

Convolutional neural network–based person tracking using overhead views

Misbah Ahmad¹ , Imran Ahmed¹, Fakhri Alam Khan¹ ,
Fawad Qayum² and Hanan Aljuaid³

Abstract

In video surveillance, person tracking is considered as challenging task. Numerous computer vision, machine and deep learning–based techniques have been developed in recent years. Majority of these techniques are based on frontal view images/video sequences. The advancement of convolutional neural network reforms the way of object tracking. The network layers of convolutional neural network models trained on a number of images or video sequences improve speed and accuracy of object tracking. In this work, the generalization performance of existing pre-trained deep learning models have investigated for overhead view person detection and tracking, under different experimental conditions. The object tracking method Generic Object Tracking Using Regression Networks (GOTURN) which has been yielding outstanding tracking results in recent years is explored for person tracking using overhead views. This work mainly focused on overhead view person tracking using Faster region convolutional neural network (Faster-RCNN) in combination with GOTURN architecture. In this way, the person is first identified in overhead view video sequences and then tracked using a GOTURN tracking algorithm. Faster-RCNN detection model achieved the true detection rate ranging from 90% to 93% with a minimum false detection rate up to 0.5%. The GOTURN tracking algorithm achieved similar results with the success rate ranging from 90% to 94%. Finally, the discussion is made on output results along with future direction.

Keywords

Convolutional neural network, person detection, person tracking, overhead views, Faster region convolutional neural network, Generic Object Tracking Using Regression Networks

Date received: 16 March 2020; accepted: 21 May 2020

Handling Editor: Manuel Mazzara

Introduction

Nowadays, convolutional neural network (CNN)-based models achieve remarkable success, particularly in the area of pattern recognition, image processing, remote sensing, data classification, computer vision, and smart surveillance analysis (specifically in object detection, tracking, and recognition). Person tracking is used for analysis of the target trajectory of person in video sequences. It is considered important because of its wide applications in numerous research areas, covering unusual event detection, fall detection in elderly humans, congestion or crowd locality evaluation, human–computer interaction, robot navigation, and so on.

¹Center of Excellence in Information Technology, Institute of Management Sciences, Peshawar, Pakistan

²Department of Computer Science and Information Technology, University of Malakand, Chakdara, Pakistan

³Computer Sciences Department, College of Computer Science and Information Sciences, Princess Nourah Bint Abdulrahman University (PNU), Riyadh, Saudi Arabia

Corresponding author:

Misbah Ahmad, Center of Excellence in Information Technology, Institute of Management Sciences, Peshawar, KPK 25000, Pakistan.
Email: misbahahmad4872@gmail.com



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work

without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

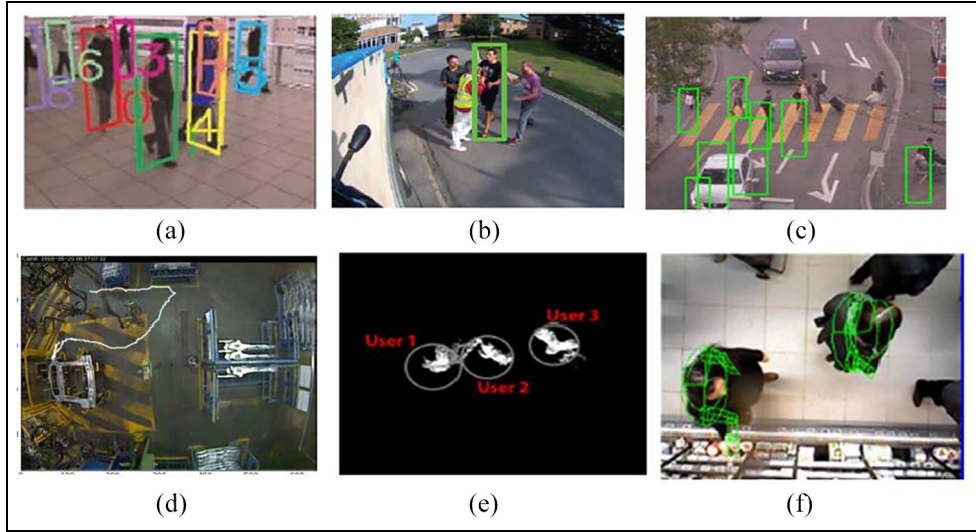


Figure 1. Sample images highlighting difference between (a–c) frontal view person tracking^{2,3} and (d–f) overhead view person tracking.^{4–6}

Person tracking is a challenging task, as the person is a deformable object and has variation in appearance, pose, scale, and size. Various person tracking methods have been developed by researchers,¹ which have shown good tracking results for normal frontal view images and video sequences. There are a number of factors which may affect the accuracy of different tracking algorithms, including variations in illumination conditions, complex backgrounds, cluttered scenes, abrupt motion, shadows, deformable nature of the person (scale and size variations), close interaction of peoples, camera perspective view, and occlusion as in Figure 1. As per first row sample images, the person shape and size are varied with respect to camera distance. The person visibility is also affected because of other person or object (occlusion), and close interaction between two persons may cause hurdles during tracking.

To overcome the problem of occlusion and better scene visibility, some researchers^{4,7–9} suggested and considered overhead/top view for person tracking and detection. The change in scene visualization due to camera perspective is highlighted in Figure 1. In case of overhead view as shown in sample images (d), (e), and (f) of Figure 1, the object appearance, visibility, and pose are significantly different from the frontal view as shown in Figure 1(a)–(c). The appearance of a person is considerably affected depending upon the location of the person with respect to the camera position. Moreover, overhead view camera covers a wide field of view and provides better visibility of scenes. Using single overhead camera may provide an efficient solution for saving energy consumption, human resource, and installation costs.¹⁰

In this work, overhead view person tracking is performed using CNN-based detection and tracking models. For overhead person tracking, Generic Object Tracking Using Regression Networks (GOTURN) algorithm¹¹ is used which takes advantage of the CNN architecture to track the object in a video sequence with high accuracy and speed. To detect or identify the object (person), Faster region convolutional neural network (Faster-RCNN)¹² is combined with a GOTURN tracking algorithm. In this way, the object (person) is first detected using Faster-RCNN and tracked using the GOTURN algorithm in overhead view video sequences. The GOTURN and Faster-RCNN models were pre-trained using normal frontal view data set, while for testing purpose, the overhead view person data set is used. The present article mainly focuses on the following:

- Performing overhead view person tracking using GOTURN tracking algorithm combined with the Faster-RCNN detection model.
- For testing purpose, overhead view person data set is used, containing video sequences having variation in person appearance (including a variety of poses, shapes, and scales of person) and different camera resolutions with indoor and outdoor backgrounds.
- To show the generalization performance of GOTURN and Faster-RCNN (pre-trained using normal or frontal view data set), testing is performed on a completely different data set, that is overhead view person data set.

- Importance of the CNN-based overhead view person tracking model is explored in contrast with conventional frontal view, particularly in the field of video surveillance. Insight discussion is made to understand the importance of CNN-based overhead view person tracking with future guidelines.

A brief review of different tracking methods, including traditional and deep learning-based techniques, is provided in the “Literature review” section. The overhead view data set is briefly discussed in the “Data set” section. The overhead view person tracking model using GOTURN along Faster-RCNN is elaborated in the successive section. Detailed explanation of output results and performance evaluation is provided in the “Experimental results” section, and finally, the “Conclusion” section concludes the discussed work.

Literature review

This section provides a brief summary of different tracking algorithms used in literature. The section is categorized into traditional generic, machine learning, features, and deep learning-based methods. A comprehensive survey of different tracking methods can be found in previous studies.^{1,13,14}

The key purpose of tracking techniques are to detect objects in a video sequence and sustain the tracking information in the successive frames to find trajectories of each detected object. Conventional techniques are usually based on motion- and observation-based models. The motion model involves the detection and prediction of object position in successive frames,^{15,16} while the observational model focused on tracked object appearance and its position across the frame.¹⁷ Some researchers¹⁸ used the template-based method for object tracking. Numerous researchers utilized machine learning-based methods for object tracking, which classifies¹⁹ the tracked object such as boosting,²⁰ random forest,²¹ Hough forest,²² structural learning,²⁰ and support vector machine (SVM).⁷ Some proposed feature-based tracking methods such as Haar-like features,²³ local binary pattern (LBP),²⁴ histogram of oriented gradient (HoG),^{25,26} scale-invariant feature transform (SFIT),²⁷ discrete cosine transform (DCT),^{28,29} and shape features.³⁰ Other techniques employed Kalman filters or Hungarian algorithm.³¹

To improve the performance of tracking methods, different researchers³² combined multiple cues information and presented methods for object tracking which combines feature-based detector with the probabilistic segmentation method. Majority of these methods are mainly developed for frontal view data set which may suffer from occlusion problems. To overcome the

occlusion problem, Ahmad et al.³³ considered overhead view scene for person detection using the background subtraction method. Ahmed et al.⁸ proposed a feature-based solution for person tracking in the top view industrial environment. Ullah et al.³⁴ provided a comparison of different traditional tracking algorithms using the overhead view data set. A rotation invariant solution⁹ is also presented for top view person tracking.

Because of recent advancement in deep learning methods such as object detection^{5,12,35–37} and image classification,^{38–41} deep learning models are now also being used in object tracking tasks. Specifically, a popular paradigm tracking by detection is used to solve the tracking problem.^{42–44} Such type of models generally used a detector to first detect the object and then subsequently a tracker is initialized to track the detected object.

Some approaches used for object tracking by detection are based on a calculation of bounding boxes in successive frames via Intersection over Union (IoU).⁴⁵ Various methods are based on recurrent neural networks (RNNs)^{46,47} or Siamese convolutional networks architectures.⁴⁸ Most of the developed methods mainly used frontal view data set. Some of the researchers developed object tracking method,^{49–52} but they use aerial or remote sensing images. Few used deep learning for detection and counting of person in the overhead view.^{53–55} In this work, the CNN-based model is used for overhead view person tracking in different indoor and outdoor environments. Using the overhead view, different problems faced during frontal view data set as discussed in the “Introduction” section may be overcome.

Data set

In this work, overhead view person data set is used, containing video sequences of person against completely different backgrounds and illumination conditions. The setup consists of a single overhead view camera which captures/records video sequences at 20 frames per second, during different day timings in different scenes. The recorded data set covers wide variety in person appearance, pose, scale, size, and orientation with respect to the camera position as depicted in Figures 5–7. From the sample frames, it can be observed that from the overhead perspective, the person appearance is totally different than frontal. The recording of the data set has been made using different camera devices and camera resolutions elaborated in Table 1. As a person is an important object in video surveillance, the data set contains person video sequences. Table 1 provides the detailed description of the data set used in this work.

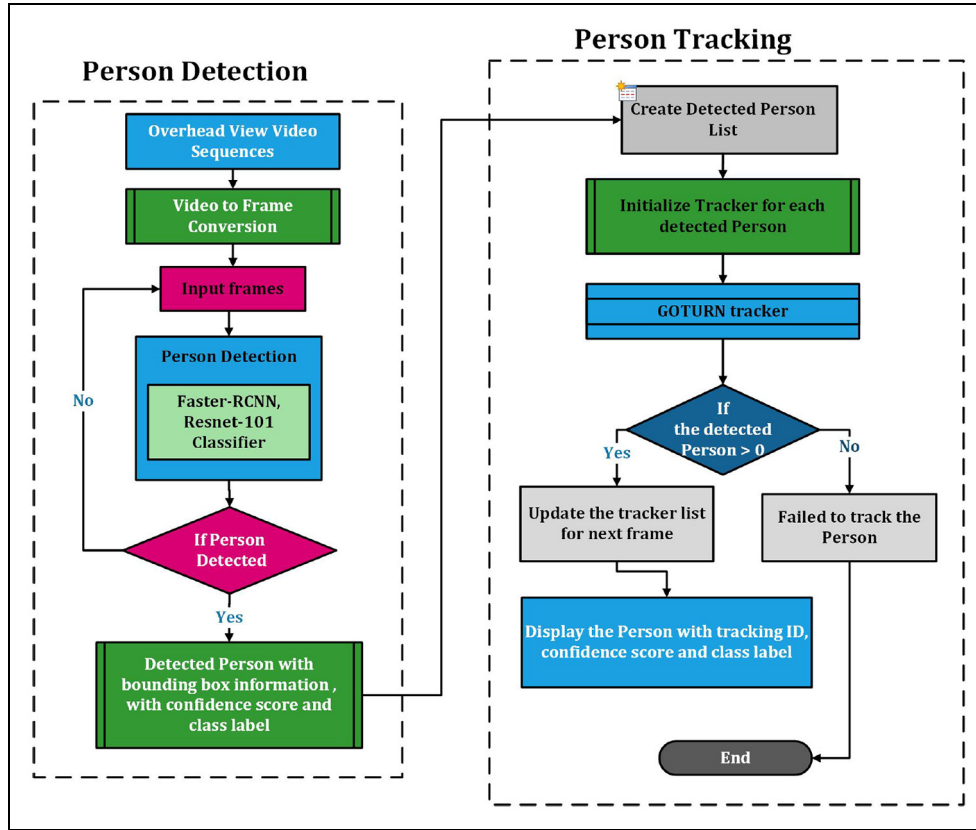


Figure 2. General framework of overhead view person tracking using Faster-RCNN detection model combined with GOTURN tracking architecture.

Table 1. Overhead view person data set.

S.no.	Description	
1	Color model	RGB
2	Duration of videos	4–30 min
3	Video type	.avi/mp4
4	Frame rate	20 frames per second
5	Height of camera	3 m from ground
6	Number of subjects	1 to many
7	Image resolution	1250 × 780 and 640 × 480
8	Location	Indoor/outdoor
9	Shadows/reflections	Yes
10	Image format	.JPG

Overhead view person detection and tracking framework

In this section, overhead view person tracking framework shown in Figure 2 is discussed. The overall framework consists of person detection and person tracking modules. In person detection module, the video sequences are converted into video frames, given to the Faster-RCNN¹² model, which detects and identifies the person in overhead view video frames. In recent years, Faster-RCNN has shown excellent performance in various applications. It uses the same conventional network

for object detection and region proposal generation which make it faster as compared to other region proposal network (RPN) models such as RCNN⁵⁶ and Faster-RCNN.³⁵ Therefore, for overhead view person detection, Faster-RCNN is chosen with Resnet-101 as depicted in Figure 2. The detection model output comprises a bounding box (containing detected person, label, and a confidence score).

The Faster-RCNN model is further combined with GOTURN¹¹ tracking architecture, which is also based on convolutions neural network layers. The detected information (containing bounding box, label, and a confidence score) is given to person tracking module. The tracking module takes the detected information and creates a list for person tracking. If the frames contain the person, the tracker starts tracking with continuously updating the tracker list information and displays person with tracking ID, otherwise it stops tracking. The following subsection explains the detection and tracking modules shown in Figure 2.

Person detection

With advancement in CNN, object detection is also gaining attention of researchers by providing an efficient solution for many object classification and

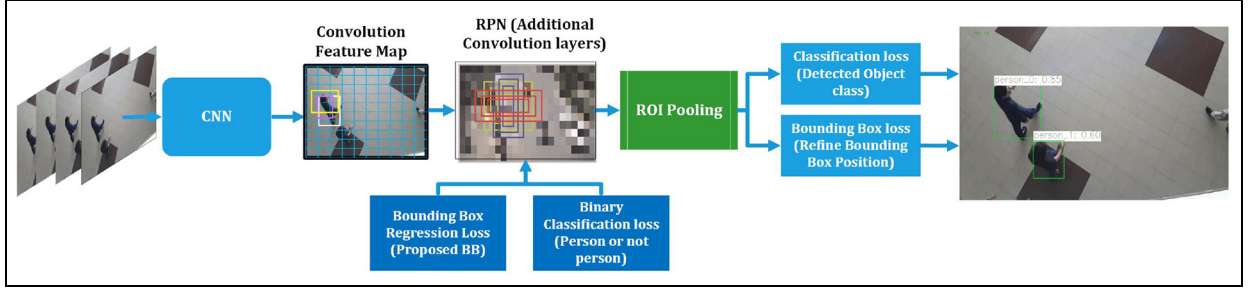


Figure 3. Complete framework diagram for overhead view person detection using Faster-RCNN.¹²

detection problems in terms of speed and computation. In this work, Faster-RCNN¹² is employed for person detection using overhead view video frames as shown in Figure 3. Faster-RCNN mainly has two stages, first stage produces region anchors (regions having high probability about occurrence of the object (person)) via RPN. The next stage classifies object (person) using detected regions and extracts bounding box information. The process shown in Figure 3 is further divided into the following three steps:

- Using convolution layers, feature extraction is performed and convolution feature map is generated at the end.
- In the second step, anchor boxes are generated using the sliding window approach, which are further refined to specify the presence of an object (person).
- In the last step, using a small network, anchors are refined and the loss function is calculated which selects best anchor regions in overhead view images containing object (person).

In Figure 3, it can be seen that overhead person frames are fed into the conventional layers, which generates feature maps that is the pre-requisite step for RPN. For feature extraction in overhead view images, Resnet-101⁵⁷ based model is used as backbone which contains 30 CNN layers. It has special residual connections in-between layers which helps to learn intermediate, local, and global features, making it effective in comparison with other CNN-based models.

There are different types of Resnet connections (for details, readers are referred to He et al.⁵⁷). In the next step, sliding window approach is used for generation of region proposal. To produce region anchors, window size $n \times n$ is slide across the feature map, where $n = 3$. Therefore, for input image, a set of nine anchors or detection boxes is produced for each pixel. All anchors have the same center (x, y) with multiple aspect ratios. Figure 3 shows that different anchor boxes are generated for a person. It can be seen that the anchors of similar color have the same aspect ratios but different

scale sizes. The bounding box coordinates⁵⁶ are calculated as follows

$$\begin{aligned} t_x &= \frac{(x-x_a)}{w_a}, & t_y &= \frac{(y-y_a)}{h_a} \\ t_w &= \log \frac{w}{w_a}, & t_h &= \log \frac{h}{h_a} \\ t_x^* &= \frac{(x^*-x_a)}{w_a}, & t_y^* &= \frac{(y^*-y_a)}{h_a} \\ t_w^* &= \log \frac{w^*}{w_a}, & t_h^* &= \log \frac{h^*}{h_a} \end{aligned} \quad (1)$$

In equation 1, bounding box central coordinates are denoted by x, y . The h is the height and w represents the width of the bounding box. In addition, ground truth GT , anchor box, and predicted bounding box are represented by x^* , x_a , and x respectively. To determine how anchor regions overlapped with ground truth bounding boxes, (IoU) intersection of the union method is used. A threshold is defined for a region containing object as person or as background. The probability for person based on the IoU value is given in the equation¹² below

$$IoU = \frac{BoundingBox(anchor) \cap GT}{BoundingBox(anchor) \cup GT} \begin{cases} > 0.7 = person \\ < 0.3 = Noperson \end{cases} \quad (2)$$

After selection of anchor regions, loss function is used for fine tuning at the end of RPN. A network is used for binary classification to classify the detected object as no person (simply background) or person. The regression function is applied to determine the positions of the predicted bounding box using four coordinates x, y, w, h as stated earlier where the center point is denoted with x, y and height and width of the anchor box is represented with h and w . To estimate the loss for regression and classification, loss function¹² is defined as

$$\begin{aligned} L(\{p_i\}, \{t_i\}) &= \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \\ &\lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \end{aligned} \quad (3)$$

The above equation is used for calculation of classification and regression loss of detected bounding box. i is used as the index for anchor, predicted probability of p_i as person is represented as λ . The ground truth is denoted with a condition, and person classification is defined as p_i^* ; if the p_i^* value equals to 1, then anchor belongs to positive class (person); otherwise, in case of 0, the anchor belongs to negative class (background). Vector t_i is used which signifies predicted bounding box coordinates, while t_i^* is used to represent ground truth coordinates. Since RPN was trained to produce regions of interest (ROIs) on convolution feature maps, which represents enclosed bounding boxes around detected object having high probability about presence of objects (persons), the selected features by RPN are given to the ROI pooling layer which classifies the detected object as person or no person and also refines the detected bounding box. For that purpose, maximum pooling is performed at inputs; for each ROI, it converts nonuniform size feature maps into fix-sized feature maps.

Now, the same size feature maps or ROIs are extracted from the ROI pooling layer. The feature maps are once again used for classification and regression. It involves two steps: in the first step, the calculation of bounding box classification and regression is made, for optimization of the loss function. At the second stage, classification shows the results of detected person class score and regression shows the resized bounding box with values x, y, w, h to cover complete person in overhead view image as shown at the output image in Figure 3. The Faster-RCNN model used in this work was pre-trained on COCO data set.⁵⁸ The model outputs the person bounding box information (containing bounding box coordinates, width and height information along with the class label and confidence score) as shown in Figures 2 and 3; this information is further used for tracking the person. Each

person's detection results are stored in list which is further used by tracking module.

Person tracking

In this section, overhead view person tracking module is briefly discussed. For tracking purpose, GOTURN¹¹ is used which is based on CNN layer architecture. The GOTURN architecture is fundamentally trained on thousands of cropped frames (mainly frontal view video sequences).¹¹ In the first frame, the location of the known ROI (person) is cropped. The cropped ROI is two times the detected bounding box, and location of the each detected ROI in the next frame is predicted as seen in Figure 4. The same information of bounding box is used to crop the person or ROI in the second frame. In the second frame, to predict bounding box location, CNN is used. The main architecture is similar to that of the Siamese CNN model⁴⁸ (trained using frontal view data set), which is used for object tracking at high speeds. The flow diagram of GOTURN architecture used for overhead view person tracking is explained using Figure 4. The GOTURN tracker takes the detected bounding box information from the first frame, and pass it through the set of convolution layers. The convolution layers are further concatenated with fully connected layers. In the next successive frame, the same detected bounding box or ROI information is used to initialize the tracker. The output of the Figure 4 is the tracked person from the overhead view. The general architecture of GOTURN model is explained in the following steps:

- The original network was entirely trained on frontal view video sequences/images. It learns the genetic relationship of the object appearance and motion.⁴⁸ At each training iteration, this

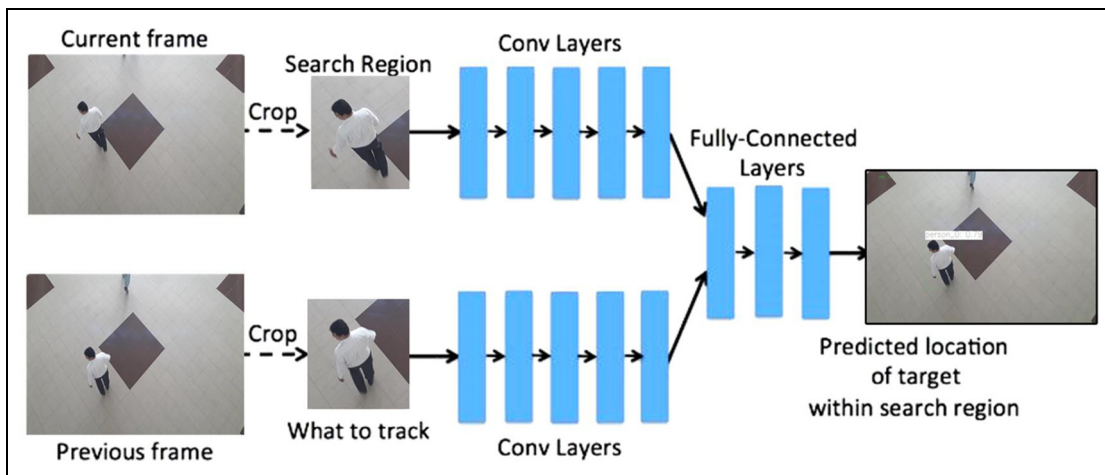


Figure 4. General architecture of GOTURN tracker.¹¹

relationship was learned by convolution network using the contextual information of the object, and it takes two inputs from the frame: crop bounding box of the previous frame at time $t - 1$ and crop bounding box of the subsequent frame at time t . The crop bounding box at time $t - 1$ usually contains the target information, that is, person as shown in Figure 4. The information of detected bounding box center at c is given in the below equation, where c_x and c_y are the center coordinates with width of w and height of h

$$c = (c_x, c_y) \quad (4)$$

- In the second frame, the same object bounding box information which needs to be searched and tracked and called target object is used by next successive frames. The bounding box width and height are automatically adjusted by the tracking algorithm according to the detecting person bounding box information as shown in Figure 4. In original algorithm, it is usually centered at the same point $c = (c_x, c_y)$ as the previous frame; although, the bounding box height and width are scaled by a factor of k .
- The input of the detection model passes through a set of convolutional layers of tracking algorithm, and the first five layers are based on CaffeNet architecture,⁵⁹ whereas the output convolutional layers (i.e. the pool5 features) are concatenated in one vector form with length of 4096 nodes.¹¹ These layers are further input to three fully connected layers. Finally, the fully convolution layer is associated with the output layer, includes four nodes representing the bottom and top coordinates of the tracking bounding box (see Figure 4).
- A condition is defined for tracking the person in the overhead view input frames: If the target person does not move too rapidly, then the target will be scaled in the present search area. Therefore, the network estimates four coordinates of bounding box directly containing tracked person in search area for the next frame. To initialize and preserve the tracker, the following rules are needed:
 - The tracker is initialized for first detection information received from the person detection module. This detection information is fed into the tracker and target person is defined as shown in Figure 4
 - If the detection information in the object list is greater than zero, the tracker is initialized. For next input, the previous frame (at time $t - 1$ frame) predicted bounding box

information and the current frame (scaled area at time t in the frame) search area is used.

- To ensure that tracker does not lose the target, in successive frames, the detection information is checked for at least 10 frames. Furthermore, the results are compared using *IoU* measure with the tracker's prediction.
- The predicted track bounding box is then swapped with the predicted detecting bounding box to improve the accuracy. When an object is detected in the field of view, it is tracked by the tracker, however, not be tracked whenever it leaves the field of view. The algorithms stop tracking an object when they do not receive any detection in the successive frames.

Experimental results

This section provides a detailed summary of experimental results using Faster-RCNN and the GOTURN model for overhead view person detection and tracking. Both models are implemented using OpenCV. The experimental results of person detection and tracking from the overhead view using different scenes with variation in background, illumination condition, person appearance (height, pose, angles, scale, and size), camera resolution, and aspect ratio are visualized in Figures 5–7. The first two scenes shown in Figures 5 and 6 mainly contain frames of the person captured from the complete overhead view (symmetric) where the person is considered below the camera in indoor and outdoor environments. As in video surveillance, the main focus is to track the person, so in almost all scenes, our focused ROI is a person. In Figure 5, the tracking and detection results of GOTURN and Faster-RCNN for overhead view person sample frames for indoor environment are depicted. In the sample images, the camera perspective is completely different than the frontal view, but still, the models perform well and give a good detection and tracking results. The green bounding box in the sample frames represents the detected bounding box along with its class label (person) and confidence score which is almost 80% in most of the time. The tracking ID can also be seen, as there is only one person in the video sequence, so ID 0 is assigned to the target person. Furthermore, during the sequence, if the same object is detected, the tracking algorithms continue tracking the object with the same tracking ID (Figure 5). Figure 6 demonstrates the overhead view person tracking and detection results for outdoor environment. The sample frames contain the results of person detection and tracking at different



Figure 5. Testing results of person detection and tracking using Faster-RCNN and GOTURN, in an indoor environment covering the symmetric overhead view.

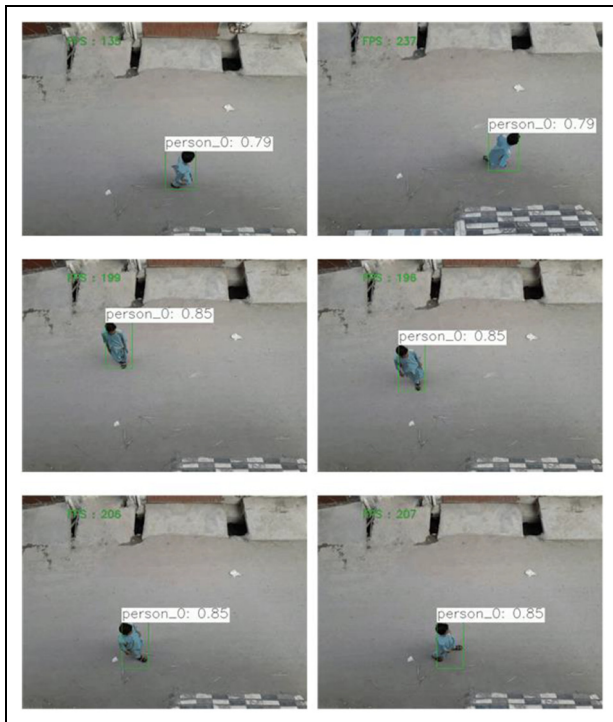


Figure 6. Testing results of person detection and tracking using Faster-RCNN and GOTURN, in an outdoor environment covering symmetric overhead view.

locations of the outdoor scene. It can be seen that the appearance of the person is significantly different in each sample frame with reference to the camera position. The person's lower body is occluded below the camera, but still, the model detects the person, classifies it to its label class, and gives good tracking results.

Similarly, the models are also tested for sample frames of person against different backgrounds covering both symmetric and asymmetric overhead view in indoor and outdoor environments as shown in Figure 7. The tracking ID for multiple persons can also be seen, as there are multiple persons in scene; therefore, unique IDs are assigned, which are further used for tracking the person in successive frames. In Figure 7, the sample frames used for testing are completely different from trained models, but still, the above discussed models achieve good results while detecting and tracking person from symmetric and asymmetric overhead views for variety of backgrounds. In Figure 7, the first row indicated the results for the symmetric overhead view, although the persons are too close, but still, the tracking and a detection model achieves good results. Along with good detection results, some false detection and not-detected results are also reported. As in Figure 7, the red detected box indicates not-detected results. The reason might be the complete change in appearance (size, shape, and scale variation) of the person from the overhead view. The overhead view results of person detection and tracking models for outdoor and indoor asymmetric environment are visualized in the second row of Figure 7.

The performance of detection and tracking models are assessed using the true detection rate (TDR) and false detection rate (FDR). For tracking purposes, the accuracy and success rate has been calculated. Table 2 demonstrates the detection results for overhead view person sample frames. The detection model achieves good results without any additional training. For person sample frames, the Faster-RCNN achieves TDR of 93%.

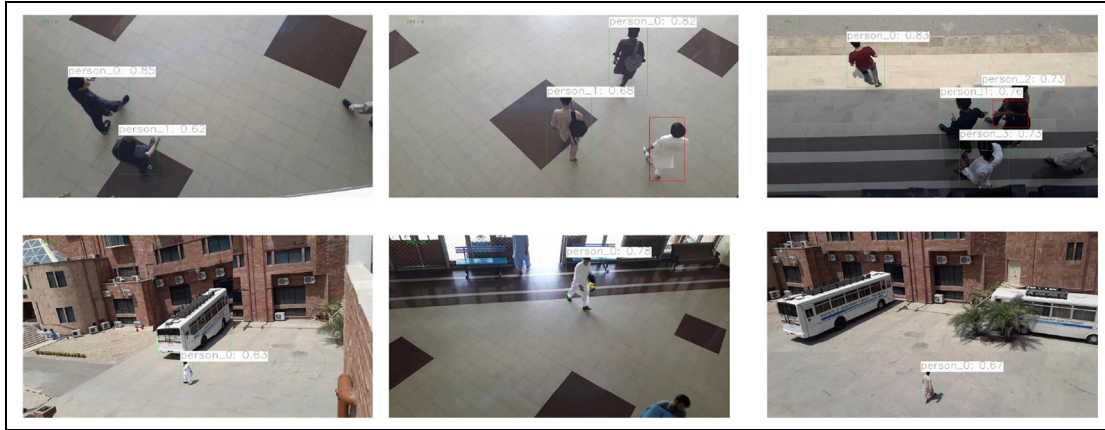


Figure 7. Testing results of person detection and tracking using Faster-RCNN and GOTURN, in an outdoor environment covering both symmetric and asymmetric overhead view.

Table 2. Overhead view person detection results.

Detected object	TDR (%)	FDR (%)
Person indoor symmetric	93	0.4
Person indoor asymmetric	94	0.4
Person outdoor symmetric	93	0.5
Person outdoor asymmetric	93	0.5
Multiple person indoor symmetric	92	0.5
Multiple person indoor asymmetric	92	0.5
Multiple person outdoor symmetric	90	0.5
Multiple person outdoor asymmetric	90	0.5

TDR: true detection rate; FDR: false detection rate.

For multiple person images, the TDR is ranging from 91% to 90% depending on illumination, background conditions, and the number of persons within the scene.

The tracking accuracy of the GOTURN tracker for overhead view person can be seen in Figure 8. The accuracy has been plotted for different numbers of persons against different outdoor and indoor conditions.

It can be seen that overall performance of the tracking algorithm is good. The accuracy is also computed for multiple persons from the overhead view. Figure 8 illustrates that the algorithm accuracy is slightly effected which depends upon the number of people in the scene. As seen from Figure 7, in case of where the persons are too close to each other, the tracking algorithm is not able to detect and track the person accurately. The GOTURN tracking algorithm results are also compared with those of other conventional tracking algorithms as shown in Figure 9. Multiple instance learning (MIL) tracker is better in accuracy but suffers

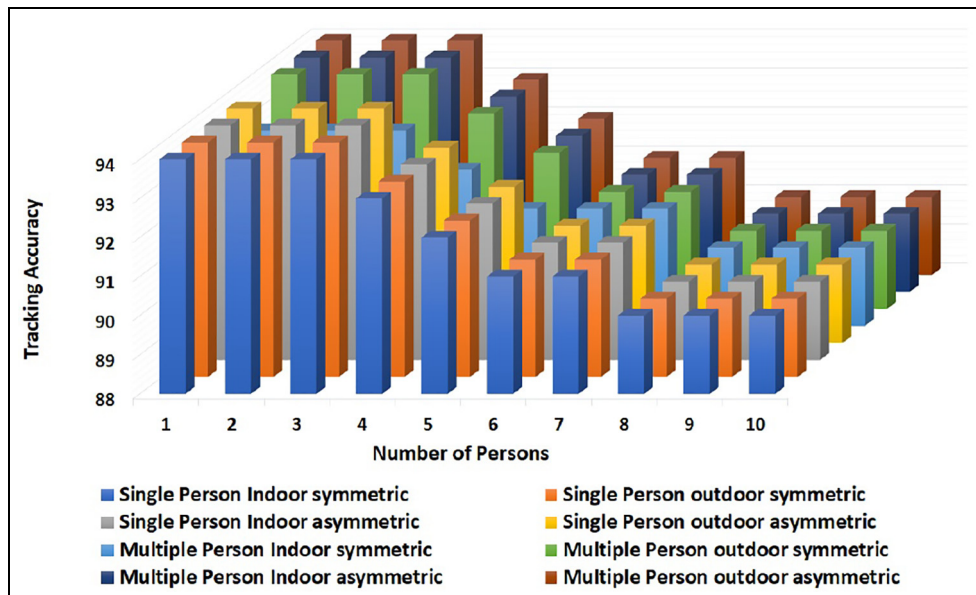


Figure 8. GOTURN tracking accuracy results for overhead view person tracking in different indoor and outdoor environments.

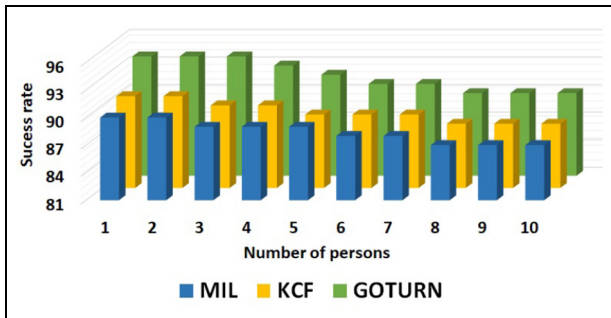


Figure 9. Tracking success rate of GOTURN as compared to other traditional tracking algorithms for overhead view multiple person tracking.

during reporting failure. Similarly, Kernelized Correlation Filter (KCF) algorithm is faster but does not handle occlusion well. On the other hand, GOTURN algorithm gives good results and better handle changes during tracking (e.g. view point changes, deformation and lighting changes).

Conclusion

In this work, overhead view person tracking is performed using CNN-based object detection model and tracking algorithm. For person detection, Faster-RCNN is used which achieves good detection results for overhead view images. Furthermore, for person tracking, the Faster-RCNN detection model is combined with GOTURN tracking algorithm. The models used in this work were pre-trained on normal frontal view images, while tested on completely different (overhead view person) data set, containing multiple persons against different backgrounds. This work is attempt to combine and used CNN-based models for person tracking and detection using the overhead view. The experimental results demonstrate the robustness and efficiency of CNN-based detection model and tracking algorithm, although there is significant variation in the data set in terms of appearance, visibility, shape, and size of the person in contrast with the normal frontal view. The results prove the performance of tracking algorithm with a success rate of 94% and detection model by achieving a TDR of 90% to 93% with an FDR of 0.5%. In future, this work might be extended by training the models on a suitable overhead view data set.

Acknowledgements

The authors would like to thank the Institute of Management Sciences (IMSciences), Hayatabad, Peshawar, Pakistan, for supporting the technical aspects of this research.


Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by the Deanship of Scientific Research at Princess Nourah Bint Abdulrahman University (PNU), Riyadh, Saudi Arabia, through the Fast-track Research Funding Program.

ORCID iDs

Misbah Ahmad  <https://orcid.org/0000-0001-7013-0159>

Fakhri Alam Khan  <https://orcid.org/0000-0002-9130-1874>

References

1. Zhou S, Ke M, Qiu J, et al. A survey of multi-object video tracking algorithms. In: Abawajy JH, Choo KKR, Islam R, et al. (eds) *International conference on applications and techniques in cyber security and intelligence*. Berlin: Springer, 2018, pp.351–369.
2. Breitenstein MD, Reichlin F, Leibe B, et al. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Trans Pattern Anal Mach Intell* 2011; 33(9): 1820–1833.
3. Khan G, Tariq Z and Khan MUG. Multi-person tracking based on faster R-CNN and deep appearance features. In: Mazzeo PL, Ramakrishnan S and Spagnolo P (eds) *Visual object tracking in the deep neural networks era*. London: IntechOpen, 2019.
4. Ahmed I and Adnan A. A robust algorithm for detecting people in overhead views. *Cluster Comput* 2018; 21(1): 633–654.
5. Van Etten A. You only look twice: rapid multi-scale object detection in satellite imagery. arXiv:1805.09512v1, 2018.
6. Migniot C and Ababsa F. Hybrid 3D–2D human tracking in a top view. *J Real Time Image Process* 2016; 11(4): 769–784.
7. Ahmed I, Ahmad A, Piccialli F, et al. A robust features-based person tracker for overhead views in industrial environment. *IEEE Internet Things J* 2017; 5(3): 1598–1605.
8. Ahmed I, Ahmad M, Adnan A, et al. Person detector for different overhead views using machine learning. *Int J Mach Lear Cyb* 2019; 10: 2657–2668.
9. Ullah K, Ahmed I, Ahmad M, et al. Rotation invariant person tracker using top view. *J Amb Intel Hum Comp* 2019; 1–17.
10. Ahmad M, Ahmed I, Ullah K, et al. Energy efficient camera solution for video surveillance. *Int J Adv Comput Sci Appl* 2019; 10(3): 522–529.
11. Held D, Thrun S and Savarese S. Learning to track at 100 fps with deep regression networks. In: *ECCV 2016*:

- 14th European conference on computer vision, Amsterdam, 11–14 October 2016, pp.749–765. Cham: Springer Nature.
12. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 2017; 39(6): 1137–1149.
13. Ahmad M, Ahmed I, Ullah K, et al. Person detection from overhead view: a survey. *Int J Adv Comput Sci Appl* 2019; 10(4): 567–577.
14. Smeulders AW, Chu DM, Cucchiara R, et al. Visual tracking: an experimental survey. *IEEE Trans Pattern Anal Mach Intell* 2013; 36(7): 1442–1468.
15. Comaniciu D, Ramesh V and Meer P. Kernel-based object tracking. *IEEE Trans Pattern Anal Mach Intell* 2003; 24(5): 564–575.
16. Li Y, Ai H, Yamashita T, et al. Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans. *IEEE Trans Pattern Anal Mach Intell* 2008; 30(10): 1728–1740.
17. Kwon J and Lee KM. Tracking by sampling and integrating multiple trackers. *IEEE Trans Pattern Anal Mach Intell* 2013; 36(7): 1428–1441.
18. Wang D, Lu H and Yang MH. Online object tracking with sparse prototypes. *IEEE Trans Image Process* 2012; 22(1): 314–325.
19. Ahmad M, Khan AM, Mazzara M, et al. Multi-layer extreme learning machine-based autoencoder for hyperspectral image classification. In: *Proceedings of the 14th international conference on computer vision theory and applications (VISAPP'19)*, Prague, Czech Republic, February 2019, pp.25–27.
20. Yao R, Shi Q, Shen C, et al. Part-based visual tracking with online latent structural learning. In: *2013 IEEE conference on computer vision and pattern recognition*, Portland, OR, 23–28 June 2013, pp.2363–2370. New York: IEEE.
21. Santner J, Leistner C, Saffari A, et al. Prost: parallel robust online simple tracking. In: *2010 IEEE conference on computer vision and pattern recognition*, San Francisco, CA, 13–18 June 2010, pp.723–730. New York: IEEE.
22. Gall J, Yao A, Razavi N, et al. Hough forests for object detection, tracking, and action recognition. *IEEE Trans Pattern Anal Mach Intell* 2011; 33(11): 2188–2202.
23. Hare S, Golodetz S, Saffari A, et al. Struck: structured output tracking with kernels. *IEEE Trans Pattern Anal Mach Intell* 2015; 38(10): 2096–2109.
24. Yang F, Lu H, Zhang W, et al. Visual tracking via bag of features. *IET Image Process* 2012; 6(2): 115–128.
25. Dalal N and Triggs B. Histograms of oriented gradients for human detection. In: *2005 IEEE conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, San Diego, CA, 20–25 June 2005, pp.886–893. New York: IEEE.
26. Lu Y, Wu T and Chun Zhu S. Online object tracking, learning and parsing with and-or graphs. In: *2014 IEEE conference on computer vision and pattern recognition*, Columbus, OH, 23–28 June 2014, pp.3462–3469. New York: IEEE.
27. Fan J, Shen X and Wu Y. Scribble tracker: a matting-based approach for robust tracking. *IEEE Trans Pattern Anal Mach Intell* 2011; 34(8): 1633–1644.
28. Li X, Dick A, Shen C, et al. Incremental learning of 3D-DCT compact representations for robust visual tracking. *IEEE Trans Pattern Anal Mach Intell* 2012; 35(4): 863–881.
29. Khan FA, Shaheen S, Asif M, et al. Towards reliable and trustful personal health record systems: a case of cloud-dew architecture based provenance framework. *J Amb Intel Hum Comp* 2019; 10(10): 3795–3808.
30. Ahmad M, Protasov S, Khan AM, et al. Fuzziness-based active learning framework to enhance hyperspectral image classification performance for discriminative and generative classifiers. *PLoS One* 2018; 13(1): e0188996.
31. Bewley A, Ge Z, Ott L, et al. Simple online and realtime tracking. In: *2016 IEEE international conference on image processing (ICIP)*, Phoenix, AZ, 25–28 September 2016, pp.3464–3468. New York: IEEE.
32. Duffner S and Garcia C. PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. In: *2013 IEEE international conference on computer vision*, Sydney, NSW, Australia, 1–8 December 2013, pp.2480–2487. New York: IEEE.
33. Ahmad M, Ahmed I, Ullah K, et al. Robust background subtraction based person's counting from overhead view. In: *2018 IEEE 9th annual ubiquitous computing, electronics & mobile communication conference (UEMCON)*, New York City, NY, 8–10 November 2018, pp.746–752. New York: IEEE.
34. Ullah K, Ahmed I, Ahmad M, et al. Comparison of person tracking algorithms using overhead view implemented in OpenCV. In: *2019 9th annual information technology, electromechanical engineering and microelectronics conference (IEMECON)*, Jaipur, India, 13–15 March 2019, pp.284–289. New York: IEEE.
35. Girshick R. Fast R-CNN. In: *2015 IEEE international conference on computer vision*, Santiago, Chile, 7–13 December 2015, pp.1440–1448. New York: IEEE.
36. Khan FA, Butt AUR, Asif M, et al. Computer-aided diagnosis for burnt skin images using deep convolutional neural network. *Multimed Tool Appl*. Epub ahead of print 16 March 2020. DOI: 10.1007/s11042-020-08768-y.
37. Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector. In: *ECCV 2016: 14th European conference on computer vision*, Amsterdam, 11–14 October 2016, pp.21–37. Cham: Springer Nature.
38. Ahmad M, Shabbir S, Oliva D, et al. Spatial-prior generalized fuzziness extreme learning machine autoencoder-based active learning for hyperspectral image classification. *Optik* 2020; 206: 163712.
39. Krizhevsky A, Sutskever I and Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 2012; 25(2): 1097–1105.
40. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556v6, 2014.
41. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: *2015 IEEE conference on computer vision and pattern recognition*, Boston, MA, 7–12 June 2015, pp.1–9. New York: IEEE.
42. Choi W. Near-online multi-target tracking with aggregated local flow descriptor. In: *2015 IEEE international conference on computer vision*, Santiago, Chile, 7–13 December 2015, pp.3029–3037. New York: IEEE.

43. Li P, Wang D, Wang L, et al. Deep visual tracking: review and experimental comparison. *Pattern Recogn* 2018; 76: 323–338.
44. Wang N and Yeung DY. Learning a deep compact image representation for visual tracking. *Adv Neural Inf Process Syst* 2013; 1: 809–817.
45. Bochinski E, Eiselein V and Sikora T. High-speed tracking-by-detection without using image information. In: *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, Lecce, 29 August–1 September 2017, pp.1–6. New York: IEEE.
46. Gan Q, Guo Q, Zhang Z, et al. First step toward model-free, anonymous object tracking with recurrent neural networks. arXiv:1511.06425v2, 2015.
47. Milan A, Leal-Taixé L, Reid I, et al. Mot16: a benchmark for multi-object tracking. arXiv:1603.00831v2, 2016.
48. Bertinetto L, Valmadre J, Henriques JF, et al. Fully-convolutional Siamese networks for object tracking. In: *ECCV 2016: 14th European conference on computer vision*, Amsterdam, 11–14 October 2016, pp.850–865. Cham: Springer Nature.
49. Du D, Qi Y, Yu H, et al. The unmanned aerial vehicle benchmark: object detection and tracking. In: *ECCV 2018: 15th European conference on computer vision*, Munich, 8–14 September 2018, pp.375–391. Cham: Springer Nature.
50. Zhu P, Wen L, Du D, et al. VisDrone-VDT2018: the vision meets drone video detection and tracking challenge results. In: *ECCV 2018: 15th European conference on computer vision*, Munich, 8–14 September 2018.
51. Qi Y, Zhang S, Zhang W, et al. Learning attribute-specific representations for visual tracking. In: *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, Honolulu, HI, 27 January–1 February 2019, pp.8835–8842. Reston, VA: AIAA.
52. Ahmed I and Carter JN. A robust person detector for overhead views. In: *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, Tsukuba, Japan, 11–15 November 2012, pp.1483–1486. New York: IEEE.
53. Ahmad M, Ahmed I and Adnan A. Overhead view person detection using YOLO. In: *2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON)*, New York City, NY, 10–12 October 2019, pp.627–633. New York: IEEE.
54. Ahmad M, Ahmed I, Ullah K, et al. A deep neural network approach for top view people detection and counting. In: *2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON)*, New York City, NY, 10–12 October 2019, pp.1082–1088. New York: IEEE.
55. Ahmed I, Din S, Jeon G, et al. Exploring deep learning models for overhead view multiple object detection. *IEEE Internet Things J*. Epub ahead of print 5 November 2019. DOI: 10.1109/JIOT.2019.2951365.
56. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *2014 IEEE conference on computer vision and pattern recognition*, Columbus, OH, 23–28 June 2014, pp.580–587. New York: IEEE.
57. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: *2016 IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, 27–30 June 2016, pp.770–778. New York: IEEE.
58. Lin TY, Maire M, Belongie S, et al. Microsoft COCO: common objects in context. In: *ECCV 2014: 13th European conference on computer vision*, Zurich, 6–12 September 2014, pp.740–755. Cham: Springer Nature.
59. Jia Y, Shelhamer E, Donahue J, et al. Caffe: convolutional architecture for fast feature embedding. In: *Proceedings of the 22nd ACM international conference on multimedia*, Orlando, FL, 3–7 November 2014, pp.675–678. New York: ACM.