International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 08 – Issue 05 || May 2023 || PP. 109-117

Motion detection using a Raspberry Pi 4 and OpenCV

José Ignacio Vega Luna[1], Francisco Javier Sánchez Rangel[1], José Francisco Cosme Aceves[1], Gerardo Salgado Guzmán[1], Víctor Noé Tapia Vargas[1]
[1]*(Área de Sistemas Digitales, Departmento de Electrónica/Universidad Autónoma Metropolitana-Azcapotzalco, México)*

Abstract: Currently, one of the most important parts of the cloud and computing platforms are the data centers, since they house the equipment where the applications and information processing are executed. The vitalkey areas of data centers are the equipment sites. In addition to the fact that access is controlled and restricted to these areas, continuous monitoring of the presence of people in these areas is critical. The data centers have closed circuit television systems that allow this activity to be carried out; however, these systems do not send alert messages that allow timely reaction to any threat from unauthorized persons who have accessed the equipment site. Similarly, presence detectors of different technologies are used, which present disadvantages compared to the system presented here. For these reasons, in this work a motion detection system was developed based on the processing of video images captured with a camera integrated into a Raspberry Pi 4 card and programmed with Python and OpenCV functions. When the system detects movement, it sends a WhatsApp alert message to a mobile phone. The tests carried out showed that the system is accurate to 98.08%.
Keywords: Data centers, OpenCV, motion detection, Python, video images.

I. INTRODUCTION

Data centers are facilities that are used to house the computer and telecommunications equipment of companies and institutions whose productivity and services they offer depend on the availability and security existing in these facilities. For this reason, one of the main needs of a data center is to have mechanisms and security measures for the physical and logical protection of the equipment, information, and infrastructure [1]. One of the areas that requires the most protection are the equipment sites, since that is where the equipment that forms the central part of the data centers is located. In these areas access is controlled by using different techniques and devices [2]. The presence of people in these areas is restricted to maintenance, operation, and surveillance personnel who carried outspecific and controlled tasks. The number of people present at equipment sites is small and, in many data centers, access is only allowed during certain hours of the day [3].In the equipment sites of the data centers, the racks of the computing and telecommunications equipment are installed in lines, or rows, strategically distributed in such a way that the air that reaches the equipment, provided by the cooling and ventilation systems, is always at the right temperature, as shown in Fig. 1.

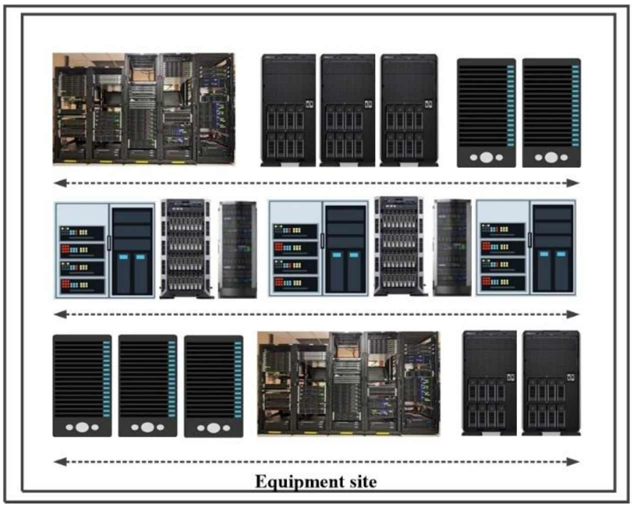

Figure 1. Rack distribution in an equipment site.

There is adequate distance to the front and rear of the racks to allow the doors to be opened, for maintenance and operation personnel to circulate, and to separate them from each other. In other words, between each row of racks there is a corridor free of obstacles in which the presence of people is permanently monitored

[4]. Additionally, the facilities of some data centers have presence and movement sensors that, in addition to detecting people, activate CCTV, lighting systems, and alarms [5].

Some types of presence and movement sensors used are vibration sensors, Passive Infrared (PIR), sound sensors, and temperature sensors, among others [6-7]. Most presence detectors, regardless of the type of technology used, have certain drawbacks that sometimes cause false positives and negatives, limited range, technology dependency, sensitivity, interference, maintenance, installation, and cost, among other deficiencies [8]. It is important that the presence and movement detection systems at the equipment sites work optimally, efficiently, and permanently since it is always necessary to be prepared for any eventuality or alert situation to safeguard access to the equipment.

Recently realized motion detection systems use different techniques and mechanisms, among which are the following:

1- Frame Difference. Compare each frame of the video with the previous frame to detect changes. In case of finding any significant difference between the frames, it is considered that there is movement [9].
2- Optical flow method. It is based on tracking the movement patterns of objects in the video over time. The speed and direction of objects in each frame are calculated and motion is detected if there is a notable change in optical flow [10-12].
3- Background Subtraction Method. It is used to extract moving objects from a scene in a video. The current frame is compared to a previous reference frame, and any difference in color, texture, or brightness is considered motion [13-14].
4- Contour-Based Motion Detection. This technique is used to detect moving objects by identifying the edges of objects in the image and their movement through the scene [15-16].
5- Gaussian Mix Model. It models the distribution of pixels in the video and detects movement when there is a significant deviation from this distribution [17].
6- Visual Background Extractor (ViBE). It is based on the analysis of movement patterns in the pixels of the video. Pixels that have a different pattern from the background are identified and motion is detected [18].
7- Object Segmentation. It focuses on the detection of individual objects in the video. Image segmentation algorithms are used to separate objects from the background, and motion is detected if there is a notable change in the position or shape of the object [19].
8- Histogram-Based Algorithms. They use the histogram of the intensity values of pixels in an image to detect motion [20].
9- Energy-Based Algorithms. They use the energy of the pixels in an image to detect movement [21].
10- Feature-Based Algorithms. They use characteristics of objects, such as their shape, size, and color, to detect movement [22].

The investigations carried out in recent years, which mark the state-of-the-art, regarding the detection of movement in a video signal, have not only been used for presence monitoring and video surveillance [23-24]. Different applications have been made using this technology among which are the following: health care and attention, surgical interventions [25], detection of epileptic seizures in neonatal intensive care units [26], cerebral palsy treatments in infants [27], and motion of vehicles and pedestrians monitoring [28], among others. Similarly, advanced research in this field makes use of models based on artificial intelligence, deep learning, convolutional neural networks [29], and change detection in multitemporal monitoring images under low illumination [30].

The system presented here aims to detect the presence of people in an equipment site and send an alert message to the mobile phone of the data center administrator. The system is based on a Raspberry Pi 4 card which was programmed in Python and OpenCV library functions. The Raspberry Pi 4 integrates a video camera which is located at the beginning and/or at the end of a corridor of the equipment site whose angle of view covers the area of the corridor. The procedure used is based on a combination of algorithms Background Subtraction, Contour-Based Motion Detection, and Gaussian Mix Model.

II. MATERIALS AND METHODS

The development of the application consists of two parts: the hardware, made up of the Raspberry Pi card, and the system software. Fig. 2 shows the way in which the developed system is installed in the corridors of the equipment site.

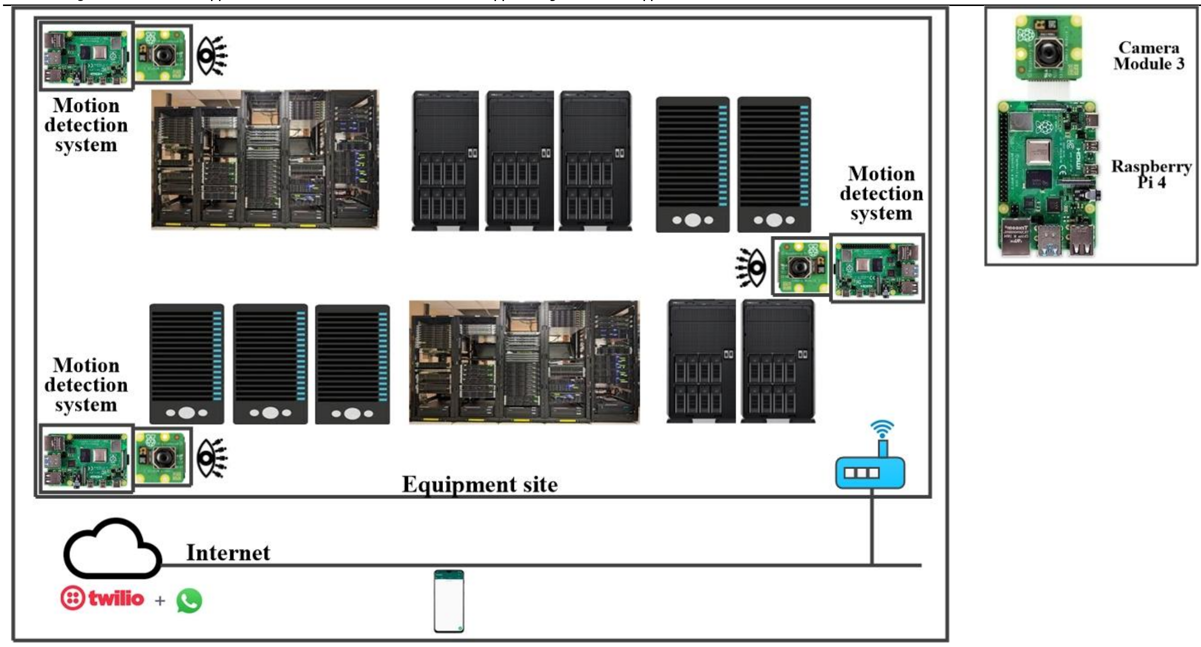

Figure 2. System block diagram.

2.1. System hardware

A Raspberry Pi 4 card was used because it integrates enough devices and hardware modules to implement the functions of the system proposed in the system design. Additionally, the Raspberry Pi runs the Raspberry Pi OS with desktop and recommended software, Release date February 21st, 2023, and Kernel version 5.15, in which the Python programming language and a considerable number of open-source function libraries can be installed for handling and processing images, one of them is OpenCV.

Regarding the hardware resources that the Raspberry Pi has, there are the following: a 1.5 GHz quad-core ARM Cortex-A72 CPU, up to 4 GB of RAM which can carry out 4K video decoding at 60 fps, VideoCore VI GPU, 1 GB to 8 GB LPDDR4 SDRAM memory, Bluetooth 5.0, Wi-Fi 802.11ac and Gigabit Ethernet communication interfaces, two USB 2.0 ports, two USB 3.0 ports, two micro HDMI ports, Camera Serial Interface (CSI), a microSD memory card slot and a 40-pin General Purpose Input Output (GPIO) connector. This variety of hardware resources is commonly used to connect input and output devices, such as video camera, SD memory, video monitor, and keyboard, as well as the performance provided by the state-of-the-art CPU.The Raspberry Pi 4 card used in this system integrates a Dynamic Range (HDR) Camera Module 3 type video camera as shown in Fig. 2. This camera, compact in size, economical, and autofocus, uses a 12 MP IMX708 Quad Bayer sensor. It provides an ultra-wide 120-degree angle of view. It can work as a night-vision camera with infrared lighting. The Camera Module 3 provides a resolution of 11.9 megapixels, 4608 pixels horizontally and 2592 pixels vertically. It can capture HD video at 50 fps and capture video with resolutions of 1080p50, 720p100, 480p120. It connects to the Raspberry Pi board via the CSI interface.

The system is in one of two states: initialization or detection. To implement these states, a push-button setup was connected to a GPIO terminal of the Raspberry Pi, configured as input, and three LEDs were connected to three GPIO terminals, configured as outputs, one red, one yellow and the third green. When the execution starts, the main program turns on the red led and waits for the administrator to press the setup push-button. Initialization is used when the system is installed for the first time or moved out of position. Once the push-button is pressed, the system turns off the red led and the yellow led blinks to allow the administrator to connect the video monitor to the HDMI port of the Raspberry Pi and place the camera in the correct position facing the rack isolate. Through the monitor you can view the image, without people, captured by the video camera to initialize the system.When the camera is in the desired position, the administrator must press the push-button a second time and the system turns off the yellow led, capture the fixed background image and enter the detection state. Upon entering this state, the system will turn on the green led to start detection.

2.2. System software

The system software was made using Python 3.3. One of the key features of the design was that it uses recent features built into the OpenCV library for handling images focused on motion detection. In this application, the background of the images is fixed, since the data center cameras used in this application are fixed, do not move and are focused on the corridor that are in front,or behind, the racks of the equipment site.

The methodology used for motion detection consisted of continuously capturing the video signal, sequentially reading the video signal frame image over time, and obtaining the position of the objects in the frames to determine if there is movement and therefore detection of people. The main process of this methodology consists in carrying out a continuous cycle where the difference between the background image and the frame read in the iteration of the cycle is obtained to remove the background, or fixed part, of the image. With this action you get the objects that have been moved. Lastly, it determines the rotated rectangle of the objects that have moved to be more certain that movement has been detected. As indicated in the introduction section, the procedure used is based on a combination of algorithms Background Subtraction, Contour-Based Motion Detection, and Gaussian Mix Model.

To carry out the actions indicated above, the main program performs the following tasks: 1-Initialize the GPIO terminals and the Wi-Fi interface to connect to the Internet, 2-Turns on the red led,3-Enter a loop in which it monitors the push-button setup,4-When the push-button is pressed, turns off the red led and the yellow led blinks, 5-Wait for the push-button to be pressed a second time, 6-Capture the fixed background image using the *cap.read*function, 6-Turn off the yellow led and turn on the green led, and 7-Call the *Detection* routine, remaining in that state for the rest of the program. In Fig. 3 the flowchart used to carry out the main program is indicated.

The tasks performed by the *Detection*routine are the following:A-Starts video capture, through the *cv2.VideoCapture* function, B-Read a frame of the captured video signal, calling the *video.read* function, C-Since the functions used in the processing work with grayscale images, it is necessary carry out the transformation from BGR to grayscale and, even better, reduce the noise of the images by applying a bilateral filter. To perform these actions, the functions *cv2.bilateralFilter* and *cv2.cvtColor(cv2.COLOR_BGR2GRAY)* were used.The bilateral filter is a non-linear filter that reduces noise by smoothening images preserving edges and keeping edges sharp. This filter uses, instead of the intensity of each pixel, the weighted average of intensity of each pixel obtained from nearby pixels. This type of filter is considered an edge-preserving filter since it does not do averaging across edges. For this reason, the bilateral filter is close to the Gaussian filter, D-Next, the routine performs the background subtraction of the filtered image using the *cv2.absdiff(background,frame)* function. With this, the unnecessary information is removed, the fixed background, of the filtered image. This function determines the absolute difference between the pixels of the two image arrays and obtains the pixels of the people, or objects, which are supposed to be moving, E-Later, it is necessary to specifically mark the people, or objects, by means of the outline to identify them as clearly as possible. However, before obtaining the contour, processing of the array containing the subtracted image is performed. The process consists of performing three actions: the morphological transformation, the binary transformation and the application of a medium filter to blur and smooth the image.The morphological transformation is performed based on the image shape. It carries out the operations of erosion and dilation. A kernel indicating the type of operation is applied to the image. The first operation erodes away the boundaries of foreground object and tries to keep foreground in white. A pixel in the original image, either 1 or 0, will be 1 if all the pixels under the kernel are 1, otherwise it is eroded, the pixel will be 0. The second operation, dilation, performs the opposite actions of erosion with the goal of increasing the white region in the image or size of foreground objects. The transformation was implemented in the system software using the *cv2.morphologyEx()* function. The binary transformation consists of thresholding the image to make the object more prominent and processing the image in black and white. This action is performed using the *cv2.threshold()* function, in which a threshold value is applied to each pixel of the image. If the pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value indicated as a parameter of this function. The result is a binary image containing the moving areas marked in white.

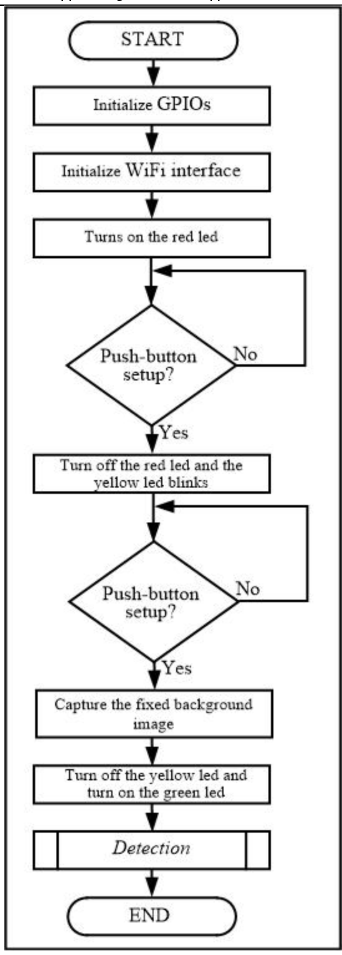

Figure 3. Main program flowchart.

The medium filter allows to reduce the noise, named in the literature about its salt-and-pepper noise. It is a non-linear filter that uses the median of all the pixels under a kernel area and replaces the central element with this median value, F-Once the subtraction image is processed, the *Detection*routine proceeds to identify and mark the contours of objects in the image that are assumed to have moved. Using the functions *cv2.findContours()* and *cv2.drawContours()*, respectively, G-In the next step, the areas marked by rectangles whose size is less than or equalthan the established size, 8,500 in the system software presented here, are excluded as a value maximum. This was implemented through the functions *cv2.contourArea()* and *cv2.boundingRect()*. The objective of this stage of the procedure is to discriminate, or ignore, small rectangles, or areas, that may be false positives, H-The next task is to determine the orientation of the contours of the rectangles that passed the previous filter, also called rotated rectangle or ellipse, by means of from the *cv2.fitEllipse()* function. This function returns the angle that contains the area where the movement was detected with respect to the x-axis, I-If the angle is equal to or greater than 1 degree, it is assumed that there was movement and, J-If movement was detected, the WhatsApp alert message is sent, through the Twilio platform, and the image read from the video signal that is being processed in the cycle interaction, and K-Finally, wait one minute and continue the cycle in taskB. In Fig. 4 the flowchart used to carry out the *Detection* routine is indicated.

International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 08 – Issue 05 || May 2023 || PP. 109-117

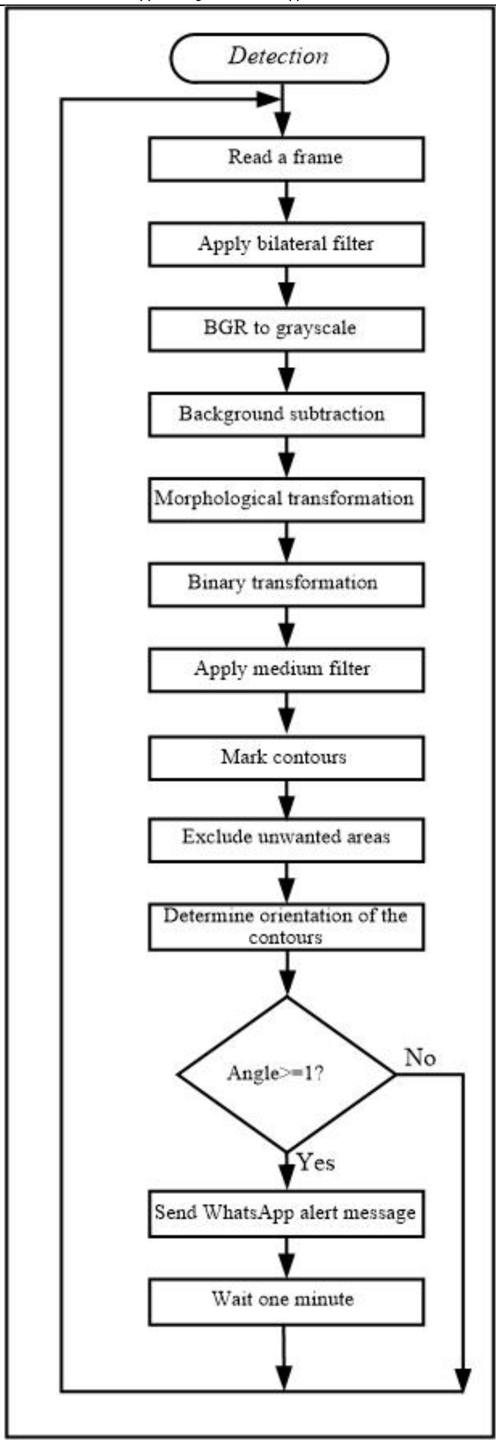

Figure 4. Detection routine flowchart.

III. RESULTS AND DISCUSSION

The most important stage of the implemented procedure was the one that discriminates the small rectangles found with the *cv2.contourArea()* and *cv2.boundingRect()* functions in the *Detection* routine. They are those small areas that can create false positives. For this reason, the first set of tests carried out had the objective of determining the optimal value used by comparing the area of the detected objects, calculated with the first function, with respect to a minimum value established in the program, to eliminate false positives. The procedure implemented by the software ignores areas equal to or smaller than this minimum valueand its effect is to tune the system to obtain better results in detection and performance. For this reason, the set consisted of thirteen tests. Initially, the background image was captured and then, in each test, the minimum value of the area was established and one person, or groups of 2 to 5 people, accessed the corridor where the system was installed to verify that the movement of one or more people. In the first test, the minimum value of the area was set at 5,000. In subsequent tests, this value was increased by 500 until it reached 12,000. In the first 7 tests, 10, 9, 8, 5, 3, 2, 1 false positive were detected, respectively, and in the remaining 6 no false positives were detected, as indicated in the graph in Fig. 5. Based on these results, it was determined to use 8,500 as the minimum value of the area.

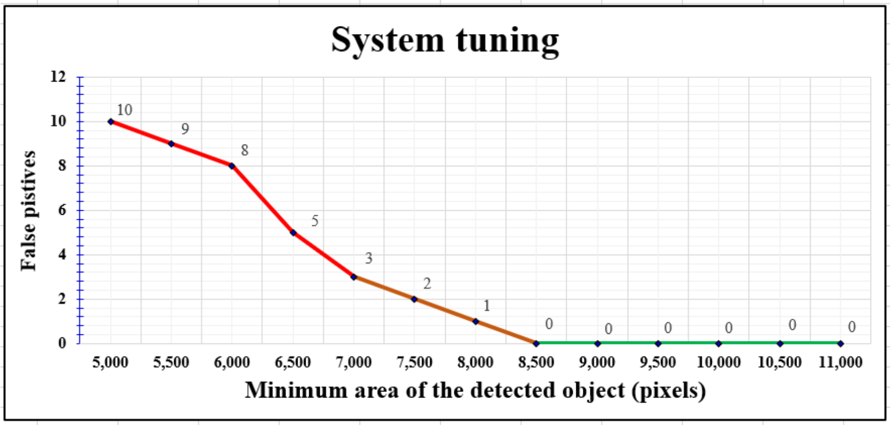

Figure 5. System tuning for detection.

The second group of tests was aimed at determining the accuracy of the system. Ten tests were carried out. In the first test, 4 people accessed the rack corridor and the system detected all 4. In the following tests, one more person accessed than in the previous test, until reaching 13 people and in each test, 5, 6, 7, 8, 9, 9, 10, 12, and 13 people, respectively. This showed that the precision of the system is 98.08%, on average, which is acceptable considering that in the corridors of the equipment sites it is not common for there to be more than 3 people.Fig. 6 shows the contour of a person whose movement was detected, and the contour marked by the system software.

Figure 6. Motion detection and contour of the marked person.

IV. CONCLUSION

A system was developed that detects the presence of people in the equipment site of a data center through movement in a video signal. The system consists of a Raspberry Pi 4 board with an integrated HDR video camera. The programming was done in Python and OpenCV functions for image processing. The system is installed at one end of the existing corridors between the equipment racks to capture the image of the fixed background, then continuously captures the video signal and subtracts the background image from each frame to detect the differences and determine if there is movement. If so, it sends an alert message to the data center administrator.

The design of the system implies that to cover the entire area of the equipment site, a system, such as the one presented here, must be installed at each end of the corridors. For this reason, as future work, a second version was started where the CCTV video cameras of the data center will be used so that the Raspberry Pi card receives the images from the cameras to perform the processing and detection. The advantages that this will represent are that only one module with Raspberry will be used and the CCTV infrastructure will be used. The only requirement is that the cameras capture HDR images for better detection results.

Another improvement that is also being worked on is for the system to work as a detection network. That is, installing the system in each rack corridor, as proposed in this work, and when each module detects movement, it sends the captured image, and its location, to a master module so that the latter is in charge of communication with the Internet platform for the transmission of alert messages. In this way, each module with Raspberry, installed in each corridor, will be a slave module.

Finally, it is recommended that when you install the system in a new position, the background image is captured, since this is the reference for the image subtraction process. If the range of the Raspberry Wi-Fi interface is not enough to communicate with the Internet access point, repeaters or routers can be installed to extend the range.

REFERENCES

[1] A. Sharma and S. Chauhan, "Sensor Fusion for Distributed Detection of Mobile Intruders in Surveillance Wireless Sensor Networks," in *IEEE Sensors Journal*, vol. 20, no. 24, pp. 15224-15231, 15 Dec.15, 2020, doi: 10.1109/JSEN.2020.3009828.

[2] Z. Á. Mann, A. Metzger, J. Prade, R. Seidl and K. Pohl, "Cost-Optimized, Data-Protection-Aware Offloading Between an Edge Data Center and the Cloud," in *IEEE Transactions on Services Computing*, vol. 16, no. 1, pp. 206-220, 1 Jan.-Feb. 2023, doi: 10.1109/TSC.2022.3144645.

[3] D. B. Rawat, R. Doku and M. Garuba, "Cybersecurity in Big Data Era: From Securing Big Data to Data-Driven Security," in *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 2055-2072, 1 Nov.-Dec. 2021, doi: 10.1109/TSC.2019.2907247.

[4] C. Jiang, Y. Qiu, H. Gao, T. Fan, K. Li and J. Wan, "An Edge Computing Platform for Intelligent Operational Monitoring in Internet Data Centers," in *IEEE Access*, vol. 7, pp. 133375-133387, 2019, doi: 10.1109/ACCESS.2019.2939614.

[5] H. Dong, A. Munir, H. Tout and Y. Ganjali, "Next-Generation Data Center Network Enabled by Machine Learning: Review, Challenges, and Opportunities," in *IEEE Access*, vol. 9, pp. 136459-136475, 2021, doi: 10.1109/ACCESS.2021.3117763.

[6] A. Naser, A. Lotfi and J. Zhong, "Adaptive Thermal Sensor Array Placement for Human Segmentation and Occupancy Estimation," in *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1993-2002, 15 Jan.15, 2021, doi: 10.1109/JSEN.2020.3020401.

[7] L. Wu, Z. Chen and Y. Wang, "Occupancy Detection Using a Temperature-Sensitive Adaptive Algorithm," in *IEEE Sensors Letters*, vol. 5, no. 12, pp. 1-4, Dec. 2021, Art no. 2501204, doi: 10.1109/LSENS.2021.3126285.

[8] L. Wu, F. Gou, S. -T. Wu and Y. Wang, "SLEEPIR: Synchronized Low-Energy Electronically Chopped PIR Sensor for True Presence Detection," in *IEEE Sensors Letters*, vol. 4, no. 3, pp. 1-4, March 2020, Art no. 2500204, doi: 10.1109/LSENS.2020.2976801.

[9] X. Sheng, K. Li, Z. Shen and G. Xiao, "A Progressive Difference Method for Capturing Visual Tempos on Action Recognition," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 3, pp. 977-987, March 2023, doi: 10.1109/TCSVT.2022.3207518.

[10] Y. Fu, B. Zhang, G. Liu, R. Zhang, Q. Liu and Y. Ye, "An Optical Flow SBAS Technique for Glacier Surface Velocity Extraction Using SAR Images," in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022, Art no. 2001505, doi: 10.1109/LGRS.2022.3200422.

[11] S. Ando and T. Kindo, "Direct Imaging of Stabilized Optical Flow and Possible Anomalies From Moving Vehicle," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24044-24056, Dec. 2022, doi: 10.1109/TITS.2022.3199203.

[12] Li, B. Zheng, L. Li and C. Zhang, "Optical Flow Estimation and Denoising of Video Images Based on Deep Learning Models," in IEEE Access, vol. 8, pp. 144122-144135, 2020, doi: 10.1109/ACCESS.2020.3014345.

[13] J. M. McHugh, J. Konrad, V. Saligrama and P. -M. Jodoin, "Foreground-Adaptive Background Subtraction," in *IEEE Signal Processing Letters*, vol. 16, no. 5, pp. 390-393, May 2009, doi: 10.1109/LSP.2009.2016447.

[14] J. Zuo, Z. Jia, J. Yang and N. Kasabov, "Moving Target Detection Based on Improved Gaussian Mixture Background Subtraction in Video Images," in *IEEE Access*, vol. 7, pp. 152612-152623, 2019, doi: 10.1109/ACCESS.2019.2946230.

[15] Y. Hu, "Image Segmentation Based on Velocity Feature Vector for Moving Target Extraction," in *IEEE Sensors Journal*, vol. 20, no. 20, pp. 11983-11991, 15 Oct.15, 2020, doi: 10.1109/JSEN.2020.2974314.

[16] Y. Liang, Y. Zhang, Y. Wu, S. Tu and C. Liu, "Robust Video Object Segmentation via Propagating Seams and Matching Superpixels," in *IEEE Access*, vol. 8, pp. 53766-53776, 2020, doi: 10.1109/ACCESS.2020.2981140.

[17] H. Xie, K. Mengersen, C. Di, Y. Zhang, J. Lipman and S. Van Huffel, "A Variational Bayesian Gaussian Mixture-Nonnegative Matrix Factorization Model to Extract Movement Primitives for Robust Control," in *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 6, pp. 1258-1270, Dec. 2022, doi: 10.1109/THMS.2022.3194593.

[18] Y. Zhang, W. Zheng, K. Leng and H. Li, "Background Subtraction Using an Adaptive Local Median Texture Feature in Illumination Changes Urban Traffic Scenes," in *IEEE Access*, vol. 8, pp. 130367-130378, 2020, doi: 10.1109/ACCESS.2020.3009104.

[19] A. Lukežič, J. Matas and M. Kristan, "A Discriminative Single-Shot Segmentation Network for Visual Object Tracking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9742-9755, 1 Dec. 2022, doi: 10.1109/TPAMI.2021.3137933.

[20] S. Unnikrishnan, J. Donovan, R. Macpherson and D. Tormey, "An Integrated Histogram-Based Vision and Machine-Learning Classification Model for Industrial Emulsion Processing," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5948-5955, Sept. 2020, doi: 10.1109/TII.2019.2959021.

[21] J. Yin et al., "Energy-Based Periodicity Mining With Deep Features for Action Repetition Counting in Unconstrained Videos," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4812-4825, Dec. 2021, doi: 10.1109/TCSVT.2021.3055220.

[22] S. Chotipant and P. Jamsri, "Differentiative Feature-based Fall Detection System," *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, Jeju, Korea, Republic of, 2021, pp. 359-364, doi: 10.23919/ICCAS52745.2021.9650068.

[23] X. Tian, P. Zheng and J. Huang, "Robust Privacy-Preserving Motion Detection and Object Tracking in Encrypted Streaming Video," in *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5381-5396, 2021, doi: 10.1109/TIFS.2021.3128817.

[24] M. Aloraini, M. Sharifzadeh and D. Schonfeld, "Sequential and Patch Analyses for Object Removal Video Forgery Detection and Localization," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 917-930, March 2021, doi: 10.1109/TCSVT.2020.2993004.

[25] K. Lam et al., "Deep Learning for Instrument Detection and Assessment of Operative Skill in Surgical Videos," in *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 4, pp. 1068-1071, Nov. 2022, doi: 10.1109/TMRB.2022.3214377.

[26] B. Olmi, L. Frassineti, A. Lanata and C. Manfredi, "Automatic Detection of Epileptic Seizures in Neonatal Intensive Care Units Through EEG, ECG and Video Recordings: A Survey," in *IEEE Access*, vol. 9, pp. 138174-138191, 2021, doi: 10.1109/ACCESS.2021.3118227.

[27] B. Nguyen-Thai, V. Le, C. Morgan, N. Badawi, T. Tran and S. Venkatesh, "A Spatio-Temporal Attention-Based Model for Infant Movement Assessment From Videos," in *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 10, pp. 3911-3920, Oct. 2021, doi: 10.1109/JBHI.2021.3077957.

[28] S. Arroyo, L. Garcia, F. Safar and D. Oliva, "Urban dual mode video detection system based on fisheye and PTZ cameras," in *IEEE Latin America Transactions*, vol. 19, no. 9, pp. 1537-1545, Sept. 2021, doi: 10.1109/TLA.2021.9468447.

[29] X. Zeng, Y. Jiang, W. Ding, H. Li, Y. Hao and Z. Qiu, "A Hierarchical Spatio-Temporal Graph Convolutional Neural Network for Anomaly Detection in Videos," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 1, pp. 200-212, Jan. 2023, doi: 10.1109/TCSVT.2021.3134410.

[30] Y. Zhu, Z. Jia, J. Yang and N. K. Kasabov, "Change Detection in Multitemporal Monitoring Images Under Low Illumination," in *IEEE Access*, vol. 8, pp. 126700-126712, 2020, doi: 10.1109/ACCESS.2020.3008262.