ISSN 2616-5775 Vol. 7, Issue 11: 78-85, DOI: 10.25236/AJCIS.2024.071111

Research on Fall Detection Based on Vision and Wearable Devices

Shan Li[1,a], Lei Ding[2,*],Yi Shi[1,3]

[1]*Electronic Science and Technology, Shaanxi University of Science and Technology, Xi'an, China*
[2]*Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China*
[3]*Computer Science and Technology, Shaanxi University of Science and Technology, Xi'an, China*
[a]*j3069651072@163.com*
[*]*lding@sust.edu.cn*
[*]*Corresponding author*

Abstract: *Falls are sudden accidental injuries, and a real-time fall detection system can mitigate the severe consequences of delayed detection by providing timely assistance to the individual who has fallen. In recent years, with the rapid development of technologies such as deep learning, various fall detection methods have been developed. Based on this, this paper proposes a multi-modal fall detection system that integrates accelerometers and video surveillance. The system simultaneously detects the target using both video surveillance and accelerometers, and then performs decision fusion to obtain the final detection result. Experimental evaluations on the UR Fall Detection dataset demonstrate that the developed multi-sensor fusion fall detection system achieves higher accuracy compared to single modality fall detection methods using either video surveillance or accelerometer sensors.*

Keywords: *YOLOv5; LSTM; Decision Fusion; Multi-modal; Fall Detection*

1. Introduction

The 21st century has witnessed a global shift towards population aging, marked by a dramatic increase in the elderly population. Consequently, the health issues of the elderly have garnered widespread attention. According to the World Health Organization, over 300,000 deaths annually are attributed to falls, with half of these occurring among individuals aged 60 and above [1]. In China, falls are the leading cause of injury-related deaths among the elderly, and the risk of fatality and injury increases with age. This poses a significant challenge to medical resources and exacerbates psychological anxiety among the elderly. Therefore, falls among the elderly represent a global health concern, significantly impacting their health and quality of life. Currently, falls among the elderly occur frequently, becoming the primary cause of death and injury among those over 65 years old.

With the rapid advancements in modern machine learning technologies, communication technologies, and micro-electromechanical systems (MEMS), integrating embedded devices equipped with inertial sensors, cameras, and other sensors into the daily lives of the elderly offers a promising solution. These devices can real-time perceive activity data, analyze potential factors leading to falls, and thereby prevent the harm caused by falls. This trend is poised to become a research focus and hotspot in fall detection technologies for the elderly.

Currently, two mainstream fall detection methods exist: sensor-based and computer vision-based approaches.

Sensor-based fall detection primarily involves wearing devices integrated with various sensors, which detect changes in human motion data such as acceleration and orientation angles to identify falls. Sucerquia A et al.[2] proposed a fall detection method using nonlinear classification features and a Kalman Filter with a periodicity detector to reduce false alarms. Wilk et al. [3] combined nonlinear acceleration time series with spatial trajectories and a Convolutional Neural Network (CNN) to extract and classify features from accelerometer data. Eyobu et al. [4] first applied Short Time Fourier Transform (STFT) to analyze Inertial Measurement Unit (IMU) data to obtain spectral features, then utilized Long Short-Term Memory Networks (LSTM) to capture sequential patterns, and finally employed a Multi-Layer Perceptron (MLP) to recognize human activities.

Computer vision-based fall detection techniques capture images of daily life activities through

ISSN 2616-5775 Vol. 7, Issue 11: 78-85, DOI: 10.25236/AJCIS.2024.071111

cameras, analyze video sequences, extract relevant human feature information, and subsequently distinguish between fall and non-fall behaviors. Ramirez H et al. [5] proposed a recognition system with four classification models (RF, SVW, MLP, and KNN) for fall detection and activity monitoring based on camera vision. They introduced pose estimation as a feature extraction mechanism, describing RGB images as a set of human skeletons represented by key nodes, enabling multi-person fall detection. Chen et al. [6] developed a two-stream CNN algorithm combining RGB and optical flow images, utilizing inter-frame information differences as input to detect falls. Núñez-Marcos A et al. [7] first applied transfer learning from action recognition to fall detection, employing optical flow images as input to the network, independent of environmental characteristics. These images solely represent motion between consecutive video frames, disregarding appearance-related information. Computer vision-based fall detection is non-invasive, requiring no real-time device wear, thus minimizing disruption to daily life. It is relatively low-cost, has a wide detection range, and has garnered considerable attention.

Currently, most fall detection methods rely heavily on single-modality data, such as solely video footage or sensor data, which possesses inherent drawbacks. The information captured by a single modality tends to be one-sided, limiting its applicability in various scenarios. For instance, sensor-based fall detection technologies utilize simulated human activities and fall data, which significantly differ from real-world scenarios. Conversely, computer vision-based fall detection methods are prone to be influenced by factors like lighting conditions and occlusions, rendering them inadequate for fulfilling fall detection needs in complex environments.

In response to these limitations, this paper proposes a fall detection framework that integrates two distinct data sources: image data and accelerometer data. By harnessing the strengths of both accelerometer readings and image surveillance footage, followed by their fusion in later stages, this approach aims to enhance the accuracy of fall detection, improve adaptability in complex environments, and address the shortcomings associated with single-modality data in fall detection.

To validate the feasibility of this framework, experiments were conducted on the UR Fall Detection [8], a publicly available multimodal dataset specifically designed for fall detection research.This empirical investigation serves to substantiate the practicality and effectiveness of the proposed method.

2. Design of a Fall Detection Algorithm Based on Multi-modal Data

A fall detection system integrating both video data and accelerometer data monitoring comprises two subsystems: a video-based subsystem utilizing the YOLOv5 algorithm, and an accelerometer-based subsystem employing a two-layer cascaded LSTM network. Subsequently, a decision-level dynamic weighted fusion algorithm is adopted for integrated detection.

2.1 Video Surveillance Data

For a video-based fall detection system, the YOLO V5 algorithm is employed for detection, which is structured into four primary components: Input, Backbone, Neck, and Head.

The Input stage initializes by accepting images, subsequently resizing or padding them to a specified dimension for the network, followed by normalization. Additionally, the Input incorporates Mosaic data augmentation to enhance both detection speed and accuracy. It also leverages adaptive anchors and image scaling for optimal performance.

The Backbone comprises the Focus slicing process and the Cross Stage Parity (CSP) structure, featuring slice-based downsampling capabilities. The innovative Cross Stage Partial Network (CSPNet) employed in the Backbone addresses gradient-related challenges, reducing model complexity while enhancing inference speed and accuracy in object detection.

The Neck module serves as a conduit, integrating feature maps of varying human motion postures from different layers.The Path Aggregation Network (PANet) facilitates feature fusion by propagating extracted features to deeper layers, generating multi-scale human motion feature maps, thereby improving target detection accuracy.

The Output stage, comprising bounding box prediction, classification loss functions, and Non-Maximum Suppression (NMS), evaluates the detection performance. It is primarily responsible for conducting multi-scale object detection on the feature maps extracted by the backbone network, assigning class confidence scores. Figure 1 below illustrates the overall architecture of YOLOv5.

ISSN 2616-5775 Vol. 7, Issue 11: 78-85, DOI: 10.25236/AJCIS.2024.071111

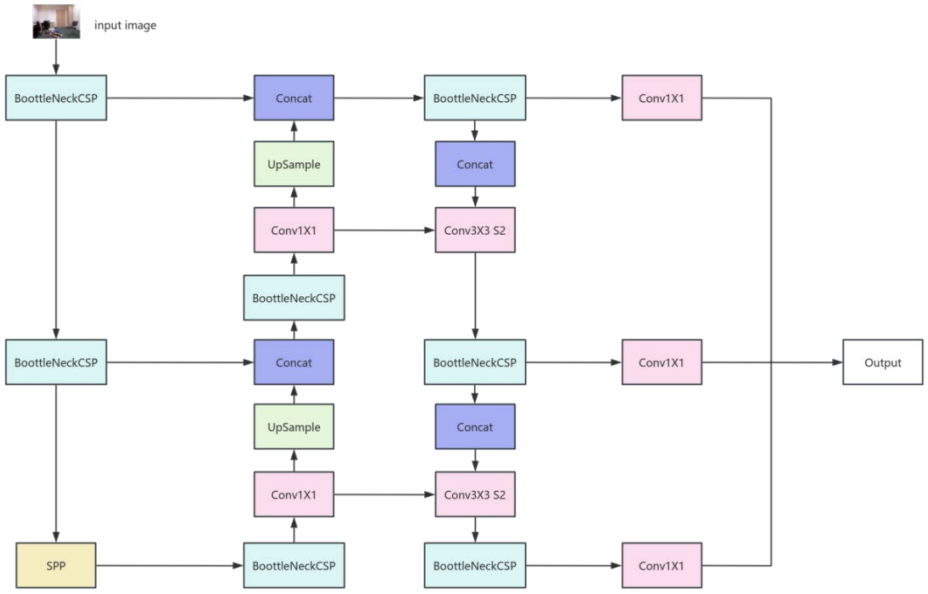

Figure 1: YOLOv5 architecture.

2.2 Accelerometer Data

Given the inherent temporal characteristics of sensor data, Recurrent Neural Networks (RNNs) are utilized for processing. However, traditional RNNs suffer from the issue of long-term dependencies, often resulting in the "forgetting" of early information as the sequence progresses, making it difficult to effectively consider the overall context. Long Short-Term Memory (LSTM) networks, a special type of RNN, aim to address this issue by mitigating the gradient explosion and vanishing problems encountered by conventional RNNs when dealing with long sequences. As an artificial neural network architecture capable of capturing long-range dependencies in time-series data, LSTM is well-suited for handling sequential data. By explicitly storing internal representations of temporal contexts, LSTM networks can effectively process time-series data. A common challenge in processing time-series is the time lag between significant pieces of information, which standard recurrent networks often struggle to handle due to their inability to process long lags, limiting their efficiency with real-world data. Most real-world information, such as voice signals and sensor data, contains temporal recurrences spanning far intervals, making them difficult to process with such models.

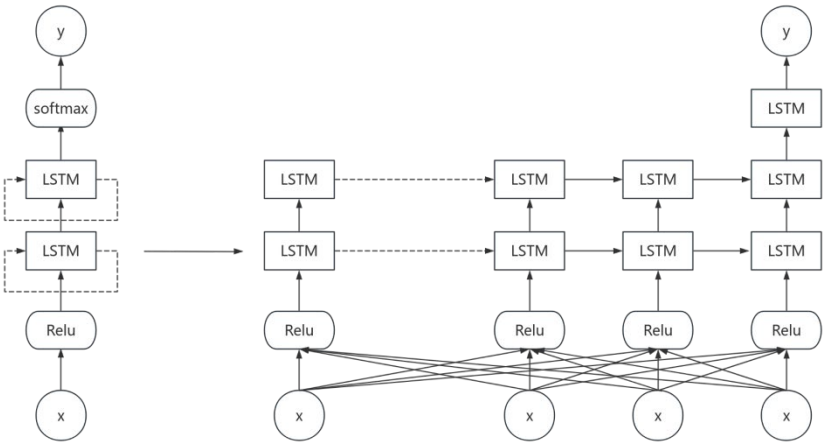

Figure 2: LSTM architecture.

To achieve end-to-end processing from raw sensor data to final decision outcomes, this paper designs a fall detection model based on LSTM networks. The model architecture, as illustrated in Figure 2, comprises four main components:

Input Layer: Receives time-series data from triaxial accelerometers. The raw data is segmented

ISSN 2616-5775 Vol. 7, Issue 11: 78-85, DOI: 10.25236/AJCIS.2024.071111

using a sliding time window to extract segments containing n sampling points, which serve as inputs to the subsequent network layers. Here, n represents the length of the data segment (i.e.the number of samples covered by the sliding time window), and denotes the data at time t. Thus, each input segment is a two-dimensional vector of dimension [3,n], accounting for the three axes (three acceleration).

Non-linear Layer: A non-linear activation layer is introduced between the input layer and the LSTM hidden layer to incorporate additional non-linear features, thereby better reflecting the data's dynamics. The output of this layer can be represented as:

$$r_t = \mathrm{ReLU}(W_n x_t + b_n) \tag{1}$$

In Equation (1), ReLU represents the activation function, W_n denotes the weight matrix for this layer, and b_n is the bias vector.

LSTM Hidden Layer: The model incorporates a two-layer cascade of LSTM networks. These networks perform a series of linear and nonlinear transformations on the input data. The two-layer cascade ensures that the model can learn sufficiently rich features. Specifically, the weight matrices within the network are iteratively updated during the training phase until they converge to optimal values. These weights represent the features extracted by the model from the training data, capturing the underlying patterns and dynamics of the time series. However, since these weights do not possess explicit meaning, we refer to such network layers as hidden layers.

Softmax Classifier: The processed data from the neural network is classified into one of two categories: fall or non-fall. The classifier comprises two main parts: Firstly, a linear transformation layer computes the unnormalized probabilities that the input data belongs to each class.

$$z = W_c h_{n-1} + b_c \tag{2}$$

In Equation (2), W_c denotes the weight matrix for this layer,h_{n-1} represents the output from the LSTM hidden layer, and$b_c$stands for the bias vector. Following this, the probability values are normalized, and the class with the maximum predicted probability is chosen as the final output.

$$p_i = \exp(z_i) \div \sum_{j=0}^{c-1} \exp(z_j) \tag{3}$$

In Equation (3),p_i represents the normalized probability value for each class, where c is the number of classes (in this paper, (c = 2)).

3. Multimodal Fusion

Due to the heterogeneity of accelerometer data and video data, there are significant differences in their data formats, rendering data-level and feature-level fusion approaches inapplicable. Therefore, this paper adopts a decision-level fusion approach, which generates a unified output from multiple distinct decision models. This method preserves the unique characteristics of different sensors, enhances the flexibility of integrating new technologies into the system, and improves robustness against sensor failures.

The dynamic weighted average fusion algorithm assigns varying weights to different data sources or elements and averages these weighted elements to obtain the fused result.Unlike traditional fixed-weight averaging methods, the weights in the dynamic weighted average fusion algorithm are dynamic, adjustable based on factors such as data characteristics, positional relationships, and temporal variations. Dynamic weighted average fusion is a crucial method in decision fusion, capable of dynamically adjusting weights according to the actual data situation, thereby improving the accuracy and reliability of the fusion results. This method integrates probability predictions from both video and accelerometers, as detailed in Algorithm 1.

```
Require: x_m, y_m, x_n, y_n, w_v
Ensure: Weighted average probability & predicted class
function WtAvgFus(x_m, y_m, x_n, y_n, w_v)
    if x_m=None ∨ y_m=None then
        Handle missing predictions
    end if
     t_m ← (x_m x_n) + (y_m y_n)
     s_n ← x_n + y_n
```

ISSN 2616-5775 Vol. 7, Issue 11: 78-85, DOI: 10.25236/AJCIS.2024.071111

```
    return  t_m/s_n if s_n>0 else t_m
end function
function UPDWTS(x_c, y_c)
    T_c=x_c+y_c
    return (x_c/T_c, y_c/T_c) if T_c>0 else(0.5,0.5)
end function
function GetConf(probs)
    Calculate avg. confidence from model probs.
    Initialize confidence ←[ ]
    for chunk in probs do
        if chunk is not empty then
            Append max(chunk) to conf
         end if
        end for
        return mean (conf) if conf not empty else 0
 end function
 Init. x_mn, y_mn with w_v
 for ∀segment do
    f_m←wtavgfus(x_m, y_m, x_n, y_n)
    pred_c ← argmax(f_m)
    c_w=c_w+1
    if c_w mod w_v=0 then
        x_c, y_c ← Getconf(x_mn, y_mn)
        x_n, y_n ← updwts(x_c, y_c)
    end if
 end for
```

Initially, each modality's prediction is assigned an equal weight of 0.5. Subsequently, these modalities are processed independently through dedicated models. Table 1 explains the abbreviations used in the pseudocode.

Table 1: Abbreviations Used in Pseudocode

Abbreviation	Description
x_m, y_m	Accelerometer, Video Probability
x_w, y_w	Accelerometer, Video Weights
w_v	Batch Size
WtAvgFus	Weighted Average Fusion Function
UpdWts	Update Weights Function
GetConf	Get Confidence Function
t_m	Weighted Probability
s_n, T_c	Total Weight, Total Confidence
f_m	Fused Probability
$pred_c$	Predicted Class
x_{mn}, y_{mn}	Audio, Video Probability Windows
conf	Confidences List
cw	Current Batch
x_c, y_c	Accelerometer, Video Confidence

The "GetConf" function plays a pivotal role by calculating the average confidence level for each modality. It achieves this by extracting the maximum confidence value from the probability distribution of each segment, providing a quantitative basis for weight adjustment. Following this, the "WtAvgFus" function combines predictions and dynamically adjusts weights to favor more confident modalities, thereby enhancing prediction performance. This dynamic adjustment is implemented through the "UpdWts" function, which updates the weights of the two sources based on their cumulative correctness (ac and vc). If the total correctness is greater than 0, the weights are updated according to the cumulative correctness; otherwise, they are set to equal weights (0.5, 0.5). This feature recalibrates the weights for the next batch of inputs based on the recent confidence levels of the modalities [9].

As defined by a weighting parameter, the approach of processing inputs in batches allows for the accumulation of necessary data to facilitate informed weight adjustments. This ensures that the final prediction not only leverages the strengths of both accelerometerand video inputs but also dynamically

ISSN 2616-5775 Vol. 7, Issue 11: 78-85, DOI: 10.25236/AJCIS.2024.071111

balances their contributions to optimize overall accuracy. The adaptive, confidence-based fusion method enables the system to achieve robust performance under varying conditions, effectively handling fluctuations in the reliability of individual modalities

4. Experiment Procedure

The experiments in this paper employ the UR Fall Detection dataset, which comprises synchronized RGB images and accelerometer data. The samples in the dataset are labeled as various types of falls (FALL) and activities of daily living (ADL). This experiment divides the dataset into an 80% training set and a 20% test set. To enhance the robustness of the system, we utilize ADL instances as negative examples and Fall instances as positive examples to train the classifier.

The dataset encompasses 70 sequences (30 falls + 40 ADL events). The fall activities are recorded using two Microsoft Kinect cameras along with corresponding accelerometer data. In contrast, the ADL events are documented using a single device (Camera 0) and an accelerometer. Sensor data is collected using PS Move (at 60Hz) and x-IMU (at 256Hz) devices.

In the case of triaxial accelerometers, variations in sensor position and orientation lead to different acceleration values along the three axes. Therefore, to reduce data volume and computational complexity, as well as to eliminate the influence of sensor orientation, this paper performs vector summation on the raw triaxial acceleration data, combining accelerations from all directions into a single resultant acceleration value. The calculation is shown in Equation (4).

$$W = \sqrt{a^2 + b^2 + c^2} \tag{4}$$

In this context, refers to the components of acceleration along the three axes. This paper employs sensitivity as the primary metric to evaluate the classification results. Sensitivity, also known as the true positive rate, represents the proportion of all positive samples that are correctly predicted. In the context of fall detection, missing even a single fall event can have irreversible consequences for elderly individuals, making sensitivity a crucial indicator for assessing the performance of a fall detection model. It reflects the model's ability to detect fall behaviors, with lower sensitivity indicating a higher rate of missed detections or false negatives. The sensitivity is calculated as shown in Equation (5).

$$\text{sensitivity} = \frac{TP}{TP + FN} \tag{5}$$

Here, TP stands for True Positives (events where a fall actually occurred and were correctly identified as falls), while FN represents False Negatives (events where a fall occurred but were incorrectly identified as non-falls).

This experiment compares the sensitivity of three types of fall detection models: a model based solely on accelerometer sensors, a model based on video data, and a model based on the fusion of heterogeneous sensors. The objective of this comparison is to evaluate the recognition capability of these fall detection models for identifying fall events. By analyzing the sensitivity of each model, we aim to gain insights into their effectiveness in detecting falls accurately, ultimately guiding the selection of the most suitable approach for real-world applications. The experimental results are shown in Table 2 below.

From analyzing the data in Table 2, it can be observed that the fall detection model based on accelerometer sensors exhibits the highest sensitivity, which is largely attributed to the variability in human activities. The sensitivity of the fall detection model based on heterogeneous sensor fusion, at 94.8%, demonstrates a notable improvement of 0.8% and 5.1% over the accelerometer-only and video-only models, respectively, highlighting its significant advantage in accuracy.

Table 2: Sensitivity experiment comparison

Data	sensitivity
Accelerometer	94%
video	89.7%
Accelerometer+video	94.8%

In the task of fall detection, specificity is another crucial metric for evaluating the performance of fall detection models. Specificity refers to the proportion of all negative samples that are correctly predicted, reflecting the model's ability to avoid misclassifying normal activities of the elderly as falls.

ISSN 2616-5775 Vol. 7, Issue 11: 78-85, DOI: 10.25236/AJCIS.2024.071111

If a fall detection model solely pursues high sensitivity at the cost of specificity, it can result in an extremely high false alarm rate. While high false alarms may not lead to severe consequences, frequent false alerts can significantly disrupt users' daily lives and erode their trust in the fall detection model. The specificity is calculated as shown in Equation (6).

$$\text{specificity} = \frac{TN}{TN + FP} \tag{6}$$

Here, TN stands for True Negatives (events where no fall occurred and were correctly identified as non-falls), while FP represents False Positives (events where no fall occurred but were incorrectly identified as falls).

This experiment compares the specificity of three types of fall detection models: a model based solely on accelerometer sensors, a model based on video data, and a model based on the fusion of heterogeneous sensors. The purpose of this comparison is to evaluate the recognition capability of these fall detection models for distinguishing between fall and non-fall events. By assessing the specificity of each model, we aim to gain insights into their ability to accurately differentiate normal activities from falls, ultimately contributing to the selection of the most suitable model for real-world applications that minimizes false alarms and maintains user trust.The experimental results are shown in Table 2 below.

Table 3: Specificity experiment comparison

Data	specificity
Accelerometer	91%
video	95.4%
Accelerometer+video	96.6%

Analyzing the data in Table 3 reveals that the fall detection model based solely on accelerometer sensors has the lowest specificity, which is closely related to the randomness of human activities. In contrast, the specificity of the fall detection model based on heterogeneous sensor fusion stands at 96.6%, representing an improvement of 5.6% and 1.2% over the other two models, respectively, demonstrating a clear advantage in reducing false alarms.

5. Conclusion

Addressing the issues of high false alarm rates and resource consumption in fall detection systems utilizing single or homogeneous sensors, this paper investigates the development of a fall detection model based on the fusion of heterogeneous sensor data, specifically incorporating accelerometer and audio data. The research focuses on both single-sensor fall detection and fall detection through heterogeneous sensor data fusion, aiming to minimize false alarms while maintaining high sensitivity in identifying fall behaviors.

For accelerometer sensor data, a sliding time window is employed to extract time-series sensor information, and the raw data undergoes minimal preprocessing such as normalization before being fed into the model. The neural network automatically learns and extracts features from the data during training, and a classifier is then utilized to determine whether the data represents a fall event. This approach eliminates the need for manual feature extraction, enabling an end-to-end processing pipeline from raw data to classification results.

For video data, the YOLO v5 network model is adopted, and the fall detection results obtained from these two distinct data sources are fused at the decision level using a weighted average. Experimental results indicate that compared to single-source data detection, the fusion of data from two different sources enhances the accuracy and robustness of the human fall detection system.

This system effectively meets the needs of intelligent monitoring for the elderly and has the potential to contribute to addressing the challenges of aging societies and enhancing social care capabilities in the future.

Acknowledgment

This work was supported by Natural Science Basic Research Program of Shaanxi (2020JQ-734).

ISSN 2616-5775 Vol. 7, Issue 11: 78-85, DOI: 10.25236/AJCIS.2024.071111

References

[1] World Health Organization. Global Health Estimates: Life expectancy and leading causes of death and disability[EB/OL]. Retrieved from https://www.who.int/data/gho,2024.
[2] SUCERQUIA, Angela; LÓPEZ, José David; VARGAS-BONILLA, Jesús Francisco. Real-life/real-time elderly fall detection with a triaxial accelerometer[J]. Sensors, Volume 18, Issue 4，pp.1101,2018.
[3] Wilk B, Augustyn M, Wilk G. Algorithm for human fall detection based on acceleration measurement[C]. Signal Processing: Algorithms, Architectures, Arrangements, and Applications. Poznan,pp.13-17,2020.
[4] Steven Eyobu O, Han D S. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network[J].Sensors, Volume 18, Issue 9,pp. 2892,2018.
[5] Ramirez H, Velastin S A, Meza I, et al. Fall detection and activity recognition using human skeleton features[J]. IEEE Access, Volume 9,pp. 33532-33542,2021.
[6] Chen E Q, Bai X, Gao L, et al. A spatiotemporal heterogeneous two-stream network for action recognition[J]. IEEE Access,Volume 7,pp.57267-57275,2019.
[7] NÚÑEZ-MARCOS, Adrián; AZKUNE, Gorka; ARGANDA-CARRERAS, Ignacio. Vision-based fall detection with convolutional neural networks[J]. Wireless communications and mobile computing, Volume 2017, Issue 1,pp.9474806,2017.
[8] KWOLEK, Bogdan; KEPSKI, Michal. Human fall detection on embedded platform using depth maps and wireless accelerometer[J]. Computer methods and programs in biomedicine, Volume 117, Issue 3, pp.489-501,2014.
[9] ALLA, Ildi, et al. From Sound to Sight: Audio-Visual Fusion and Deep Learning for Drone Detection[C]. In: Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 123-133,2024