Low-cost intelligent surveillance system based on fast CNN

Zaid Saeb Sabri [1,2], **Zhiyong Li** [Corresp. 1]

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan, China

[2] Department of Computer Science & Information Systems, Al-Mansour University College, Baghdad, Al-Andalus Square, Iraq

Corresponding Author: Zhiyong Li
Email address: zhiyong.li@hnu.edu.cn

Smart surveillance systems are used to monitor specific areas, such as homes, buildings, and borders, and these systems can effectively detect any threats. In this work, we investigate the design of low-cost multiunit surveillance systems that can control numerous surveillance cameras to track multiple objects (i.e., people, cars, and guns) and promptly detect human activity in real time using low computational systems, such as compact or single board computers. Deep learning techniques are employed to detect certain objects to surveil homes/buildings and recognize suspicious and vital events to ensure that the system can alarm officers of relevant events, such as stranger intrusions, the presence of guns, suspicious movements, and identified fugitives. The proposed model is tested on two computational systems, specifically, a single board computer (Raspberry Pi) with the Raspbian OS and a compact computer (Intel NUC) with the Windows OS. In both systems, we employ components, such as a camera to stream real-time video and an ultrasonic sensor to alarm personnel of threats when movement is detected in restricted areas or near walls. The system program is coded in Python, and a convolutional neural network (CNN) is used to perform recognition. The program is optimized by using a foreground object detection algorithm to improve recognition in terms of both accuracy and speed. The saliency algorithm is used to slice certain required objects from scenes, such as humans, cars, and airplanes. In this regard, two saliency algorithms, based on local and global patch saliency detection are considered. We develop a system that combines two saliency approaches and recognizes the features extracted using these saliency techniques with a conventional neural network. The field results demonstrate a significant improvement in detection, ranging between 34% and 99.9% for different situations. The low percentage is related to the presence of unclear objects or activities that are different from those involving humans. However, even in the case of low accuracy, recognition and threat identification are performed with an accuracy of 100% in approximately 0.7 s, even when using computer systems with relatively weak hardware specifications, such as a single board computer (Raspberry Pi). These results prove that

the proposed system can be practically used to design a low-cost and intelligent security
and tracking system.

1 Low-cost intelligent surveillance system based on fast
2 CNN

3   Zaid Saeb Sabri[1,2], Zhiyong Li[1]

4   [1] College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan

5   410082, China; zaidsaeb@hnu.edu.cn

6   [2] Department of Computer Science & Information Systems, AL-Mansour University College,

7   Al-Andalus Square, Baghdad, Iraq; zaid.saeb@muc.edu.iq

8   Corresponding Author: Zhiyong Li[1]

9   Changsha, Hunan 410082, China

10 Email address: zhiyong.li@hnu.edu.cn

12 Abstract

13 Smart surveillance systems are used to monitor specific areas, such as homes, buildings, and
14 borders, and these systems can effectively detect any threats. In this work, we investigate the
15 design of low-cost multiunit surveillance systems that can control numerous surveillance cameras
16 to track multiple objects (i.e., people, cars, and guns) and promptly detect human activity in real
17 time using low computational systems, such as compact or single board computers. Deep learning
18 techniques are employed to detect certain objects to surveil homes/buildings and recognize
19 suspicious and vital events to ensure that the system can alarm officers of relevant events, such as
20 stranger intrusions, the presence of guns, suspicious movements, and identified fugitives. The
21 proposed model is tested on two computational systems, specifically, a single board computer
22 (Raspberry Pi) with the Raspbian OS and a compact computer (Intel NUC) with the Windows OS.
23 In both systems, we employ components, such as a camera to stream real-time video and an
24 ultrasonic sensor to alarm personnel of threats when movement is detected in restricted areas or
25 near walls. The system program is coded in Python, and a convolutional neural network (CNN) is
26 used to perform recognition. The program is optimized by using a foreground object detection
27 algorithm to improve recognition in terms of both accuracy and speed. The saliency algorithm is
28 used to slice certain required objects from scenes, such as humans, cars, and airplanes. In this
29 regard, two saliency algorithms, based on local and global patch saliency detection are considered.
30 We develop a system that combines two saliency approaches and recognizes the features extracted
31 using these saliency techniques with a conventional neural network. The field results demonstrate
32 a significant improvement in detection, ranging between 34% and 99.9% for different situations.
33 The low percentage is related to the presence of unclear objects or activities that are different from
34 those involving humans. However, even in the case of low accuracy, recognition and threat
35 identification are performed with an accuracy of 100% in approximately 0.7 s, even when using
36 computer systems with relatively weak hardware specifications, such as a single board computer
37 (Raspberry Pi). These results prove that the proposed system can be practically used to design a
38 low-cost and intelligent security and tracking system.

39 Introduction

40 Most traditional video surveillance systems use surveillance cameras connected to monitors and
41 screens. However, in recent times, the need to detect and classify normal or abnormal events to
42 suitably assess a given situation and to adopt security measures has emerged. Typically, in standard
43 surveillance systems that involve a large number of surveillance cameras covering a large area,
44 certain operators must continuously check the real-time footage recorded by the cameras (Figure
45 1). In the event of an undesirable incident, the operators must alert security or the police. In certain
46 surveillance camera systems, the monitors display the video stream from a single camera.
47 However, in most cases, a single monitor displays multiple streams from several cameras, such as
48 4, 8, or 16 cameras, in a sequential or simultaneous manner (Aldasouqi & Hassan, 2010).
49 Furthermore, practically, the operators cannot monitor all the screens all the time; instead, the
50 camera output is recorded by using video recorders, such as digital video recorder (DVR) or
51 network video recorder(NVR). If an incident occurs, video footage is used as evidence. One
52 drawback of this strategy is that the operators cannot address the incidents or prevent any related
53 damage in real time as the recordings can only be viewed at a later time. Moreover, considerable
54 time is required to find the relevant section of the recording as a suspect is often at the scene long
55 before an incident occurs, and the recording may correspond to multiple cameras. Consequently,
56 it is necessary to develop a method or technique that can instantly analyze and detect threats based
57 on the detection of humans and their activities (Salahat et al., 2013; Troscianko et al., 2004).
58
59   **Figure 1.** A typical control room for traditional surveillance systems

60 In the last decade, modern video surveillance systems have attracted increasing interest with
61 several studies focusing on automated video surveillance systems, which involve a network of
62 surveillance cameras with sensors that can monitor human and nonhuman objects in a specific
63 environment. Pattern recognition can be used to find specific arrangements of features or data,
64 which usually yield details regarding a presented system or data set. In a technical context, a pattern
65 can involve repeating sequences of data with time, and patterns can be utilized to predict trends
66 and specific featural configurations in images to recognize objects. Many recognition approaches
67 involving the use of the support vector machine (SVM) (Junoh et al., 2012), artificial neural
68 network (ANN) (Petrosino & Maddalena, 2012), deep learning (Wang et al., 2019), and other rule-
69 based classification systems have been developed. Performing classification using an ANN is a
70 supervised practical strategy that has achieved satisfactory results in many classification tasks. The
71 SVM requires fewer computational requirements than the ANN; however, the SVM provides
72 lower recognition accuracy than the ANN. In recent years, networks have played a significant role
73 in a wide range of applications, and they have been applied to surveillance systems. In recent years,
74 as the amounts of unstructured and structured data have increased to big data levels, researchers
75 have developed deep learning systems that are basically neural networks with several layers. Deep
76 learning allows one to capture and mine larger amounts of data, including unstructured data. This
77 approach can be used to model complicated relationships between inputs and outputs or to find
78 patterns. However, the associated accuracy and classification efficiency are generally low (Liu J
79 and an FP, 2020). Many strategies have been developed to increase the recognition accuracy. In

80 this work, we discuss the accuracy gains from adopting certain saliency methods to improve the
81 recognition and detection of an object and isolate it from a scene.

82

83 The performance efficiency of existing surveillance systems is highly dependent on the activity of
84 human operators who are responsible for monitoring the camera footage (Sedky, Moniri and
85 Chibelushi, 2005). In general, most medium and large surveillance systems involve numerous
86 screens (approximately 50 or more) that display the streams captured by numerous cameras. As
87 the number of simultaneous video streams to be viewed increases, the work of surveillance
88 operators becomes considerably challenging and fatiguing. Practically, after twenty minutes of
89 continuous work, the attention of the operators is expected to degrade considerably. In general, the
90 operators check for the absence or presence of objects (i.e., people and vehicles) in surveillance
91 areas and ensure that the maximum capacity of a place remains intact, such as by ensuring that no
92 unauthorized people are present in restricted areas and no objects are present in unexpected places.
93 The failures of such systems in alarming authorities can be attributed to the limitations of manual
94 processing. Generally, most traditional methods used to obtain evidence depend heavily on the
95 records of the security camera systems in or near accident sites. Practically, when an incident
96 occurs in a vast space or considerable time has elapsed since its occurrence, it is difficult to find
97 any valuable evidence pertaining to the perpetrators from the large number of surveillance videos,
98 which hinders the resolution of the cases. Thus, to minimize the mental burden of the operators
99 and enhance their attention spans, it is desirable that an automated system that can reliably alert an
100 operator of the presence of target objects (e.g., a human) or the occurrence of an anomalous event
101 be developed.
102

103 Pattern recognition, which is widely used in many recognition applications, can be performed to
104 find arrangements of features or data, and this technique can be applied in the surveillance domain.
105 Several recognition approaches involving the support vector machine, artificial neural networks,
106 decision trees, and other rule-based classification systems have been proposed. Machine learning
107 typically uses two types of approaches, namely, supervised and unsupervised learning. Using these
108 approaches, especially supervised learning, we can train a model with known input and output data
109 to ensure that it can estimate any future output. Moreover, in some existing systems, an artificial
110 immune system (AIS)-inspired framework, where the AIS is a computational paradigm that is a
111 part of the computational intelligence family and is inspired by the biological immune system that
112 can reliably identify unknown patterns within sequences of input images, has been utilized to
113 achieve real-time vision analysis designed for surveillance applications (Cserey, Porod and Roska,
114 2004).

115

Literature Survey

117 The field of video surveillance is very wide. Active research is ongoing in subjects, such as
118 automatic thread detection and alarms, large-scale video surveillance systems, face recognition
119 and license plate recognition systems, and human behavior analysis (Mabrouk & Zagrouba, 2018).
120 Intelligent video surveillance (Singh & Kankanhalli, 2009) is of significant interest in industry

121 applications because of the increased requirement to decrease the time it takes to analyze large-
122 scale video data. Relating to the terminology, Elliott (Elliott, 2007) recently described an
123 intelligent video system (termed IVS) as "any kind of video surveillance method that makes use
124 of technology to automatically manipulate process and/or achieved actions, detection, alarming
125 and stored video images without human intervention." Academic and industry studies are focused
126 on developing key technologies for designing powerful intelligent surveillance systems along with
127 low-cost computing hardware; and the applications include object tracking (Khan & Gu, 2005;
128 Avidan, 2007), pedestrian detection (Dalal & Triggs, 2005), gait analysis (Wang, 2006), vehicle
129 recognition (Wang, 2007), privacy protection (Yu et al., 2008), face and iris recognition (Park &
130 Jain, 2010), video summarization (Cong, Yuan & Luo, 2012) and crowd counting (Cong et al.,
131 2009). Nguyen (Nguyen et al, 2015) described the implementation and design of an intelligent
132 low-cost monitoring system using a Raspberry Pi and a motion detection algorithm programmed
133 in Python as a traditional programming environment. Additionally, the system utilizes the motion
134 detection algorithm to considerably reduce storage usage and save expense costs. The motion
135 detection algorithm is executed on a Raspberry Pi that enables live streaming cameras together
136 with motion detection. The real-time video camera can be viewed from almost any web browser,
137 even by mobile devices. (Sabri et al., 2018) present a real-time intruder monitoring system based
138 on a Raspberry Pi to deploy a surveillance system that is effective in remote and scattered places,
139 such as universities. The system hardware consists of a Raspberry Pi, long-distance sensors,
140 cameras, a wireless module and alert circuitry; and the detection algorithm is designed in Python.
141 This system is a novel cost-effective solution with good flexibility and improvement needed for
142 monitoring pervasive remote locations. The results show that the system has high reliability for
143 smooth working while using web applications; in addition, it is cost-effective. Therefore, it can be
144 integrated as several units to catch and concisely monitor remote and scattered areas. Their system
145 can also be controlled by a remote user geographically or sparsely far from any networked
146 workstation. The recognition results prove that the system efficiently recognized intruders and
147 provided alerts when detecting intruders at distances between one to three meters from the system
148 camera. The recognition accuracy is between 83% and 95% and the reliable warning alert is in the
149 range of 86-97%. Turchini (Turchini et al, 2018) proposes an object tracking system that was
150 merged with their lately developed abnormality detection system to provide protection and
151 intelligence for critical regions.
152 In recent years, many studies have focused on using artificial intelligence for intelligence
153 surveillance systems. These techniques involve different approaches, such as the SVM, the ANN,
154 and the latest developed types based on deep learning techniques. However, deep neural networks
155 are computationally challenging and memory hungry; therefore, it is difficult to run these models
156 in low computational systems, such as single board computers (Verhelst & Moons, 2018). Several
157 approaches have been utilized to address this problem. Many approaches have reduced the size of
158 neural networks and maintained the accuracy, such as MobileNet, while other approaches
159 minimize the number of parameters or the size (Véstias, 2019).
160

161 **System Concepts**
162 We designed a robust surveillance system based on the faster RCNN and enhanced it by utilizing
163 a saliency algorithm. The following equation can be used to determine the dimensions of the
164 activation maps (O'Shea & Nash, 2015; Aggarwal, 2018):
165 $(D_i + 2P_a - D_f)/S_t + 1;$ (1)
166 where D_i = Image dimension (input file)
167 • P_a = Padding
168 • D_f = Filter dimension
169 • S_t = Stride
170 A CNN has a certain activation range. In this work, we used a rectified linear unit or ReLU
171 function. Currently, the ReLU is one of the most commonly used activation functions in NNs. One
172 of the most significant advantages of the ReLU over other activation functions is that it is unable
173 to activate all neurons at the same time. The ReLU function transforms all the negative inputs to
174 0, and no neuron is activated. Consequently, the function is computationally efficient since only a
175 few neurons are activated over time. Practically, the ReLU converges six times faster than the
176 sigmoid and tanh activation functions. One of the disadvantages of the ReLU is that it is saturated
177 in the negative region, which means that the gradient in that region is 0. In this case, all the weights
178 are not updated through backpropagation (BP), and a leaky ReLU can be used to overcome this
179 limitation. In addition, ReLU functions are not centered at zero, which means that a random and
180 thus longer path is often adopted for the functions to reach their optimal points. In addition, a
181 pooling layer is placed between the convolution layers. The pooling layer fundamentally
182 minimizes the amount of computation and number of parameters in the network and controls the
183 overfitting by progressively minimizing the spatial size of the network. Generally, two operations
184 are performed in this layer: maximum and average pooling. In this work, we utilize the max
185 pooling technique. Specifically, only the maximum value is obtained from the pool by using filters
186 sliding throughout the input; and at each stride, the maximum parameter is extracted, and the
187 remaining values are not considered. This technique practically downsamples the network.
188 Compared with the convolution layer, this layer does not alter the network, and the depth
189 dimension remains unchanged (Shang et al., 2016).
190 The output after performing max pooling can be determined as
191 $(D_i - D_f)/S_t + 1$ (2)
192 where
193 • D_i = Dimension of input (image) to pooling layer
194 • F = Filter dimension
195 • S_t = Stride
196 In the fully connected layer, all the neurons are fully connected to each activation from the prior
197 layers. These activation values can be computed via matrix multiplication and then a bias offset,
198 which is the last phase of the CNN. The CNN is constructed using hidden layers and fully
199 connected layers.

200
201   The RCNN (Girshick et al., 2014) extracts many parts from the presented image utilizing selective
202   search and then investigates whether any of these boxes has an object. First, the model extracts all
203   these regions; and for every region, a CNN is utilized to extract specific features. Finally, these
204   features are later used to detect objects. However, the RCNN is slow because of these multiple
205   steps included in the process. The fast RCNN (Girshick, 2015), alternatively, passes the entire
206   image towards the convolutional net that yields regions of interest (rather than transferring the
207   extracted areas from the image). Additionally, rather than making use of three different models (as
208   in the RCNN), it utilizes a single model that extracts features out of the areas, classifies them into
209   several classes, and applies bounding boxes. Each of these steps is conducted at the same time,
210   hence making it execute quicker than the RCNN. However, the fast RCNN is not fast enough in
211   cases where it is applied to a large dataset since it also uses selective search for region extraction.
212   The faster RCNN (Ren et al., 2016) improves upon the fast RCNN. In the faster RCNN method,
213   the "Selective Search" method was replaced by the Region Proposal Network (known as RPN),
214   which is a network to present regions (Brandenburg et al, 2018) and is faster than the RCNN and
215   fast RCNN.
216
217   To improve the recognition process and reduce features, we utilize a saliency algorithm to enhance
218   the image maps as the algorithm maps the images to indicate the unique quality of each pixel.
219
220   In computer vision, the Saliency map is defined as an image in which every pixel in the image has
221   unique quality. The aim of a saliency map is to make the image simpler and/or change it is
222   representation (Daniilidis, Maragos & Paragios, 2010). The saliency detection approach is
223   commonly used in the areas of cognition and target detection (Moosmann, Larlus, & Jurie, 2006;
224   Zhaoyu, Pingping, & Changjiu, 2009; Kanan C. & Cottrell, 2010; Borji et al., 2019), object
225   discovery (Frintrop, García, & Cremers, 2014), image segmentation (Kang et al., 2012;
226   Yanulevskaya, Uijlings & Geusebroek, 2013), visual tracking (Klein et al., 2010; Borji et al., 2012;
227   Stalder, Grabner & Gool, 2012), etc. The saliency represents a type of image segmentation
228   technique. The saliency map aims to simplify and adjust the image representation to a more
229   substantial form that is faster and easier to analyze. For example, when a pixel has a considerably
230   large gray level or different color values in a color image, the quality of the pixel can be identified
231   easily in the saliency map.
232
233   The saliency can be local or global (Borji & Itti, 2012). In the local domain, the contrast
234   corresponds to the saliency of the image patches in the local neighborhoods. In contrast, in the
235   global domain, to determine the saliency of an image patch, the contrast is computed using the
236   patch statistics along the whole image. In this work, we utilized the global saliency approach. The
237   local patch is identical to its neighbors. However, the whole area (that is, the local domain and its
238   surroundings) exhibits a global characteristic in a scene. If only the local saliency is considered,
239   the areas may be reduced to a homogeneous area, which causes blank regions and impedes the
240   realization of object-based focus (for example, a uniformly textured object could solely be salient

241 at its edges). To overcome this limitation, in this work, global saliency is adopted, which is
242 established by guiding an operator through the saliency measure of data. Instead of using each
243 pixel, we calculate the possibility of each patch $P(p_i)$ across the whole scene and determine its
244 inverse to obtain the global saliency S_g as follows (Borji & Itti, 2012; Ming-Ming et al., 2015):
245 1.

246 $$\log\left(S_g{}^c(p_i)\right) = -\log\left(P(p_i)\right) \tag{3}$$

247 and

248 $$\log\left(S_g{}^c(p_i)\right) = -\sum_{j=1}^{n}\log\left(P(\alpha_{ij})\right) \tag{4}$$

250 To calculate $P(p_i)$, the coefficients α are considered to be conditionally independent. This aspect,
251 to some extent, is performed by using a sparse coding algorithm. The description vector of each
252 patch coefficient, that is, the initial binned histogram (100 bins), is determined from each of the
253 patches in the scene and transformed to $(P(\alpha_{ij}))$ by dividing by the sum. In cases in which the
254 patch is difficult to find in one of the features, the previous product is assigned a small value,
255 which leads to a higher global saliency for the entire patch.

257 The proposed method is based upon the measurement of the saliency in each color space. After
258 the measurement, the saliency values are merged into a final saliency map. For every color channel,
259 first, the input image is separated into a nonoverlapping patch. Each patch is symbolized via a
260 coefficient vector of the saliency from the index tree of patches derived from natural scenes.
261 Subsequently, the global and local saliency are determined and combined to represent the saliency
262 of each patch.

264 The saliency contrast maps are consolidated, and the output is calculated as follows:

266 $$S_{lg} = \int S(p)p(Bp\,|\,I)dp \tag{5}$$

268 where I is the input image, and S_{lg} denotes the final saliency map.

270 The local and global saliency maps can be normalized and merged as

272 $$S_{lg}{}^c(p_i) = N\left(S_l^c(p_i)\right) \text{o} \, N\left(S_g^c(p_i)\right) \tag{6}$$

274 where o is an integration structure (scheme, such as -, +, *, min, or max). The saliency values of
275 the image patch in every channel are normalized and summed iteratively to determine the saliency
276 of a patch in every color system. Figure 2 illustrates the global and local saliency of an image
277 patch.

279   **Figure 2.** Illustration of the global and local saliency of an image patch

283 ## System Architecture

284 The objective of this work is to design a smart, low-cost surveillance system that can control
285 surveillance units by using certain devices. The system is expected to monitor and control one or
286 multiple surveillance cameras. The system should be able to detect certain aspects in a smart and
287 automated manner for closed or open areas, such as regions between or outside buildings, and the
288 surrounding areas. The proposed system is composed of a control unit, which controls all the
289 processes, sensors (i.e., ultrasonic sensors to detect motion and distance), and a camera to output
290 continuous video to identify the presence of humans and their activities. Figure 3 illustrates an
291 example of the indoor surveillance system of buildings. The surveillance system includes cameras
292 and ultrasonic sensors distributed to cover the main areas. The system first gathers data from the
293 ultrasonic sensor model (HC-SR04) in real time. When the sensor detects movement in its range,
294 it relays a signal to the camera located in that zone along with the computed distance. The camera
295 recognizes the movements and assesses the threat. In case of any threat, the system alarms the
296 security officer and informs him/her of the type of threat, such as if intruders (or even employees)
297 are present in regions that are out of bounds. Figure 3 illustrates the detection operation.
298 **Figure 3.** Method to activate the camera to detect motion
299
300 The system can be scheduled to perform multiple tasks multiple times by using the recognition
301 process to intelligently analyze moving objects and activity. For example, the system can activate
302 all cameras during working times and recognize faces to identify employers and any authorized
303 guests. If the system detects any face that is not authorized or suspicious, the surveillance officers
304 are alarmed, and the system tracks the suspects. In addition, the system can alert officers if any
305 employer enters an area at a prohibited time. The main concept is to implement a fast recognition
306 system that can perform key recognition even when using compact and low computational units,
307 such as a Raspberry Pi. Such a system can be implemented by utilizing a conventional neural
308 network and computer vision technique known as the saliency algorithm.
309

Hardware Design

311 We employed two systems: a PC-based system, and a single board microcontroller system. The
312 hardware specs of each system are described in Table 1.
313
314 **Table 1.** Hardware specs of each system
315
316 Computers and single-board computers are used as processing units in most processing operations.
317 Cameras are used to stream video from the desired area to process the images and recognize the
318 presence of a person(s) within the camera's field of view. The ultrasonic sensor is used to
319 determine the distance.
320
321 For the single board microcontroller system, the following connections were made with the
322 Raspberry Pi:
323   1- The camera is connected through the CSI camera port.

324   2- The ultrasonic sensor is linked to Raspberry Pi through 4 pins, where VCC is linked to
325   pin 2, GND is linked to ground pin, echo is linked to GPIO 12, and trig is linked to
326   GPIO 16.

328   The connection scheme for the proposed surveillance system based on a single board is shown in
329   Figure 4.

331   **Figure 4.** Connection scheme for the proposed surveillance system based on a single board.

Implementation of the System Program

334   The proposed system program is designed in Python. For both systems, Python version 3.7
335   installed on the operating system (OS: Windows 10 for the PC system and Raspbian for the
336   Raspberry Pi) is used. The program is designed to manage the overall processes, starting from
337   gathering information from the camera(s) (streaming video) and sensor(s) (signals). This program,
338   which uses a CNN, is enhanced by using the saliency map algorithm. The algorithm analyzes the
339   scenes continuously and classifies any detected motion. Subsequently, the algorithm isolates
340   humans and detects threats based on human face recognition and human activity. Face recognition
341   is performed to identify any suspicious person and alarm the observer. Moreover, the system
342   analyzes human activity; and thus, officers are alarmed in the event of any suspicious activity,
343   such as the presence of a gun. However, we designed our model based on work done by (Rosebrock
344   A., 2019) as a reference for designing a CNN model.

Methodology

347   As described previously, the system program is based on the CNN algorithm that is optimized
348   using a computer vision technique known as the saliency algorithm. The proposed system involves
349   the following steps:
350
351   **Data Collection:** Approximately 3450 images (of humans) and 10014 images for three categories
352   of weapons (knife, small gun, and large gun) are collected to perform the classification task by
353   using transfer learning to reduce false positives. For humans, a total of 3450 images were collected
354   and divided into two categories: the training set and the testing set. The captured images of humans
355   included various poses, perspectives and orientations. This can help the deep learning CNN learn
356   the required objects in an efficient way. From the 3450 images of humans, 2761 images were
357   selected for training, and 689 were selected for testing. For weapons, 10014 images are used to
358   classify the three categories of weapons (knife, small gun, and large gun). Of these 8,011 images
359   were used for training, and 2,003 images were used for testing. To repurpose a pretrained model,
360   we fine-tune our model by training certain layers and freezing other layers.

362   **Preprocessing Images:** This step includes several processes, such as augmentation (shift and
363   flip), resizing, rotation, zooming and introducing Gaussian noise. The images were cropped to a
364   square ratio and then resized to 800×800 pixels.

365

Optimization with Saliency Algorithm

367
368
The stepwise procedure of the proposed saliency algorithm can be described as follows:

369 **Step 1:** Image preprocessing: In this part, we first streamed live video as FPS images and later
370 converted it to grayscale.

371 **Step 2:** Image separation: In this part, we segmented the image by using a superpixel algorithm
372 (Achanta et al., 2010; Zhang, Malmberg & Sclaroff, 2019), which is the simple linear
373 iterative clustering (SLIC) algorithm and is based on the typical k-means method in
374 order to group pixels for conventional color areas. SLIC superpixels are made based on
375 two criteria: one criterion is the spectral similarity (limited to 3 channels), and the second
376 criterion is the spatial proximity.

377 **Step 3:** Extracting features from an image: In this stage, the input image is portioned to make it
378 perceptually homogeneous and obtain tiny features by using two algorithms: the
379 Boolean Map Saliency algorithm (BMS) (Zhang, Malmberg & Sclaroff. 2019) and
380 applied LG (Local + Global).

381 **Step 4:** Create index tree: Along with superpixels, a particular index tree is generated to encode
382 the construction information through hierarchical separation. Consequently, we first
383 calculate the gain of every surrounding patch by obtaining the 1st- and 2nd-order
384 reachable matrix (Peng H, et al, 2016).

385 **Step 5:** Recombining: In this part, we recombine all the patches and execute context-based
386 propagation to obtain the final saliency map.

387 **Step 6:** Recognition: The CNN is applied to recognize the separated features and classify the
388 specified objects in the saliency map.

389

390 These images are then annotated and stored in XML format. Then, this XML file is transformed
391 into CSV format and then transformed into TF data that will be input into the deep learning
392 framework. After the TF data have been generated, the training phase is started.

393

394 **Modeling with MobileNet and Faster RCNN**

395 Transfer learning is used to build appropriate models while reducing the time consumption. In this
396 work, pretrained models are used to execute transfer learning. Due to the high computational costs
397 of training complex models, it is a regular practice to import and use existing models (e.g., VGG
398 16 (Simonyan & Zisserman, 2014), Inception (Szegedy, 2014; Szegedy, 2015) or MobileNet
399 (Howard, 2017)). In this work, we utilized MobileNet as a feature extractor.

400

401 The RPN is composed of two layers used to find the areas that can include objects in an image
402 (feature maps). The network utilizes the ROI pooling layer to minimize and resize resource maps
403 depending on proposals from that area. The maps make use of the new features of every area to
404 select frames through three fully connected layers (FCLs). MobileNet has been used as a CNN that
405 takes the layers as learning functions; hence, the original feature extraction has several layers.

406 However, the first convolution stack structures are obtained via transfer learning through the use
407 of MobileNet. The method includes two steps forming the current surveillance detection. The first
408 step determines the ROIs in images. All these ROIs are considered references indicating several
409 possible object sites that are created in the second step. Figure 5 shows the proposed model that
410 consists of five convolutional layers and three FC layers. The faster RCNN mainly uses the last
411 convolutional layer features to perform classification and localization. After the two convolutional
412 layers, the outputs of the last three convolutional layers (layers three, four and five) are utilized as
413 input data to the 3 levels of pooling of the ROI and the related normalization levels. For every
414 RPN anchor forming a fully convolutional network, a degree is forecasted that makes it able to
415 determine the probability that this anchor has the element of interest. Moreover, the RPN offers
416 the acceleration and measurement coefficients for every anchor, which is a part of the peripheral
417 regression mechanism, and thus enhances the position of the object.

418

419 **Figure 5.** The proposed faster RCNN approach. The structure consists of 5 convolutional layers
420 and 3 fully connected layers.

421

422 As an illustration, the architecture includes two steps. First, the RPN presents a set of bounding
423 boxes having a trusted rating related to a potential human image. The second step defines the
424 analysis of those fully convolution architectures using MobileNet as a feature extractor after
425 obtaining the output feature map from a pretrained model, which is MobileNet. As we used an
426 input image with a resolution of 800x800 in x3 dimensions, the output feature map should be
427 50x50x256 dimensions. Every point in a 50x50 area represents an anchor. Hens, we must specify
428 sizes and specific ratios for every anchor, which are 128^2, 256^2, and 512^2 for the three respective
429 sizes and 1:1, 1:2, and 2:1 for the three respective ratios, in the original image. Then, the RPN is
430 linked to a conv layer using 3×3 filters, 1 padding, and 512 output channels. Then, the output is
431 linked to two 1×1 convolutional layers for box regression and classification (where the
432 classification is used to verify if the box is an object or is not). In such a case, each anchor will
433 have 9 corresponding boxes from the original image, which means there are 50x50x9 = 22,500
434 boxes in the original image. We only select 256 of these 22,500 boxes to be a minibatch that has
435 128 backgrounds (neg) and 128 foregrounds (pos). Simultaneously, nonmaximal suppression is
436 implemented to ensure that there is zero overlap for those proposed regions. When the previous
437 steps are finished, then the RPN is finished. In the second stage of the RCNN, similar to the fast
438 RCNN, ROI pooling is utilized for these proposed areas (ROIs). Then, we flatten this layer using
439 some fully connected layers. The last step is a softmax function for linear regression and
440 classification to fix the boxes' locations. Figure 6 shows the FRCNN/RPN structures

441

442   (a)
443   (b)
444 **Figure 6.** (a) First step of the FRCNN; and (b) the RPN structures, where k is the number of
445 anchors.

446
447 **Results**
448 This section describes the accuracy gain when the saliency method is applied to generate a saliency
449 map, which is used as an input for the CNN to detect related objects. Figure 7 and Figure 8 show
450 the saliency maps generated from a live streamed video.
451   (a)
452   (b)
453   **Figure 7.** Saliency map results for humans (in real-time streaming). (A) and (B) Streamed
454   images; and (C) and (D) The output (saliency results) of streamed images (A) and (B),
455   respectively
456
457   **Figure 8.** Saliency map results for humans with a gun (in real-time streaming). (A) Streamed
458   images, and (B) The output (saliency results) of the streamed image (A).
459
460 As shown in Figure 7 and Figure 8, the human body and gun are extracted from a highly detailed
461 image that involves many objects. The results indicate that the proposed method efficiently
462 removes foreground objects (humans and guns) from other objects in a scene with sufficient detail.
463 The saliency map is passed to the RCNN to recognize the human and gun. The recognition process
464 results are shown in Figure 9.
465
466   (a)
467   (b)
468   **Figure 9.** Detection results (in real time streaming**. (A) Human and gun detection results). (A)
469   Detection of two humans in this scene, (B) Scene output for human detection, (C) Detection of
470   humans with guns, and (D) Scene output for human and gun detection.
471
472 As shown in Figure 9, the CNN can successfully recognize humans in real time. The recognition
473 had an accuracy of 99.2%, and it can detect an object with no slowdown or missing object failure.
474 In particular, the system tracked multiple humans in only 0.9 s. The mean relative error was
475 computed by normalizing the given values through the following formula:
476
477        $\text{metric} = \text{mean}(|y_{pred} - y_{true}|/normalizer))$
478
479 which is based on the TensorFlow metric "tf.keras.metrics.MeanRelativeError", (TensorFlow,
480 website, 2020). The mean relative errors of recognition are presented in Table 2.
481
482   **Table 2.** Recognition results
483
484 By using transfer learning, we reduced the false positives, as indicated by the metrics. Specifically,
485 the validation mask loss is approximately 0.475, and the validation class loss is approximately
486 0.0383.

487

488 We tested the system to recognize and detect a person in real working operations. The system first
489 detects if there is any movement in the range of the ultrasonic sensor. If there is movement, the
490 system then opens the camera and detects if there is a human in the field of view of the camera. If
491 yes, then the system turns the light on and shows a screen and alarm to the surveillance officer.
492 Figure 10 shows the detection process in a room.

493

494

495   **Figure 10.** Detection process of the proposed system**.** (A) Human detection screen. The green
496   square represents the detected human. (B) Human distance calculation. The system computes the
497   distance of the human from a wall when entering the surveillance area, activates the light (light
498   text in Figure) at a distance of 5 m and sounds an alarm while showing the camera view on the
499   monitor screen where the human is detected (TV text in Figure).

500

501 As shown in Figure 10, we tested the system's operations when a human enters the surveillance
502 area. When an ultrasonic sensor detects movement, it computes the distance and switches on the
503 camera, thereby initiating the recognition process. If the system detects humans, the camera is
504 continuously switched on to track the movement of the person and analyze the activity while
505 continuously computing the distance. If the human enters a restricted area or has a weapon, the
506 system alarms the observer and displays the stream from the camera. The system thus successfully
507 detects humans and guns and isolates them from other objects in the area. The results indicate that
508 the system successfully and effectively detects humans and guns with a high accuracy between
509 16% and 99% with a low response time of approximately 0.9 s. However, even in the case with
510 low accuracy, detection and isolation are successfully performed for humans, even when a part of
511 the human body is hidden. The algorithm in such cases exhibits reasonable performance for
512 detection movement and computes distance with the error ranging between 0 and 5 m. Overall, the
513 system can perform 100% detection for objects and can track humans and guns.
514 We also compared the recognition processes of the systems based on the PC and Raspberry Pi.
515 The PC system has high computational hardware (core i7 8750 3.9 GHz CPU, 16 GB DDR4 RAM,
516 a 256 GB NVMe SSD HDD, and an HD webcam). We compared the detection time and
517 recognition percentage in a room with the same lighting conditions. The results are presented in
518 Table 3 and Figure 11.

519

520   **Table 3.** Recognition results for the PC-based system and the Raspberry Pi-based system

521

522

523   **Figure 11.** Detection results for the PC-based system and Raspberry Pi-based systems. (A) The
524   speed of the system's response to the number of people in the scene. (B) Detection efficiency of
525   the number of people in the scene

526

Conclusions

In this work, we designed a smart surveillance system utilizing a low-cost computing unit and a CNN to monitor certain aspects and alarm observers automatically. A single board computer, a Raspberry Pi 3 version B, is used as the central controller that manages several tasks at the same time. For distance detection, a low-cost ultrasonic sensor type (HC-SR04) is used to sense motion and compute the distance from moving objects within the monitoring area. The recognition model can recognize and track the desired moving object (human) in real time, detect his/her activity (in this work, we focused on gun detection) and alarm officers if the situation is critical. The model is based on the faster RCNN optimized by using a saliency algorithm for feature extraction. Compared with the existing saliency methods, the proposed method does not require a database to identify objects, and it uses the local and global approach to generate a saliency map that enables fast and accurate feature extraction. The main results that have been achieved are described as follows:

- The overall system works smoothly and efficiently, and the controller can successfully control multiple tasks simultaneously with no failure.
- The recognition model operates promptly and accurately.
- The RCNN part of the proposed model is different from other surveillance approaches in that this model can use a low computational component, such as a Raspberry Pi, to perform multiple tasks with accurate and fast recognition. In this manner, a compact dedicated smart surveillance camera can be used to integrate this system to establish a control room to control a large number of surveillance cameras to perform multiple tasks, such as the surveillance of institutes, military bases, and cities.
- The recognition process is fast and highly accurate owing to the use of saliency algorithms with the RCNN. The main advantage of using a saliency algorithm is the reduction of the image details due to the removal of undesired features from the image and retaining only the critical objects in the scene. In this manner, the system successfully isolates the essential features and uses these features in the training/recognition process. The removal of unimportant objects along with a reduction in the image details can reduce the computational requirements and increase both the accuracy and speed of the training/recognition process.
- The most critical achievement of this work is the reduction in the computational requirements and improvement in the recognition process in both speed and accuracy. The results show that the system can recognize humans and threats (e.g., a human handling a gun) in any situation with recognition rates ranging from 16% to 99.4%.
- Even with a low recognition percentage, the system can successfully detect and classify humans and guns with an accuracy of nearly 100% in different situations (for instance, in cases in which a human is partially hidden behind certain objects).
- The model can work on low computational systems, such as a single board computer, with a fast processing time.

566   •   The system achieved real-time detection of humans in less than 1 s when using both
567   Raspberry Pi and PC models.

568

569 In summary, the model can perform fast recognition, which is essential in surveillance systems.
570 However, the model needs more research and improvement, and I suggest the following:

571   •   Use other architectures, such as the CNN, YOLO, SSD, mask RCNN, etc.
572   •   Extend the system to detect other types of objects or even behaviors (e.g., theft or violence).
573   •   Complement the system with another PC with more resources capable of performing online
574   learning to retrain the system with new images.

575

Acknowledgments

577 This work was partially supported by the National Natural Science Foundation of China (Nos.
578 61672215, and 61976086), the National Key R&D Program of China (No. 2018YFB1308604),
579 and the Hunan Science and Technology Innovation Project (No. 2017XK2102).

580

References

582 **Achanta R, Shaji A, Smith K, Lucchi A, Fua P and Susstrunk S. 2012.** SLIC superpixels
583 compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell.
584 34(11):2274-82. DOI: 10.1109/TPAMI.2012.120.
585 **Aggarwal CC. 2018.** Neural Networks and Deep Learning: A Textbook. Springer International
586 Publishing. DOI: 10.1007/978-3-319-94463-0
587 **Aldasouqi I and Hassan M. 2010.** Human face detection system using HSV. CSECS '10:
588 Proceedings of the 9th WSEAS international conference on Circuits, systems, electronics, control
589 & signal processing. pp.13-16.
590 **Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam**
591 **H. 2017.** MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.
592 Computer Vision and Pattern Recognition. arXiv:1704.04861.
593 **Avidan, S. 2007.** Ensemble tracking, IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 2, pp.
594 261– 271.
595 **Borji A, Cheng MM, Hou Q, Jiang H, and Li J. 2019.** Salient object detection: A survey.
596 Computational Visual Media. Computational Visual Media, Springer. 5:117–150.
597 **Borji A, Frintrop S, Sihite DN, and Itti L. 2012.** Adaptive object tracking by learning
598 background context. In Proceedings of the IEEE Computer Society Conference on Computer
599 Vision and Pattern Recognition Workshops, pp. 23–30.
600 **Borji A and Itti L. 2012.** Exploiting local and global patch rarities for saliency detection. 2012
601 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA. DOI:
602 10.1109/CVPR.2012.6247711.
603 **Cong, Y, Gong, H, Zhu S and Tang Y.**: Flow mosaicking: Real-time pedestrian counting without
604 scene-specific learning, in Proc. CVPR, 2009, pp. 1093–1100.

605 **Cong, Y, Yuan J, Luo J. 2012.** Towards scalable summarization of consumer videos via sparse
606 dictionary selection, IEEE Trans.Multimedia, vol. 14, no. 1, pp. 66–75.

607 **Cserey G, Porod W and Roska T.** 2004. An Artificial Immune System Based Visual Analysis
608 Model and Its Real-Time Terrain Surveillance Application. International Conference on Artificial
609 Immune Systems. DOI: 10.1007/978-3-540-30220-9_21.

610 **Dalal N and Triggs B. 2005.** Histograms of oriented gradients for human detection, in Proc.
611 CVPR, pp. 886–893.

612 **Daniilidis K, Maragos P and Paragios N. 2010.** Computer Vision. 11th European Conference
613 on Computer Vision, Proceedings, Part II, Heraklion, Crete, Greece, Proceedings, Part V.   DOI:
614 10.1007/978-3-642-15552-9.

615 **Elliott D. 2010.** Intelligent video solution: A definition, Security, pp. 46–48.

616 **Frintrop S, García GM, and Cremers AB. 2014.** A cognitive approach for object discovery. in
617 Proceedings of the IEEE 22nd International Conference on Pattern Recognition (ICPR '14).
618 pp.2329–2334. DOI: 10.1109/ICPR.2014.404.

619 **Girshick R. 2015.** Fast R-CNN. Computer Vision and Pattern Recognition. Cite as:
620 arXiv:1504.08083.

621 **Girshick R., Donahue J., Darrell T., and Malik J. 2014.** "Rich feature hierarchies for accurate
622 object detection and semantic segmentation. Computer Vision and Pattern Recognition (CVPR).
623 Cite as: arXiv:1311.2524.

624 **Junoh AK, Mansor MN, Abu SA, and Ahmad WZW. 2012.** SVM Classifier for Automatic
625 Surveillance System. Procedia Engineering, Elsevier Ltd. 38:1806-1810. DOI:
626 10.1016/j.proeng.2012.06.222.

627 **Kanan C. and Cottrell G. 2010.** Robust classification of objects, faces, and flowers using natural
628 image statistics," in Proceedings of the IEEE Computer Society Conference on Computer Vision
629 and Pattern Recognition (CVPR '10), IEEE, San Francisco, USA. pp.2472–2479.

630 **Kang S, Lee H, Kim J and Kim J. 2012.** Automatic Image Segmentation Using Saliency
631 Detection and Superpixel Graph Cuts. Springer, Berlin, Heidelberg. Robot Intelligence
632 Technology and Applications. 1023-1034. DOI: 10.1007/978-3-642-37374-9_99.

633 **Khan Z and Gu, I. 2010.** Joint feature correspondences and appearance similarity for robust
634 visual object tracking, IEEE Trans. Inf. Forensics Security, vol. 5, no. 3, pp. 591–606.

635 **Klein DA, Schulz D, Frintrop S, and Cremers AB. 2010.** Adaptive real-time video-tracking for
636 arbitrary objects," in Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent
637 Robots and Systems (IROS '10), Taipei, Taiwan. pp. 772–777.

638 **Liu J and An FP. 2020.** Image Classification Algorithm Based on Deep Learning-Kernel
639 Function. Hindawi, Scientific Programming. vol. 2020, Article ID 7607612, pp1-14.

640 **Ming-Ming C, Niloy JM, Xiaolei H, Philip HSTand Shi-Min H. 2015.** Global Contrast Based
641 Salient Region Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 37
642 (3): 569-582. DOI: 10.1109/TPAMI.2014.2345401.

643 **Moosmann F., Larlus D., and Jurie F. 2006.** Learning saliency maps for object categorization,"
644 in Proceedings of the Workshop on the Representation & Use of Prior Knowledge in Vision
645 (ECCV '06).

646 **Nguyen HQ, Loan TTK, Mao BD and Huh EN. 2015.** Low-cost real-time system monitoring
647 using Raspberry Pi. 2015 Seventh International Conference on Ubiquitous and Future Networks,
648 Sapporo, Japan, IEEE. 857-859. DOI: 10.1109/ICUFN.2015.7182665.

649   **Park U and Jain A. 2010**. Face matching and retrieval using soft biometrics, IEEE Trans. Inf.
650   Forensics Security, vol. 5, no. 3, pp. 406–415.

651   **Peng H, Li B, Ling H, Hu W, Xiong W, Maybank SJ. 2016**. Salient object detection via
652   structured matrix decomposition. IEEE transactions on pattern analysis and machine intelligence.
653   vol.39, no.4, pp.818-32.

654   **Petrosino A. and Maddalena L. 2012.** Neural Networks in Video Surveillance: A Perspective
655   View. Handbook on Soft Computing for Video Surveillance, Volume: Chapman & Hall/CRC. pp.
656   59-78. DOI: 10.1201/b11631-4.

657   **Ren S, He K, Girshick R and Sun J. 2016.** Faster R-CNN: Towards Real-Time Object Detection
658   with Region Proposal Networks. Advances in Neural Information Processing Systems 28 (NIPS
659   2015). pp.1-9.

660   **Sabri N, Salim MS, Fouad S, Aljunid SA, AL-Dhief FT and Rashidi CBM. 2018.** Design and
661   Implementation of an Embedded Smart Intruder Surveillance System. MATEC Web of
662   Conferences 150 Malaysia Technical Universities Conference on Engineering and Technology
663   (MUCET 2017). 150:1-6. DOI: 10.1051/matecconf/201815006019.

664   **Salahat E, Saleh H, Mohammad B, Al-Qutayri M, Sluzek A and Ismail M. 2013.** IEEE 20th
665   International Conference on Electronics, Circuits, and Systems (ICECS), Abu Dhabi, United Arab
666   Emirates. DOI: 10.1109/ICECS.2013.6815354.

667   **Sedky M, Moniri M and Chibelushi CC. 2005.** Classification of smart video surveillance
668   systems for commercial applications. IEEE Conference on Advanced Video and Signal Based
669   Surveillance (AVSS).

670   **Shang W, Sohn K, Almeida D, and Lee H. 2016.** Understanding and improving convolutional
671   neural networks via concatenated rectified linear units. in ICML.

672   **Singh, V., Kankanhalli, M.** 2009Adversary aware surveillance systems, IEEE Trans. Inf.
673   Forensics Security, vol. 4, no. 3, pp.552–563.

674   **Simonyan K and Zisserman A. 2015.** Very Deep Convolutional Networks for Large-Scale Image
675   Recognition. Computer Vision and Pattern Recognition. arXiv:1409.1556.

676   **Rosebrock A. 2019**. Live video streaming over network with OpenCV and ImageZMQ. Available
677   from: https://www.pyimagesearch.com/2019/04/15/live-video-streaming-over-network-with-
678   opencv-and-imagezmq.

679   **Stalder S, Grabner H, and Gool LV. 2012.** Dynamic objectness for adaptive tracking. In
680   Proceedings of the Asian Conference on Computer Vision. pp. 1–14.

681   **Sun F and Li W. 2018.** Saliency detection based on aggregated Wasserstein distance. J. of
682   Electronic Imaging, SPIE. 27(4): 043014. DOI: 10.1117/1.JEI.27.4.043014.

683   **Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V,**
684   **Rabinovich A.   2014.** Going Deeper with Convolutions. Computer Vision and Pattern
685   Recognition.   arXiv:1409.4842.

686   **Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. 2015.** Rethinking the Inception
687   Architecture for Computer Vision. arXiv:1512.00567.

688   **TensorFlow. 2020.** Mean Relative Error. Available from link: https://www.tensorflow.org/
689   api_docs/python/tf/keras/metrics/MeanRelativeError

690   **Troscianko T, Holmes A, Stillman J, Mirmehdi M, Wright D, Wilson A.** 2004. What happens
691   next? The predictability of natural behaviour viewed through CCTV cameras. Perception. 33(1)
692   :87-101. DOI : 10.1068/p3402

693 **Turchini F, Seidenar L, Uricchio T and Bimbo A. 2018.** Deep Learning-Based Surveillance
694 System for Open Critical Areas. Inventions. 3(4):69. DOI:10.3390/inventions3040069.
695 **Verhelst M and Moons B. 2018.** Embedded Deep Neural Network Processing: Algorithmic and
696 Processor Techniques Bring Deep Learning to IoT and Edge Devices. IEEE Solid-State Circuits
697 Magazine 9(4):55-65. DOI: 10.1109/MSSC.2017.2745818.
698 **Véstias MP. 2019.** A Survey of Convolutional Neural Networks on Edge with Reconfigurable
699 Computing. Algorithms, MDPI. 12(8), 154, pp.1-24. DOI: 10.3390/a12080154.
700 **Wang L. 2006.** Abnormal walking gait analysis using silhouette-masked flow histograms, in Proc.
701 ICPR, vol. 3, pp. 473–476.
702 **Wang S and Lee H. 2007.** A cascade framework for a real-time statistical plate recognition
703 system, IEEE Trans. Inf. Forensics Security, vol. 2, no. 2, pp. 267–282.
704 **Wang W, Shen J, Xie J, Cheng MM, Ling H, Borji A. 2019.** Revisiting Video Saliency
705 Prediction in the Deep Learning Era. IEEE Transactions on Pattern Analysis and Machine
706 Intelligence. DOI: 10.1109/TPAMI.2019.2924417.
707 **Yanulevskaya V, Uijlings JRR and Geusebroek JM. 2013.** Salient object detection: from pixels
708 to segments. Image and Vision Computing 31(1):31–42. DOI: 10.1016/j.imavis.2012.09.009.
709 **Yu X, Chinomi K, Koshimizu T, Nitta N, Ito Y and Babaguchi N. 2008.** Privacy protecting
710 visual processing for secure video surveillance, in Proc. ICIP, pp. 1672–1675.
711 **Zhang J, Malmberg F, Sclaroff S. 2019.** Visual Saliency: From Pixel-Level to Object-Level
712 Analysis. Springer International Publishing. DOI: 10.1007/978-3-030-04831-0.
713 **Zhaoyu P, Pingping L and Changjiu L.** 2009. Automatic Detection of Salient Object based on
714 Multi-features 2008 Second International Symposium on Intelligent Information Technology
715 Application, IEEE. DOI: 10.1109/IITA.2008.466.
716
717
718
719

Hardware specs of each system

PC-Based System			Single Board Microcontroller System		
Item	Specifications	Cost	Item	Specifications	Cost
System	Intel NUC NUC7CJYH 1. 2GHz Intel Celeron Processor 2. 8 GB Ram 3. 128 GB SSD 4.	$139	System	Raspberry Pi 3 version B+ 1. 1.2 GHz Broadcom BCM2837CPU 2. 1 GB of RAM	47$
Camera	Logitech HD Webcam C615	85$	Room	32 GB SanDisk Ultra Micro SDHC	10$
	Or, Commercial fast low cost 1080p Webcam	13$	Camera	Raspberry Pi Camera Version 2	24$
Sensor	MaxBotix MB1043 HRLV MaxSonar Ultrasonic Range Finder	20$	Sensor	HC-SR04 ultrasonic sensor	2$
Total Cost	244$ or 172$		Total Cost	83$	

Recognition results

Object	Error
Human	4.3536820e-05
Knife	2.6346240e-04
Large Gun	9.1683286e-01
Small Gun	8.2903586e-02

Recognition results for the systems-based PC and Raspberry Pi based System

Number of persons in view	Detection Time for System Based on PC		Detection Time for System Based on Raspberry Pi	
	(s)	Accuracy (%)	(s)	Accuracy (%)
1	0.69	99.4	0.71	99.1
2	0.7	98.45	0.73	98.7
3	0.7	96.4	0.79	97.8
4	0.7	92.5	0.8	95.4
5	0.73	93.1	0.86	94.3
6	0.75	98.5	0.94	98.9
7	0.74	99.4	0.98	98.7
8	0.74	99.4	1.3	93.1
9	0.75	99.4	1.98	91

Figure 1

A typical control room pertaining to traditional surveillance systems.

© 123rf.com
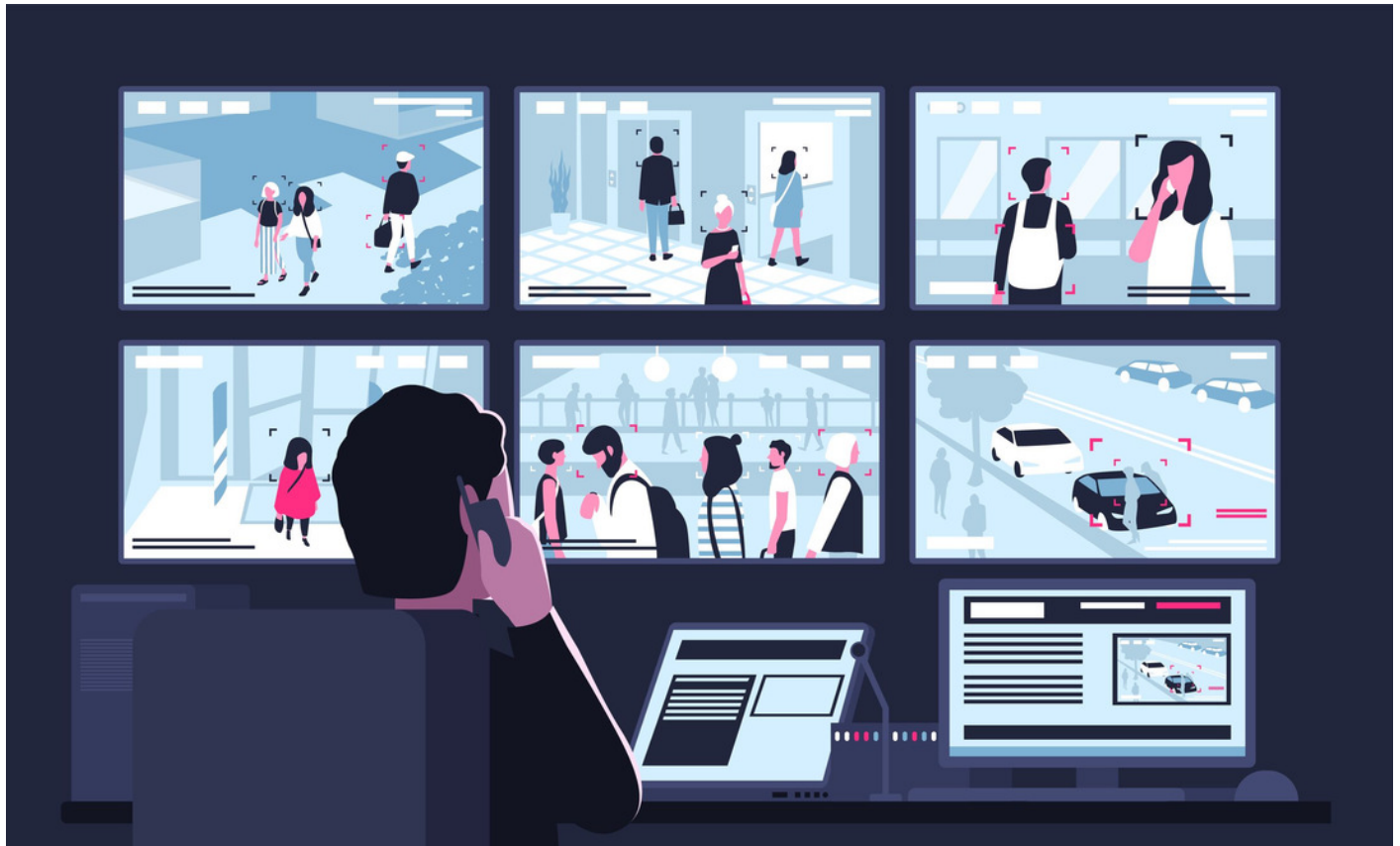
Figure 2

Illustration of global and local saliency of an image patch.
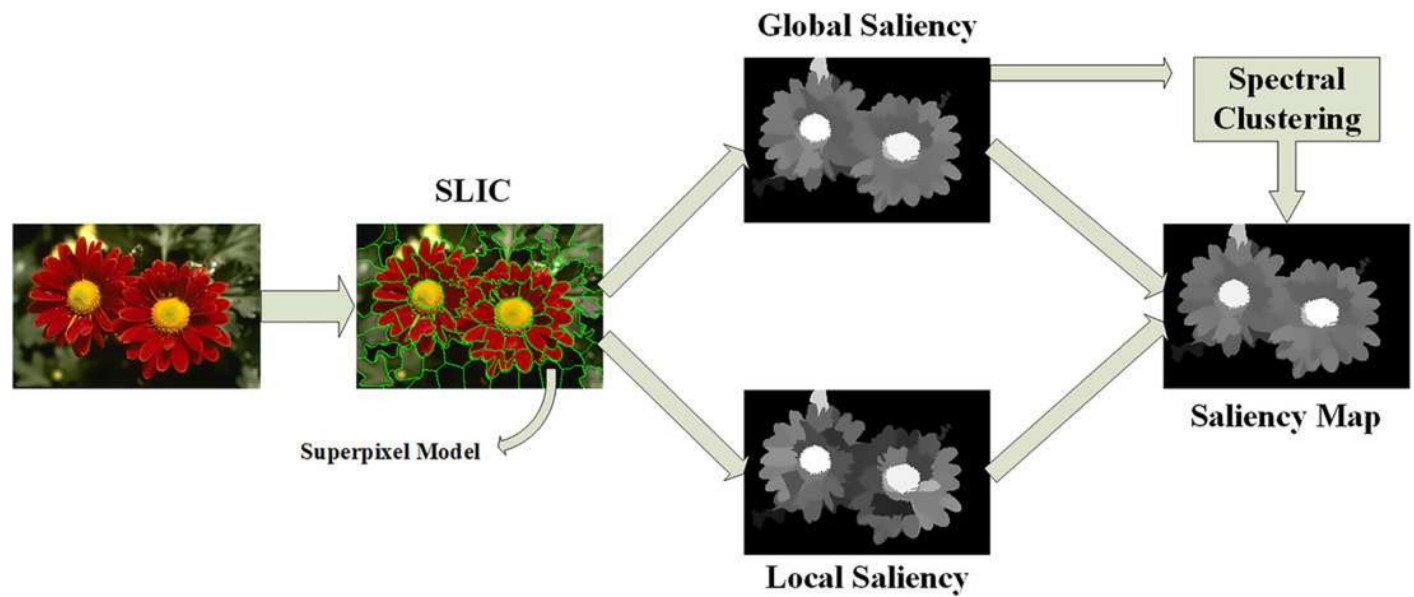
Figure 3

Method to activate the camera on detecting motion.
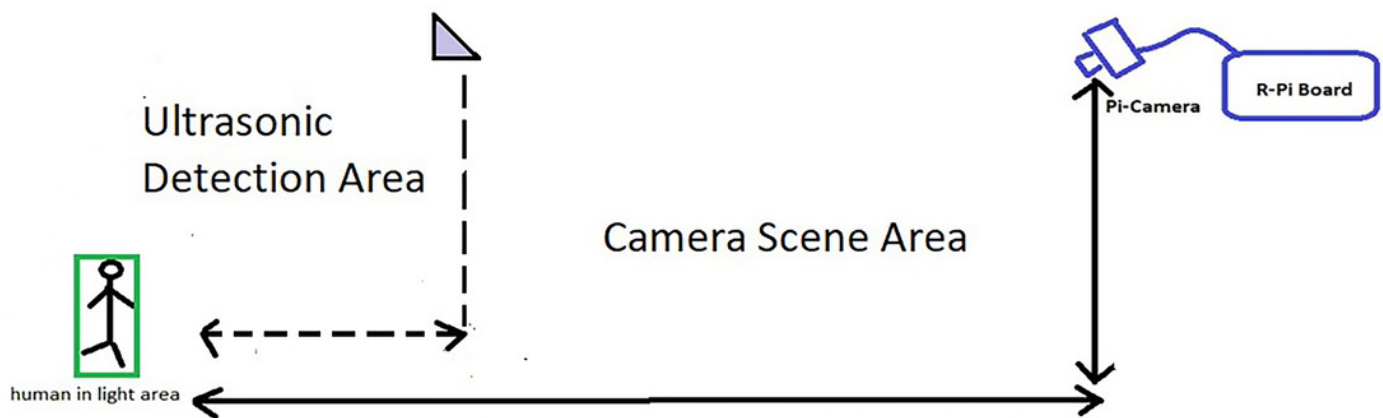
Figure 4

Connection scheme for the proposed surveillance system based on a single board.
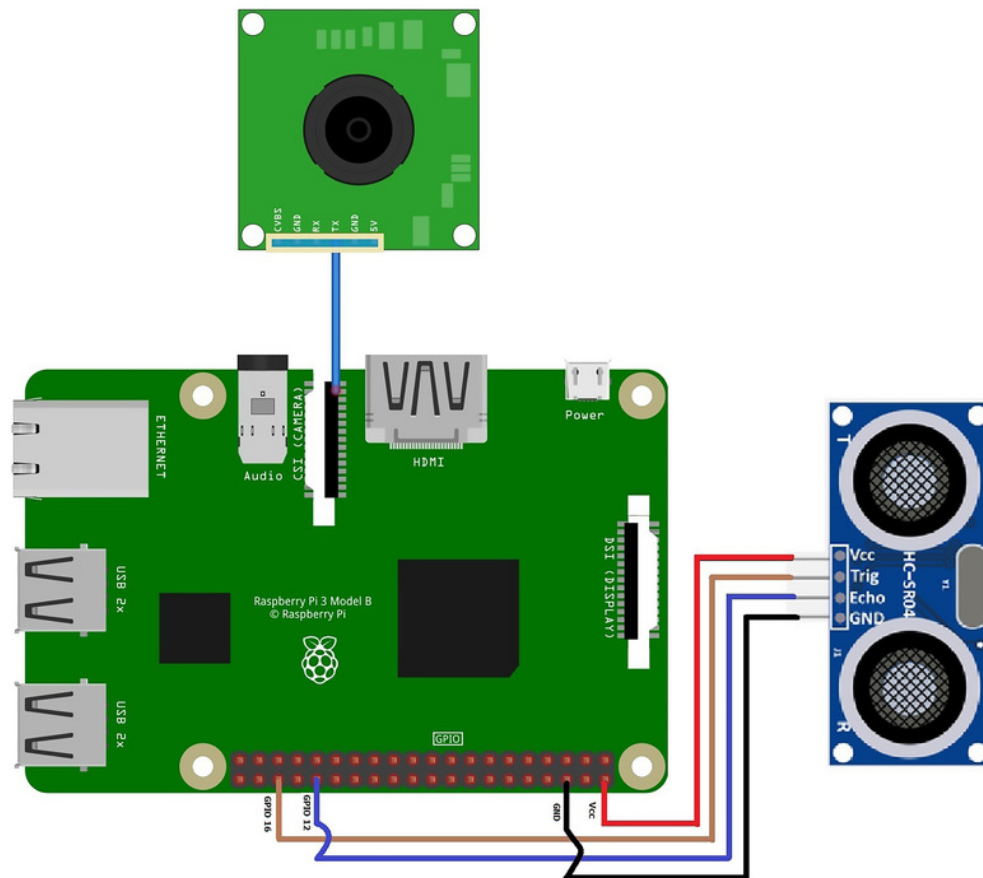
Figure 5

The proposed Approach of Faster RCNN. The structure is consisted from 5 convolutional layers and 3 fully connected layers.
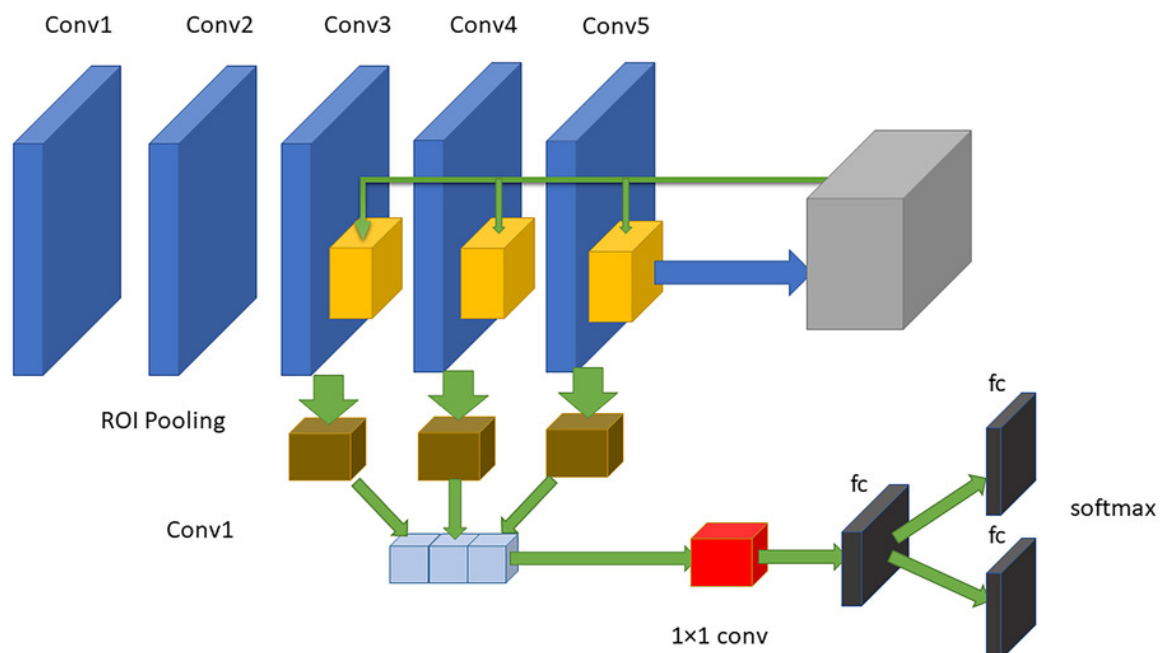
Figure 6

Figure 6. (a) First step of FRCNN , (b) RPN structures, in which k is the anchors number.
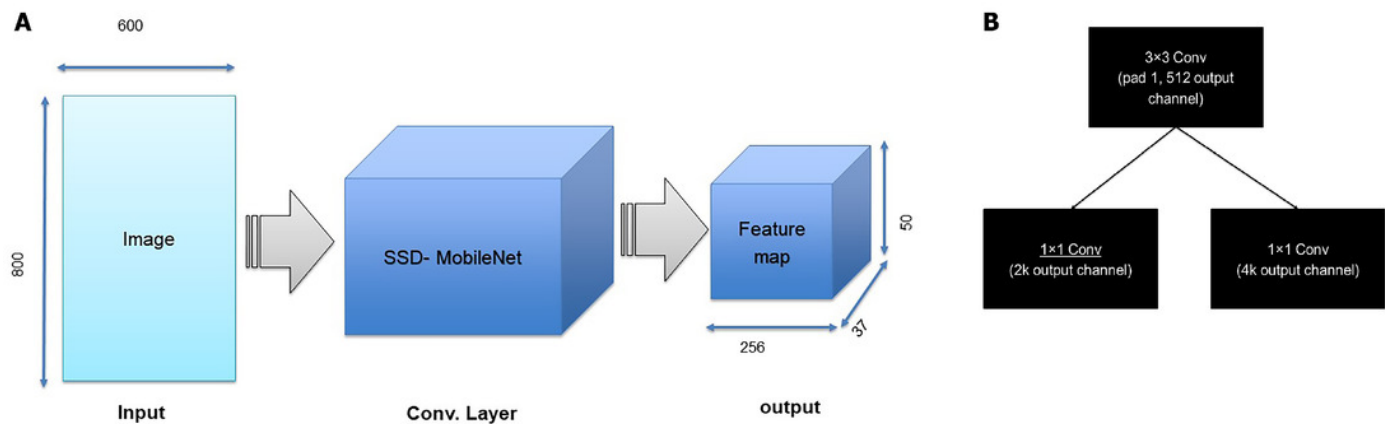
Figure 7

Figure 7. Saliency map results for humans (in real time streaming) .

(A), (B) Streamed Images, (C), (D) The output (Saliency Results) of streamed image (A) and (B) respectively
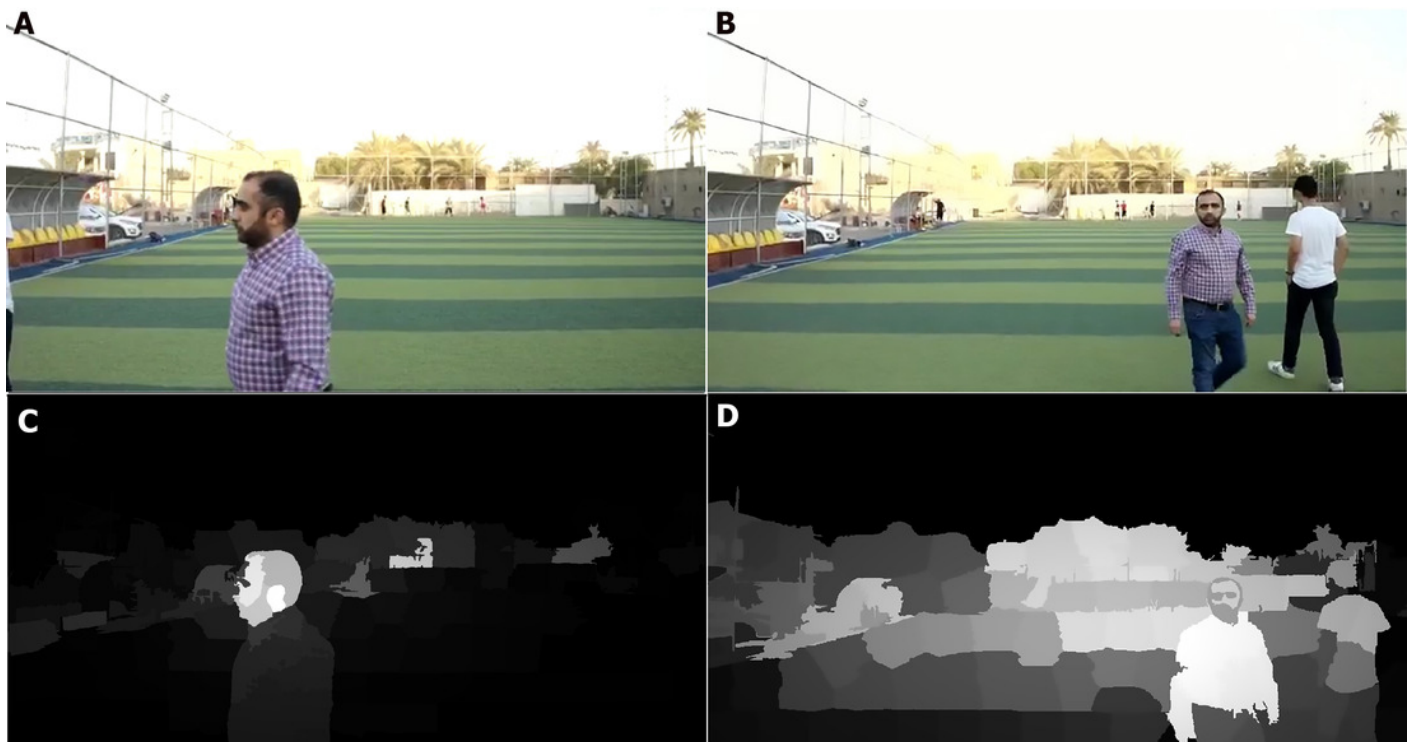
Figure 8

Figure 8. Saliency map results for humans with a gun (in real time streaming) .

(A) Streamed Images, (B) The output (Saliency Results) of streamed image (A).

Figure 9

Figure 9. Detection results (in real time streaming. (A) Human and gun detection results
).

(A) Detection of two humans in the sense, (B) Sense output for humans detection(C)

Detection of human with gun, (D) Sense output for humans and gun detection.
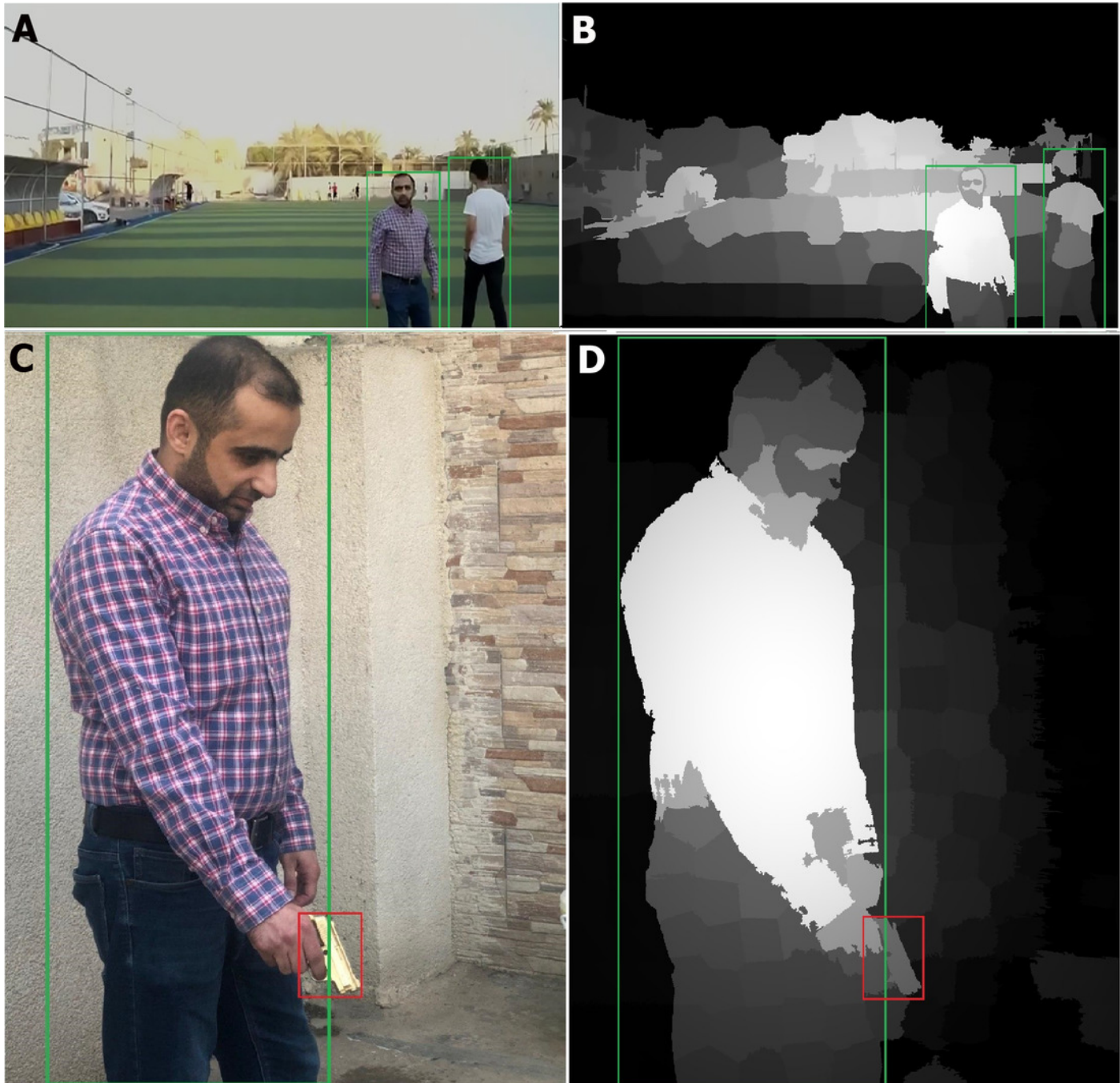
Figure 10

Figure 10. Detection process of the proposed system.

(A)Human detection screen, a green square represents the detected human. (B) Human distance calculation (the system computes the distance of the human from a wall when entering the surveillance area and activates the light (light text in figure) at distance of 5 meter and making alarm with showing the camera view in monitor screen where human detect (TV text in figure).

Figure 11

Figure 11. Results of detection for PC based systems and Raspberry Pi based System.

(A)The speed of the system response to the number of people in the scene. (B)Detection efficiency to the number of people in the scene