scale CCTV systems to be processed in real-time and ensures scalability and flexibility, enabling application in various environments.

3.2. Application of the Proposed System

In this section, we describe the application scenario considered for the proposed system. We investigated the Distinction Control Center in Seoul, Korea, to build the application scenario. The existing control system used by the center to monitor the city consists of four people working in three shifts of eight hours each, and each person monitors a program that contains 16 small screens that show different CCTV sections every five seconds. Moreover, when a police report or request is made, one of the controllers monitors the CCTV in the area where the incident occurred. In order to apply pedestrian anomaly analysis to such a system, it is necessary to utilize existing CCTV, so the installation of additional intelligent CCTV devices at an additional cost is not preferable for control agencies. Therefore, it is necessary to connect the edge devices to the existing CCTV or to connect an analysis server for analysis. Therefore, in this paper, we consider a scenario in which the edge devices first analyze the video streams of the CCTV of such an existing system. Only the time sections of the CCTV where the pedestrian abnormal behavior events occurred are analyzed by the analysis module of the GPU server. Finally, the existing control system is notified when pedestrian abnormal behavior occurs. In previous studies, methods that utilized only edge devices showed fast processing speeds but low accuracy, and frequent alarms could lead to increased fatigue and distraction, which could slow down the response to actual important events. In addition, the existing server-based system provides high accuracy. However, it requires server expansion as the number of CCTV devices increases; moreover, as the number of CCTV devices increases, the number of analyzable servers must be increased, which leads to increased costs. The system proposed in this paper has a cost-saving effect, scalability, and efficiency because it analyzes only the temporal segment of the CCTV with a GPU server only when the edge device detects pedestrian abnormal behavior. In addition, the analysis module used by the edge device can process multiple CCTV images on one edge, depending on the frame throughput required by the analysis engine, and the server analysis module performs analysis only when a pedestrian abnormal behavior event occurs as a result of the edge analysis. This fast preprocessing at the edge and the transmission of only the necessary data to the server maximize the analysis efficiency, reduce the load on the server, and reduce the cost. Therefore, the proposed system can enhance pedestrian safety and detect abnormal real-time behavior in various urban environments. In urban areas, thousands of individuals walk on the streets each day, which brings a variety of hazards. Through real-time video analysis, the system can reduce the risk of accidents at intersections and pedestrian walkways, detect dangerous situations in front of crosswalks in advance, and send appropriate warning signals to draw

the attention of both drivers and pedestrians. In educational institutions such as schools and university campuses, protecting the safety of students is a major priority. This system can help to identify conflicts between students, urgent health issues, or other emergencies on campus in real time so that authorities can respond immediately. For example, if a potential fight or medical emergency is detected on campus, the system can automatically alert the relevant authorities for a quick response. Shopping malls and commercial facilities also often see large volumes of customers, which can lead to congestion and incidents such as theft. This system analyzes customer movement routes and congestion levels and notifies managers so that they can efficiently manage congestion and prevent theft. It also ensures safety in commercial facilities by responding quickly in an emergency and providing customers with a safe escape route. This system can function as a critical element of public safety in smart cities. Deployed in different parts of the city, the system can detect abnormal behavior in public spaces, anomalies in traffic flow, and safety issues in public facilities in real time. This allows city managers to quickly respond to issues and take the necessary actions to maintain safety and order in the city, including reducing the incidence of accidents and crime. Applying these systems will contribute toward improving people's safety in their daily lives and preventing dangerous situations from occurring. These systems can be widely used in various areas of society, from city centers to schools, commercial facilities, and smart cities.

4. Experiments

In this section, we evaluate the environmental and quantitative performance to validate the efficiency and scalability of the system described in Section 3, as well as describe the datasets and evaluation metrics used for validation.

4.1. Experimental Environment

In this section, we describe the environment for the validation of the efficiency and scalability of the system proposed in this paper. Figure 4 shows the environment configured to experiment with the proposed system.

For the experiments, the CCTV environment was configured with 20 video streams using 20 RTSP video streaming servers. The videos streamed using the video streaming servers were generated by stitching videos from multiple datasets and labeling the regions in which events occurred; this is described in detail in Section 4.1. The edge environment was configured using 10 Jetson Xavier NX-based edge devices and 5 Jetson AGX Orin-based edge devices, while the ICS used a single server with a CPU i9-10940X, with 256 GB RAM, 1 TB SSD, and 20 TB HDD. The GPU server was configured using three servers with a CPU i9-10940X, with 256 GB of RAM, 1TB of SSD, and RTX 3090 $\times$ 2. The analytics modules on both the edge devices and GPU servers analyze assault events, which can occur between pedestrians during pedestrian anomalies, and fall events, which indicate when a pedestrian has fallen or fainted. A total of 10 Jetson Xavier NX-based edge devices were used with 1:1 mapping with CCTV, and 5 Jetson AGX Orin-based edge devices were used with 1:2 mapping with CCTV. The analytics module of the edge devices performs pedestrian abnormal behavior analysis through object detection, two types of object tracking, and two pedestrian abnormal behavior detection engines. Object detection uses YOLOv7-tiny, the lightest model in the deep learning-based YOLOv7 [13] family. The YOLOv7-tiny model is trained on CCTV video datasets based on the learning method in [18,19] and optimized with FP16 lightweight and TensorRT [20] conversion. Assault and fall event detection for pedestrian safety leverages the assault analysis engine from [11] and the fall analysis engine from [12], respectively. Both pedestrian event analysis engines are rule-based and can be processed in real time. To meet the requirements of different object tracking models for pedestrian anomaly analysis, the ByteTrack [21] model is used in [11] and the SORT [22] model is used in [12]. The development environment for the edge device includes JetPack 4.5, Python 3.6, DeepStream SDK 5.1 [23], and TensorRT 8.4, utilizing the YOLOv7 TensorRT model. When an edge device connects, the ICS's VSM creates threads

using Celery to receive and store the stream in the video database. These threads are responsible for receiving and storing continuous streams from the edge devices. When a particular edge device is removed from the connection list, the associated threads are also terminated to manage the resources efficiently. The ABER scans for pedestrian anomaly events from the edge devices in a standby state. When an event occurs, it requests the GPU server to analyze the video from the CCTV at the specified period via the SATS. A message queuing system utilizing Celery, Redis, and RabbitMQ allows real-time events to be queued and processed sequentially to maximize the system's processing power. The GPU server only performs analysis when an assault or fall event is detected, and notifications are sent to the ICS's RND module when the analysis is complete. When the GPU server receives an analysis request, it requests the video from the ICS's VSM using the CCTV information and time information included in the analysis request, receives the video, and starts analyzing it for assaults and falls. The model of [24] is used for assault analysis. The model of [17] is used for fall analysis. Figures 5 and 6 show the structures of the pedestrian anomaly analysis engines of [17,24]. For the assault detection model, the pose estimation results for each frame are used to generate pose joint images and stacked by T to determine the presence of an assault through the network shown in Figure 5.

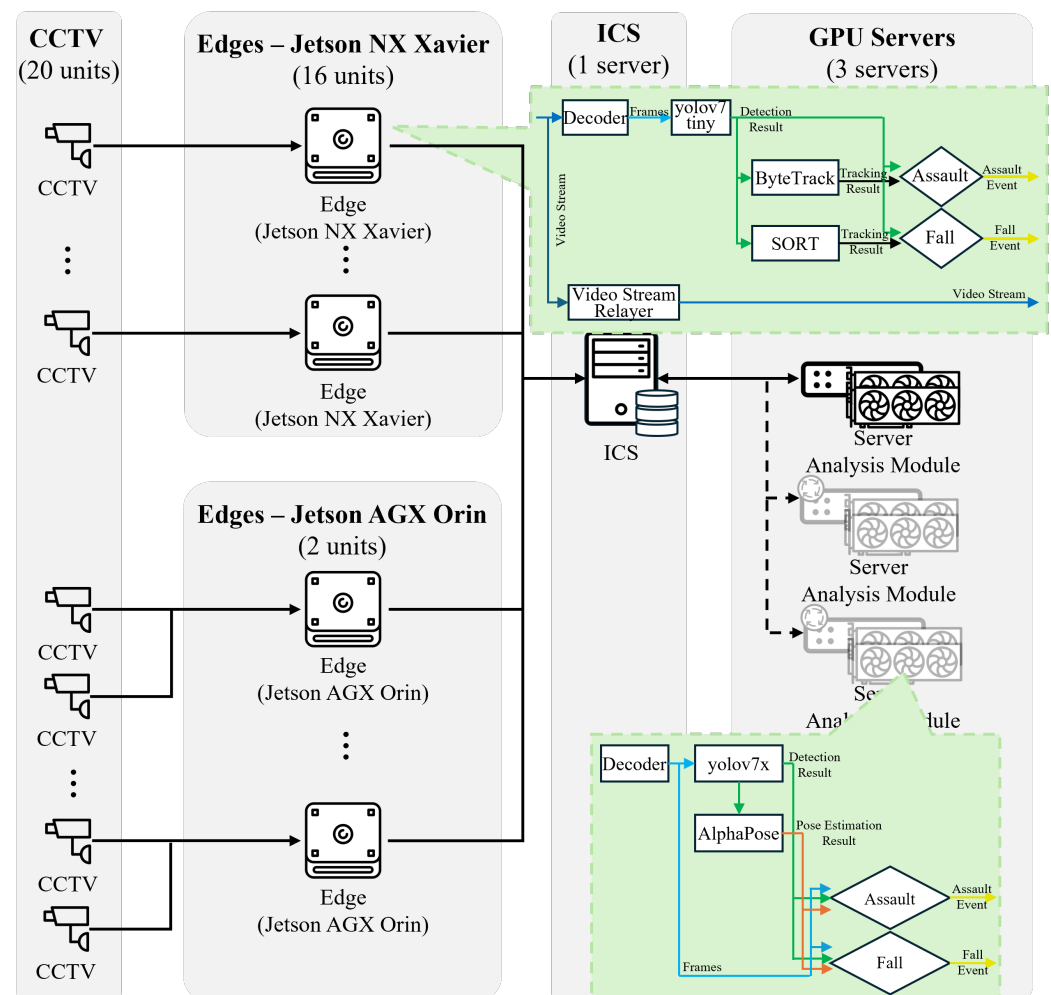


Figure 4. Configuration of the experimental environment for the efficiency and scalability evaluation of the proposed system. The architecture of the experimental environment shows the configuration of the CCTV cameras, the edge devices based on Jetson NX Xavier and Jetson AGX Orin, the integration control server (ICS), and the GPU servers. The green callouts represent the modules dedicated to analyzing pedestrian abnormal behavior within each unit.

For the fall detection model, we stack T vectors containing the coordinates of the pose estimation results for each frame into a matrix and analyze the presence of falls through the network shown in Figure 6.

The common preprocessing model for both models, object detection, uses the YOLOv7x model, a more accurate member of the YOLOv7 model family than the model used in the edge. As in the edge, YOLOv7x is trained based on the learning method of [18,19]. Since the assault and fall analysis models require preprocessing with a pose estimation model, we add the AlphaPose [25] model for real-time analysis. Since each GPU server is equipped with two GPUs, two analysis modules can be uploaded per GPU server for six analysis modules.

The scenarios in this experiment are designed to evaluate the ability of the proposed system to accurately detect pedestrian misbehavior and provide appropriate notifications under various environmental conditions and real-world situations. For example, we examine how the system can effectively respond to various situations, including an assault scenario in a park at night and an older adult falling. In these situations, the system detects the events in real time and immediately generates notifications so that users or administrators can quickly recognize them. One of the primary metrics measured in the experiments is the latency from event detection to notification, which is an essential criterion in evaluating the responsiveness and efficiency of the system. This is discussed in detail in Section 4.2. With this approach, the proposed system is expected to exhibit practicality and effectiveness by demonstrating how quickly and accurately it can respond to various safety threats that may occur in real-world environments.

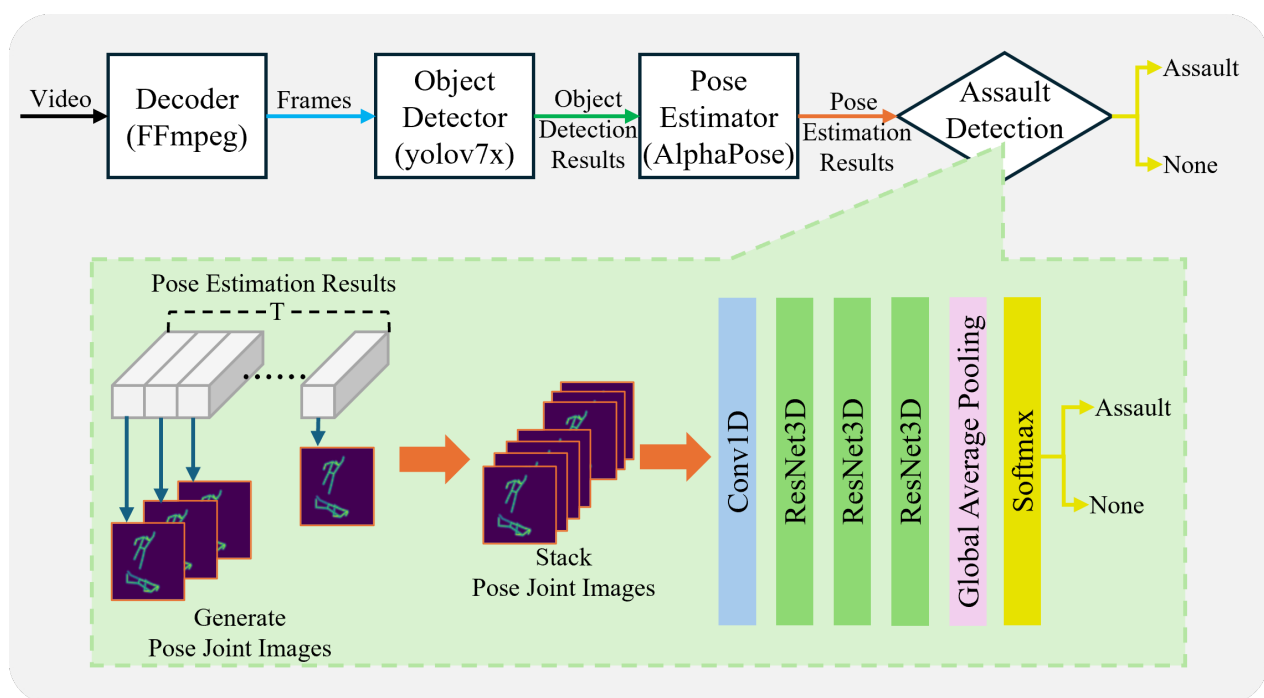


Figure 5. Network structure of the assault event detection model.

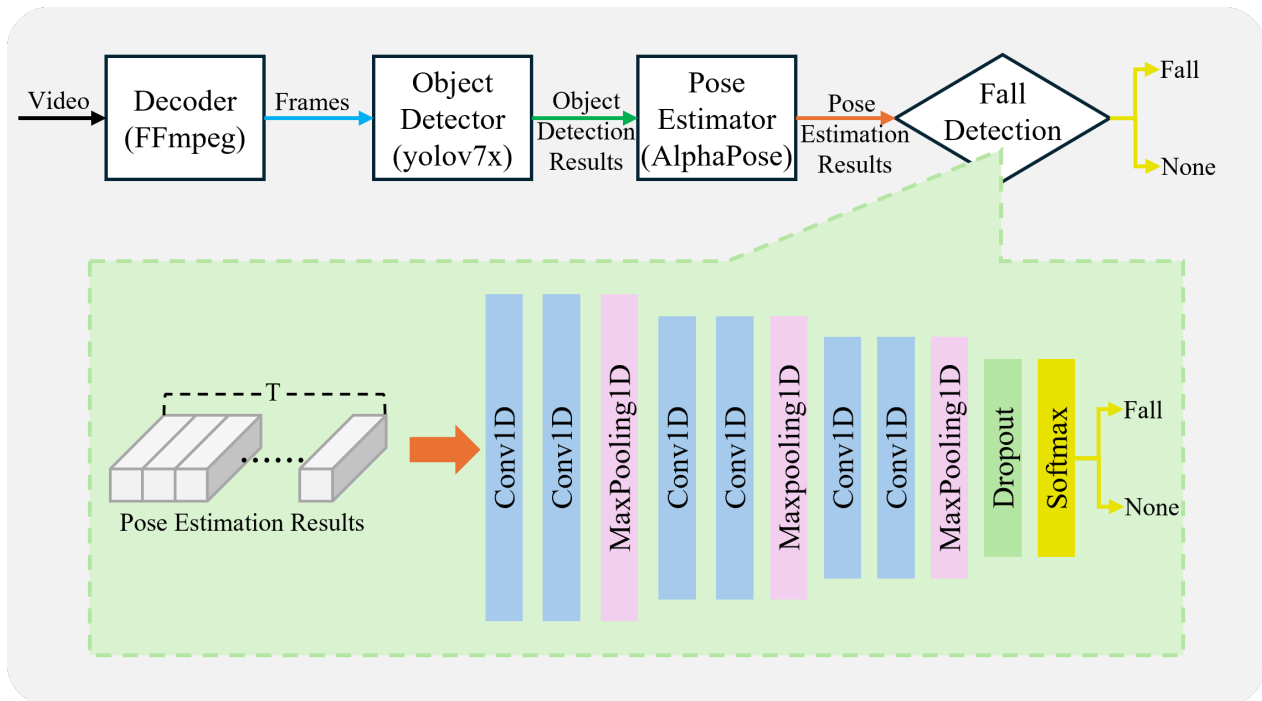


Figure 6. Network structure of the fall event detection model.

4.2. Datasets

In this paper, we evaluate the datasets used in the studies [11,12] to validate the proposed system by creating a combined video, as shown in Figure 7, and labeling the pedestrian misbehavior events and their onset and end times in the label file. Each combined movie contains ten clips of assault and fall events.

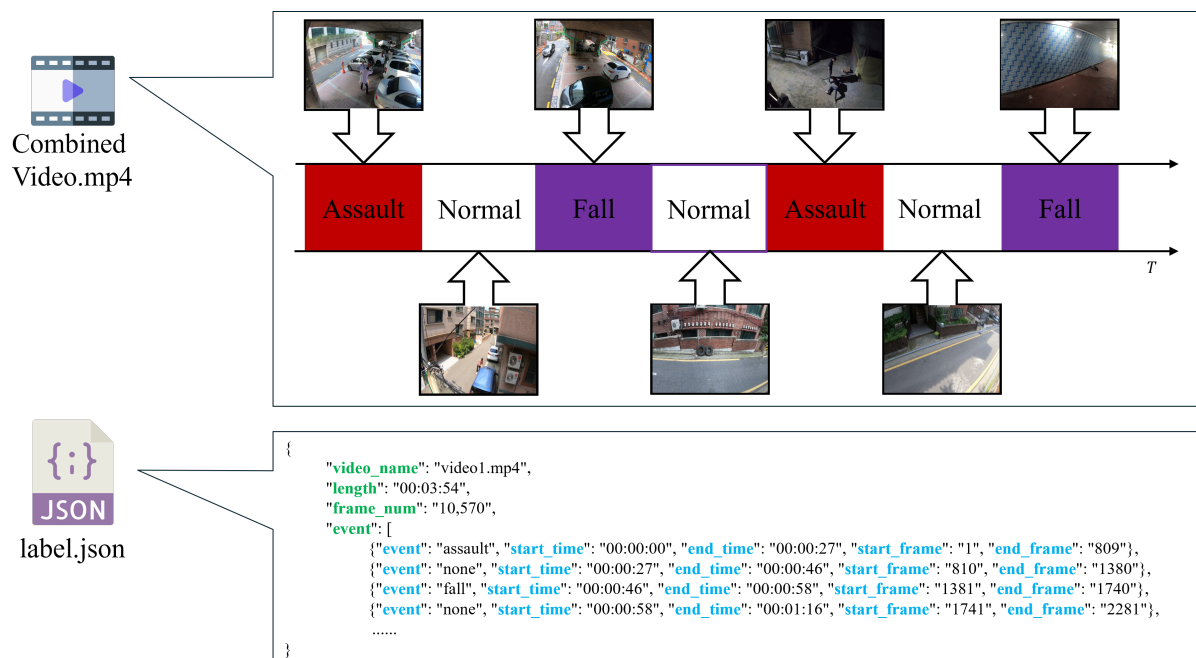


Figure 7. Structure of combined pedestrian abnormal behavior video and labels for experiments. The combined video contains assault, fall, and normal events, which are labeled with the start and end times in a JSON-formatted label file.

Table 1 shows the datasets used in this study, and Table 2 shows information about the combined videos created. The assault videos were cut into clips from the datasets in [26–29] and added to the combined video, while the fall videos were cut into clips from the datasets in [26,30,31] and added to the combined video. For videos with no events, we cut clips from the assaults and falls dataset during times at which no events occurred. The datasets used were as close as possible to the environment captured by the CCTV cameras and included daytime and nighttime footage. As shown in Table 2, only the AI Hub CCTV Abnormal Behavior dataset has 200 videos because it contains both assaults and falls, so we used 100 of each. The combined video data generated in this way were continuously streamed from 20 streaming servers to verify the efficiency and scalability of the proposed system by providing input to the system and further measuring the edge-only performance, server-only performance, and edge–server coupling performance.

Table 1. Video dataset information related to assault and fall events.

Event	Dataset	Number of Videos	Video Length
Assault	AI Hub CCTV Abnormal Behavior [26]	2780	235 h
	AI Hub Subway [27]	6510	98 h
	AI Hub Residential [28]	3579	95 h
	RWF-2000 [29]	2000	3 h
Fall	VFP290K [30]	178	2 h
	URFD [31]	70	0.1 h
	AI Hub Abnormal Behavior CCTV [26]	912	84 h

Table 2. Dataset information for combined videos.

Dataset	AI Hub Abnormal Behavior CCTV [26]	AI Hub Subway [27]	AI Hub Residential [28]	RWF-2000 [29]	VFP290K [30]	URFD [31]	Combined Video
Number of videos	190	90	90	100	100	70	64

4.3. System Efficiency Analysis and Performance Comparison

In the experiments, we continuously streamed the combined video created as described in Section 4.2 on 20 streaming servers to provide input to the system and measure the analysis time of the analysis modules used on the edge and server of the proposed system. This enabled us to obtain the processing speed on each device and check the GPU activation time in order to show that real-time processing is possible in the actual experimental environment. We also measured the performance of the edge-alone, server-alone, and edge–server coupling systems. Table 3 shows the time taken for each engine to process one frame on the edge and GPU server and the overall processing time and frames per second (fps) that can be processed. The edge is represented by both the Jetson NX Xavier and Jetson AGX Orin. Since the edge’s assault and fall models require 20 fps for peak performance, the experimental criteria involve connecting one camera for the Jetson NX Xavier and two for the Jetson AGX Orin. On the other hand, the GPU server’s analysis module requires 10 fps to achieve the best performance in the assault and fall model, and the GPU server’s analysis module, which can process 21 frames, can process events from two CCTV cameras simultaneously in real time.

The scalability of the proposed system in an experimental environment can be verified using (2) in Section 3.1. We can see that the GPU server’s $f_{\text{servercap}}$ is 21 and $f_{\text{servercap}}$ is 10, as shown in Table 3. Moreover, given λ , T , and d , we can determine the required GPU server’s analysis module. For example, if we assume that the experimental time (T) is 1 h, 120 pedestrian anomaly events with an average duration of 15 s occur in each CCTV section in 1 h, and there are 20 such CCTV sections, we can predict that λ is 2400,

T is 3600, and d is 15, so approximately five analysis modules are required. Figure 8 shows the activation of the six server analytics modules loaded on three GPU servers when processing 20 video streams. The experiment involved 120 pedestrian anomaly events, each averaging approximately 15 seconds, over the course of an hour, based on the assumptions outlined above. Each GPU server has two GPUs, so two analytics modules are loaded for six. When an assault or fall occurs due to the edge's pedestrian anomaly analysis, an analysis request is sent to the GPU server's analysis module, and the server analysis module that received the analysis request is activated. When the analysis is completed, the server analysis module is deactivated. The activation status of the six server modules in the graph shows that each module is not always active, has idle periods, and is shut down in real time without exceeding the video streaming time when running for one hour. Thus, we can see that 20 CCTV cameras can be covered by 15 edge devices and 3 GPU servers.

Table 4 shows the performance of the edge and server alone and after edge–server linkage. The pedestrian abnormal behavior analysis model of the edge is a rule-based model. However, it is designed with higher recall than precision, and the server's pedestrian abnormal behavior analysis model is designed with lower recall but higher precision. Pedestrian anomaly analysis using only general edges has a high recall, so frequent alarms are generated; when the controllers in the control center receive frequent alarms, fatigue, and distraction can slow down the response to actual important events. This, in turn, reduces the overall effectiveness of the system and risks reducing the ability to respond appropriately to emergencies. If the server is used alone, the frequency of alarms is reduced because the pedestrian anomaly analysis model is designed to prioritize precision. However, the server must process all connected CCTV cameras, which creates a significant load. Consequently, a server expansion cost may be incurred, as the server needs to be upgraded to connect additional CCTV cameras. Table 4 shows the processing time versus the video length used in the experiment. In the table, in the term video length $\times$ CCTV, video length is the average length of each video, and CCTV is the number of video streams. The experimental results show that processing 20 videos with the edge alone takes 294 s, with an average of 313 s, but the F1 score is the lowest. The server alone takes over three times as long to process 20 videos but performs better than the edge. On the other hand, we can see that using both the edge and server results in near real-time processing but with improved performance over the edge and server alone. Based on these experimental results, the edge–server system is designed to combine the advantages of high recall from the edge and high precision from the server, thus minimizing the occurrence of unnecessary alarms while not missing important events. The results for the edge–server system show that the recall performance is improved compared to that of the server alone due to the reduction in false alarms in the combined system. Therefore, it can be seen that the system proposed in this paper can help controllers focus on critical events and respond quickly, improving the overall system's performance and reliability.

Table 5 compares the performance of the edge and server alone and that after edge–server coupling for day and night videos. The experimental results clearly show that the performance varies between day and night; they provide an essential baseline for the evaluation of the adaptability and effectiveness of each system under different environmental conditions. When the edge is used independently, it tends to have high recall, which enables it to detect pedestrian anomalies without missing them. When the server is used independently, it tends to have high precision, which enables it to detect pedestrian anomalies with high accuracy. On the other hand, the integrated edge–server architecture shows a large performance improvement in terms of recall and precision compared to the standalone system. This proves that the proposed system is able to minimize unnecessary daytime and nighttime alarms while not missing important events. These results demonstrate the applicability and practicality of the integrated system in real-world urban environments while contributing to improved pedestrian safety and public security.

Table 3. Overall processing time to process one frame of a video on the edge and server.

Device	Engine			Processing Time		
	Type	Name	Minimum Required fps	1 Frame (ms)	Total	
					Time (ms)	fps
Edge (Jetsont NX Xavier)	Decoder	NVDEC	-	0.925	47.430	21.083
	Object Detection	YOLOv7-tiny TRT	-	43.171		
	Object Tracking	ByteTrack	-	1.736		
		SORT	-	1.112		
	Abnormal Behavior Detection	Assault	20	0.352		
		Fall	20	0.134		
Edge (Jetson AGX Orin)	Decoder	NVDEC	-	0.925	23.998	41.670
	Object Detection	YOLOv7-tiny TRT	-	19.702		
	Object Tracking	ByteTrack	-	1.714		
		SORT	-	1.143		
	Abnormal Behavior Detection	Assault	20	0.345		
		Fall	20	0.169		
GPU Server	Decoder	FFmpeg	-	1.133	47.163	21.203
	Object Detection	YOLOv7x	-	7.441		
	Pose Estimation	AlphaPose	-	12.279		
	Abnormal Behavior Detection	Assault	10	25.293		
		Fall	10	1.017		

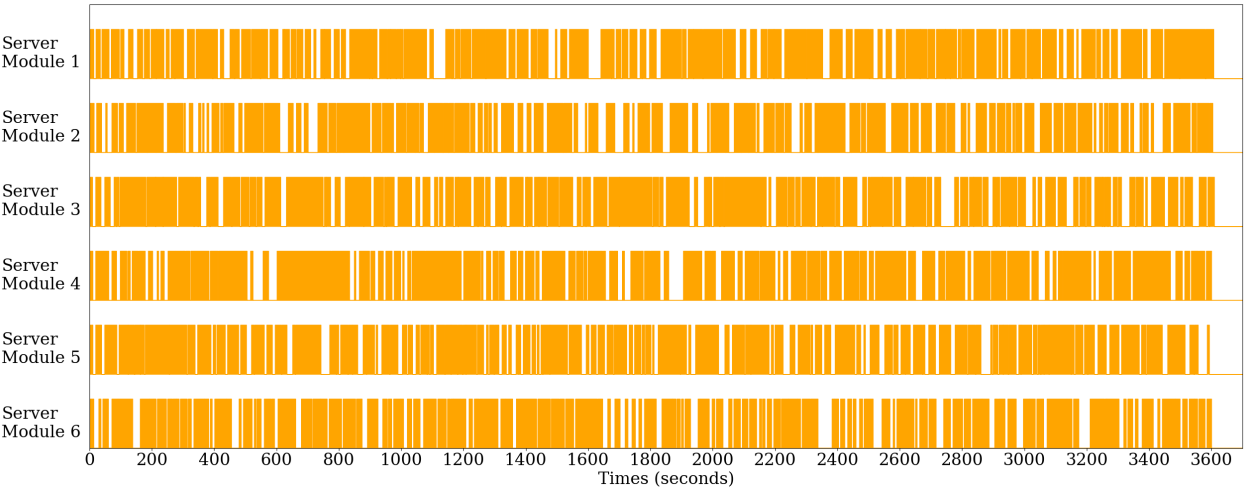


Figure 8. Visualization of server module activation states in GPU servers within the experimental environment. The activation statuses of 6 server analysis modules loaded on 3 GPU servers to process real-time video streaming from 20 CCTV cameras for 1 h are shown. These data show that the proposed system is capable of real-time processing and can effectively cover 20 CCTV cameras with the selected configuration.

Table 4. Performance comparison: standalone edge and server vs. integrated edge–server architecture.

Abnormal Behavior	Type	Number of Units	Video Length × CCTV	Proc. Time (s)	TP	FP	FN	Recall	Precision	F1
Assault	Standalone Edge	18 Edge	313×20	294	288	70	32	0.899	0.803	0.848
	Standalone Server	3 Server	313×20	1040	297	51	23	0.929	0.853	0.889
	Integrated edge–server (Ours)	18 Edge + 3 Server	313×20	314	286	40	21	0.932	0.877	0.904
Fall	Standalone Edge	18 Edge	313×20	294	249	21	23	0.915	0.921	0.918
	Standalone Server	3 Server	313×20	1040	241	15	21	0.906	0.941	0.923
	Integrated edge–server (Ours)	18 Edge + 3 Server	313×20	314	225	10	18	0.926	0.958	0.942

Table 5. Day and night performance comparison: standalone edge and server vs. integrated edge–server architecture.

Abnormal Behavior	Day or Night	Type	TP	FP	FN	Recall	Precision	F1 Score
Assault	Day	Standalone Edge	192	41	27	0.875	0.824	0.849
		Standalone Server	199	39	10	0.954	0.836	0.891
		Integrated edge–server (Ours)	190	27	14	0.931	0.8756	0.902
	Night	Standalone Edge	96	29	5	0.950	0.764	0.847
		Standalone Server	98	12	13	0.881	0.888	0.884
		Integrated edge–server (Ours)	96	13	7	0.933	0.880	0.906
Fall	Day	Standalone Edge	166	14	15	0.915	0.922	0.918
		Standalone Server	157	12	16	0.909	0.929	0.919
		Integrated edge–server (Ours)	146	5	18	0.891	0.967	0.928
	Night	Standalone Edge	83	7	8	0.917	0.919	0.918
		Standalone Server	84	3	5	0.940	0.966	0.953
		Integrated edge–server (Ours)	79	5	0	0.997	0.941	0.968

5. Discussion

Through the experiments described in this paper, we demonstrate that the proposed system with an edge–server structure is more scalable and efficient for pedestrian abnormal behavior detection compared to the edge and server alone. However, during the research process, we needed assistance in collecting and processing the data under various conditions similar to natural urban environments. In particular, the video resolution of the published datasets was low, such as 320×240 or 640×360 . In CCTV images, human objects occupy less than 1/10 of the area in the frame, so high-resolution videos are required to improve the accuracy of pedestrian abnormal behavior analysis. It was time-consuming to create combined videos, including standard videos with no events other than assaults and falls, and to label the times at which the events occurred. In addition, to test real-time streaming data processing, it was challenging to design and implement a streaming environment with 20 streams and to ensure seamless data communication and processing between the edge and ICS and the ICS and GPU server. In particular, much effort was required to develop a mechanism to transmit the data without network delays or data loss and promptly deliver the analysis results. The limited computing power of edge devices complicates the running of complex analytics algorithms that require high accuracy, so much experimentation was needed to select the available models.

The limitations of the proposed system include the fact that we used the combined videos generated in this study for video streaming. Although the combined videos consisted of publicly available data, they were specific to the experimental environment, which may limit their generalizability to various real-world environments. In particular, this study targeted pedestrian anomalies such as assaults and falls. However, the definition and

detection criteria of actual anomalies may vary from environment to environment, and it is a great challenge to develop a model that can analyze various anomalies. In addition, the analysis of real-time CCTV video streaming may raise concerns about individuals' privacy and data security. The system proposed in this study may not provide a sufficient solution to these problems, which need to be addressed separately. In addition, most systems that utilize CCTV operate on a private network, and the proposed system must be applied as an addition to a control system that uses a private network. However, data encryption is essential in the communication between CCTV and edge devices, between edge devices and the ICS, and between the ICS and GPU servers, to protect the data from unauthorized access and prevent sensitive information from being leaked. In particular, the VSM of the ICS should ensure that the video data stored in the video database are accurately defined for the retention period and deleted and that faces and license plates that can be used to identify individuals in the video are de-identified as much as possible during storage to enhance individuals' privacy.

Although the proposed system is focused on pedestrian abnormal behavior analysis in a large-scale CCTV environment, it is expected to be used in various domains that require control, such as traffic control, urban safety, and environmental monitoring similar to pedestrian abnormal behavior analysis, as well as in domains that require advanced analysis capabilities. For example, suppose there is a scenario in which analysis or data generation is performed on the smartphone itself. In that case, the results of analyzing or generating data through the models available on the smartphone are viewed. The user requests the server for higher-performance analysis or generation results, and the proposed system can contribute to efficiency and scalability in such a scenario. In this regard, the proposed system will have many possible domains to utilize, and if applied, it will be possible to create a scalable and efficient environment.

In this study, we constructed and evaluated the performance of an edge-server-based pedestrian abnormal behavior analysis system. However, continued research and improvement efforts are required to enhance the system's performance further and increase its applicability in real-world environments. In particular, obtaining high-resolution datasets, optimizing the real-time processing, enhancing the model diversity and flexibility, and addressing privacy and data security concerns are vital issues to be addressed in future research. Through these efforts, the proposed system is expected to be effectively applied to a broader range of real-world environments.

6. Conclusions

The edge-server architecture-based pedestrian anomaly detection system developed in this paper provides key contributions that significantly improve the efficiency and scalability of existing systems. The innovation lies in the dual-stage data pre-processing approach on edge devices and deep analysis on GPU servers. This structure allows each edge device to process the initial data independently and only send the information to the GPU server when a significant event is detected, significantly reducing the throughput of the entire system and improving the response time. This innovative approach provides the ability to process data from large-scale CCTV systems in real time, which is a key contribution to enabling its application in various environments. In particular, the system management via ICS in the proposed system minimizes network transmission data and effectively balances the load on the server. These technical innovations ensure the system's scalability and flexibility, making it easier to adapt to more diverse environments. This research also points to important ways to improve the accuracy of real-time data processing in large-scale networks and reduce operational costs. However, since the current system focuses on a specific urban environment, further research is needed to confirm its applicability in different environments. In the future, it is necessary to improve the accuracy and reliability of the analytical model by obtaining high-resolution datasets and expanding the model's scope to various types of anomalies. Solutions to privacy and data security issues must be developed to strengthen technical and legal accountability. These ongoing

research and development efforts are expected to contribute significantly to realizing safe and efficient urban environments.

Author Contributions: Conceptualization, J.S.; methodology, J.S.; software, J.S.; validation, J.S.; formal analysis, J.S.; investigation, J.S. and J.N.; resources, J.S.; data curation, J.S.; writing—original draft preparation, J.S.; writing—review and editing, J.S.; visualization, J.S.; supervision, J.N.; project administration, J.N.; funding acquisition, J.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2023-00259099) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation)

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used for the training and evaluation of our models, including the AI Hub Abnormal Behavior CCTV [26], AI Hub Subway [27], AI Hub Residential [28], RWF-2000 [29], VFP290K [30], and URFD [31] datasets, are publicly accessible. The AI Hub Abnormal Behavior CCTV dataset is available at <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=171> (accessed on 3 April 2024). The AI Hub Subway dataset is available at <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=174> (accessed on 3 April 2024). The AI Hub Subway Residential is available at <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=515> (accessed on 3 April 2024). The RWF-2000 dataset can be found at <https://github.com/mchengny/RWF2000-Video-Database-for-Violence-Detection> (accessed on 3 April 2024). The VFP290K dataset can be found at <https://sites.google.com/view/dash-vfp300k/> (accessed on 3 April 2024). The URFD dataset can be found at <http://fenix.ur.edu.pl/~mkepski/ds/uf.html> (accessed on 3 April 2024). Each dataset's usage in this study was in accordance with the terms set by the respective provider.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CCTV	closed-circuit television
CNN	convolutional neural network
3D CNN	three-dimensional convolutional neural network
ABER	abnormal behavior event receiver
ICS	integration control server
SATS	server analysis task scheduler
VSM	video storage manager
RND	result notification dispatcher

References

1. Ugli, D.B.R.; Kim, J.; Mohammed, A.F.; Lee, J. Cognitive video surveillance management in hierarchical edge computing system with long short-term memory model. *Sensors* **2023**, *23*, 2869. [\[CrossRef\]](#)
2. Song, L.K.; Li, X.Q.; Zhu, S.P.; Choy, Y.S. Cascade ensemble learning for multi-level reliability evaluation. *Aerosp. Sci. Technol.* **2024**, *148*, 109101. [\[CrossRef\]](#)
3. Zhang, J.; Tian, J.; Alcaide, A.M.; Leon, J.I.; Vazquez, S.; Franquelo, L.G.; Luo, H.; Yin, S. Lifetime extension approach based on levenberg-marquardt neural network and power routing of dc-dc converters. *IEEE Trans. Power Electron.* **2023**, *38*, 10280–10291. [\[CrossRef\]](#)
4. Zhang, J.; Tian, J.; Yan, P.; Wu, S.; Luo, H.; Yin, S. Multi-hop graph pooling adversarial network for cross-domain remaining useful life prediction: A distributed federated learning perspective. *Reliab. Eng. Syst. Saf.* **2024**, *244*, 109950. [\[CrossRef\]](#)
5. Li, X.; Lv, S.; Zhang, J.; Li, M.; Rodriguez-Andina, J.J.; Qin, Y.; Yin, S.; Luo, H. FDGR-Net: Feature Decouple and Gated Recalibration Network for medical image landmark detection. *Expert Syst. Appl.* **2024**, *238*, 121746. [\[CrossRef\]](#)
6. Ali, M.; Goyal, L.; Sharma, C.M.; Kumar, S. Edge-Computing-Enabled Abnormal Activity Recognition for Visual Surveillance. *Electronics* **2024**, *13*, 251. [\[CrossRef\]](#)

7. Lee, H.; Cho, H.; Noh, B.; Yeo, H. NAVIBox: Real-Time Vehicle–Pedestrian Risk Prediction System in an Edge Vision Environment. *Electronics* **2023**, *12*, 4311. [CrossRef]
8. Graves, A. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.
9. Yao, S.; Ardabili, B.R.; Pazho, A.D.; Noghre, G.A.; Neff, C.; Tabkhi, H. Integrating AI into CCTV Systems: A Comprehensive Evaluation of Smart Video Surveillance in Community Space. *arXiv* **2023**, arXiv:2312.02078.
10. Zahrawi, M.; Shaalan, K. Improving video surveillance systems in banks using deep learning techniques. *Sci. Rep.* **2023**, *13*, 7911. [CrossRef] [PubMed]
11. Lee, S.; Park, R.; Park, H.M. Robust and Efficient Road-view CCTV Video Violence Detection Method. *J. Inst. Electron. Inf. Eng.* **2023**, *60*, 631–641.
12. Moon, S.; Yang, C.; Kang, S.J. Real-time Detection of Specific Events: A Case Study of Detecting Falls. *IEIE Trans. Smart Process. Comput.* **2023**, *12*, 171–177. [CrossRef]
13. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
14. Huang, T.; Han, Q.; Min, W.; Li, X.; Yu, Y.; Zhang, Y. Loitering detection based on pedestrian activity area classification. *Appl. Sci.* **2019**, *9*, 1866. [CrossRef]
15. Mehmood, A. Abnormal behavior detection in uncrowded videos with two-stream 3D convolutional neural networks. *Appl. Sci.* **2021**, *11*, 3523. [CrossRef]
16. Ullah, F.U.M.; Ullah, A.; Muhammad, K.; Haq, I.U.; Baik, S.W. Violence detection using spatiotemporal features with 3D convolutional neural network. *Sensors* **2019**, *19*, 2472. [CrossRef] [PubMed]
17. Salimi, M.; Machado, J.J.; Tavares, J.M.R. Using deep neural networks for human fall detection based on pose estimation. *Sensors* **2022**, *22*, 4544. [CrossRef] [PubMed]
18. Hwang, Y.; Song, J.; Nang, J. Development of Risky Objects to Pedestrian Detector based on Deep-Learning for Night Time CCTV Video Analysis. In Proceedings of the KIISE Korea Software Congress, Pyeongchang, Republic of Korea, 20–22 December, 2021; pp. 407–409.
19. Moon, H.; Song, J.; Nang, J. How to Augment your Dataset to Recognize Small Objects in CCTV Footage Dataset Enrichment Methods for Recognizing Small Objects. In Proceedings of the the KIISE Korea Computer Congress, Jeju, Republic of Korea, 18–20 June, 2023; pp. 970–972.
20. NVIDIA Corporation. NVIDIA TensorRT Documentation. 2024. Available online: <https://docs.nvidia.com/deeplearning/tensorrt/> (accessed on 3 April 2024).
21. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Bytetrack: Multi-object tracking by associating every detection box. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 1–21.
22. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE international conference on image processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 3464–3468.
23. NVIDIA Corporation. NVIDIA DeepStream SDK Documentation. 2024. Available online: <https://developer.nvidia.com/deepstream-sdk> (accessed on 3 April 2024).
24. Hwang, Y.; Song, J.; Nang, J. Development of Pose-based CCTV Video Behavior Classifier for Detecting Violent Events. In Proceedings of the the KIISE Korea Computer Congress, Jeju, Republic of Korea, 29 June–1 July 2022; pp. 948–950.
25. Fang, H.S.; Li, J.; Tang, H.; Xu, C.; Zhu, H.; Xiu, Y.; Li, Y.L.; Lu, C. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 7157–7173. [CrossRef] [PubMed]
26. Artificial Intelligence Hub Korea. AIhub Abnormal Behavior CCTV Dataset. 2021 Available online: <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=171> (accessed on 1 April 2024).
27. Artificial Intelligence Hub Korea. AIhub Subway Dataset. 2021. Available online: <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=174> (accessed on 1 April 2024).
28. Artificial Intelligence Hub Korea. AIhub Abnormal Behavior in Residential and Public Spaces Dataset. 2021. Available online: <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=515> (accessed on 1 April 2024).
29. Cheng, M.; Cai, K.; Li, M. RWF-2000: An Open Large Scale Video Database for Violence Detection. *arXiv* **2019**, arXiv:1911.05913.
30. An, J.; Kim, J.; Lee, H.; Kim, J.; Kang, J.; Shin, S.; Kim, M.; Hong, D.; Woo, S.S. VFP290k: A large-scale benchmark dataset for vision-based fallen person detection. In Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), Virtual, 6 December, 2021.
31. University of Rzeszow. UR Fall Detection Dataset. Available online: <http://fenix.ur.edu.pl/~mkepski/ds/uf.html> (accessed on 1 April 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.