Journal of Big Data

RESEARCH

Open Access

A hybrid framework combining background subtraction and deep neural networks for rapid person detection

Chulyeon Kim[1], Jiyoung Lee[2], Taekjin Han[1] and Young-Min Kim[1]*

*Correspondence: yngmnkim@hanyang.ac.kr
[1] Graduate School of Technology & Innovation Management, Hanyang University, Seoul, Republic of Korea
Full list of author information is available at the end of the article

Abstract

Currently, the number of surveillance cameras is rapidly increasing responding to security issues. But constructing an intelligent detection system is not easy because it needs high computing performance. This study aims to construct a real-world video surveillance system that can effectively detect moving person using limited resources. To this end, we propose a simple framework to detect and recognize moving objects using outdoor CCTV video footages by combining background subtraction and Convolutional Neural Networks (CNNs). A background subtraction algorithm is first applied to each video frame to find the regions of interest (ROIs). A CNN classification is then carried out to classify the obtained ROIs into one of the predefined classes. Our approach much reduces the computation complexity in comparison to other object detection algorithms. For the experiments, new datasets are constructed by filming alleys and playgrounds, places where crimes are likely to occur. Different image sizes and experimental settings are tested to construct the best classifier for detecting people. The best classification accuracy of 0.85 was obtained for a test set from the same camera with training set and 0.82 with different cameras.

Keywords: Convolutional Neural Network, Background subtraction, Object detection, CCTV

Introduction

With the rapid increase in violent crimes, an effective surveillance system has become a necessity. Early detection of crime using video surveillance is important but the control center cannot keep track of extended regions owing to the lack of manpower. It is not practically possible to manually track all locations and events considering the number of CCTV cameras managed by one person. For example, in Korea, one CCTV operator manages more than 100 cameras on an average at the local government's control center [1]. At the end of 2016, the total number of cameras connected to the national CCTV control centers in Korea was about 174,400 while the number of operators was only 3600 [2]. Despite this shortage, real-time surveillance significantly helps in arresting criminals. In Korea, the number of criminals arrested with the usage of the real-time monitoring of CCTV cameras has increased by 12 times over the past 3 years [3]. Hence, automatic video analytics is essential for reducing the workload of operators. However, existing automatic event detection capabilities have limitations such as a high possibility of false positives.

A (semi-)automated surveillance system should have the capability of rapid recognition of moving objects, even before it can detect the dangerous behaviour of objects. The current methods of detecting moving objects are mostly based on the background subtraction so far because of their fast processing speed [4, 5]. However, thanks to the recent advances in object detection methods such as YOLO [6] and SSD [7], which work well in real-time, we may attempt to use these techniques for the the detection of moving objects.

Typically, real-world videos are difficult to deal with due to external factors such as noise, shadows, poor illumination, low resolution, etc. Most object detection techniques do not account for all these problems and the state-of-the-art methods usually use deep Convolutional Neural Networks (CNNs), which are computationally expensive [8]. While a detailed analysis based on CNNs requires powerful machines, a typical surveillance system has limitations in terms of computational power and memory. On the other hand, the number of CCTV cameras around the world is rapidly increasing. The top two cities with the largest surveillance camera networks are Beijing and London [9, 10]. In London, there are more than 500,000 cameras [11]. It means that there are on average 15,200 cameras to manage in a district in London. Because a commonly used server that offers video analytics covers about 24 channels, we need 630 servers to manage the cameras for a district. In these circumstances, it becomes essential to assess the trade-off between detection performance and computational cost. A suitable surveillance system should not only detect well suspicious objects properly; it should be able to do with limited resources.

Toward this end, we propose a simple framework to recognize moving objects. Our proposition aims at reducing both training and inference costs as well as manual annotation cost while retaining an acceptable detection performance, comparable to other object detection methods in the recent past. The framework consists of two steps that are sequentially applied to each video frame. In the first step, we find regions of interest (ROIs) in a video frame using simple background subtraction. In the second step, we use a CNN to classify the detected ROIs based on a set of predefined criteria. This simple combination operates well in real-time and delivers a detection performance comparable to other modern object detection techniques. Our framework also offers the benefit of reusing background subtraction. In fact, the use of change detection techniques such as background subtraction is indispensable in video surveillance applications to detect anomalies such as abandoned objects or abnormal movements in video streams [12]. This means that the real processing time dedicated only to object detection in our framework is the CNN inference time.

The organization of the paper is as follows. The we introduce the motivation of our research in the next section. "Related works" section presents the related works in four different aspects, video surveillance, background subtraction, object detection, and pedestrian detection. "Methods" section presents our framework consisting of two steps including a detailed description of the extracted images from first step. We also introduce the surveillance video data we filmed for the experiments. "Results" section reports the experimental result and the last section presents the conclusion and discussion.

Motivation

Current object detection techniques are based on deep learning and the latest algorithms such as have significantly enhanced the detection speed. However the training step still requires considerable computational resources and through a computational downgrade does reduce expenses; it also hampers the performance as compared to the original version. CNN-type models suffer from the over-fitting that becomes more problematic when the models are applied to CCTV data in which the quality, color, angle, noise, etc. much vary as different cameras are used. Therefore, for the video surveillance, it might be necessary to train a classifier for each camera type or even for each camera. In such a case, reducing cost in training as well as in manual annotation would be an important factor when constructing an object detection system.

A weakness of modern object detection methods is that they often miss detecting small objects. Figure 1 shows a result of a state-of-the-art object detection technique, YOLOv2(YOLO below) with a set of pre-trained weights. An image frame with a size of 1280 × 720 pixels extracted from a real-world CCTV camera is tested. In this example, the model can detect well the cars but not the people.

Figure 2 shows the people left undetected, classified into different groups. The individuals in (a) are relatively small and are hidden by a car. Persons in (b) are also small and not well separated. The size of the individuals in (c) is not small but they are not distinguishable from the dark background. As seen in this example, there are many such cases encountered in real-world CCTV footage. The objects to be identified are often very small and partially hidden. Dark objects at night are usually indistinguishable from the background.

In a real-world video surveillance application, an appropriate trade-off between the processing speed and system cost is required. As surveillance systems only focus on suspicious objects, we do not need to detect all the objects in video streams, unlike usual object detection. The application would work more efficiently if the focus was just trained on moving objects. Traditional moving object detection methods in videosurveillance basically use background subtraction. Unlike in case of object detection, recognizing the object class is not important in this step. With a controllable environment

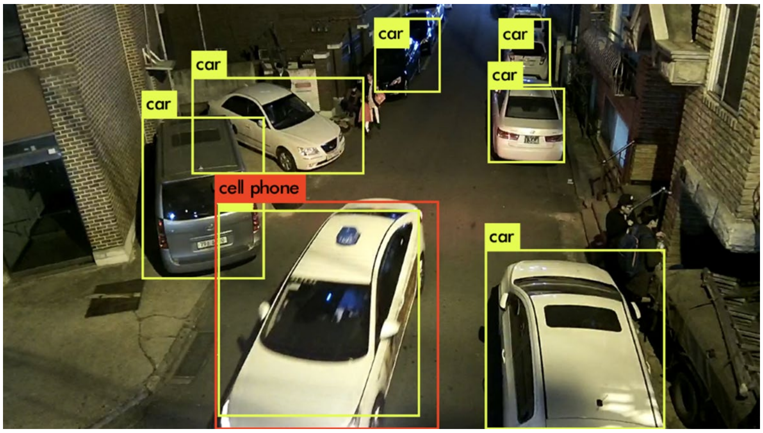

Fig. 1 Object detection with a pretrained YOLO model

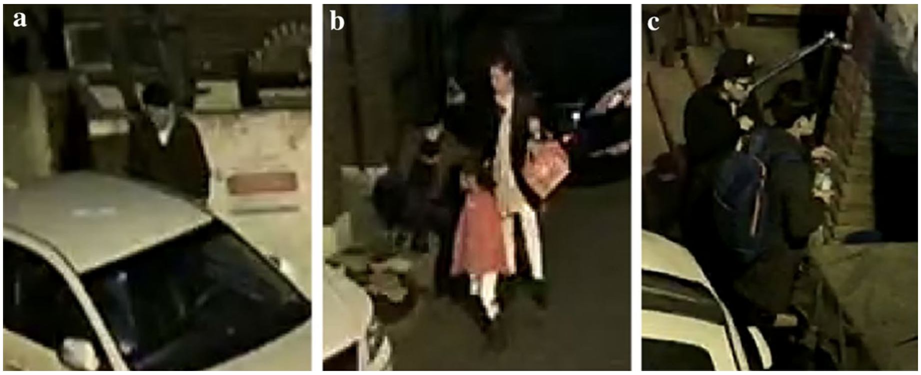

Fig. 2 Persons not detected by YOLO. **a** The individuals are relatively small and are hidden by a car. **b** The individuals are small and not well separated. **c** The individuals are not distinguishable from the dark background

such as an indoor scene, classifying objects may not be necessary. But in outdoor environments, moving object types vary and false detection increases because of noises. Therefore, to identify a meaningful event related to a moving object in noisy environment, recognizing well the class of the object appropriately is critical.

This study largely leverages two different technologies to devise an efficient object detection method: background subtraction and object recognition. The most time-consuming component of current object detection processes is the repeated testing of hundreds of region proposals by scanning the whole image. By using background subtraction to find ROIs, we can significantly reduce the number of proposals that are then used as the input for a simple CNN classifier for object recognition. In our framework, recognition accuracy would depend on subtraction quality to a large extent. However, as our primary objective is to verify the effectiveness of combining subtraction and CNN classification, we do not consider optimizing a background subtraction algorithm. Instead, showing that even a low quality of subtraction result leads to a rough-and-ready classifier, we will demonstrate the utility of our framework.

In our approach, a widely-used method, the Gaussian Mixture Model (GMM), is selected for the background subtraction. Then, a post-processing such as shadow removal and morphological transformation is conducted to extract the object area. In fact, the algorithm can be replaced by any other moving object detection algorithm because the next classification step is distinct from this first step.

Related works

Video surveillance

With the rapid increase of the installation of advanced cameras and CCTV infrastructures, intelligent video surveillance systems to automatically monitor specific environments have emerged. The primary objective of a video surveillance system is to provide an automatic interpretation of a scene by analysing motions and interactions of objects for preventing future undesirable incidents. Video surveillance systems have been adopted mainly for security and safety purposes. Especially, abnormal behavior detection is receiving attention in research [13]. For example [14], developed a monitoring system for the elderly in home environments focusing on fall detection. This system, shaped like an ellipse, is custom-fitted to the subject's body and the head position is tracked to identify any change in posture. The automated surveillance system proposed by [15] detects robbery and burglary through posture and motion analysis using low-cost hardware, i.e., a consumer camera. The system performs by converting 2-D results into the 3-D space [16] introduced abnormal crowd motion detection methods in an uncontrolled outdoor environment. The orientation, position, and size of the cluster created by the crowd are used to predict the future behavior of the crowd. If the subsequent motion of the cluster is not matching the prediction, it indicates the likelihood of abnormal events. More recently [17], developed a real-time suspicious behavior detection system for shopping malls. This system captures suspicious circumstances such as a loitering customer or an unattended cash desk. Sidhu and Sharad [18] proposed a smart surveillance system for detecting interpersonal crime such as unauthorized access, violence, and harassment in public places and transport by tracking the speed of positional change of the human subject.

In general, a video surveillance system is composed of object detection, object recognition, object tracking, and behavior analysis [19]. In this respect, object detection is the step for identifying the region of interest (ROI), such as moving objects, including human and vehicles. After the objects are detected, they are classified into predefined categories where they belong in object recognition step. For example, moving objects are recognized and classified into 'human', 'car', or 'bicycle' according to their features such as shape, pattern, color and so on. Then, the objects are tracked and their behavior is analyzed to capture potential suspicious events. Object detection and recognition are very critical to the performance of the overall surveillance system because the subsequent steps are highly dependent on their results.

Despite the wide attentions it has received, video surveillance is still faced with several challenges. Objects in the video are frequently occluded that causes difficulties. Significant noise occurs owing to illumination changes. These difficulties significantly increase in outdoor environments [19]. The quality of the video recorded in outdoors is usually very poor because objects are often detected at a large distance from camera and the video are more sensitive to illumination changes. Therefore, various human detection algorithms built in controlled conditions exhibit poor performance when applied to real-world scenarios [20]. For example, most existing methods are designed for daytime surveillance because their performance relies heavily on light condition [21], even though a large proportion of abnormal incidents (e.g., crime) occur during the night-time. Therefore, developing effective methods to cope with these difficulties is essential for developing advanced video surveillance systems that can be used for outdoor applications.

Background subtraction

Background subtraction is one of the crucial techniques of computer vision systems. It was first used to detect moving objects in video streams. The main purpose of the various background subtraction algorithms is to distinguish a moving object, which is called foreground, from the background of the sequence within the video stream without having any advance information about the object [22]. Background subtraction has been widely researched for video analysis, especially for video-surveillance application since the 1990s, because it can detect the moving objects such as humans, vehicles, and animals from the background before performing complicated detection processes such as invasion, detection, tracing, and people counting [23].

Generally, the background subtraction algorithms comprise the following three stages [24]: (1) Background initialization: the goal of this stage is to build a background model by using a specific number of frames and it can be designed in various ways (statistical, fuzzy, neural network, etc.). (2) Foreground detection: in this stage, the present frame and the background model are compared. It is connected to compute the foreground (e.g. detected object) of the scene by this subtraction. (3) Background maintenance: during the detection process, images for updating the background model, which was trained in the initialization step with respect to the learning rate, are additionally analyzed. For instance, a moving object that moves for a long time should be absorbed into the background.

Background subtraction algorithms can be classified on the basis of the method of developing a background model. The algorithms are classified into basic modeling, statistical modeling, modeling based on clustering, fuzzy modeling, modeling that applies neural and neuro-fuzzy methods, etc.

In basic modeling, once the model is once computed, the difference between the background image and the present frame is adjusted based on a threshold. When the computed value is bigger than the threshold value, pixels of the present frame are segmented into the foreground. Basic modeling uses a static image or the previous frame as a background. Some authors suggested building a background model by using the arithmetic mean of the continuous images [25], median [26] or based on the result of the histogram analysis over a period [27]. Basic modeling is relatively simple and useful for making background models but it suffers from limitations such as failure of foreground separation with the introduction or removal of objects or when the object suddenly stops.

In the statistical modeling, the distribution of each pixel color appears as the sum of a Gaussian distribution that is defined in the given color spaces. The most common background subtraction algorithm is based on the parametric stochastic background model suggested by [28], which was later improved by [29]. In some studies, a single Gaussian, a mixture of Gaussians, or a Kernel Density Estimation [30] during background modeling were also used for pixel color distribution.

Gaussian Mixture Models (GMMs) demonstrated a good performance in outdoor scene analysis, and it became the most widely used technique when dealing with moving backgrounds. As one of the most popular background subtraction algorithms, the model shows displays ability to process changes even in low lighting. However, the algorithm does not provide appropriate results in case of radical changes due to camera jittering, lighting changes, and appearance of shadows or movements in the background. Also, when the background model is established by a video frame that is characterized by a lot of noise in the learning step, this method can be inefficient. To solve these problems, many authors researched various methods to improve background subtraction method based on GMMs. To improve the adaptability of the system to illumination changes [31], modified the update equation [32], explored 3D multi-variable Gaussian distribution, and [33] suggested a method that can automatically calculate the number of proper Gaussian distributions of each pixel instead of setting a constant. A recent method [34] proposed a new framework integrating hysteresis thresholding and motion compensation. They used two background modeling, GMM and a texture modeling to reduce false positive cases.

In the cluster-based background modeling, each pixel of the frame can be expressed according to the time by clustering. The entered pixels are classified by whether the congruous cluster is considered as part of the background when it is compared to the relevant cluster group. As a clustering approach, a method of using K-mean algorithm or Codebook was suggested. A background modeling technique based on the fuzzy concept that the application's boundaries can vary considerably depending on the contexts or conditions has also been suggested. Zhang and Xu [35] performed background subtraction by using a similarity criterion, which has the characteristics of color and texture of the input image. El Baf et al. [36] obtained satisfactory results by using the Choquet Integral, which has the characteristics of color, edge and texture.

In neural network background modeling, the background is represented by the average of the weights of properly trained neural networks for N numbers of clean frames. The networks are trained how to classify each pixel into background or foreground [37]. The neural network decides if the pixel belongs to the foreground or background [38]. Culibrk et al. [37] suggested a segmentation approach by using an adaptive form of PNNs (Probabilistic Neural Networks). Maddalena and Petrosino [39] used a SOM (Self-Organizing Map) network to perform background subtraction. They improved upon their previous research by adding a fuzzy function at background learning stage [40].

Object detection

Object detection is one of the fields that has shown significant progress among the applications that actively use deep learning. The difference from object recognition is that while object recognition aims to classify each image into one of the pre-defined classes, object detection aims to detect objects in each image by localizing them.

Most of the recent object detection methods are based on deep learning adopting CNNs [41–43]. Since the advent of the R-CNN [44], combining region proposal and CNN classification became a the preferred framework for object detection. Instead of using handcrafted features such as HoG [45], R-CNN uses CNN features for a more effective representation. In R-CNN, thousands of bounding boxes called region proposals are created through selective search, and the proposals become the inputs for CNN classification. Fast R-CNN [46] complements R-CNN in terms of both efficiency and accuracy. First, the proposals share the weights of the forward pass in CNN via region of interest (ROI) pooling technique to reduce computation. In addition, convolutional features, classifier, and bounding box regressor are connected in a single network to speed up the system. However, there is still an inefficient part, which is the selective search for region proposals. Faster R-CNN [47] improved the formers by integrating region proposal process with detection networks. Instead of selective search, convolutional layers and region proposal network are used to create the proposals. As a result, the detection process became faster than Fast R-CNN. However, the detection speed is still far from real-time, about 5 fps on a GPU.

YOLO [6, 48], a state-of-the-art detection method, surpassed the aforementioned methods. It is also based on CNN but uses a totally different framework. It works by dividing an image into grid cells and predicting the coordinates of bounding boxes and the probabilities of each cell. The individual box confidence is calculated by aggregating the probabilities. This framework significantly speeds up the processing time so that it can process images at 40–90 fps on a GPU and a tiny version can do it at more than 200 fps.

Pedestrian detection

Pedestrian (human) detection is one of the main tasks of video surveillance. Pedestrian detection methods can generally be grouped into two categories: hand-crafted feature-based methods and learning-based methods. Histograms of oriented gradients (HoG) is a method representative of the former category, in which uses local object appearance and shapes for detection [49]. Currently, HoG is mainly used as a baseline for developing many extended algorithms. For example, [50] used a combination of LBP and HoG features to cope with the occlusion problem in pedestrian detection and achieved a 91.3 % detection rate with the INRIA dataset [49]. Felzenszwalb et al. [45] proposed a part-based model to improve both detection efficiency and accuracy of algorithms. HoG with additional LUV color channels (HoG + LUV) is also widely used [51, 52]. Haar-like features are also used for pedestrian detection. For example, [53] introduced a simple and efficient informed-Haar detector designed exclusively for humans standing upright. The test results obtained from the INRIA and Caltech dataset [54] showed a 14.43 and 34.6% of miss rate, respectively. Note that the subjects in INRIA are always standing upright and annotated in high-quality. On the other hand, the Caltech dataset is usually considered as predominant benchmark including various and challenging images [55]. These methods are followed by a classifier such as SVM [56] or a boosted classifiers [57]. There are also other frameworks using the other hand-crafted features such as HSG-HIK [58]. In the recent past, the learning-based methods have received great attention. These methods learn features from the pixels in an image and the state-of-the-art detection methods are based on deep CNNs [59]. Sermanet et al. [60] applied unsupervised convolutional sparse auto-encoders for pre-training features and used end-to-end supervised training for classification. They used the INRIA dataset and ConvNet with multi-stage features to obtain an average error rate of 10.55%. Ouyang and Wang [61] designed a unified CNN-based deep model, which uses a learning process to enable interactions between feature extraction, part deformation, occlusion, and classification components. To cope with the complex variations in pedestrian appearances, TA-CNN, introduced by [62], optimizes pedestrian classification with auxiliary semantic tasks, including pedestrian and scene attributes, and reduces miss rates in the Caltech dataset and ETH datasets [63]. The DeepParts [64] method employs part detectors for solving the occlusion problem, thereby reducing the miss rate by 11.89% in the Caltech dataset. R-CNN (Regions with CNN features) has been also used for person detection. It reaches 53.9% accuracy of human classification accuracy in the VOC2011 dataset [65], while other region-based methods such as 'Regions and Parts' [66] deliver a slightly lower accuracy. MixedPeds algorithm [67] is a quite original approach. The algorithm produces a mixed reality dataset combining real background and synthetically generated human-agents. Using Faster R-CNN, their approach improved the detection precision over previous detectors. Though R-CNN improves accuracy, it requires significant memory and time, thus reducing detection speed.

Methods

Our object detection framework benefits from the background subtraction procedure. For each video frame, a GMM is first applied to extract moving object areas and then a CNN is used to classify these ROIs. Figure 3 summarizes the entire architecture of our approach. This section describes our approach in detail including the characteristics of the extracted ROIs. We also introduce the video surveillance data we filmed for the experiments.

Video surveillance data

To verify the effectiveness of our framework, a dataset that reflects well the real surveillance environment is required. We constructed new data to assess how our object detection framework is appropriate for outdoor surveillance scenarios. The data is introduced briefly in this subsection.

We filmed some scenarios using the existing CCTV cameras with permission and cooperation from authorities in city A in Korea to obtain data similar to real-life surveillance data. We acquired the data from CCTV cameras in places characterized by frequent incidents of crime. The dataset was acquired from three places: a playground, an alley, and a lonely walkway/road. The existing CCTV cameras with resolution 1280 × 720 pixels were placed at a height significantly above the average human height.

The scenarios typically involved loitering, which usually precedes a crime. The recording was conducted in early spring 2017 and it was a fine day, though a bit chilly. Seven men and three women participated in the shooting of the scenario, dressed according to the parts they were playing. They were then asked to loiter around the three identified places: a playground, an alley, and a lonely walkway/road changing their clothes according to the location. The number of actual objects was observed to be much more than 10 because arbitrary pedestrians passing through the area during the shooting were also pulled into the experiments. Note that most arbitrary pedestrians were unidentifiable due to the resolution.

Figure 4 shows raw video frame examples obtained from eight different CCTV cameras. Figure 4a–e show the images obtained from five cameras used for both training and test. The frames correspond to a place for outdoor fitness, a playground, and the

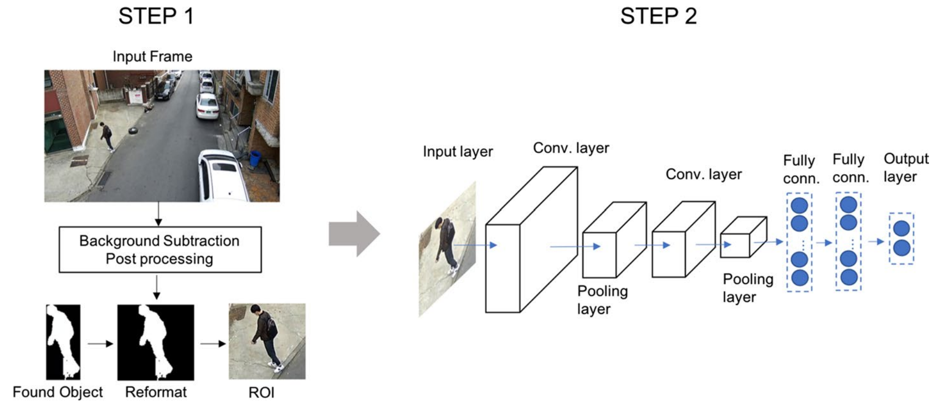

Fig. 3 Entire architecture of our approach

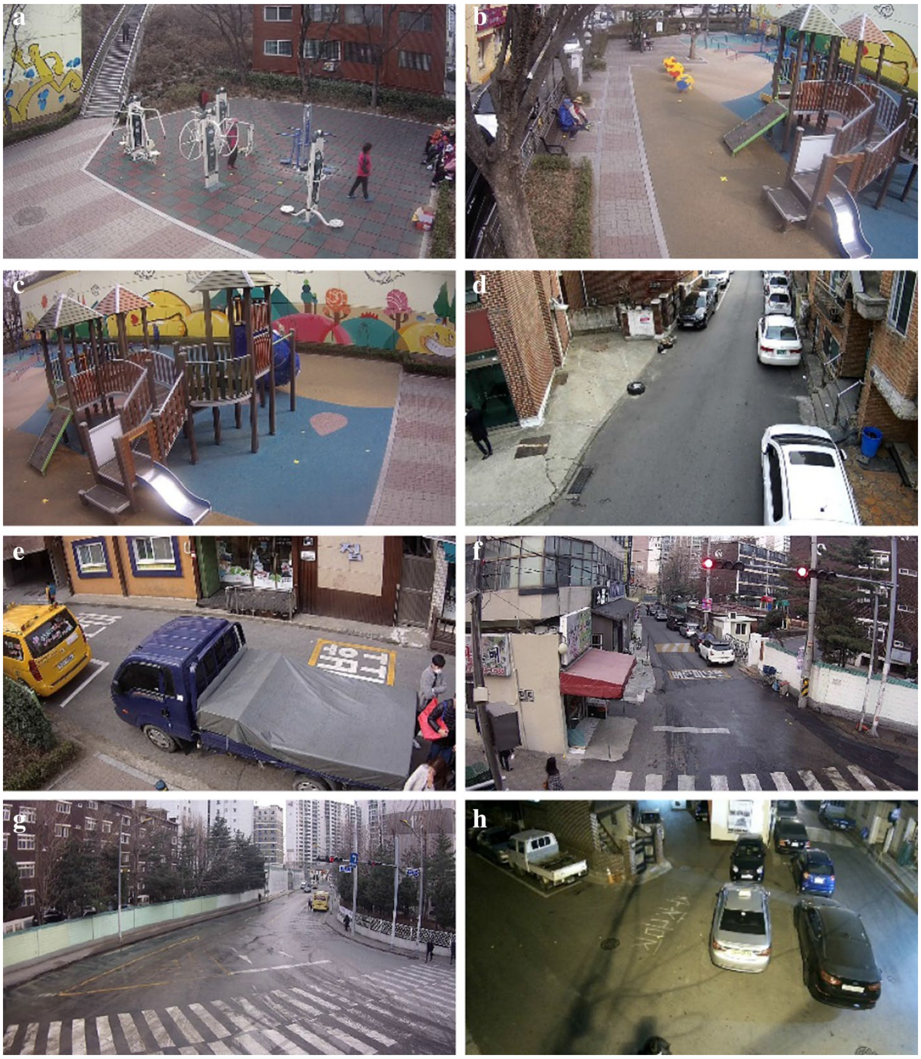

Fig. 4 Video frames from CCTV cameras in playground and alley. **a** Place for outdoor fitness. **b** Playground. **c** Playground. **d** Residential alley. **e** Residential alley. **f** Intersection near the residential area. **g** Main street intersection. **h** Residential alley intersection captured at night

same playground with different view, a residential alley and another alley, respectively. Figure 4f–h show images obtained from three cameras used only for test. The frames correspond to an intersection near the residential area, a main street intersection, and a residential alley intersection captured at night, respectively. To test the generalizability of the trained model, the different places (f, g, h) selected were completely different from the places selected for the training data.

Step 1: Extraction of moving objects with background subtraction

In Step 1 of our framework, we extract the moving objects from each video frame with background subtraction. Despite several drawbacks of GMM, such as noises from camera jitter, shadows, lightning etc., the algorithm is one of the most widely used methods because of its easy accessibility and fast processing time. We choose GMM as a primary algorithm to show that our approach works well even with a simple background subtraction method. It can be replaced by any another algorithm.

After we apply the GMM algorithm to each video frame, several post-processing procedures are conducted to extract objects. Because the algorithm cannot handle shadows, we first try to eliminate shadows based on their positional and color characteristics. Then we perform morphological filtering such as closing and opening to eliminate noises. Finally, we obtain the desired object areas that are represented by rectangles. Too small objects that are less than 5×5 pixels are eliminated because it is highly probable that could just be noise. We also use object tracking to take the objects which exists for a period.

With all the operations, step 1 processes the images of 1280×720 pixels at a rate higher than 30 fps on a CPU. Thus, we can regard this step as a real-time procedure. The extracted objects become ROIs that are used as input images for step 2, CNN classification. To this end, we need to annotate a set of images to make a training set. An ROI example is given in Fig. 3.

Annotation of ROIs extracted in Step 1

Because we use the result of background subtraction as input images for classification, we need to carry out step 1 before manually annotating images. The object areas identified in step 1 are edited as square images to obtain a regular input form for a standard CNN classifier. As described in "Video surveillance data" section, the image sources are eight CCTV cameras installed in a playground or an alley.

Figure 5 shows some examples of the extracted ROIs. Figure 5a, b correspond to an individual but not the whole body. In many cases, even when the whole body can be seen in the image, only a part of the body can be detected because of a similar background color or stationary body part. When the target person is partially covered by other objects or moves in the blind spots of the camera, only certain parts of the body are likely to be detected. Figure 5c, d show moving objects, which are not people but a roof cover flapping in the wind and moving sports equipment. In addition, glittering surfaces or objects in the light are often captured, especially when artificial lighting exists. They are not the objects of interest, so we want to eliminate them using CNN classification.

There are some notable characteristics of the extracted ROIs. First, the image size varies significantly because all moving objects are the target of extraction. In crime detection, the movement of a suspicious person is often observed from a distance. Considering that modern object detection using deep learning usually fails to detect

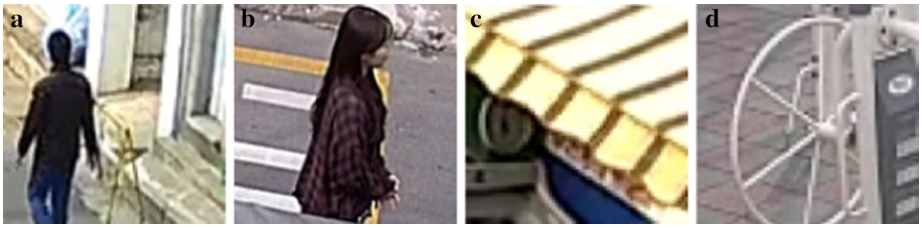

Fig. 5 Extracted images from background subtraction. **a** Individual but not the whole body. **b** Individual but not the whole body. **c** Roof cover flapping in the wind. **d** Moving sports equipment

small objects, this characteristic would be an important merit. Second, some parts of the body are often extracted, as shown in Fig. 5a, b. Especially when a person stops walking, only the moving portion of the body is detected. Sometimes, it is difficult to accurately classify these images even using manual annotation. When the images representing a tiny part of the body are extracted and used as training samples, they can downgrade the classification performance. Third, many representations of the same object with different or similar poses and backgrounds are captured, especially when the target moves around in front of the camera. These duplicate images should be reduced when making a training set.

Accounting for these characteristics, we set up the strategies for the training set of CNN classification. The default input image size is set to 64 × 64 pixels because more than 40% of the extracted images are smaller. Table 1 shows the cumulative proportion of the images for each size. Meanwhile, different types of body parts are annotated separately. Images corresponding to persons are sub-tagged as one of the 'full', 'bend', 'upper', 'head', 'cluster', and 'etc'. classes. In the last class, tiny parts of the human body that cannot be immediately identified as persons are included. To handle the problem of duplication, the subtraction results are saved once every 2 s. Because the duplications usually occur in case of persons, the person to non-person ratio within the collected images is about 8 to 1. To solve this categorical imbalance and also to reduce duplication, we take only 25% of person images.

In manual annotation, each person image is annotated as one of the above classes. Some duplicate images as well as images that are difficult to identify are are again removed. For example, human heads that show only hair are all removed. Non-person objects are annotated as 'car', 'part of car', 'animal', or 'etc.' classes. In the last 'etc.' class, all the non-person images except the ones included in the previous three categories are classified. For example, an object waving in the wind, glittering surface in the light, shadows that are not eliminated, glistening surface of objects, etc. We use five out of the eight cameras to make a pair of training and test sets. The remaining cameras are used to make the other test sets.

Table 2 presents the annotated data statistics. The sub-table (a) represents the extracted and selected images obtained from five different cameras that are used for both training and test. The sub-classes within the person class indicate full body, bent body, upper body, head, person group, etc. in order. 'Full', 'cluster' and 'upper' classes are the majority ones, whereas the remaining three classes occupy only 16%. As we mentioned above, 'etc' class includes images that are difficult to classify and the other class images are also not easy to be automatically identified. Among the classes in non-person, 'etc.' class is remarkably large, followed by the class 'Car-part'. The other two classes include only 6% of the total images. It means that most moving objects are either persons or cars and the remaining images caught by background subtraction

Table 1 Cumulative proportion of the extracted images of different sizes

Image size (smaller than)	32	64	96	128	160	192	224	256	320	800
Proportion (%)	2	41	65	77	86	92	96	98	99	100

Table 2 Extracted and selected data after background subtraction

(a) Extracted and selected data from 5 cameras—dataset A

Person class	# of images	Non-person class	# of images
Person-full	484	Car-part	333
Person-bend	60	Car	41
Person-upper	211	Animals	29
Person-head	60	etc.	772
Person-cluster	356		
Person-etc.	84		
Total	1255	Total	1125

(b) Extracted and selected data from different camera

Roadset1		Roadset2		Nightset	
Class	# of images	Class	# of images	Class	# of images
Person	100	Person	103	Person	153
Non-person	141	Non-person	107	Non-person	305
Total	241	Total	210	Total	458

are mostly wrongly detected objects. Considering the difficulty of image collection, our goal in this study is to classify images into two classes, person and non-person.

Table 2b presents three test sets extracted from the other cameras. These are used only for testing. As the training and test sets extracted from same data pool are likely to have similar objects and backgrounds, we also test the trained models with images obtained from completely different cameras. Two cameras for two different intersections and one camera for an alley at night are used as data sources. In this case, we only annotate them as one of the two classes: person and non-person. Sample images for each class are given in the Additional file 1. Because we crop images into a rectangle, unnecessary backgrounds or objects are often included in the images.

Step2: CNN object classification

The resolution of the images extracted from background subtraction varies considerably, as shown in Table 1. Therefore, we need to choose the appropriate size of input images for training an accurate CNN model. The default size is first set to 64 × 64 pixels, based on the median image size. However, this does not guarantee the best performance. Therefore, an optimum size is chosen by experiments.

Our purpose is to construct a light model to classify the extracted ROIs. As we already filtered out many uninteresting parts of the images in Step 1, we expect that a simple CNN architecture can easily handle the object classification. Inspired by the other pioneer networks such as AlexNet and LeNet, we use a basic architecture described in the right side of Fig. 3. The input images are 64 × 64 pixels of RGB color images of two categories: person and non-person. The network has two convolutional layers, two pooling layers, two fully connected layers, and an output layer at the end. At the end of a convolutional layer, an ReLU activation function is applied and it is followed by a pooling layer. Each convolutional layer has 64 feature maps, both with a 5 patch and the stride is set to

1. In the pooling layer, a max pool with 2 × 2 filters with stride 2 is used and the image size is then reduced to half. Thus, the final image size obtained after the second pooling layer is 16 × 16 pixels. At each fully connected layer, the ReLU function and dropout are applied. The first fully connected layer has 384 nodes and the second one has 192. These numbers are chosen by experiments. To find an appropriate image size, images with three different resolutions, 32 × 32, 64 × 64 and 128 × 128 pixels are tested. In the network, the number of feature maps and the number of nodes in fully connected layers change in proportion to the input size as depicted in Table 3.

Results

In this section, we describe the experimental results of CNN classification. The experiments are conducted using two types of datasets as described in Table 2. The first type corresponds to dataset A, which is constructed from a group of five cameras, and is prepared to train and test a base model. The second type including three datasets from the other cameras is used to retest the trained base model. In the latter case, we check the availability of the trained model for any change int the data source. Through the experiments, we are trying to show the effectiveness of our approach even with the limited data we constructed. We expect that our simple and fast framework would offer some ideas for handling practical issues related to computing power and resources in video surveillance processing. Considering the diversity and difficulty of the extracted ROI images, we identify objects simply into two classes: 'person' and 'non-person.'

Empirical setting

After the background subtraction, we obtain 2380 images for dataset A (Table 2a). For the training, we use two different data compositions. First, we split data into training and test sets using all the images. Second, we compose training and test sets by excluding two sub-class images of persons that are difficult to identify automatically: 'person-bend' and 'person-etc.' The former is denominated as 'full-set,' and the latter as 'part-set.' We train a model for each of the two sets separately. As a result, we get two models with the dataset A for a given image size. In the case of different camera test sets, there are 241, 210 and 458 images for roadset 1, roadset 2 and nightset respectively. These sets are used only for the evaluation.

The CNN code is written in python with TensorFlow. The experiments are carried out on a NVIDIA Tesla M40 24GB GPU. For an experiment, 80% of data are randomly selected as training set and the remainder is used as test set. In the case of different camera sets, all the images in a set are used for a validation. The detailed setting for the network is as follows: The mini-batch size is set to 50, with 30,000 iterations. The dropout

Table 3 Number of feature maps and fully connected layers

Image size	32 × 32	64 × 64	128 × 128
# of first feature maps	32	64	128
# of second feature maps	32	64	128
# of first fully connected layers	192	384	768
# of second fully connected layers	96	192	384

rate of the fully connected layer is 10%. For the convolution layers, 5 × 5 sliding windows are used. Experiments are repeated by changing the input image size and repeated random subsampling is used for each input size.

Evaluation with images obtained from the same group of cameras

Table 4 presents the experimental results with dataset A. The experiments are conducted separately on the full-set and part-set. For each input image size, we repeat the experiments 10 times using random subsampling. Each value is the average of the 10 experiments and accuracy, precision, and recall are used as measures. On both sets, the accuracy as well as other measures increase with increasing image size. The experiments performed on the part-set show better results than those on the full-set. The value in italic in each column indicates the best result obtained from among the experiments of three input sizes.

For a full-set with an image size 32 × 32 pixels, the precision for person class is 0.80 and that of a non-person is 0.83. We obtain 0.82 and 0.81 for person and non-person recalls and 0.81 for accuracy. When the image size increases to 64 × 64 pixels, the values of each measure value increase by 2–3% points except non-person precision and person recall. In these exceptional cases, the difference is statistically insignificant. With an image size 128 × 128 pixels, the accuracy increases to 0.84, which is 1% point higher than that obtained for an image size of 64 × 64 pixels. However, the person precision and non-person recall values do not change. The recall gap for a person class is comparatively high (4 points) but that of a non-person does not change. In summary, the performance improves with image size but there is comparatively little enhancement between 64 × 64 pixels and 128 × 128 pixels. Person precision and non-person recall mainly improve when the size becomes 64 × 64 pixels.

On the part-set, we get an improved result with a similar pattern. Because we eliminated difficult images from the full-set, the improvement was predictable. The results with an image size 64 × 64 pixels are clearly better than that of the image size 32 × 32 pixels. The improvement in person precision and non-person recall values is particularly high (5% points increase). When the image size becomes 128 × 128, we get the best accuracy, 0.85. However, there is little or no improvement for person precision and non-person recall values that had already attained highly improved values when the size was 64 × 64 pixels. The best precision of 0.87 is found with size 64 × 64 pixels. From these

Table 4 Evaluation with dataset A

Input size	Precision		Recall		Accuracy
	Person	Non-person	Person	Non-person	
(a) Full-set					
32 × 32	0.80	0.83	0.82	0.81	0.81
64 × 64	*0.83*	0.83	0.81	*0.84*	0.83
128 × 128	*0.83*	*0.85*	*0.85*	*0.84*	*0.84*
(b) Part-set					
32 × 32	0.82	0.82	0.82	0.81	0.82
64 × 64	*0.87*	0.82	0.83	*0.86*	0.84
128 × 128	0.86	*0.84*	*0.84*	*0.86*	*0.85*

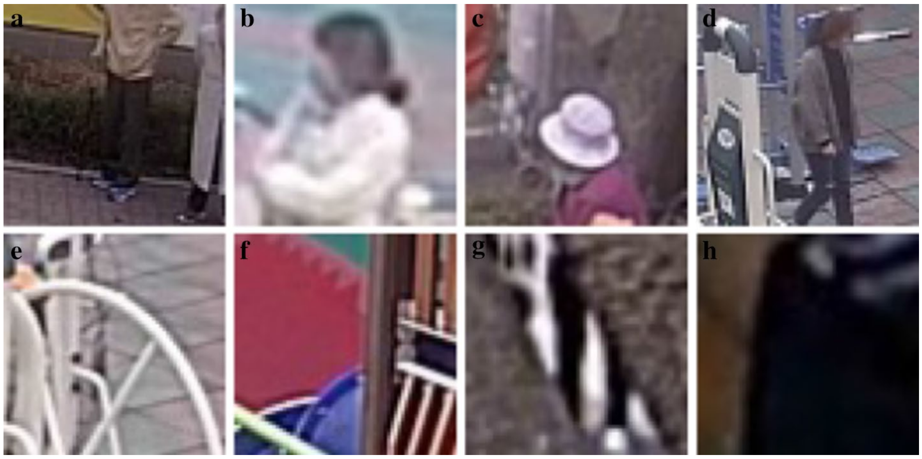

Fig. 6 Misclassified images for person class (**a**–**d**) and non-person class (**e**–**h**)

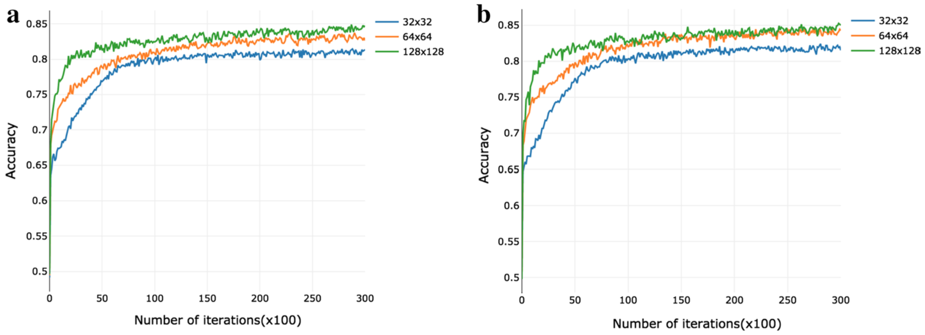

Fig. 7 Accuracy curves. **a** Averaged accuracy versus the number of iterations for the full-set. **b** Averaged accuracy versus the number of iterations for the part-set

results, we can conclude that a complicated network with too high a resolution does not significantly affect the classification performance.

When we verified the classification results in detail, we could find a pattern. Most of the full-body images were well detected, unlike head or upper body images. Some examples of misclassification are shown in Fig. 6. Figure 6b shows the upper body image with low resolution and (c) shows a person's head with a hat. An image of a person against a complex background (d) was also not well detected. Sometimes, certain objects or backgrounds, which had often co-occurred with the person in the training set (e, f), have been misclassified as persons. Simple background images had been well classified as non-persons but sometimes misclassified (h) because of the unusual images of the person in training set. For example, an extreme close-up of the person's head or upper body can be confused with a simple background.

Figure 7 shows the curves representing averaged accuracy versus the number of iterations for the full-set and part-set. We repeated 30,000 iterations with a mini-batch size 50 for input sizes of 32 × 32 and 64 × 64 pixels, and the same iterations with a mini-batch size 30 for 128 × 128 pixels because of memory constraints. The performance rapidly improves when the image size is bigger, but converges to similar values for the size 64 × 64 and 128 × 128 pixels.

As we measure training speeds, we observe that it takes about 4 min for the size 32 × 32 pixels. When the input size increases, the training time rapidly increases, to 22 min for 64 × 64 and 110 min for 128 × 128 pixels. The inference time also increases proportionally to the training time. Therefore, although the larger input size significantly enhances the classification performance, we choose the size 64 × 64 as optimum, considering the processing speed. With a size of 64 × 64 pixels, our entire framework processes images at 15fps on a NVIDIA Tesla M40. With the same setting, the processing speed of Faster R-CNN is 2.7 fps and that of YOLO is the same as that obtained by our method.

Table 5 compares the efficiency of our CNN model and that of YOLO in terms of computational complexity and total memory usage per image during training. We counted the number of parameters to be updated for the complexity. At each convolutional layer, the number of parameters is calculated as follows:

$$F^2 \cdot C \cdot K, \tag{1}$$

where F is filter size, C is the number of input channels, and K is the number of filters. Our approach significantly reduced the complexity compared to the state-of-the-art object detection method. The number of parameters is 209 K that is only 0.3% of YOLO parameters. The memory usage per image was greatly reduced as well from 53.6 to 1.7 MB. Actually, the training time of YOLO is tens of hours on a GPU. The low complexity of our approach is a great advantage when dealing with huge video data.

Despite the superiority of our approach over the benchmark in terms of computational complexity, the inference time in the experiments was not different. We suppose that there was a time delay when transferring the background subtraction result to the CNN in the code. In the current version, the subtraction algorithm is written in C whereas the main CNN classifier is written in Python. We can further reduce the processing time by optimizing the code.

Evaluation with images from different cameras

In this subsection, we evaluate the previously trained models using three datasets constructed from different cameras. For each input size, two models trained with the full-set and part-set are tested. We call them the full-set model and part-set model, respectively.

Table 5 Complexity and memory usage of our approach and a state-of-the-art object detection method (YOLO)

	Our CNN model	YOLO
Input size	64 × 64	224 × 224
# of conv. layers	2	24
# of parameters	209 K	66,000 K
Memory usage per image	1690 KB	53,606 KB

Roadsets 1 and 2 are constructed from CCTV images images obtained from two street roads, a residential area road (Fig. 4f) and a main street intersection (Fig. 4g). Roadset 1 includes many hard cases such as parts of the human body, which were eliminated while constructing the part-set. Roadset 2 includes images of a small size because the camera covers a wide area. The nightset is constructed from an alley camera filmed at night (Fig. 4h). It includes many low-resolution images.

Table 6 presents the test results with these three datasets. The measured values for all datasets are found to be lower than the results of dataset A. Let us consider the results with a full-set model (Table 6a) first. The best accuracies highlighted in italic for the full-set model are 0.72, 0.82, and 0.66 for roadset 1, roadset 2, and nightset, respectively. The performance with roadset 2 is comparable to that of data A, for which the best accuracy was 0.84.

In roadset 1, some difficult images such as body parts and dark images due to shadows might have a negative influence on performance. Moreover, new types of backgrounds could disturb the classification. However, an accuracy of 0.72 for the full-set is not bad considering that the test images are from a different camera with the training set. Person precision (0.79) is higher than the other measures whereas non-person precision (0.65) is lower. It means that the pre-trained model can predict a person appropriately but is not suitable for accurately classifying unusual person images. An interesting aspect is that high resolution does not guarantee better performance. On the contrary, the best accuracy is obtained with size 32×32 pixels.

Table 6 Evaluation with different test datasets

Test set	Input	Precision		Recall		Accuracy
		Person	Non-pers.	Person	Non-pers.	
(a) Full-set model						
Roadset1	32 × 32	0.78	*0.65*	*0.73*	0.71	*0.72*
	64 × 64	*0.79*	0.64	0.70	0.73	0.71
	128 × 128	*0.79*	0.63	0.69	*0.75*	0.71
Roadset2	32 × 32	*0.84*	0.80	0.80	*0.84*	*0.82*
	64 × 64	0.82	*0.81*	*0.81*	0.81	0.81
	128 × 128	0.83	0.79	0.78	*0.84*	0.80
Nightset	32 × 32	0.76	*0.50*	*0.72*	0.55	*0.66*
	64 × 64	*0.77*	*0.50*	*0.72*	0.56	*0.66*
	128 × 128	0.76	0.48	0.68	*0.58*	0.65
(b) Part-set model						
Roadset1	32 × 32	0.79	0.65	*0.73*	0.72	0.72
	64 × 64	*0.81*	*0.66*	*0.73*	*0.75*	*0.74*
	128 × 128	0.80	0.64	0.71	0.73	0.72
Roadset2	32 × 32	0.82	0.81	0.81	*0.82*	0.81
	64 × 64	*0.83*	*0.81*	*0.82*	*0.82*	*0.82*
	128 × 128	0.82	0.80	0.80	*0.82*	0.81
Nightset	32 × 32	0.76	0.51	*0.74*	0.54	0.67
	64 × 64	*0.78*	*0.52*	0.73	*0.58*	*0.68*
	128 × 128	0.76	0.49	0.71	0.55	0.65

In the case of roadset 2, the reason for the good performance might be the characteristics of the extracted ROIs from the video frames. As the video was taken from a larger distance compared to the others, fewer small movements, which are usually noises, were captured when processing background subtraction. Therefore, the extracted ROIs are mostly persons, cars, traffic signals, or reflected roads. There are not many complicated backgrounds, but there are many low-resolution objects. With the full-set model, the best accuracy (0.82) is observed with 32 × 32 pixels, as in roadset 1. However, the difference among the measured values is not much, unlike roadset 2. The main reason is probably that there are not many noisy backgrounds in the test images. From these results, we can infer that high-resolution models may degrade classification performance when the test images are significantly different from the training data.

With nightset, the performance is clearly poorer than the other sets. The best accuracy, person precision, and non-person precision are 0.66, 0.77, and 0.50 respectively. However, we observe similar patterns as the other sets. First, person precision is comparatively higher while non-person precision is comparatively lower as in roadset 1. The classifier could appropriately detect clear person images, but dark and unclear person images have been ignored. Second, the models trained with low-resolution images work better. However, there is not much difference between the results with different sizes.

Now, let us briefly introduce the results with a part-set model. Interestingly, they show quite a different pattern from the full-set model. On all three test sets, the model trained with the 64 × 64 pixels size images outperforms the others. The performance gaps are significant, but are higher than those observed in full-set models. Moreover, the part-set model's overall performance is better than the full-set model. We suppose the reason is the simplicity of the training data of the part-set. Because the part-set did not include difficult images such as bent body or a small part of the body, the classifier was able to focus on the general pattern of a person. Therefore, when test images are obtained from a different data pool, a simple part-set model can better classify them than a full-set model, even when the test set includes difficult images like roadset 1. From this observation, we can get some guidelines for constructing a training set. Instead of including all cases of images, filtering out too difficult cases could rather enhance the classification performance. Another conclusion is that the proper size of input images for classification in our framework is 64 × 64 pixels. The size has been chosen by considering the trade-off between computational time and classification performance. Although the size was determined from the datasets we used, it could be generalized because typical real-world cameras compositions are similar to the cameras we used for our experiments.

Conclusions and discussion

We proposed a simple framework to detect objects using outdoor CCTV video footage by combining background subtraction and CNN classification method. This study was initiated as the first phase of a roadmap for constructing an effective video surveillance system to battle crime. Therefore, it was necessary to devise a system that was fast as well as one that optimizes resource utilization. Because background subtraction is usually included as an essential feature in video surveillance applications, we can reuse the results for object detection in our framework. And for the practical experiments, we constructed datasets from various real-world CCTV cameras.

We found that the test accuracy was 0.85 when we used images from the same data pool with training data. This was comparable to the recent pedestrian detection algorithms' performance. Considering that our dataset contains much more difficult cases than usual pedestrian detection datasets, the result demonstrates a major success. For the other test sets constructed from three different cameras, the accuracies were found to be 0.74, 0.82, and 0.68 respectively. These results are encouraging because the detection accuracy usually decreases significantly when the target camera changes in real-world applications. A relatively high recall value of a person class when testing nightset data is also a positive sign because person detection is most important for video surveillance.

Our approach offers significant advantages in terms of processing, training, and manual annotation speed. The computational complexity of our CNN classifier was much lower than a state-of-the-art object detection method. Our model has 209 K parameters to be updated that are only 0.3% of the YOLO parameters. The processing speed of our entire framework on a GPU was 15 fps, which was equivalent to YOLO and 5.6 times faster than Faster R-CNN when testing the same image frames. Although the latter two models tested here can detect multiple classes, our primary objective is to identify moving objects only, especially people. Meanwhile, training speed is also important in practice because we often need to train a model fit for new environments instead of using a pre-trained one. Thanks to the simple CNN structure of our approach, the training speed is much faster than the others. In general, it takes anywhere from several hours to tens of hours to train a model on a GPU for both YOLO and Faster R-CNN. The manual annotation speed of our system is incomparable because we do not need to specify the location of objects for training. We only need to label the extracted regions from background subtraction as one of the predefined classes. The low complexity is an important merit when introducing an automatic detection system into a surveillance control center. We expect that our light detection system would work well on the existing servers. And with the system, an operator would more efficiently monitor the cameras by verifying only the person movement detected by a machine.

Another merit is the ability to detect small objects. It is useful in crime detection because suspicious objects are often seen from a distance. With outdoor surveillance cameras, the recorded objects of interest can be very small or blurred as criminals naturally prefer a CCTV blind spot. Our method first detects moving objects from a video frame and then classifies them. Therefore, any moving small objects can be caught in the first step. Considering that the limitation of other modern algorithms such as YOLO is in dealing with small objects, this characteristic would be a major advantage in video surveillance.

There are some data issues that can reduce detection performance. First, lack of training data might disturb the training procedure. A training set of 1900 examples is not small for binary classification but not enough considering that the images include various patterns. Second, same object images can cause over-fitting. After background subtraction, the case of capturing the same person multiple times occurred frequently, especially in the case of the playground video. Although we reduced these instances by taking only 25% of person images, there was still a risk of performance degradation. Third, background images are much included in training data. Because we mainly took the videos of people walking in the target areas, other moving objects were not much detected by background subtraction. On the other hand, in non-person class, background images such as glittering ground or reflecting objects have been captured very often. This made training difficult because these images usually did not have particular patterns.

Despite these limitations, we have the potential for enhancing performance. Currently, we use object tracking only to take the objects, which are continuously caught during background subtraction. If we efficiently use the tracking result for classification, the accuracy would increase. For example, instead of labeling each of the input images, we can label each tracked object by aggregating the classification result of the images corresponding to the object. Replacement of the background subtraction algorithm is also possible. GMM is a widely used method, but recently proposed advanced algorithms are expected to decrease noise detection. Finally, by optimizing the dataset, we can further enhance the performance. As mentioned above, there are some drawbacks related to training data. In future studies, we will re-construct the training dataset by adding video frames recorded from other cameras from various locations. Moreover, other open datasets can be used to expand the training data range.

Additional file

Additional file 1. Annotated image examples per class.

Authors' contributions

All authors equally contributed to the main idea of this paper and the construction of datasets. CK is in charge of a project that funds this work. YMK initially wrote the paper and conducted experiments including code implementation. JL and TH contributed to the related works especially. All authors read and approved the final manuscript.

Authors' information

Chulyeon Kim is a research assistant professor at Hanyang University, South Korea since Jan. 2014. The main research interest is to develop practical surveillance systems based on intelligent video analytics and IoTs. He received Ph.D. in industrial engineering from Hanyang University by studying an efficient approximation algorithm to solve large scale combinatorial optimization problems. He also worked as a business consultant and software engineer to redesign and implement processes for manufacturing Innovations from 2001 to 2013.

Jiyoung Lee is a Ph.D. student at Hanyang University, South Korea (M.S.-Ph.D. integrated program). She received her B.S. in industrial engineering from Hanyang University at Feb. 2014. Her research interest is in mathematical optimization, data mining, machine learning, and management of technology.

Taekjin Han is a Ph.D. student at Hanyang University, South Korea (Major is Management of Technology) He received his B.S. in Mechanical engineering from Hanyang University at Feb. 2012. He received his M.S. in Mechanical engineering from Korea Advanced Institute of Science and Technology at Feb. 2014. His research interest is in innovation system, catch-up cycle in industry, machine learning, and management of technology.

Young-Min Kim is an assistant professor at Hanyang University, South Korea since Sep. 2016. Her research background is in machine learning, probabilistic models and unsupervised learning. She received her B.S. and M.S. in industrial engineering from Hanyang University and earned her second M.S. and Ph.D. in computer science from University Paris 6, France. She completed two post-doctoral fellowships at University of Avignon and at University of Lyon 2. She was a senior researcher at Korea Institute of Science and Technology Information (KISTI) from Feb. 2014 to Aug. 2016.

Author details

[1] Graduate School of Technology & Innovation Management, Hanyang University, Seoul, Republic of Korea. [2] Department of Industrial Engineering, Hanyang University, Seoul, Republic of Korea.

Acknowledgements

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets will be available online when this work is published.

Funding

This work is partially supported by two projects, Classification of The Artists using Deep Neural Networks, funded by Hanyang University (201600000002255) and Smart Multimodal Environment of AI Chatbot Robots for Digital Healthcare (P0000536), funded by the Ministry of Trade, Industry and Energy (MOTIE).

Publisher's Note

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

Received: 30 April 2018 Accepted: 26 June 2018
Published online: 10 July 2018

References

1. "smartly" increases the cctv control efficiency. http://www.boannews.com/media/view.asp?idx=67319. Accessed 10 June 2018.
2. The government focuses on implementing intelligent cctv control center in 2017. http://www.boannews.com/media/view.asp?idx=52904. Accessed 10 June 2018.
3. The arrest rate for cctv has increased by 12 times over three years. http://news.joins.com/article/20634296. Accessed 10 June 2018.
4. Bouwmans T. Traditional and recent approaches in background modeling for foreground detection: an overview. Comput Sci Rev. 2014;11–12:31–66.
5. Wang Y, Luo Z, Jodoin PM. Interactive deep learning method for segmenting moving objects. Pattern Recogn Lett. 2017;96(C):66–75.
6. Redmon J, Divvala SK, Girshick RB, Farhadi A. You only look once: unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016; 2016. p. 779–88.
7. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: single shot multibox detector. Comput Vis ECCV. 2016;2016:21–37.
8. Bouwmans T, Zahzah EH. Robust pca via principal component pursuit: a review for a comparative evaluation in video surveillance. Comput Vis Image Underst. 2014;122:22–34.
9. The most spied upon cities in the world. https://www.worldatlas.com/articles/most-spied-on-cities-in-the-world.html. Accessed 10 June 2018.
10. West DM, Bernstein D. Benefits and best practices of safe city innovation. Washington, DC: The Brookings Institution; 2017.
11. How many cctv cameras in london? https://www.caughtoncamera.net/news/how-many-cctv-cameras-in-london/. Accessed 10 June 2018.
12. Bianco S, Ciocca G, Schettini R. How far can you get by combining change detection algorithms? 2015. CoRR, abs/1505.02921.
13. Mabrouk AB, Zagrouba E. Abnormal behavior recognition for intelligent video surveillance systems: a review. Expert Syst Appl. 2018;91:480–91.
14. Foroughi H, Aski BS, Pourreza H. Intelligent video surveillance for monitoring fall detection of elderly in home environments. In: 11th international conference on computer and information technology, 2008. ICCIT 2008. New York: IEEE; 2008. p. 219–24.
15. Lao W, Han J, De With PH. Automatic video-based human motion analyzer for consumer surveillance system. IEEE Trans Consum Electron. 2009;55(2):591–8.
16. Chen DY, Huang PC. Motion-based unusual event detection in human crowds. J Vis Commun Image Represent. 2011;22(2):178–86.
17. Arroyo R, Yebes JJ, Bergasa LM, Daza IG, Almazán J. Expert video-surveillance system for real-time detection of suspicious behaviors in shopping malls. Expert Syst Appl. 2015;42(21):7991–8005.
18. Sidhu RS, Sharad M. Smart surveillance system for detecting interpersonal crime. In: 2016 International Conference on communication and signal processing (ICCSP). New York: IEEE; 2016. p. 2003–7.
19. Valera M, Velastin SA. Intelligent distributed surveillance systems: a review. IEEE Proc Vis Image Signal Process. 2005;152(2):192–204.
20. Conde C, Moctezuma D, De Diego IM, Cabello E. Hogg: Gabor and hog-based human detection for surveillance in non-controlled environments. Neurocomputing. 2013;100:19–30.
21. Huang K, Wang L, Tan T, Maybank S. A real-time object detecting and tracking system for outdoor night surveillance. Pattern Recog. 2008;41(1):432–44.
22. Toyama K, Krumm J, Brumitt B, Meyers B. Wallflower: principles and practice of background maintenance. In: The Proceedings of the seventh IEEE international conference on computer vision, 1999, vol. 1. New York: IEEE; 1999. p. 255–61.
23. Sobral A, Vacavant A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Comput Vis Image Underst. 2014;122:4–21.
24. Bouwmans T. Background subtraction for visual surveillance: a fuzzy approach. Handb Soft Comput Video Surveill. 2012;5:103–38.
25. Lee B, Hedley M. Background estimation for video surveillance. In: Image & Vision Computing New Zealand (IVCNZ '02). Auckland, NZ; 2002. p. 315–20.
26. McFarlane NJ, Schofield CP. Segmentation and tracking of piglets in images. Mach Vis Appl. 1995;8(3):187–93.

27. Zheng J, Wang Y, Nihan N, Hallenbeck M. Extracting roadway background image: mode-based approach. Transp Res Rec J Transp ResBoard. 1944;82–88:2006.
28. Stauffer C, Grimson WEL. Adaptive background mixture models for real-time tracking. In: IEEE computer society conference on computer vision and pattern recognition, vol. 2. New York: IEEE; 1999. p. 246–52.
29. Hayman E, Eklundh JO. Statistical background subtraction for a mobile observer. In: Proceedings of the international conference on computer vision. New York: IEEE; 2003. p. 67–74.
30. Elgammal A, Harwood D, Davis L. Non-parametric model for background subtraction. In: Proceedings of the European conference on computer vision. Berlin: Springer; 2000. p. 751–67.
31. Kaewtrakulpong P, Bowden R. An improved adaptive background mixture model for realtime tracking with shadow detection. In: Proceedings of 2nd European workshop on advanced video based surveillance systems. Dordrecht: Brunel University; 2001.
32. Conaire C, Cooke E, O'Connor N, Murphy N, Smearson A. Background modelling in infrared and visible spectrum video for people tracking. In: CVPR'05 Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition—workshops. CVPR workshops. New York: IEEE; 2005. p. 20.
33. Zivkovic Z, Van Der Heijden F. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recogn Lett. 2006;27(7):773–80.
34. Yeh C-H, Lin C-Y, Muchtar K, Lai H-E, Sun M-T. Three-pronged compensation and hysteresis thresholding for moving object detection in real-time video surveillance. IEEE Trans Ind Electron. 2017;64:4945–55.
35. Zhang H, Xu D. Fusing color and texture features for background model. In: Proceedings 3 of the third international conference fuzzy systems and knowledge discovery, FSKD 2006, Xi'an, China, September 24–28, 2006. Berlin: Springer; 2006. p. 887–93.
36. El Baf F, Bouwmans T, Vachon B. Foreground detection using the choquet integral. In: WIAMIS'08 Proceedings of the 2008 ninth international workshop on image analysis for multimedia interactive services. New York: IEEE; 2008. p. 187–90.
37. Culibrk D, Marques O, Socek D, Kalva H, Furht B. Neural network approach to background modeling for video object segmentation. IEEE Trans Neural Netw. 2007;18(6):1614–27.
38. Bouwmans T. Recent advanced statistical background modeling for foreground detection—a systematic survey. Recent Pat Comput Sci. 2011;4(3):147–76.
39. Maddalena L, Petrosino A. A self-organizing approach to background subtraction for visual surveillance applications. IEEE Trans Image Process. 2008;17(7):1168–77.
40. Maddalena L, Petrosino A. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. Neural Comput Appl. 2010;19(2):179–86.
41. Gkioxari G, Girshick RB, Malik J. Actions and attributes from wholes and parts; 2014. CoRR. abs/1412.2604.
42. Kong T, Yao A, Chen Y, Sun F. Hypernet: towards accurate region proposal generation and joint object detection. In: The IEEE conference on computer vision and pattern recognition (CVPR). Las Vegas, NV; 2016. p. 845–53.
43. Yang F, Choi W, Lin Y. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In: The IEEE conference on computer vision and pattern recognition (CVPR); 2016.
44. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. p. 580–7.
45. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell. 2010;32(9):1627–45.
46. Girshick R. Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision. New York: IEEE; 2015. p. 1440–8.
47. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. In: The conference on advances in neural information processing systems. Montréal: Curran Associates; 2015. p. 91–9.
48. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: 2017 IEEE conference on computer vision and pattern recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017; 2017. p. 6517–25.
49. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: CVPR'05 Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition. CVPR 2005, vol. 1. New York: IEEE; 2005. p. 886–93.
50. Wang X, Han TX, Yan S. An hog-lbp human detector with partial occlusion handling. In: 2009 IEEE 12th international conference on computer vision. New York: IEEE; 2009. p. 32–9.
51. Dollár P, Appel R, Belongie S, Perona P. Fast feature pyramids for object detection. IEEE Trans Pattern Anal Mach Intell. 2014;36(8):1532–45.
52. Dollár P, Appel R, Kienzle W. Crosstalk cascades for frame-rate pedestrian detection. In: Proceedings of the 12th European conference on computer vision (ECCV) 2012. Berlin: Springer; 2012. p. 645–59.
53. Zhang S, Bauckhage C, Cremers AB. Informed haar-like features improve pedestrian detection. In: 2014 IEEE conference on computer vision and pattern recognition. p. 947–54; 2014.
54. Luo P, Tian Y, Wang X, Tang X. Switchable deep network for pedestrian detection. In: 2014 IEEE conference on computer vision and pattern recognition; 2014. p. 899–906.
55. Benenson R, Omran M, Hosang JH, Schiele B. Ten years of pedestrian detection, what have we learned? 2014. CoRR, abs/1411.4304.
56. Maji S, Berg AC, Malik J. Classification using intersection kernel support vector machines is efficient. In: IEEE conference on computer vision and pattern recognition, 2008. CVPR 2008. New York: IEEE; 2008. p. 1–8.
57. Dollár P, Tu Z, Perona P, Belongie S. Integral channel features. In: Cavallaro A, Prince S, Alexander D, editors. Proceedings of the British Machine Vision Conference. BMVA Press; 2009. p. 91.1–11.
58. Bilal M, Khan A, Khan MUK, Kyung CM. A low-complexity pedestrian detection framework for smart video surveillance systems. IEEE Trans Circuits Syst Video Technol. 2016;27:2260–73.
59. Kang K, Ouyang W, Li H, Wang X. Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 817–25.

60. Sermanet P, Kavukcuoglu K, Chintala S, LeCun Y. Pedestrian detection with unsupervised multi-stage feature learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2013. p. 3626–33.
61. Ouyang W, Wang X. Joint deep learning for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision; 2013. p. 2056–63.
62. Tian Y, Luo P, Wang X, Tang X. Pedestrian detection aided by deep learning semantic tasks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 5079–87.
63. Luo P, Wang X, Tang X. Pedestrian parsing via deep decompositional network. In: 2013 IEEE international conference on computer vision; 2013. p. 2648–55.
64. Tian Y, Luo P, Wang X, Tang X. Deep learning strong parts for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1904–12.
65. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The PASCAL visual object classes challenge 2011 (VOC2011) results. http://host.robots.ox.ac.uk/pascal/VOC/voc2011/results/index.html. Accessed 10 June 2018.
66. Arbeláez P, Hariharan B, Gu C, Gupta S, Bourdev L, Malik J. Semantic segmentation using regions and parts. In: 2012 IEEE conference on computer vision and pattern recognition; 2012. p. 3378–85.
67. Cheung E, Wong A, Bera A, Manocha D. Mixedpeds: pedestrian detection in unannotated videos using synthetically generated human-agents for training. In: Proceedings of the AAAI conference on artificial intelligence. New Orleans, Louisiana, USA; 2018.