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 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 (such as people, cars, and guns) and promptly detect human activity in real time by using low computational systems such as compact or single board computers. Deep learning techniques are employed to detect certain objects to realize the surveillance of homes/buildings and recognize suspicious and vital events to ensure that the system can alarm officers of relevant events such as stranger intrusions, presence of guns, suspicious movement, and fugitive identification. The proposed model is tested on two computational systems, specifically, a single board computer (Raspberry PI) with Raspbian OS and a compact computer (Intel NUC) with 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 realize recognition. The program is optimized by using a foreground object detection algorithm to accelerate the recognition in terms of both the accuracy and speed. The saliency algorithm is used to slice certain required objects from the scenes, such as humans, cars, and airplanes. In this regard, two saliency algorithms are considered, based on the local and global patch saliency detection. 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 the 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. Nevertheless, even in the case of low accuracy, the recognition and threat identification are realized with an accuracy of 100% in approximately 0.7 s, even when using a single board computer. These results indicate that the proposed system can be practically used to design a low cost and intelligent security

and tracking system.

Low cost intelligent surveillance system based on fast CNN

Zaid Saeb Sabri[1,2], Zhiyong Li[1]
[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China; zaidsaeb@hnu.edu.cn
[2] Department of Computer Science & Information Systems, AL-Mansour University College, Al-Andalus Square, Baghdad, Iraq; zaid.saeb@muc.edu.iq
Corresponding Author:
Zhiyong Li[1]
Changsha, Hunan 410082, China
Email address: zhiyong.li@hnu.edu.cn

Abstract

Smart surveillance systems are used to monitor specific areas, such as homes, buildings, and borders, and 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 (such as people, cars, and guns) and promptly detect human activity in real time by using low computational systems such as compact or single board computers. Deep learning techniques are employed to detect certain objects to perform the surveillance of homes/buildings and recognize suspicious and vital events to ensure that the system can alarm officers of relevant events such as stranger intrusions, presence of guns, suspicious movement, and fugitive identification. The proposed model is tested on two computational systems, specifically, a single board computer (Raspberry PI) with Raspbian OS and a compact computer (Intel NUC) with 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 accelerate the recognition in terms of both the accuracy and speed. The saliency algorithm is used to slice certain required objects from the scenes, such as humans, cars, and airplanes. In this regard, two saliency algorithms are considered, based on the local and global patch saliency detection. 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 the 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. Nevertheless, even in the case of low accuracy, the 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 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.

Introduction

Most traditional video surveillance systems are based on the use of surveillance cameras connected to monitor screens. However, in recent times, a need to detect and classify normal or abnormal events has emerged to suitably assess a given situation to adopt security measures. Typically, in standard surveillance systems that involve a large number of surveillance cameras covering a large area, certain operators must continuously check the real time footage recorded by the cameras (Figure 1). In the event of an undesirable incident, the operators must alert the security or police. In certain surveillance camera systems, the monitors display the video stream from a single camera. However, in most cases, a single monitor displays multiple streams from several cameras such as 4, 8, or 16 cameras in a sequentially or simultaneous manner (Aldasouqi & Hassan, 2010).

Furthermore, practically, the operators cannot monitor all the screens all the time; instead, the camera output is recorded by using video recorders such as DVRs or NVRs. If an incident occurs, such video footage is used as evidence. One drawback of this strategy is that the operators cannot address the incidents or prevent any related damage in real time, as the recordings can only be viewed at a later time. Moreover, considerable time is required to find the relevant section of the recording, as often, the suspect is at the scene long before the incident occurs and the recording may correspond to multiple cameras. Consequently, it is necessary to develop a method or technique that can instantly analyze and detect threats based on the detection of humans and their activities (Salahat et al., 2013; Troscianko et al., 2004).

Figure 1. A typical control room pertaining to traditional surveillance systems

In the last decade, modern video surveillance systems attracted increasing interest, with several studies focusing on automated video surveillance systems, which involve a network of surveillance cameras with sensors that can monitor human and nonhuman objects in a specific environment. Pattern recognition can be used to find specific arrangements of features or data, which usually yield details regarding a presented system or data set. In a technical context, a pattern can involve repeating sequences of data with time, which can be utilized to predict trends and specific featural configurations in images to recognize objects. Many recognition approaches involving the use of the support vector machine (SVM) (Junoh et al., 2012), artificial neural network (ANN) (Petrosino & Maddalena, 2012), deep learning (Wang et al., 2019), and other rule-based classification systems have been developed. Performing classification using ANN is a supervised practical strategy that has achieved satisfactory results in many classification tasks. The SVM is required less computational requirement compared to ANN however, it provides lower recognition accuracy in comparison with ANN. In recent years, the network has played a significant role in a wide area of application and it has been employee to serve the surveillance systems. In last years, as unstructured and structured data sizes enlarged to big data levels, researchers had developed deep learning systems that are basically neural networks having several layers. Deep learning allows to capture and mining of greater and larger data, including unstructured data. This approach can be used to model complicated relationships between inputs and outputs or to find patterns. However, the associated accuracy and classification efficiency are generally low (Liu J and an FP, 2020). Many strategies have been developed to increase the recognition accuracy. In this work, we discuss the use of the accuracy gain by adopting certain saliency methods to improve the recognition and detection of an object and perform its isolation from a scene.

The performance efficiency of the existing surveillance systems is highly depending on the activity of human operators, that are responsible on monitoring the camera footage (Sedky, Moniri and Chibelushi, 2005). In general, most medium and large surveillance systems involve numerous screens (approximately 50 and above) that display the streams captured by numerous cameras. With the increase in the number of simultaneous video streams to be viewed, the work of surveillance operators becomes considerably challenging and fatiguing. Practically, after twenty minutes of continuous work, the attention of the operators is expected to degrade considerably. In general, the operators check for the absence or presence of objects (such as people and vehicles) in surveillance areas and ensure that the maximum capacity of a place remains intact, for example, by ensuring that no unauthorized people are present in restricted areas and no objects are present in unexpected places. The failures of such systems in alarming the authorities can be attributed to the limitations of manual processing. Generally, most traditional methods to obtain evidence depend heavily on the records of the security camera systems in or near the accident site. Practically, when an incident occurs in a vast space or considerable time has elapsed since its occurrence, it is difficult to find any valuable evidence pertaining to the perpetrators from the large number of surveillance videos, which hinders the resolution of the cases. Thus, to minimize the mental burden of the operators and enhance their attention span, it is desirable to develop an automated system that can reliably alert the operator of the presence of target objects (such as a human) or the occurrence of an anomalous event.

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

Literature Survey

The field of video surveillance is very wide. Active research is going on in subjects like automatic thread detection and alarming, large-scale video surveillance systems, face recognition and license plate recognition system, and human behavior analysis (Mabrouk & Zagrouba, 2018). Intelligent video surveillance (Singh & Kankanhalli, 2009) is of significant interest in industry applications because of the expanded request for the decrease of the time of analyzing large-scale video data. Relating to the terminology, Elliott (Elliott, 2007), has recently described an intelligent video system (termed IVS) as “any kind of video surveillance method that makes use of technology to automatically manipulate process and/or achieved actions, detection, alarming and stored video images without human intervention. The academics and industry researches being focused on developed the key technologies for design intelligent surveillance system that are powerful along with low cost computing hardware, which include object tracking (Khan & Gu, 2005; Avidan, 2007), pedestrian detection (Dalal & Triggs, 2005), gait analysis (Wang, 2006), vehicle recognition (Wang, 2007), privacy protection (Yu et al., 2008), face and iris recognition (Park & Jain, 2010), video summarization (Cong, Yuan & Luo, 2012) and crowd counting (Cong et al., 2009). Nguyen (Nguyen et al, 2015), describe the implementation and design an intelligent low-cost monitoring system using a Raspberry Pi, that uses the Motion Detection algorithm programmed in Python as a traditional programming environment. Additionally, the system utilizes the motion detection algorithm to considerably reduce storage usage and save expense costs. The motion detection algorithm is being executed on Raspberry Pi that enables live streaming cameras together with the motion detection. The real time video camera is able to be viewed from almost any web browser, even by mobile. (Sabri et al, 2018), presents a Real-Time intruder monitoring system based on a using a Raspberry Pi in order to deployed surveillance system that is effective in remote and scattered places such as universities. The system hardware is consisted from a Raspberry Pi, Long distance sensors, cameras, wireless module and alerting circuitry, while the detection algorithm is designed in python in order to presents a novel cost-effective solution having a good flexibility and improvement needed for monitoring pervasive remote locations. The results show that the system has high reliability of smooth working while using web application, in addition it cost-effective as a result it can be integrated as several units to catch and concisely monitor remote and scattered areas. Their system is also can be controlled by a remote user geographically or sparsely far from any networked workstation. The recognition results prove that the system is efficiently recognized intruder and making alert when detect intruder at distance between one to three meters from system camera, in which the recognition accuracy is between of 83% and 95% and the reliable warning alert had been in the range of 86-97%. Turchini (Turchini et al, 2018), propose an object tracking system that merged with their lately developed abnormality detection system which can present protection and intelligence for critical regions.

In last years, there are many studies focused on using the artificial intelligence for intelligence surveillance system. These techniques involve different approches such as SVM, ANN, and the last developed types based on deep learning techniques. However, deep neural network is computationally challenging and memory hungry therefore it is a difficulty to run these models in low computational systems such as single board computer (Verhelst & Moons, 2018). Several approaches have been utilized to deal with this problem. A lot of approaches have reduced the size of neural networks and even so keep the accuracy, such as MobileNet, while other approaches minimize the number of parameters or the size (Véstias, 2019).

System Concepts

We designed a robust surveillance system based on Faster RCNN and enhanced it by utilizing a saliency algorithm. The following equation can be used to determine the dimensions of the activation maps (O'Shea & Nash, 2015; Aggarwal, 2018):

$$(D_i + 2P_a — D_f)/ S_t + 1; \quad (1)$$

where D_i = Image dimension (input file)

- P_a = Padding
- D_f = Filter dimension
- S_t= Stride

For CNN, there exists a certain range of activation. In this work, we used a rectified linear unit or (ReLU) function. Currently, ReLU is one of the most commonly used activation functions in NNs. One of the most significant advantages of ReLU over other activation functions is the fact that it is unable to activate all the neurons at the same time. The ReLU function transforms all the negative inputs to 0, and no neuron is activated. Consequently, the function is computationally efficient since only a few neurons are activated over time. Practically, ReLU converges six times faster than the sigmoid and tanh activation functions. One of the disadvantages of ReLU is that it is saturated in the negative region, which means that the gradient in that region is 0. In this case, all the weights are not updated through backpropagation (BP), and a leaky ReLU can be used to overcome this limitation. In addition, ReLU functions are not centered at zero, which means that a random and thus longer path is often adopted for the functions to reach their optimal points. In addition, a pooling layer is placed between the convolution layers. The pooling layer fundamentally minimizes the amount of computation and number of parameters in the network and controls the overfitting by progressively minimizing the spatial size of the network. Generally, two operations are performed in this layer: maximum and average pooling. In this work, we utilize the max pooling technique; specifically, only the maximum value is obtained from the pool by using filters sliding throughout the input, and at each stride, the maximum parameter is extracted and the remaining values are not considered. This technique practically down samples the network. Compared with the convolution layer, this layer does not alter the network, and the depth dimension remains unchanged (Shang et al., 2016).

The output after performing max pooling can be determined as ()

$$(D_i— D_f)/ S_t + 1 \quad (2)$$

where

- D_i = Dimension of input (image) to pooling layer
- F = Filter dimension
- S_t = Stride

In the fully connected layer, all the neurons are fully connected to each activation from the prior layers. These activation values can be computed via matrix multiplication and then a bias offset, which is the last phase of the CNN network. The CNN is constructed using hidden layers and fully connected layers.

RCNN (Girshick et al., 2014) extracts a lot of parts from the presented image utilizing selective search, and after that investigate if any of these boxes has an object. At first, the model extracts all these regions, as well as for every region, CNN is utilized to extract specific features. At last, these features are later used to detect objects. However, RCNN turns into slow because of these multiple steps included in the process. Fast RCNN (Girshick, 2015), alternatively, passes the entire image towards the convolutional Net that yields regions of interest (rather than transferring the extracted areas from the image). Also, rather than making use of three different models (like as in RCNN), it utilizes a single model that extracts features out of the areas, classifies them to several classes, and proceeds the bounding boxes. Each one of these steps are carried out at the same time, hence making it execute quickly in contrast to RCNN. However, fast RCNN isn't fast enough in cases where applied on a big dataset since it also makes use of selective search for regions extraction. Faster RCNN (Ren et al., 2016) makes much progress than Fast RCNN. In Faster RCNN method, the “Search Selective” method was replaced by Region Proposal Network (known as RPN), which is a network to present regions (Brandenburg et al, 2018) and it faster than RCNN and Fast RCNN.

To improve the recognition process and reduce features, we intend to utilized saliency algorithm to enhance the image maps, as the algorithm maps the images to indicate the unique quality of each pixel.

Saliency map defined in computer vision as an image, where every pixels of the image have unique quality. The aim of a saliency map is to make image simpler and/or change it is representation (Daniilidis, Maragos & Paragios, 2010). Saliency detection approach is commonly used in the areas of cognition and target detection (Moosmann, Larlus, & Jurie, 2006; Zhaoyu, Pingping, & Changjiu, 2009; Kanan C. & Cottrell, 2010; Borji et al., 2019), object discovery (Frintrop, García, & Cremers, 2014), image segmentation (Kang et al., 2012; Yanulevskaya, Uijlings & Geusebroek, 2013), visual tracking (Klein et al., 2010; Borji et al., 2012; Stalder, Grabner & Gool, 2012), etc. The saliency represents a type of image segmentation technique. The saliency map aims to simplify and adjust the image representation to a more substantial form that is faster and easier to analyze. For example, when a pixel has a considerably large gray level or different color values in a color image, the quality of the pixel can be identified easily in the saliency map.

The saliency can be local or global (Borji & Itti, 2012). In the local domain, the contrast corresponds to the saliency of the image patches in the local neighborhoods. In contrast, in the global domain, to determine the saliency of an image patch, the contrast is computed with regard to the patch statistics along the whole image. In this work, we utilized the global saliency approach. The local patch is identical to its neighbors. However, the whole area (that is, the local domain and its surrounding) exhibits a global characteristic in the scene. If only the local saliency is considered, the areas may be reduced to a homogeneous area, which causes blank regions and impedes the realization of object-based focus (for example, a uniformly textured object could solely be salient at its edges). To overcome this limitation, in this work, the global saliency is adopted, which is established by guiding an operator through the saliency measure of data. Instead of using each pixel, we calculate the possibility of each patch $P(p_i)$ across the whole scene and determine its inverse to obtain the global saliency S_g as follows (Borji & Itti, 2012; Ming-Ming et al., 2015):

1.

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

and

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

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

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

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

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

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

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

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

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

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

System Architecture

The objective of this work is to design a smart, low cost surveillance system that can control surveillance units by using certain devices. The system is expected to monitor and control one or multiple surveillance cameras. The system should be able to detect certain aspects in a smart and automated manner for closed or open areas such as regions between or outside buildings and the surrounding areas. The proposed system is composed of a control unit, which controls all the processes, sensors (such as ultrasonic sensors to detect motion and distance), and a camera to output continuous video to identify the presence of humans and their activities. Figure 3 illustrates an example of the indoor surveillance system of buildings. The surveillance system includes cameras and ultrasonic sensors distributed to cover the main areas. The system first gathers data from the ultrasonic sensor model (HC-SR04) in real time. When the sensor detects movement in its range, it relays a signal to the camera located in that zone along with the computed distance. The camera recognizes the movements and assesses the threat. In case of any threat, the system alarms the security officer and informs him/her of the type of threat, for instance, if intruders (or even employees) are present in regions that are out of bounds. Figure 3 illustrates the detection operation.

Figure 3. Method to activate the camera on detecting motion

The system can be scheduled to perform multiple tasks multiple times by using the recognition process to intelligently analyze moving objects and activity. For example, the system can activate all the cameras during working times and recognize faces to identify the employers and any authorized guests. If the system detects any face that is not authorized or suspicious, the surveillance officers are alarmed, and the system tracks the suspects. In addition, the system can alert the officers if any employer enters an area at a prohibited time. The main concept is to perform a fast recognition system that can perform key recognition even under compact and low computational units such as Raspberry Pi. Such a system can be performed by utilizing a conventional neural network and computer vision technique known as the saliency algorithm.

Hardware Design

We employed two systems: a PC-based system, and a single board microcontroller system. The hardware specs of each system are described in table 1.

Table 1. Hardware specs of each system

The computer and single board computer are used as a processing unit that most processing operations done with it. Cameras is used to stream video from the desired area to process the images and recognize the presence of a person(s) within the camera's field of view. While the ultrasonic sensor is used to determine the distance.

For the single board microcontroller system, the following connections were made with the Raspberry Pi:

1- The camera is connected through the CSI camera port
2- The ultrasonic sensor is linked to Raspberry Pi through 4 pins, where VCC is linked to pin 2, GND is linked to Ground pin, echo is linked to GPIO 12, and trig is linked to GPIO 16.

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

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

Implementation of System Program

The proposed system program is designed in Python. For both systems, Python version 3.7 installed on the operation system (OS: Windows 10 for PC system and Raspbian for Raspberry Pi) is used. The program is designed to manage the overall processes, starting from gathering information from the camera(s) (streaming video) and sensor(s) (signals). This program, which uses CNN, is enhanced by using the saliency map algorithm. The algorithm analyzes the scenes continuously, and classifies any detected motion. Subsequently, the algorithm isolates the humans and detect threats based on human face recognition and human activity. The face recognition is performed to identify any suspicious person and alarm the observer. Moreover, the system analyzes human activity, and thus, the officers are alarmed in the event of any suspicious activity, such as gun handling. However, I utilized works done by (Rosebrock A., 2019) as a reference for design CNN model.

Methodology

As described previously, the system program is based on the CNN algorithm that is optimized using a computer vision technique known as the saliency algorithm. The proposed system involves the following steps:

Data Collection: Approximately 3450 images (human) and 10014 images for three categories of weapons (knife, small gun, large gun) are collected to perform the classification task by using transfer learning to reduce false positives. For human, A total of 3450 images have been collected and divided into two categories: training sets and testing sets, the images of human were captured at various pose, perspective and orientation. This can help the deep learning CNN to learn the required object in an efficient way. From the overall of 3450 images, 2761 images had been selected for training and 689 had been selected for testing. For weapons, 10014 images that classify three categories of weapons (knife, small gun, large gun). In which 8,011 were selected for training and 2,003 were selected for testing. To repurpose a pretrained model, we finetune our model by training certain layers and freezing other layers.

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

Optimization with Saliency Algorithm

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

Step 1: Image preprocessing: In this part, we first streamed live video as an FPS image and later convert it to grayscale.

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

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

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

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

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

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

Modeling with MobileNet and Faster RCNN

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

RPN that composed of two layers to look for the areas that can include objects in image (feature maps). The network utilizes the RoI pool layer to minimize and resize resource maps depending on proposals from that area. The maps make use of the new features of every area to select frame through three fully connected layers (FCL). MobileNet has been used as a CNN that took the layers to be learning functions; hence, the original feature extraction has several layers However, the first convolution stack structures obtained via transfer learning through the use of MobileNet. The method includes two steps forming the current surveillance detection: The first one is depending on determining ROI from images. All these ROI is considered as references in indicating several possible object sites which are created in the second step. Figure 5 shows the proposed model that consisted from Five convolutional layers and three FC layers. Faster RCNN mainly uses the last convolutional layer features in order to be classify and localize. After the two convolutional layers, then the outputs of the last three convolutional Layers (layers three, four and five) are utilized as input data to the 3 levels of pooling of the ROI and the related normalization levels. For every RPN anchor formatting a fully convolutional network, a degree is forecasted that makes it able to determine the probability of this anchor that has the element of interest. Moreover, the RPN offers the acceleration and measurement coefficients for every anchor which is a part of the peripheral regression mechanism, and thus enhancing the position of the object.

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

As an illustration, the architecture includes Two steps: At First, the RPN has presented a set of bounding boxes having a trusted rating related to potential human image. The second step is to defined analysis of those fully convolution architectures through the use of MobileNet to be as feature- extractor. after obtaining the output feature map coming from a pretrained model which is (MobileNet). As we used input image of resolution of 800x800 in x3 dimensions, then the output feature map should be 50x50x256 dimensions. Every point in 50x50 represented an anchor. Hens, we require to specify sizes and specific ratios for every anchor which are (128^2, 256^2, 512^2) for three sizes and (1:1, 1:2, 2:1) for three ratios, in the original image. After that, RPN is linked to a Conv layer using 3×3 filters, 1 padding, and 512 output channels. Then the output is linked to two 1×1convolutional layer for box-regression and classification (where the classification is used to verify if the box is an object or isn't). In such a case, each anchor will have 9 corresponding boxes from the original image, that means there are 50x50x9 = 22,500 boxes in the original image. We only select 256 of these 22,500 boxes to be a mini batch that has 128 backgrounds (neg) and 128 foregrounds (pos). Simultaneously, nonmaximal suppression is implemented to be sure there is zero overlapping for those proposed regions. When finish previous steps, then the RPN is finished. In second stage of RCNN, same as Fast RCNN, ROI pooling is utilized for these proposed areas (ROIs). After that, we flatten this layer by some of fully connected layers. The last step is a softmax function for linear regression and classification to fix the boxes' location. Figure 6 shows the FRCNN/ RPN structures

(a)

(b)

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

Results

This section describes the accuracy gain when the saliency method is applied to generate a saliency map, which is used as an input for the CNN to detect related objects. Figure 7 and figure 8 shows the saliency map generated from a live streamed video.

(a)

(b)

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. Saliency map results for humans with a gun (in real time streaming). (A) Streamed Images, (B) The output (Saliency Results) of streamed image (A).

As shown from, Figures 7 and Figures 8 the human body and gun extracted from a highly detailed image that involves many objects. The results indicate that the proposed method efficiently removes the foreground objects (humans and guns) from other objects in a scene with sufficient detail. The saliency map is passed to the RCNN to recognize the human and gun. The recognition process results are shown in Figure 9.

(a)

(b)

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.

As shown in Figure 9, the CNN can successfully recognize the human in real time. The recognition had an accuracy of 99.2%, and it can detect an object with no slowdown or missing object failure. In particular, the system tracked multiple humans in only 0.9 s. The mean relative error has been computed by normalizing with the given values through following formula:

$$\text{metric} = \text{mean}(|y_{pred} - y_{true}|/normalizer))$$

Which based on TensorFlow metric "tf.keras.metrics.MeanRelativeError", (Tensorflow, website, 2020). The mean relative errors of recognition are presented in Table 2

Table 2. Recognition results

By using transfer learning, we could reduce the false positives, as indicated by the metrics. Specifically, the validation mask loss is approximately 0.475, and the validation class loss is approximately 0.0383.

We tested the system to recognize and detect a person in real working operations. The system is first detect if there is any movement in the range of ultrasonic sensor, if there is a movement, the system then open camera and detect if there is a human in the field of view of camera, if yes, then turn light on and show screen and alarm to surveillance officer. Figures 10 show the detection process in room.

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).

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

Table 3. Recognition results for the systems-based PC and Raspberry Pi based System

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

Conclusions

In this work, we designed a smart surveillance system utilizing a low-cost computer unit and CNN to monitor certain aspects and alarm observers automatically. A single board computer, 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 the 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 enables fast and accurate feature extraction. The main results that have been achieved is described in 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 Raspberry Pi to perform a multiple task 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 the saliency algorithms with the RCNN. The main advantage of using a saliency algorithm is the reduction of the image details by the removal of undesired features from the image and retaining of 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 nonimportant objects along with a reduction in the image details can reduce the computational requirement 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 requirement and improvement in the recognition process both in terms of the speed and accuracy. The results show that the system can recognize humans and threats (such as human handling guns) in any situation with the recognition percentage ranging from 16% to 99.4%.

- Even with a low recognition percentage, the system can successfully detect and classify the human and gun 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.
- The system achieved real time detection of humans in less than 1s when using both Raspberry Pi and PC models.

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

- Using other architectures for the CNN, YOLO, SSD, Mask RCNN, etc.
- Extend the system to detect other type of objects or even behaviors (like theft or violence).
- Complement the system with another PC with more resources capable of performing online learning to re-train the system with new images.

Acknowledgments

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

References

Achanta R, Shaji A, Smith K, Lucchi A, Fua P and Susstrunk S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell. 34(11):2274-82. DOI: 10.1109/TPAMI.2012.120.

Aggarwal CC. 2018. Neural Networks and Deep Learning: A Textbook. Springer International Publishing. DOI: 10.1007/978-3-319-94463-0

Aldasouqi I and Hassan M. 2010. Human face detection system using HSV. CSECS '10: Proceedings of the 9th WSEAS international conference on Circuits, systems, electronics, control & signal processing. pp.13-16.

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Computer Vision and Pattern Recognition. arXiv:1704.04861.

Avidan, S. 2007. Ensemble tracking, IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 2, pp. 261– 271.

Borji A, Cheng MM, Hou Q, Jiang H, and Li J. 2019. Salient object detection: A survey. Computational Visual Media. Computational Visual Media, Springer. 5:117–150.

Borji A, Frintrop S, Sihite DN, and Itti L. 2012. Adaptive object tracking by learning background context. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 23–30.

Borji A and Itti L. 2012. Exploiting local and global patch rarities for saliency detection. 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA. DOI: 10.1109/CVPR.2012.6247711.

Cong, Y, Gong, H, Zhu S and Tang Y.: Flow mosaicking: Real-time pedestrian counting without scene-specific learning, in Proc. CVPR, 2009, pp. 1093–1100.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Turchini F, Seidenar L, Uricchio T and Bimbo A. 2018. Deep Learning-Based Surveillance System for Open Critical Areas. Inventions. 3(4):69. DOI:10.3390/inventions3040069.

Verhelst M and Moons B. 2018. Embedded Deep Neural Network Processing: Algorithmic and Processor Techniques Bring Deep Learning to IoT and Edge Devices. IEEE Solid-State Circuits Magazine 9(4):55-65. DOI: 10.1109/MSSC.2017.2745818.

Véstias MP. 2019. A Survey of Convolutional Neural Networks on Edge with Reconfigurable Computing. Algorithms, MDPI. 12(8), 154, pp.1-24. DOI: 10.3390/a12080154.

Wang L. 2006. Abnormal walking gait analysis using silhouette-masked flow histograms, in Proc. ICPR, vol. 3, pp. 473–476.

Wang S and Lee H. 2007. A cascade framework for a real-time statistical plate recognition system, IEEE Trans. Inf. Forensics Security, vol. 2, no. 2, pp. 267–282.

Wang W, Shen J, Xie J, Cheng MM, Ling H, Borji A. 2019. Revisiting Video Saliency Prediction in the Deep Learning Era. IEEE Transactions on Pattern Analysis and Machine Intelligence. DOI: 10.1109/TPAMI.2019.2924417.

Yanulevskaya V, Uijlings JRR and Geusebroek JM. 2013. Salient object detection: from pixels to segments. Image and Vision Computing 31(1):31–42. DOI: 10.1016/j.imavis.2012.09.009.

Yu X, Chinomi K, Koshimizu T, Nitta N, Ito Y and Babaguchi N. 2008. Privacy protecting visual processing for secure video surveillance, in Proc. ICIP, pp. 1672–1675.

Zhang J, Malmberg F, Sclaroff S. 2019. Visual Saliency: From Pixel-Level to Object-Level Analysis. Springer International Publishing. DOI: 10.1007/978-3-030-04831-0.

Zhaoyu P, Pingping L and Changjiu L. 2009. Automatic Detection of Salient Object based on Multi-features 2008 Second International Symposium on Intelligent Information Technology Application, IEEE. DOI: 10.1109/IITA.2008.466.

Table 1 (on next page)

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$	

Table 2 (on next page)

Recognition results

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

Table 3(on next page)

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

Number of persons in view	Detection Time for System Based on PC (s)	Accuracy (%)	Detection Time for System Based on Raspberry Pi (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.

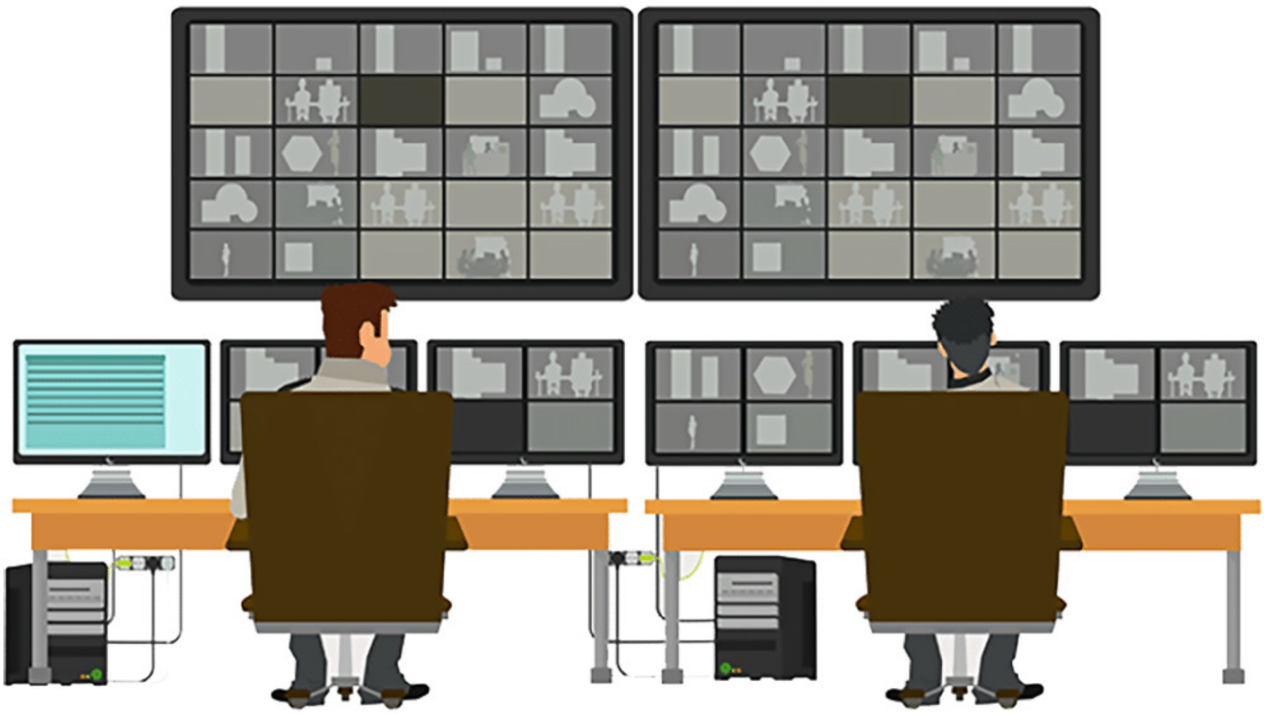

Figure 2

Illustration of global and local saliency of an image patch.

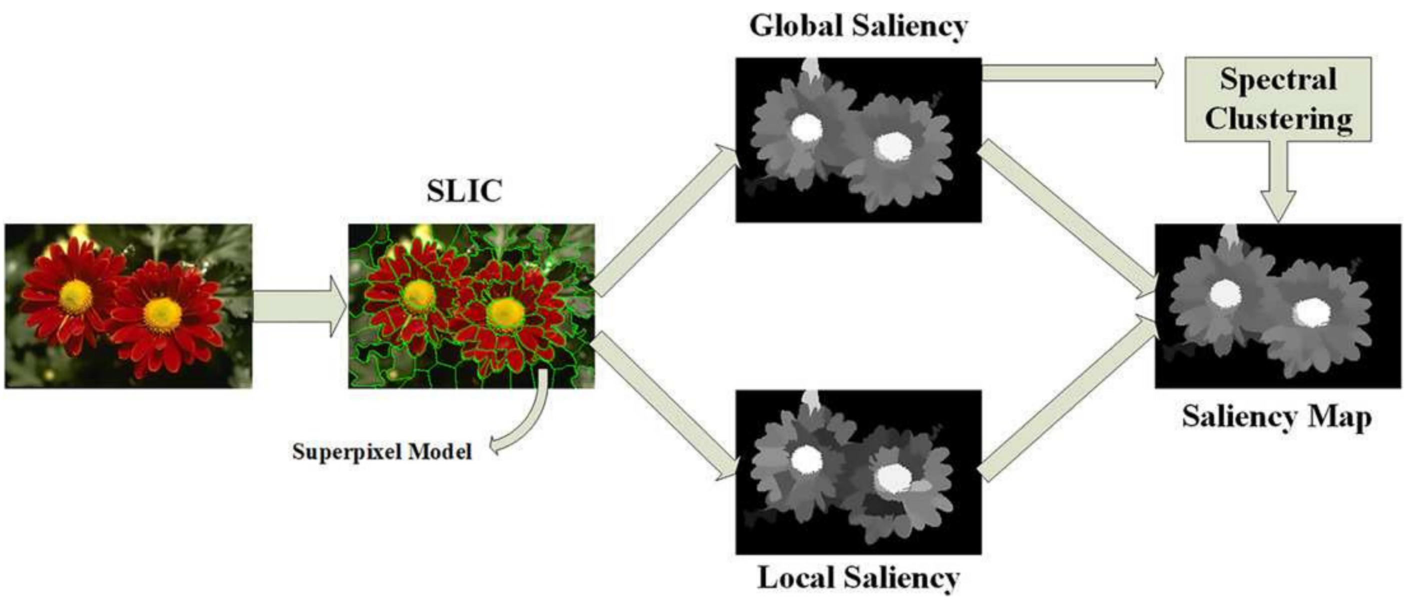

Figure 3

Method to activate the camera on detecting motion.

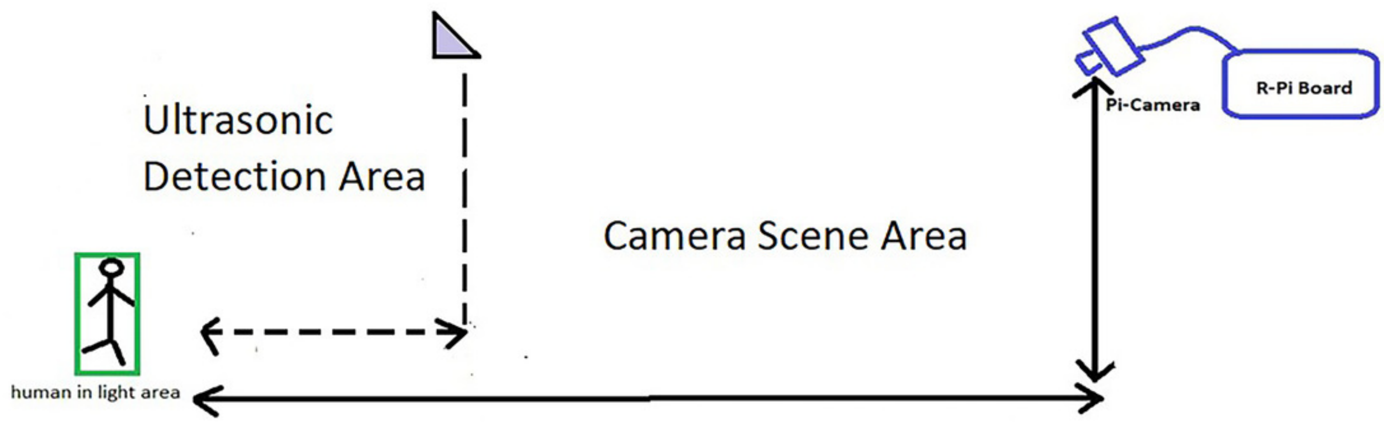

Figure 4

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

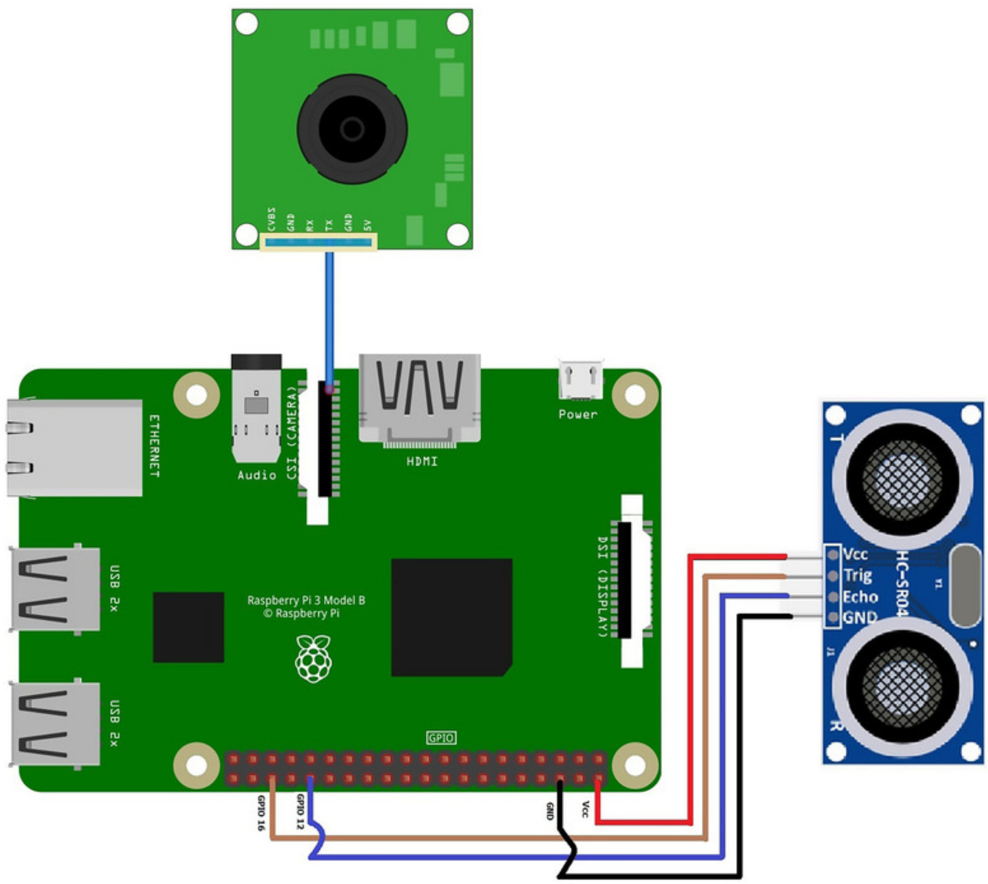

Figure 5

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

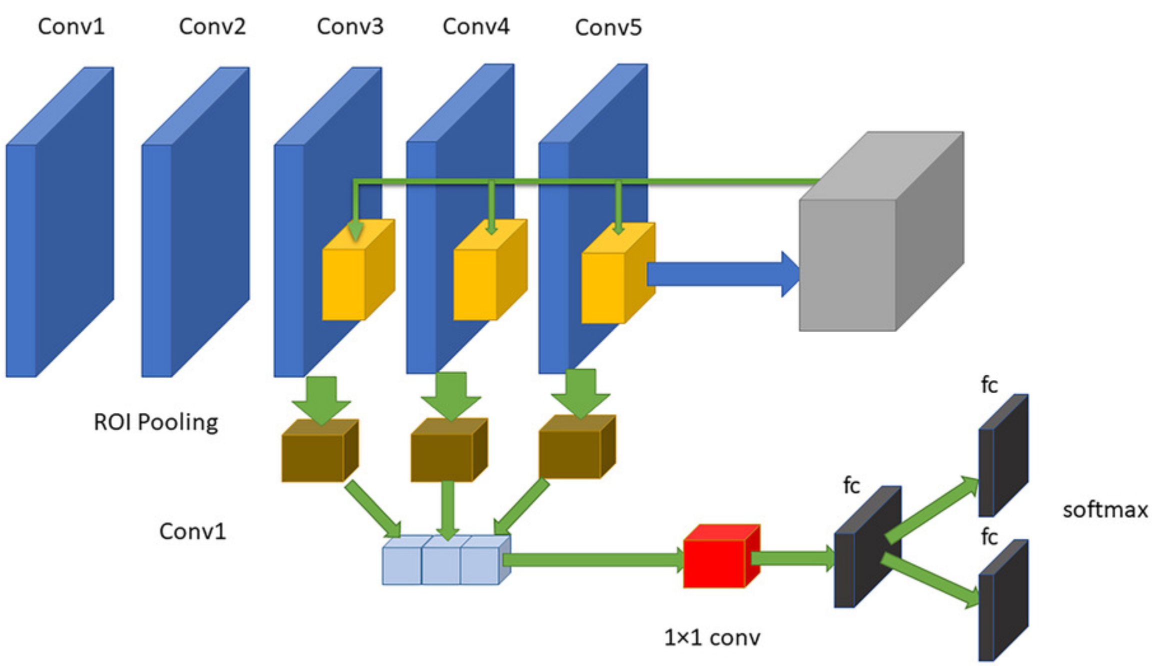

Figure 6

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

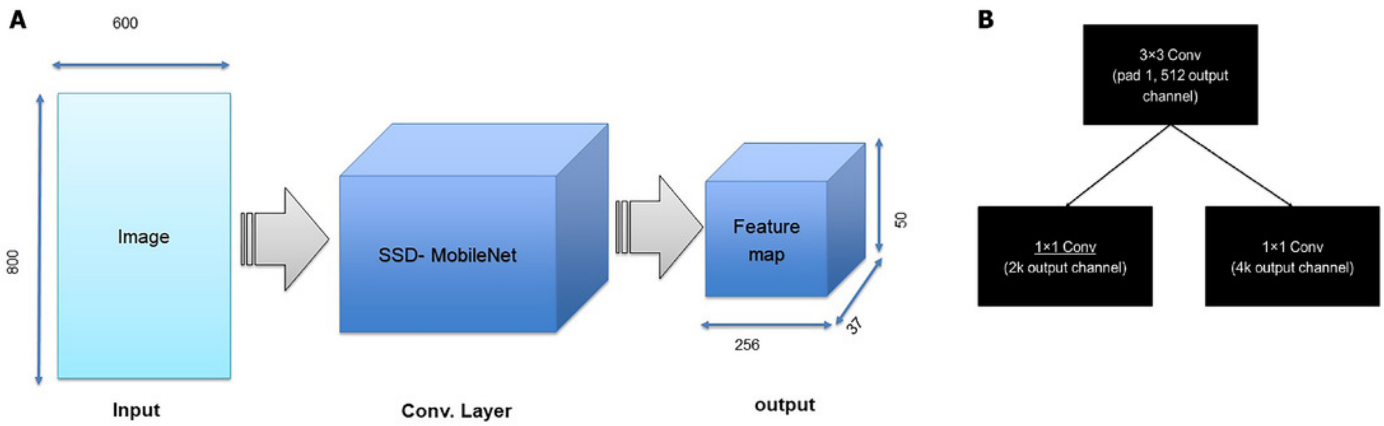

Figure 7

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

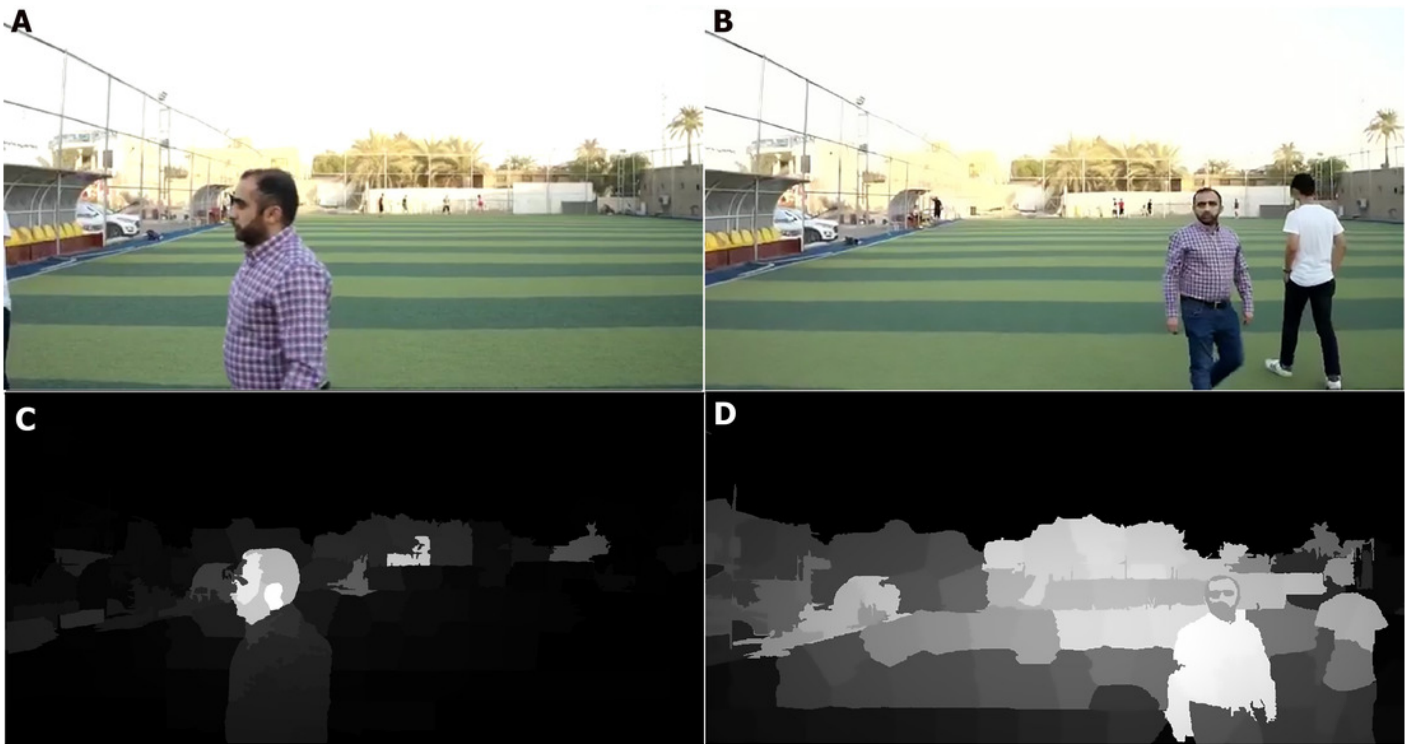

Figure 8

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.


A
B
C
D

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