

RESEARCH

Open Access



A study on implementation of real-time intelligent video surveillance system based on embedded module

Jin Su Kim¹, Min-Gu Kim¹ and Sung Bum Pan^{2*}

*Correspondence:

sbpan@chosun.ac.kr

² Department of Electronics Engineering, Chosun University, Gwangju, Korea
Full list of author information is available at the end of the article

Abstract

Conventional surveillance systems for preventing accidents and incidents do not identify 95% thereof after 22 min when one person monitors a plurality of closed circuit televisions (CCTV). To address this issue, while computer-based intelligent video surveillance systems have been studied to notify users of abnormal situations when they happen, it is not commonly used in real environment because of weakness of personal information leaks and high power consumption. To address this issue, intelligent video surveillance systems based on small devices have been studied. This paper suggests implement an intelligent video surveillance system based on embedded modules for intruder detection based on information learning, fire detection based on color and motion information, and loitering and fall detection based on human body motion. Moreover, an algorithm and an embedded module optimization method are applied for real-time processing. The implemented algorithm showed performance of 88.51% for intruder detection, 92.63% for fire detection, 80% for loitering detection and 93.54% for fall detection. The result of comparison before and after optimization about the algorithm processing time showed 50.53% of decrease, implying potential real-time driving of the intelligent image monitoring system based on embedded modules.

Keywords: Intelligent video surveillance system, Intruder detection, Fire detection, Loitering detection, Fall detection

1 Introduction

Recently, as incidents and accidents are increasing, for example, murders and domestic fires and fall accidents of elderly people happening indoors and outdoors, people are thus increasingly interested in their safety. Therefore, more and more CCTVs are installed to prevent such incidents and accidents and take action quickly, and integrated control centers are established for integrated management of CCTVs operated for various reasons to increase efficiency [1].

The conventional CCTV surveillance system in an integrated control center is a method for one person to monitor a plurality of monitors by means of CCTVs. However, when a person monitors a plurality of CCTVs, 45% of incidents occurring after 10 min are not identified, and 95% thereof are not identified after 22 min [2]. The statistics

of CCTVs and monitoring personnel in each area show that one person monitors 40 CCTVs on the average, implying difficulty in providing effective monitoring service (Table 1) [1].

With respect to 456 dongs where crimes occurred in Seoul, 113 dongs show high crime rate, and investigation shows 113 dongs have installed a smaller number of CCTVs. This implies that crime rates are related to the number of CCTVs, and CCTVs can contribute to preventing crimes [3]. However, an important thing is crime prevention by taking measures after crimes occur, not crime prevention through monitoring CCTVs by CCTV monitoring personnel before crimes occur.

An intelligent video surveillance system as a method for detecting abnormal activities before damages by crimes happen has been studied for notifying abnormal situations when they happen in video without monitoring CCTV by monitoring personnel. The intelligent video surveillance system aims detections including intruder detection, fire detection, loitering detection, fall detection, and assault detection. Among other detections, intruder and fire detections are studied more. However, the activity carried out before most crimes happen is the relevant criminal loitering around the environment for the crime. Keeping an eye on loitering can prevent various crimes beforehand [4]. Moreover, if there is no system for notifying fall accidents when they occur, worse things can happen [5].

At present, although the intelligent video surveillance system is studied on the basis of computers, it involves weakness of high power consumption and personal information leaks, system construction and maintenance costs, and it is thus highly difficult to install the system for real application. To address the issues, an intelligent video surveillance system based on small devices has been studied because of its low cost, light weight and small power consumption for driving it.

This paper suggests an intelligent video surveillance system based on embedded modules. The system uses information it learns to detect intruders, color and motion information to detect fires, balance points to detect loitering, and human body motion information to detect fall. A library is configured in order to use embedded modules, and conducts 5-algorithm optimization and 3-embedded module optimization to reduce processing time for the intelligent surveillance system. The result of driving the intelligent video surveillance system showed performances of 88.51% for intruder detection, 92.63% for fire detection, 80% for loitering detection, and 93.54% for fall detection. It was shown that the algorithm for detecting abnormal situations by the intelligent video surveillance system reduced 50.53% of processing time before and after optimization.

This paper is configured to have the following chapters. Section 2 describes the conventional abnormal situation detection algorithm and the intelligent video surveillance system. Section 3 describes the abnormal situation detection algorithm using multiple

Table 1 Status of public CCTV and control personnel

List	Gunpo	Gwangmyeong	Anseong	Suwon
Public CCTV	705	1527	574	1131
Control personnel	13	33	18	36
Monitor per person CCTV	54.23	46.27	31.89	31.42

sources. Section 4 describes implementation of the suggested intelligent video surveillance system. Section 5 describes the result of experiment for the implemented intelligent video surveillance system. Section 6 describes conclusion of this paper.

2 Related works

An abnormal situation is defined as a situation different from general behaviors and environment from the standard personal, statistic, socio-cultural and professional viewpoint. It is classified into intrusion, arson, loitering, fall and assault. There are various conventional algorithms for detecting abnormal situations depending on the type of detection, and conventional intelligent surveillance systems use computers or small devices.

2.1 Abnormal situation detection algorithm

The initial face recognition method for detecting intruder is to use principal component analysis (PCA) to recognize faces by using the eigenface obtained through PCA in the face image [6]. Other exemplary methods include the method for using images reconfigured with the discrete cosine transform (DCT) coefficient for the face images to improve PCA performance [7]; the method for using combination of PCA with linear discriminate analysis (LDA) [8]; and the method using PCA and LDA for the face image by the distances of 1 to 5 m [9]. Another method is to extract features with local binary pattern (LBP) to classify them by using Convolution Neural Networks (CNN) [10]. The last method is to recognize faces with Restricted Boltzmann Machines (RBM) of deep learning to use face expression change, lighting change and changing angular images [11].

The method for detecting fires is divided into the method for using sensors and the method for using images. The method for using sensors is divided into the method for detecting fires by using the values obtained by using temperature sensors, smoke sensors and CO₂ sensors as parameters of the Fuzzy logic [12], and the method for using a device made by combining 8 sensors of AMS MOX sensors, PID sensors, and NDIR CO₂ sensors et [13].

The color-based fire detection method is divided into the method for using additional information including spreading fires after detecting fire colors in the RGB color space [14], the method for detecting fires by calculating standard deviations of colors in various fire environments [15], and the method for detecting fires by using features of HIS color space and optical flows [16]. Another method is to detect fires with support vector machine (SVM) by using features of HIS color space, 2-dimensional discrete wavelet transform (DWT), pixel ratios and optical flow [17].

The method for identifying loitering is divided into the method for using time to measure the time of an object staying in an image [1, 18, 19], and the method dividing an image inputted into blocks of n by m to measure the time in each block [20, 20]. This is a method for using the characteristics that a loitering person shows more directional changes than normal people to measure object angles [22]. The method for identifying loitering by using 2 conditions for determining loitering is divided into the method for measuring the object time and the block time [23], the method for measuring the object time and using angles [24], and the method for using angles and the object movement distance [25].

The method for detecting fall is divided into the method for using sensors and the method for using images. The method for using sensors is divided into the method for attaching 3-axis accelerometer sensors to a body to use sensor values and acceleration values [26], and the method for using accelerometer sensors and pressure sensors [27]. Moreover, a method for using accelerometer sensors of mobile phones is the method for saving normal acceleration patterns as Activities of Daily Living (ADL) to compare them with ADL on the basis of real-time nearest neighbor rule (NNR) [28].

The method for using images is divided into the method for setting up a circle changing depending on object motion, and using circle changes, vertical and horizontal histogram feature values to determine fall with SVM [29], the method for setting up a bounding box in a detected object and using changing acceleration of the bounding box [30], and the method for using aspect ratios, effective area ratios of a concerned object, object feature points, axis angles, and contour ratios to determine fall with SVM [31].

2.2 Intelligent video surveillance system

The intelligent video surveillance system for issuing an alert when an abnormal situation occurs in video is divided into the method for using computers and the method for using small devices. The computer-based intelligent video surveillance system conducts various detections, and have been studied. However, they involve installation and maintenance costs, high power consumption and personal information leaks, and are thus not ideal to be used in real environment.

An exemplary conventional computer-based intelligent video surveillance system is an object tracking system by using a plurality of cameras. The videos inputted by digital signal processor (DSP) based IP cameras are encoded by using audio video coding standard (AVS) and sent to the IP network by means of real-time streaming protocol (RTSP). The IP network uses GPU for reducing processing time to conduct distributed processing [32].

Second, another system is the fall detection system based on body contours. This is a system for using videos inputted by cameras in real time so that computers can detect fall and send the information to a hospital server through its network to save and monitor the information. The fall detection system uses Gaussian mixture model (GMM) in the inputted images to detect objects and uses aspect ratios and tilt angle features of a human body contour with less computation for determining the fall [33]. The computer-based intelligent video surveillance system sends the videos inputted by each camera to the central server. The central server conducts various detections in combination, for example, intruder detection, fire detection, loitering detection, and fall detection.

An exemplary conventional intelligent surveillance system based on small devices is the intruder detection system using Raspberry Pi and Arduino. The system detects motions in inputted images by using MOG2, and determines human bodies by using sizes thereof. The system detects faces by using Haar-like features, and detects intruders through fisherfaces-based face recognition. When an intruder is identified, a relevant user is issued with an alert through e-mail, and can view the video remotely through a web interface [34].

Second, another exemplary system for detecting intruders uses Raspberry Pi. When a motion is sensed in an inputted video, this system saves the sensed video in the cloud

server for later examination. This system detects objects by using differential images, and uses Haar-like features to determine human bodies. When the object is identified as a human body, this system uses the GSM module to send messages to a relevant user in order to notify the user of the abnormal situation [35].

Last, the unmanned aerial vehicle (UAV) fire detection system uses QuaRC-based single Gumstix. This system uses fire color information and motion information to determine fires, and uses the Lab color space to use color information. It also uses optical flow to use motion information. This system uses sliding mode control (SMC) and linear quadratic regulator (LQR) to reduce calculation time and prevent chattering [36].

2.3 Optimization

There are some studies that reduce processing time by using various optimization methods when using small devices with lower specifications than a computer. First, there is a study that reduces processing time in low specification mobile environments by changing the front-end browser loading method. By classifying data types such as text and images, a text layout with a relatively low load is displayed to improve the experience speed, while rendering reconstruction is performed after the screen is displayed. In addition, image size is determined to reduce processing time by reducing or decreasing image quality when rendering takes a long time [37].

T.W. Lee conducted research to reduce embedded module boot time and speed processing of applications. Software suspend using the principle that the memory state and registers are changed when the program is changed from the operating state to the suspend state, the root file system that improves the decompression efficiency to reduce weight, the JFFS2 file system with high compression efficiency and low process usage, etc. The embedded modules used in the experiment is the XP-100 model manufactured by Huins [38].

J.W. Kang analyzed the structural features and speed-reduction factors of mobile enterprise application platform (MEAP) and then performed research to reduce the processing speed of mobile applications using front-end optimization speed improvement techniques. Add an expansion or a cache-control header is carried out to request process resources from a local location, not from a server. Saving server resources required for compression using gzip components, optimization using minimize HTTP requests is conducted to reduce HTTP requests by merging scripts divided [39].

Finally, there is a study by C. C. Paglinawan, who performed optimization on raspberry pi. C. C. Paglinawan performed optimization for the real-time operation of the vehicle speed calculation system, and shortened the vehicle detection time by developing the GMM for vehicle detection and the Kalman Filter (KF) for vehicle tracking in OpenCV. In addition, sparse random projection (SRP) using scikit-learn reduced processing time through image compression that projects high-dimensional video frames into low-dimensional partial spaces [40].

The intelligent video surveillance system based on small devices uses videos inputted by cameras to conduct single detections, for example, intruder detection, fire detection and motion detection. It is required to further study the intelligent video surveillance system based on small devices which conducts integrated detection in order to address weakness of personal information leaks and high power consumption

of the computer-based intelligent video surveillance system and use the system efficiently in real environment. In addition, the optimization process is required for the real-time operation of intelligent video surveillance system in low specification devices.

3 Proposed method for intelligent video surveillance system

Figure 1 shows a flowchart of the suggested intelligent video surveillance system. This system detects a moving object with the adaptive differential images in the video inputted by the embedded module, and conducts erosion for accuracy. When an object is not detected for a given time, the background image is updated, and the object size information is used to determine a human body when an object is detected. If it is not a human body, color information and motion information is used to detect fires. When it is determined as a human body, the system conducts intruder detection by using taught information, loitering detection by using balance point angle changes and movement distances, and fall detection by using human body motion information and acceleration information. When an abnormal situation is identified, this system uses smtp and the mock library to send the images and thus notify a relevance of the identified abnormal situation.

This paper uses TensorFlow and AlexNet for face recognition. AlexNet based on GPU is composed of 5 convolution layers, 3 max pooling, and 3 fully connected layers. The convolution layers extracts features, and is composed of filters for extracting features and an activation function for changing filter values into non-linear values. Max pooling is for reducing the size of data obtained by applying the filters in the convolution layers. The fully connected layers use extracted feature values to classify data.

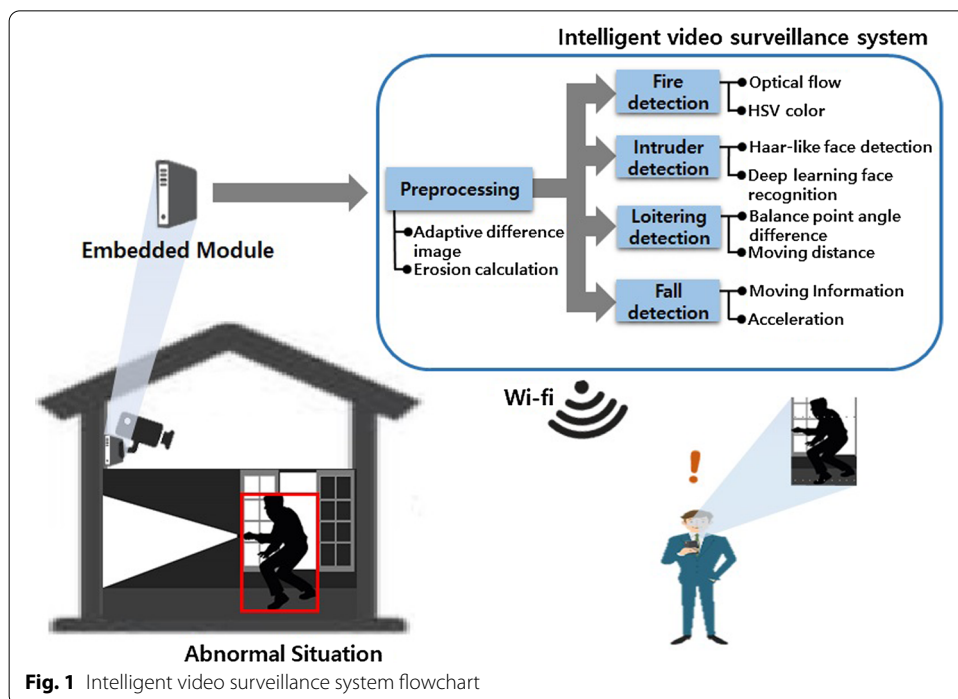


Fig. 1 Intelligent video surveillance system flowchart

3.1 Intruder detection

As shown in Fig. 1, the system conducts face detection based on Haar-like features to detect intruders. When a face is detected, the system uses deep learning to conduct face recognition based on the taught information. Deep learning is a type of machine learning for image recognition through combination of various non-linear transformation techniques, and examples thereof include TensorFlow, Cafe and Torch. TensorFlow is an open source library made by Google, and globally used at present. Because data flow of TensorFlow shows the structural view of parameter changes through graphs, it is easy to know connections and data flows.

This paper uses TensorFlow and AlexNet for face recognition. AlexNet based on GPU is composed of 5 convolution layers, 3 max pooling, and 3 fully connected layers. The convolution layers extract features, and is composed of filters for extracting features and an activation function for changing filter values into non-linear values. Max pooling is for reducing the size of data obtained by applying the filters in the convolution layers. The fully connected layers use extracted feature values to classify data (Fig. 2).

3.2 Fire detection

In this paper, fires are detected by using fire motion information using fire color information which uses the HSV color space like the flow shown in Fig. 1, and optical flow [16]. While the primary color information RGB color space has values ranging from 0 to 255, non-uniform surface light can result in different brightness or chroma on the entire surface even with the same color. It is necessary to convert the HSV color space composed of colors, chroma and brightness for detection by using colors [41].

The optical flow is a method for using object to estimate the object moving in an image. A light and dark patterned motion in an image is defined with speed vectors u and v , to add them to the object in the video, and the area at the same speed in the videos is determined as the information of one object for detection [42]. For detecting fire motion, the feature extraction method by Shi and Tomas is used to detect feature points in a video, and tracking points repeat while using the optical flow by Lucas-Kanade. The motion speeds u and v of light dark patterns are used to calculate Eqs. (1) to (3) for determining fires:

$$Pu = u_{t-1}, Pv = v_{t-1}, Nu = u_t, Nv = v_t, \tag{1}$$

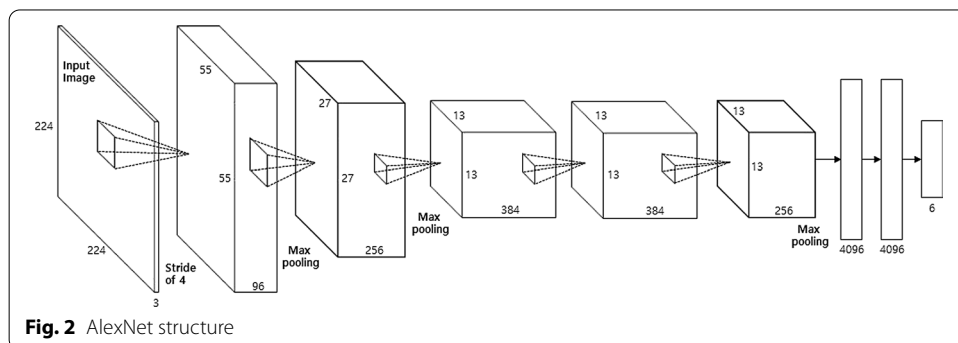


Fig. 2 AlexNet structure

$$U = P_u - 3 \left(\sqrt{(P_v - N_v)^2 + (P_u - N_u)^2} \right) \cos \left(\arctan \frac{P_v - N_v}{P_u - N_u} \right), \tag{2}$$

$$V = P_v - 3 \left(\sqrt{(P_v - N_v)^2 + (P_u - N_u)^2} \right) \sin \left(\arctan \frac{P_v - N_v}{P_u - N_u} \right). \tag{3}$$

Equation 4 is for using fire color information and the optical flow using the HSV color space to determine fires with fire motion information. In Eq. 4, TH_H is a threshold for H representing colors.

$$H > TH_H \text{ and } S > TH_S \text{ and } V > TH_V \text{ and } P_v - V > TH_v \text{ and } P_u - U > TH_u \tag{4}$$

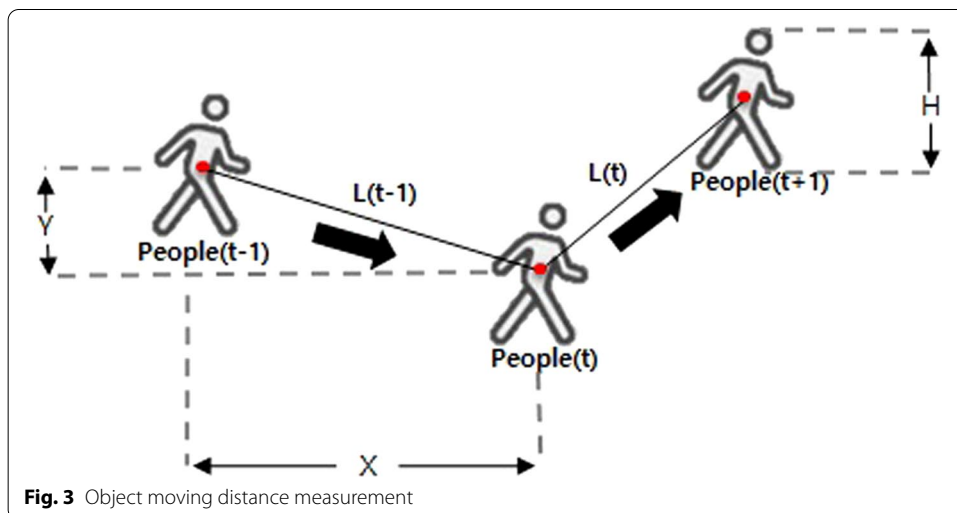
3.3 Loitering detection

Loitering is defined as moving from place to place of a special space without a fixed plan [3], and the activity before most crimes happen is that the relevant criminal loiter around the environment surrounding a crime target location. Therefore, crimes can be prevented by detecting and closely looking into loitering people to prevent crimes beforehand [4]. In this paper, the following 2 conditions are applied to using walking patterns of abnormal pedestrians described above and thus detect wandering people [43].

Condition 1: Distance of balance point movement of object.

Condition 2: Balance point angle changes of object.

Figure 3 shows the method for measuring movement distance of the object. If the movement distance from $t-1$ to t in the object is $L(t)$, $L(t-1)$ and $L(t)$ are calculated by using coordinate changes of the balance point of the object. The balance point of object normalizes the trajectory and uses the current center of gravity in every predetermined section to reduce wrong detection caused by noise. Equation 5 is used to calculate the distance $L(t-1)$ moving from $t-1$ to t in the object. X is the object length change; Y is the object height change; H is the object height for correcting wrong calculation of different movement distance measurement depending on the inputted video and the distance of the object:



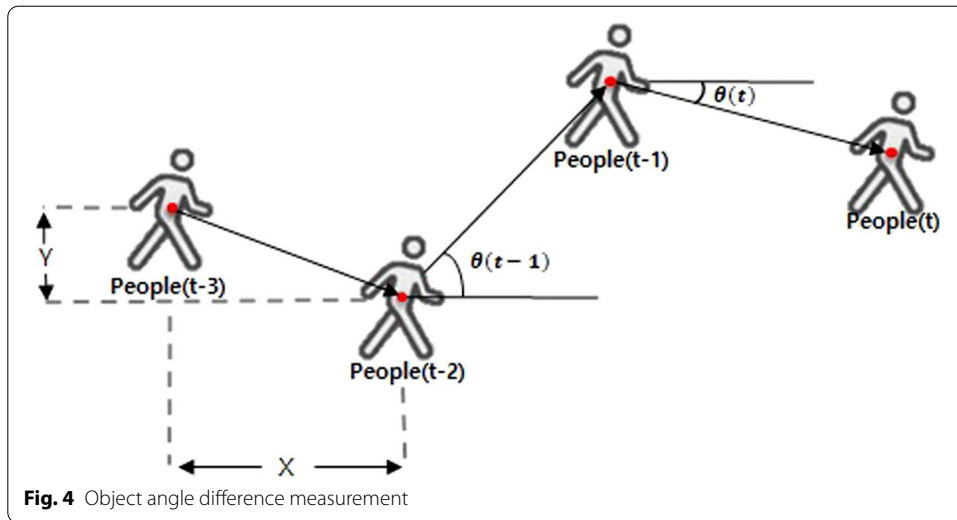


Table 2 Weight according to angle difference

Angle difference	Weight
- 5~0 or 0~5	0.01
- 25~- 5 or 5~25	0.02
- 65~- 25 or 25~65	0.04
- 125~- 65 or 65 or 125	0.07
Under - 125 or Above 125	0.10

$$L(t - 1) = \frac{\sqrt{X^2 + Y^2}}{H} \tag{5}$$

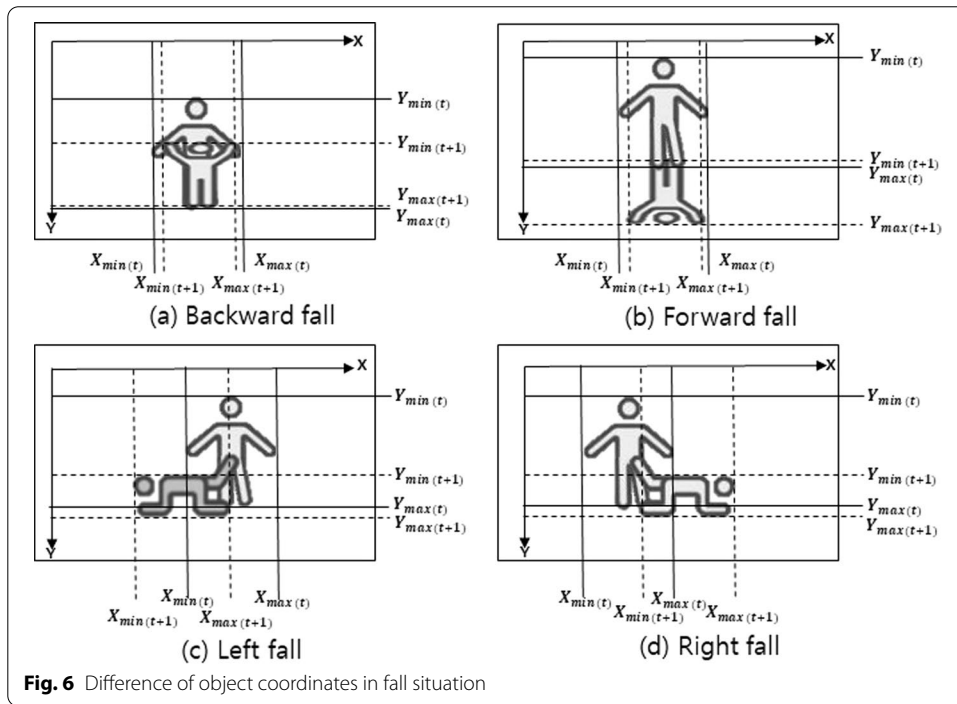
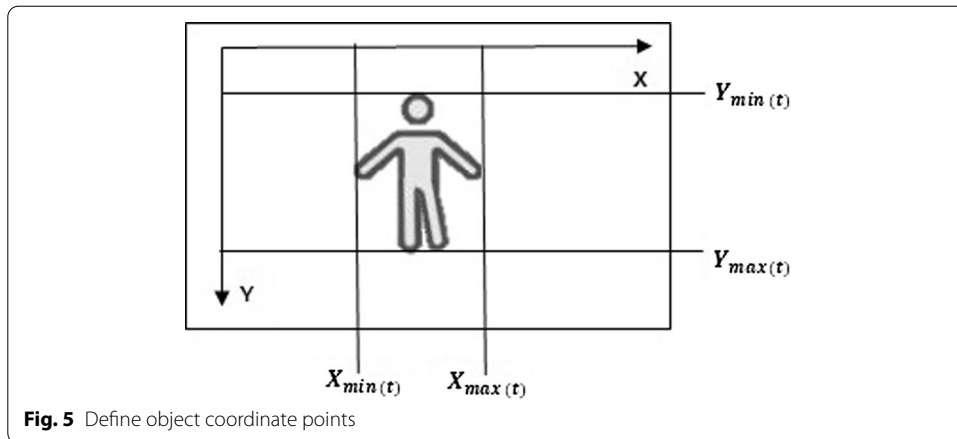
Figure 4 is a method for measuring changing angles of the object. The angle created when the object is moving from $t-2$ to $t-1$ is $\theta(t-1)$, and the angle created when it is moving from $t-1$ to t is $\theta(t)$. The angle is calculated by using X and Y changes of the balance point of the object. Equation 6 is used to calculate the angles. Angle changes are calculated by using the difference between $\theta(t-1)$ and $\theta(t)$:

$$\theta(t) = \tan^{-1}\left(\frac{Y}{X}\right). \tag{6}$$

Table 2 illustrates angle changes and weight standard of the object. The angles changing right and left from the direction of progress of the current object are grouped in 5 levels to calculate the weight corresponding to each group. Small weights are given to small angle changes, and great weights are given to great angle changes to achieve sensitivity to directional changes of the object.

3.4 Fall detection

As shown in the flow of Fig. 1 to detect fall, the motion information and the acceleration information of the object are used [5]. The reference points used for calculating the



motion information and the acceleration information are Y_{min} , Y_{max} , X_{min} and X_{max} , among which Y_{min} is the minimum coordinate of axis Y; Y_{max} is the maximum coordinate of axis Y; X_{min} is the minimum coordinate of axis X; and X_{max} is the maximum coordinate of axis X as shown in Fig. 5.

Figure 6 shows coordinate changes when the normal state changes into a fall state. Figure 6a shows a backward-fall with increasing Y_{min} and Y_{max} . Figure 6b shows a forward-fall with increasing Y_{min} . Figure 6c shows a left-fall with increasing Y_{min} but decreasing X_{min} and X_{max} . Figure 6d shows a right-fall with increasing Y_{min} , X_{min} and X_{max} . The fall state shows increasing Y_{min} , and the right and left-fall state shows apparently increasing and decreasing X_{min} and X_{max} . These features are used to detect fall based on motion information obtained by Eq. 7:

$$F_a = \frac{X_{\max} + Y_{\min}}{2}, \quad F_b = \frac{X_{\min} + Y_{\max}}{2} \quad (7)$$

Equation 8 is used to calculate the acceleration information of the object by using the acceleration information of the object as a condition for determining fall. In Eq. 8, t is time, and V_t is an acceleration at t . Equation 9 is used to determine fall by using the method for using motion information of the object and acceleration information:

$$V_t = Y_{\min}(t) - Y_{\min}(t - 1) \quad (8)$$

$$F_a > F_b \text{ and } V_t > TH_V \quad (9)$$

4 Embedded module implementation and optimization

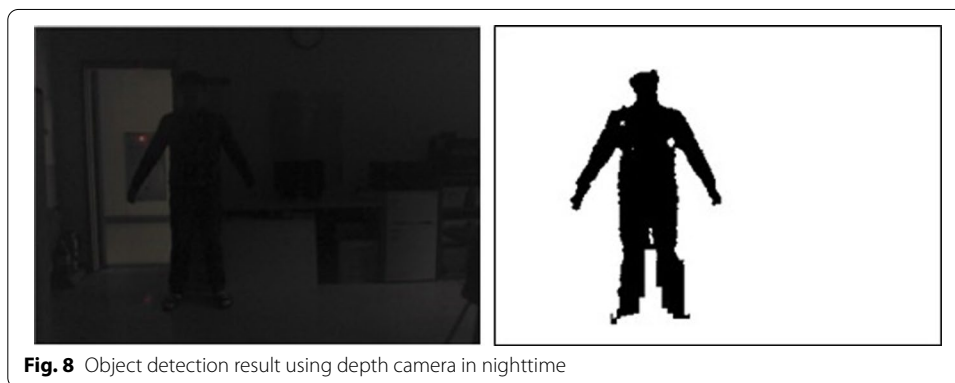
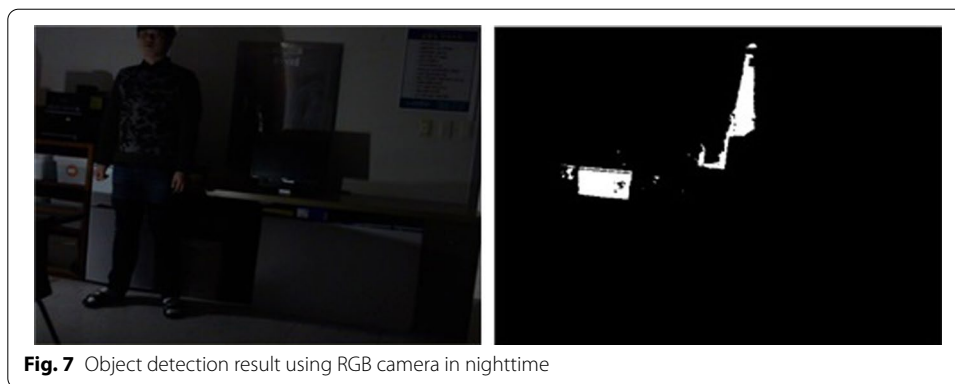
The intelligent video surveillance system is classified into the computer-based method and the method based on small devices. Although the computer-based intelligent video surveillance system is actively studied at present, it involves weakness of high power consumption and personal information leaks, and is thus not preferred in actual application. Furthermore, the intelligent video surveillance system based on small devices currently conducts non-combination detection, for example, motion detection, or person detection. Therefore, the system implemented in this paper is an intelligent video surveillance system for intruder detection, fire detection, loitering detection and fall detection by using small devices driven with strength of low cost, light weight and small power consumption. Five algorithm optimization methods and 3 embedded module optimization methods are applied in order to reduce processing time of the implemented intelligent video surveillance system.

4.1 Embedded module implementation

An embedded module is defined as a system designed by mounting a microprocessor which is a brain of a machine or device to perform specific tasks. The embedded module features driving with low cost, light weight and small power consumption. Although available open source hardware types are Raspberry Pi, Orange Pi, BeagleBoard, and Arduino, Raspberry Pi is used the most in many fields because of its superior universality and performance. Raspberry Pi is a credit card-sized single-board computer, and ideal as a small device of the real-time intelligent video surveillance system because Raspberry Pi 3 increased its processor from 32 to 64 bits, and its speed from 0.9 GHz to 1.2 GHz in comparison with Raspberry Pi 2 [44].

While object detection is carried out for the videos inputted by means of general RGB cameras, objects cannot be detected or wrong detection can occur in the environment of low lighting conditions and at night as shown in Fig. 7. To address this issue, the Kinect depth camera is used as a device for inputting videos in the embedded module (Fig. 8).

Libfreenect is used in order to use Kinect as a video input device of the embedded module. Libfreenect is a library developed to use Kinect in the Linux environment by openkinect. Five required libraries including python-dev and ipython related to python and opencv are configured to use Libfreenect, and 11 related libraries are configured.



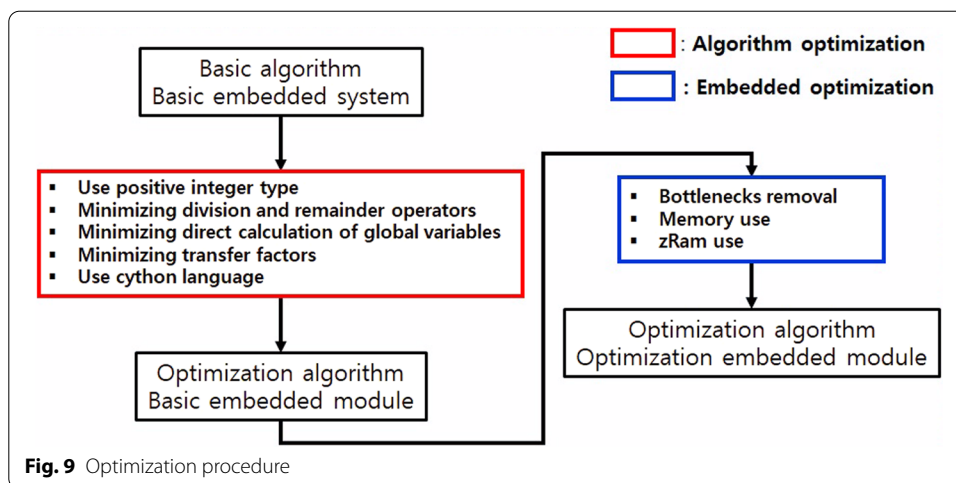
The TensorFlow environment is built by configuring 5 required libraries including pip and linux-armv7l to conduct intruder detection based on face recognition using an embedded module. Fifteen libraries including matplotlib and pkg-config are configured to conduct fire detection, loitering detection and fall detection.

It is required to configure load balancing, swapping, overclocking and Cython to optimize the embedded module and algorithms. cgroup-bin is used for load balancing; swappiness is used for swapping; arm-freq of config is used for changing overclocking; and setuptools is used for using cython. Ssmtp for sending alert when an abnormal situation occurs and the mock library for sending videos are configured.

4.2 Algorithm optimization

It is required the embedded module of lower performance than computers conducts the algorithm and embedded module optimization process to detect abnormal situations in real time. In this paper, 5 algorithm optimization methods and 3 embedded module optimization methods are applied to enhancing processing performance of the embedded module and reducing algorithm processing time as shown in Fig. 9.

The first algorithm optimization method is to use positive integers. The basic operators of the program is integer type, and processing time increases because of the unnecessary process for declaring it in a different variable type, converting and calculating it in the calculation process of driving algorithms and then converting it back to the original variable type. Moreover, because the process is faster than the



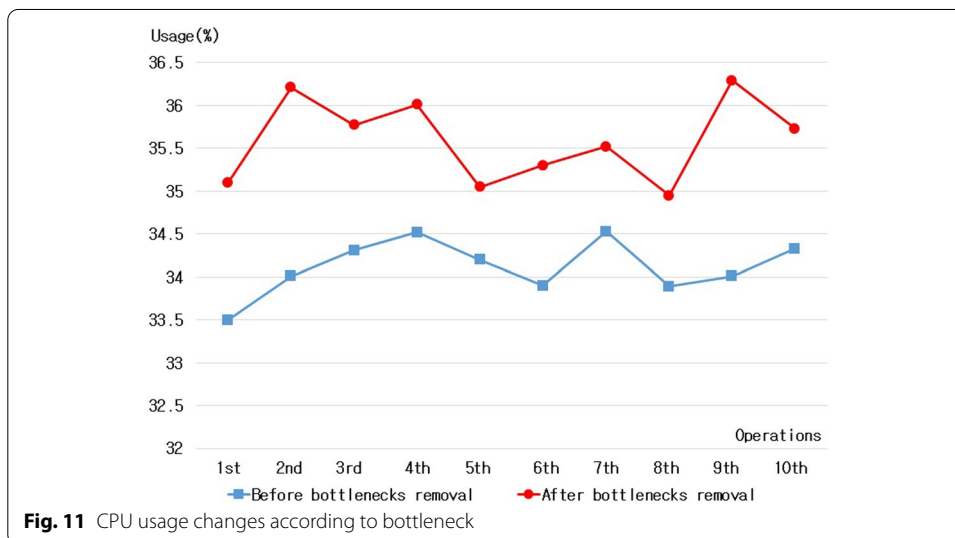
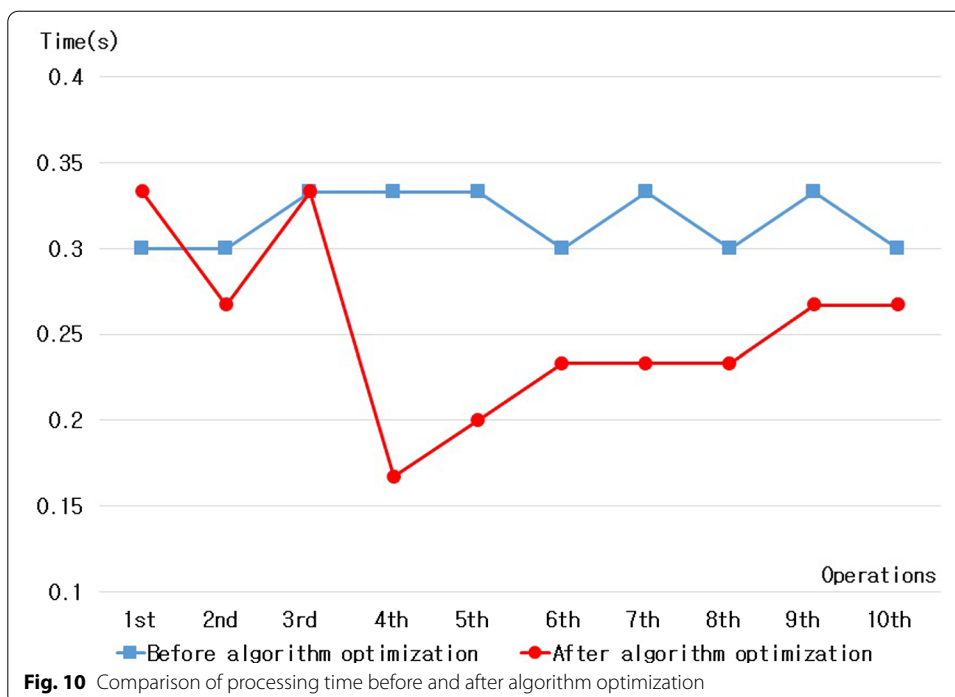
operation case of using both negative integers and positive integers, it is declared as positive integers to be used if the used values are not negatives.

Second, the method is for minimizing division and remainder operators. Because division of a standard process has an execution cycle of 20 to 140 for 32 bits of denominators and numerators, it takes longer time than other operators. Therefore, one shift operation is used among fast operators for operation with 2's multipliers, and multiplication instead of division is otherwise used.

Third, the method is for minimizing direct calculation of global variables. Because global variables cannot be allocated to registers for use thereof, they can be used by the method for allocating pointers indirectly or using function calls. Invoking pointers and function calls cause memory waste to increase processing time, and the compiler experiences overload to repeat reading global variables whenever they are directly used. Therefore, a method is used to avoid the use of global variables, and calculate local variables after substituting values for the local variables to use the global variables.

Fourth, a method is used to minimize transfer factors in function calls. Where there are at least 4 transfer factors in function calls, the factors are sent through a stack to enable memory access as big as the stack. Moreover, when a structure is sent as a transfer factor, all values of the structure are sent through the stack to enable memory access. Therefore, a method is used to declare a structure when there are at least 4 transfer factors, and send pointers for the structure.

Last, a method for using Cython language is used. The Cython language is a type of combining the strength of fast productivity of python language with execution speed of C language, and is Progressively Typed Language or Gradual Typing [45]. Figure 10 shows the processing time of the fire detection algorithm depending on whether the algorithm is optimized, in which the vertical axis depicts processing time, and the horizontal axis depicts the n-th operation. Processing time comparison reveals that the average time is 0.32 s before optimizing the algorithm, and 0.24 s after optimizing the algorithm, suggesting reduced processing time by 25%.



4.3 Embedded module optimization

The first method for optimizing embedded modules is to remove bottleneck. A bottleneck is the situation that performance or capacity of a system is limited by one component, and lowers speed if it occurs in one core of CPU. Moreover, because the number of clocks of CPU is limited, processing speed is thus limited. Therefore, load balancing and overclocking are applied and used to avoid CPU cores to gather in one of them. The arm-freq used in the experiment was established to be 1300. Figure 11 shows CPU use while driving algorithms, and an increase in CPU use after bottlenecks removal in comparison with before state.

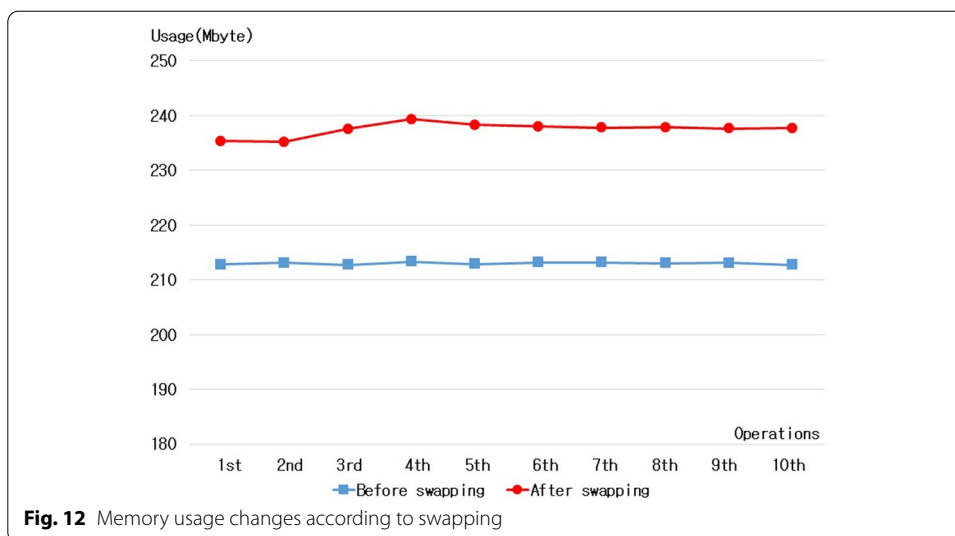


Table 3 Setting value of zRam

```

modprobe zram num_devices=4
totalmem=$(free | grep -e '^Mem:' | awk '{print $2}')
mem=$((totalmem/4)*1024)
echo $mem > /sys/block/zram(0~3)/disksize
mkswap /dev/zram(0~3)
swapon -p 5 /dev/zram(0~3)
    
```

Second, a method is to use memory. The method is divided into a disk-based calculation method and a memory-based calculation method when driving algorithms. However, because the memory access speed is even faster than the disk access speed, swapping is used to conduct memory-based calculation. The experiment was conducted after specifying swapping as 20. Figure 12 shows memory use when driving algorithms, and an increase in memory use after swapping.

Last, a method for using zRam is used. zRam is a method for moving the data in the memory domain to the zRam domain after compression thereof when the memory use is more than a specified level [46]. Although the data movement process can lower speed when a computer with enough memory is used, this is a method for addressing lower speed due to a lack of memory when used in an embedded module with insufficient memory to address the low speed issue (Table 3).

Figure 13 shows the result of processing time comparison depending on the use of embedded optimization. When embedded optimization is not used, the average processing time is 0.25 s, but it is 0.24 s when embedded optimization is used, implying a decrease by 4% in terms of processing time.

5 Experimental results and discussion

The intelligent video surveillance system based on embedded modules and suggested in this paper was implemented by using Raspberry Pi 3, and Kinect v1 was used as a video input device to use a depth camera. Table 4 shows the performance of the abnormal

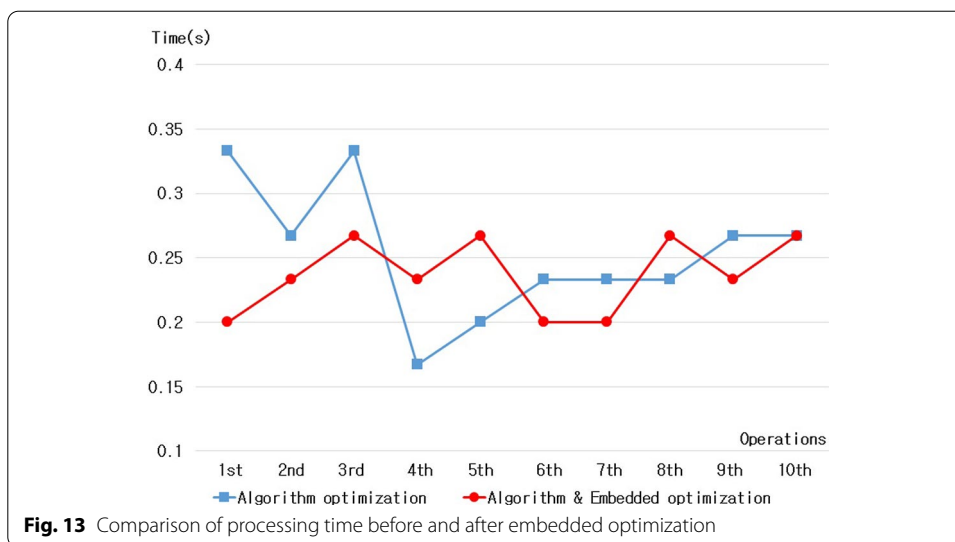
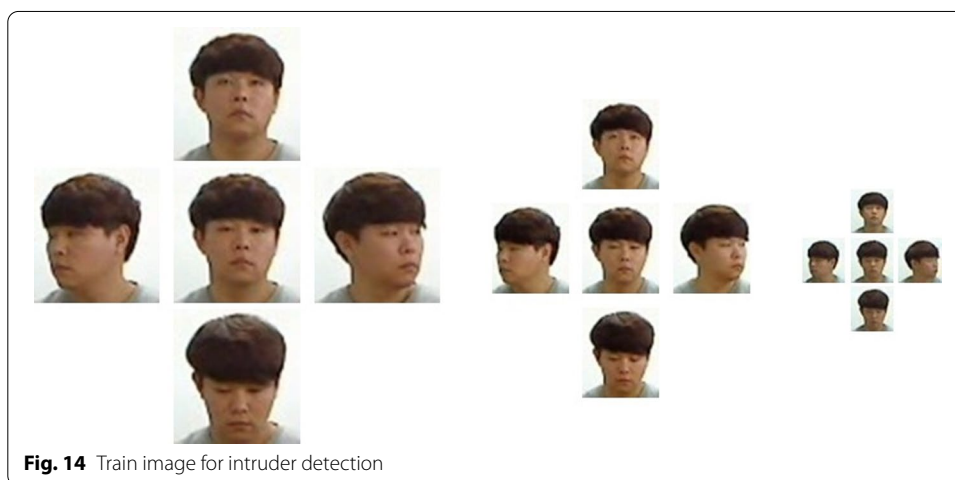


Table 4 Abnormal situation detection algorithm performance

Detection	Database	Performance
Intruder	Real-time experiment	88.51%
Fire	NIST fire database	92.63%
Loitering	PETS database + making database	80%
Fall	Real-time experiment	93.54%



situation detection algorithms, performance of embedded module and PC is the same. For intruder detection, face rotation images for each distance were used to conduct learning with original images and the images when lighting changed as shown in Fig. 14. The intelligent video surveillance system was used to evaluate performance through real-time face recognition experiments. For evaluating fire detection performance, the fire database of NIST was used, and composed of burning sofa and dry wood videos in an indoor residential environment. The PETS database and self-created database were

used to evaluate the loitering detection performance. The PETS database shows at least one person in the video showing indoor and outdoor environments, and specified people among them are shown loitering. The self-created database shows one object loitering in indoor and outdoor environments. For evaluating the performance of fall detection, the intelligent video surveillance system was used to evaluate the performance while driving the system in real time (Figs. 15, 16).

Figures 17, 18, 19, 20 show the result of detected abnormal situation when an abnormal situation of intruders, fires, loitering and fall occurs, and the abnormal situation is remotely sent to notify a relevant user of the abnormal situations.

Table 5 illustrates the performance compared before and after optimizing the intelligent video surveillance system which conducted the method for optimizing 5 algorithms and 3 embedded modules applied in this paper. The integrated processing



Fig. 15 Abnormal situation (fire) database



Fig. 16 Abnormal situation (loitering) database

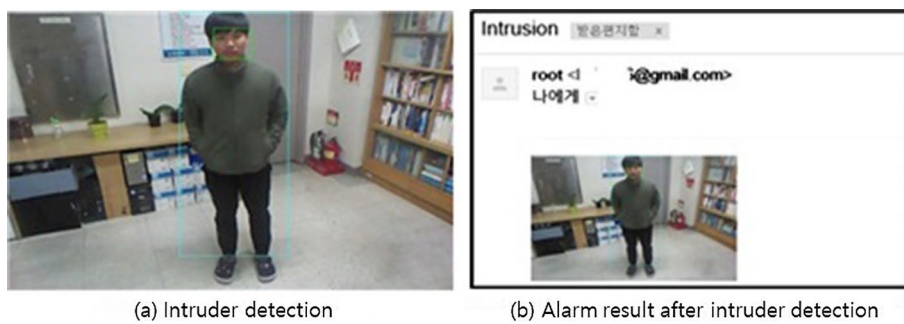


Fig. 17 Notify in case of an abnormal situation (intruder)



time of the abnormal situation detection algorithms was 0.95 s before optimization but 0.47 s after optimization, implying a decrease by 50.53%. When each abnormal situation detection algorithm was driven, the processing time of intruder detection algorithm was 0.58 s before optimization but 0.55 s after optimization, implying a decrease by 5.17%. The processing time of fire detection algorithm was 0.32 s before optimization but 0.24 s after optimization, implying a decrease by 25%. The processing time of loitering detection algorithm was 0.21 s before optimization but 0.15 s after optimization, implying a decrease by 28.57%. The processing time of fall detection algorithm was 0.2 s before optimization but 0.15 s after optimization, implying a decrease by 25%. Therefore, we confirmed the real-time operation and various

Table 5 Comparison of performance before and after optimization of embedded module

Number	Total algorithm		Intruder detection		Fire detection		Loitering detection		Fall detection	
	Before	After	Before	After	Before	After	Before	After	Before	After
1	0.87	0.47	0.57	0.53	0.3	0.2	0.23	0.17	0.17	0.17
2	1.03	0.37	0.57	0.57	0.3	0.23	0.23	0.13	0.17	0.17
3	0.9	0.4	0.6	0.57	0.33	0.27	0.33	0.17	0.17	0.13
4	1	0.53	0.6	0.57	0.33	0.23	0.2	0.17	0.27	0.13
5	0.93	0.6	0.57	0.57	0.33	0.27	0.23	0.13	0.17	0.17
6	0.9	0.4	0.57	0.53	0.3	0.2	0.23	0.17	0.27	0.13
7	0.93	0.53	0.6	0.57	0.33	0.2	0.13	0.13	0.13	0.13
8	0.97	0.5	0.57	0.57	0.3	0.27	0.17	0.13	0.27	0.17
9	0.93	0.43	0.55	0.53	0.33	0.23	0.17	0.2	0.17	0.17
10	1.03	0.47	0.6	0.53	0.3	0.27	0.2	0.13	0.17	0.17
Average	0.95	0.47	0.58	0.55	0.32	0.24	0.21	0.15	0.2	0.15

Table 6 Comparison of processing time of PC and embedded module

PC		Embedded module			
Image I/O	Algorithm	Before optimization		After optimization	
		Image I/O	Algorithm	Image I/O	Algorithm
0.03	0.05	0.21	0.95	0.14	0.47

detection possibility of intelligent video surveillance system based on embedded module.

Table 6 illustrates the result of comparing the processing time between the embedded module-based system and the PC by using the same abnormal situation detection algorithm. The image input/output time in the PC showed a difference of 6.46 times faster than the embedded module-based system before optimization, and 4.12 times faster than the embedded module-based system after optimization. The algorithm driving time in the PC showed a difference of 17.9 times faster than the embedded module-based system before optimization, and 8.86 times faster than the embedded module-based system after optimization.

Table 7 illustrates the result of comparing performance between the embedded module-based system and the PC used in the experiment. Between the embedded module and the PC, the PC showed better performance by 12 times because of the CPU, number of core and clocks. However, because the embedded module is 44 times smaller with respect to consumed power, 230 times smaller with respect to the weight, and 365 times smaller with respect to volume, it can be used without environmental restrictions.

Table 8 shows the comparison of processing times before and after optimization of small devices. System optimization was performed using algorithm optimization and embedded module optimization methods, and processing time was reduced by more than 25%. In the case of using the optimization method proposed in this paper, Intelligent video surveillance system reduced processing time by 50.53%, but accurate comparison is impossible since they are not the same device.

Table 7 Comparison of performance of PC and embedded module

List	PC	Embedded module (raspberry pi)
CPU	Intel i7-6700 K	BCM2837 64Bit QUAD Core
Graphic	GTX980Ti	Videocore IV
Speed	4.0 GHz	1.2 GHz
Memory	16 GB DDL4L	1 GB SDRAM 400 MHz
Power	220 V	5 V
Weight	10.52 kg	45 g
Size	42 × 16.5 × 40.1 cm	8.5 × 5.6 × 1.6 cm

Table 8 Comparison of processing time before and after optimization

Author	Device	Method	Result
[37]	Mobile	Algorithm modification, image transform, etc.	Reduced 26.32%
[38]	Embedded module (XP-100)	Software suspend, JFFS2, etc.	Reduced 38.5%
[39]	Mobile	Minimize HTTP, Gzip Components, etc.	Reduced 30%
[40]	Embedded module (raspberry pi)	Algorithm modification, SRP	Reduced 46.47%
Proposed	Embedded module (raspberry pi)	Algorithm modification, using zRam, etc.	Reduced 50.53%

6 Conclusions

This paper suggests the intelligent video surveillance system based on embedded modules, and applies 5 algorithm optimization and 3 embedded module optimization methods. It is to recognize faces based in taught information to detect intruders, and fires by using color information and motion information. Loitering are detected by using balance point change angles and movement distances of objects, and fall are detected by using motion features and acceleration of the object. The system conducts 5 algorithm optimization methods including the use of positive integers for real-time processing, minimization of remainder and division operators, minimizing direct calculation of global variables, minimization of transfer factors and recursive functions, and using Cython language, and 3 embedded module optimization methods including removing bottleneck, and the use of memory and zRam.

The algorithm performance of the suggested intelligent video surveillance system was 88.51% for intruder detection, 92.63% for fire detection, 80% for loitering detection, and 93.54% for fall detection. Moreover, the result of comparing processing time before and after optimization of the intelligent video surveillance system showed that processing time of the integrated algorithm decreased by 50.53%. Driving each abnormal situation algorithm showed decreases of 5.17% for intruder detection, 25% for fire detection, 28.57% for loitering detection, and 25% for fall detection in terms of processing time.

The reason for the least processing time change of intruder detection among 4 algorithms is because optimization was not conducted for the inputted image of the Haar-like feature-based learning file for face detection and for the inputted image of the deep learning file for face recognition. The future plan is to optimize learning files for

reducing processing time of the intelligent video surveillance system, and study parallel processing of the TCP/IP-based embedded module.

Abbreviations

CCTV: Closed circuit television; PCA: Principal component analysis; DCT: Discrete cosine transform; LDA: Linear discriminate analysis; LBP: Local binary pattern; CNN: Convolution Neural Networks; RBM: Restricted Boltzmann machines; DWT: Discrete wavelet transform; SVM: Support vector machine; ADL: Activities of Daily Living; NNR: Nearest neighbor rule; DSP: Digital signal processor; AVS: Audio video coding standard; RTSP: Real-time streaming protocol; UAV: Unmanned aerial vehicle; SMC: Sliding mode control; LQR: Linear quadratic regulator; GMM: Gaussian mixture model; SRP: Sparse random projection; MEAP: Mobile enterprise application platform; KF: Kalman filter.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2017R1A6A1A03015496) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2018R1A2B6001984).

Authors' contributions

All authors have contributed significantly to the technique or methods used, to the research concept, to the data collection, to the experiment design, and to the critical revision of the article. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The open datasets supporting the conclusions of this article are available at NIST (<https://www.nist.gov/>) and PETS (<http://www.cvg.reading.ac.uk/PETS2007/>). For other datasets, please contact author for data requests.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Control and Instrumentation Engineering, Chosun University, Gwangju, Korea. ²Department of Electronics Engineering, Chosun University, Gwangju, Korea.

Received: 4 June 2018 Accepted: 25 August 2019

Published online: 21 November 2021

References

1. H.J. Park, A study on monitoring system for an abnormal behaviors by object's tracking. *J Digit Contents Soc.* **14**(4), 589–596 (2013)
2. I.S. Chang et al., A study of scenario and trends in intelligent surveillance camera. *J Korea Inst Intell Transp Syst.* **8**(4), 93–101 (2009)
3. W.J. Kim, CCTV market trends and forecasts. *Electronic and Information Research Information Center* (2011)
4. J.H. Kang, S.Y. Kwak, Loitering, sudden running and intruder detection for intelligent surveillance system. *Korea Inf Sci Soc.* **31**(1), 353–355 (2012)
5. S.H. Hwang, S.B. Pan, Fall detection system using the open source hardware and RGB camera. *J Korean Inst Inf Technol.* **14**(4), 19–24 (2016)
6. M.A. Turk, A.P. Pentland, Face recognition using eigenfaces. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 586–591 (1991)
7. J.Z. He, Q.H. Zhu, M.H. Du, Face recognition using PCA on enhanced image for single training image. *IEEE International Conference on Machine Learning and Cybernetics*, 3218–3221 (2006)
8. J. Li, B. Zhao, H. Zhang, Face recognition based on PCA and LDA combination feature extraction. *IEEE International Conference on Information Science and Engineering*, 1240–1243 (2009)
9. H.M. Moon, D.J. Choi, P.K. Kim, S.B. Pan, LDA-based face recognition using multiple distance training face images with low user cooperation. *IEEE International Conference on Consumer Electronics*, 7–8 (2015)
10. H. Zhang, Z. Qu, L. Yuan, G. Li, A face recognition method based on LBP feature for CNN. *IEEE Advanced Information Technology, Electronic and Automation Control Conference*, 544–547 (2017)
11. J. Zeng, Y. Zhai, J. Gan, A novel sparse representation classification face recognition based on deep learning. *UIC-ATC-ScalCom*, 1520–1523 (2015).
12. G.J. Jang et al., Recognition of fire levels using fuzzy reasoning. *IEEE International Conference on Industrial Mechatronics and Automation.* **2**, 557–560 (2010)
13. A. Solozano et al., Fire detection using a gas sensor array with sensor fusion algorithms. *ISOCS/IEEE International Symposium on Olfaction and Electronic Nose*, 1–3 (2017)

14. W. Wang, H. Zhou, Fire detection based on flame color and area. *IEEE International Conference on Computer Science and Automation Engineering*, **3**, 222–226 (2012)
15. T. Wang, L. Bu, Q. Zhou, Z. Yang, A new fire recognition model based on the dispersion of color component. *IEEE International Conference on Progress in Informatics and Computing*, 138–141 (2015)
16. M.G. Kim, S.B. Pan, A study on the flame detection system using color-information based on Raspberry Pi. *J Korean Inst Inf Technol.* **14**(6), 87–93 (2016)
17. L. Wang, A. Li, Early fire recognition based on multi-feature fusion of video smoke. *Chinese Control Conference*, 5318–5323 (2017)
18. N.D. Bird, O. Masoud, M. Papanikolopoulos, A. Isaacs, Detection of loitering individuals in public transportation areas. *Trans Intell Transp Syst.* **6**(2), 167–177 (2005)
19. M. Elhamod, M. Levine, Automated real-time detection of potentially suspicious behavior in public transport areas. *Trans Intell Transp Syst.* **14**(2), 688–699 (2013)
20. S.W. Lee, T.K. Kim, J.H. Yoo, J.K. Paik, Abnormal behavior detection based on adaptive background generation for intelligent video analysis. *J Inst Electron Eng Korea SP.* **48**(1), 111–121 (2011)
21. T.W. Jang, Y.T. Shin, J.B. Kim, A study on the object extraction and tracking system for intelligent surveillance. *J Korean Inst Commun Sci.* **38**(7), 589–595 (2013)
22. J.G. Ko, J.H. Yoo, Rectified trajectory analysis based abnormal loitering detection for video surveillance. *IEEE International Conference on Artificial Intelligence, Modelling and Simulation*, 289–293 (2013).
23. E.S. Park et al., Loitering behavior detection using shadow removal and chromaticity histogram matching. *J Korea Inst Inf Secur Cryptol.* **21**(6), 171–181 (2011)
24. W. Li et al., Loitering detection based on trajectory analysis. *IEEE International Conference on Intelligent Computation Technology and Automation*, 530–533 (2015)
25. J.H. Kang, S.Y. Kwak, Loitering detection solution for CCTV security system. *J. Korea Multimed. Soc.* **17**(1), 15–25 (2014)
26. U. Lindemann et al., Evaluation of a fall detector based on accelerometers: a pilot study. *Med. Biol. Eng. Comput.* **43**(5), 548–551 (2005)
27. Z. Liu, Y. Song, Y. Shang, J. Wang, Posture recognition algorithm for the elderly based on BP Neural Networks. *IEEE Chinese Control and Decision Conference*, 1446–1449 (2015)
28. C. Medrano et al., Personalizable smartphone application for detecting falls. *IEEE-EMBS International Conference on Biomedical and Health Informatics*, 169–172 (2014)
29. H. Foroughi, A. Rezvani, A. Paziraee, Robust fall detection using human shape and multi-class support vector machine. *IEEE Indian Conference on Computer Vision, Graphics and Image Processing*, 413–420 (2008)
30. V. Bevilacqua et al., Fall detection in indoor environment with kinect sensor. *IEEE International Symposium on Innovation in Intelligent Systems and Applications Proceedings*, 319–324 (2014)
31. J. Dongyao et al., A method of detecting human body falling action in a complex background. *IEEE Jubilee International Conference on Intelligent Engineering Systems*, 51–56 (2016)
32. S. Zhang et al., On the design and implementation of a high definition multi-view intelligent video surveillance system. *IEEE International Conference on Signal Processing, Communication and Computing*, 353–357 (2012)
33. B.S. Lin, J.S. Su, H. Chen, C.Y. Jan, A fall detection system based on human body silhouette. *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 49–50 (2013)
34. M. Tsourma, M. Dasygenis, Development of a hybrid defensive embedded system with face recognition. *DCABES*, 154–157 (2016)
35. V.V. Shete, N. Ukunde, Intelligent embedded video monitoring system for home surveillance. *IEEE International Conference on Inventive Computation Technologies.* **1**, 1–4 (2016)
36. C. Yuan, K.A. Ghamry, Z. Lui, Y. Zhang, Unmanned aerial vehicle based forest fire monitoring and detection using image processing technique. *IEEE Chinese Guidance, Navigation and Control Conference*, 1870–1875 (2016)
37. S.H. Kim, S.J. Koh, Loading speed improvement of web browsers on mobile devices. *Telecommun Rev.* **22**(1), 139–152 (2012)
38. T.W. Lee, J.Y. Cho, Y.H. Cho, A method of embedded linux light-weight for efficient application execution. *J Korea Soc Comput Inf.* **18**(3), 1–10 (2013)
39. J.W. Kang, Dissertation, Seoul National University of Science and Technology (2015)
40. C.C. Paglinawan et al., Optimization of vehicle speed calculation on Raspberry Pi using sparse random projection. *IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management* (2018)
41. S. Trambadia, H. Mayatra, Food detection on plate based on the HSV color model. *IEEE International Conference on Green Engineering and Technologies*, 1–6 (2016).
42. B.K. Horn, B.G. Schunck, Determining optical flow. *Artif. Intell.* **17**, 185–203 (1981)
43. J.S. Kim, M.G. Kim, B.R. Cha, S.B. Pan, in *Advances in Computer Science and Ubiquitous Computing*. A study of determining abnormal behaviors by using system for preventing agricultural product theft. 937–943 (2016)
44. Wikipedia Raspberry Pi. https://en.wikipedia.org/wiki/Raspberry_pi. Accessed 3 Nov 2017
45. Wikipedia Cython. <https://en.wikipedia.org/wiki/Cython>. Accessed 3 Sept 2017
46. A. Marzetta, ZRAM: a library of parallel search algorithms and its use in enumeration and combinatorial optimization (Hartung Gorre, 1998), p. 119

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.