Human-Object Interaction: Application to Abandoned Luggage Detection in Video Surveillance Scenarios

Mihai Dogariu[1], Liviu-Daniel Ştefan[1], Mihai Gabriel Constantin[1] and Bogdan Ionescu[1]
[1] University Politehnica of Bucharest, Romania
Contact author e-mail: mdogariu@imag.pub.ro

***Abstract*—CCTV systems bring numerous advantages to security systems, but they require notable efforts from human operators in case of alarming events in order to detect the precise triggering moments. This paper proposes a system that can automatically trigger alarms when it detects abandoned luggage, detects the person that left the baggage and then tracks the suspicious person throughout the perimeter covered by a CCTV system. The system is based on Mask R-CNN and has been tested with several backbone configurations. We evaluate each subsystem independently on datasets specific for their task. The network model proves to be robust enough to carry on all of the three different tasks as demonstrated by tests.**

***Index Terms*—object detection, unattended baggage detection, video surveillance.**

I. Introduction

Modern days saw an increase in the rate of events that are intended to harm civilians through the means of terror acts. One such method is creating home-made explosive devices and deploying them in strategic places, such as crowded areas. It is customary for suspects to hide and transport these devices in ordinary packages, e.g., backpacks, suitcases, bags etc. Airports also pay a great deal of attention to unattended baggage as they have to deal with them on a regular basis. In such cases it is critical to find the baggage's owner in a short time to limit the potential threat and panic that it may cause.

Most such events are caught on camera so it becomes a retrieval problem for the authorities to extract the exact sequences concerning the event from the vast amount of feeds that the cameras capture at each moment. However, it becomes extremely tedious to manually analyze the entirety of the recorded sequences that capture the moment when an event occurred. This process can last from minutes up to several days, slowing down investigations in moments when time is of the essence. One such tragic event took place in Boston, Massachusetts in 2013, when it took the police officials more than 4 days to identify the suspects of a bomb planted in a public place. Having at hand an automated, real-time, detection system would have been definitively more effective.

This paper proposes a system that manages to quickly identify abandoned luggage, the suspected person who placed it there and track the person's path on a CCTV system. To the best of our knowledge, this is the first system to perform all these actions in an end-to-end system. The use-case of this system is represented by a human operator monitoring live feeds from a large number of surveillance cameras. We aim to help the operator by detecting unattended objects and finding the baggage owner in a short amount of time.

As Tripathi et al. [1] point out, most approaches in the literature [2]–[5] focus on semantically separating the background from the foreground and then tracking both static and moving objects. Unlike them, we propose a system composed of three modules: an object detection component, a suspect detection subsystem and a person re-identification component. In order to limit the resources needed to run the entire system we decided to use the feature extraction part from the object detection system to run person re-identification as well. Thus, our system successfully performs all three tasks without any additional cost. We evaluate the average precision and inference time of the object detection system on the MS-COCO dataset [6] and the accuracy of the person re-identification system on CUHK03 dataset [7].

The rest of the paper is structured as follows. In Section II we present the current progress in the object detection field, in Section III we explain our system's architecture, in Section IV we give details about the practical implementation, in Section V we discuss the results obtained with this algorithm and in Section VI we present the conclusions.

II. Related Work

The modern history of object detection started in 2014, when Girshick et al. [8] proposed the use of Region-based Convolutional Neural Nets (R-CNN) for accurate detection. This was done by proposing several regions of different shapes and sizes from an image and classifying their content with convolution layers. However, the feature extraction had to be run independently on each extracted region.

He et al. [9] proposed SPP Net, which acts as an improved R-CNN by introducing adaptively-sized pooling with spatial pyramid pooling and computing feature volume only once. Later progress from Girshick saw Fast R-CNN [10] bring a performance increase to R-CNN by first performing feature extraction and then proposing regions for object classification.

Faster R-CNN [11] improved on previous version by addin a region proposal network in charge for simultaneously predicting object bounds and objectness scores at each position. This

978-1-7281-5611-8/20/$31.00 ©2020 IEEE

netowrk uses already computed features from the detection network. Thus, its addition to the system's pipeline is almost cost-free.

Redmon et al. [12] proposed a different approach in YOLO (You Only Look Once). Their idea was to divide the image into a mesh of 7x7 pixels cells. Each cell goes through the object classifier and a regressor merges the cells such that objects are fully bounded by a rectangle box.

The Single Shot multibox Detector (SSD) [13] is a network that generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Moreover, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes.

Lastly, Mask R-CNN [14] extends Faster R-CNN to Instance Segmentation. Additionally to object detection, the network also predicts class-specific object masks. This approach also solves the misalignment problem of the RoI pooling layer by introducing the RoI align layer which uses bilinear interpolation to compute the exact values of the input features. They also explore different backbone architectures, such as Feature Pyramid Network (FPN) [15] and ResNet [16] to obtain state of the art results for object detection.

A similar work to ours was conducted by Intel [17], but they used Mask R-CNN only to detect objects. This limitation can be found throughout the literature, where systems are in charge solely with detecting abandoned objects. Our method, however, goes beyond this point, up to retrieving images of the unattended object and of the person that left it there.

III. System Architecture

In this section we provide information on the proposed system's architecture. As our approach relies on Mask R-CNN for object detection and feature extraction, we will briefly present its working principle. The Mask R-CNN architecture can be split into two main stages: the region proposal network (RPN) and the network head, as illustrated in Fig. 1. We explain these two stages separately.

In the region proposal stage, each image is passed through a backbone feature extractor. This is usually ResNet but it can be replaced by any other multi-stage architecture. These stages are marked as $C_1, ..., C_5$ in Fig. 1 and they represent feature maps of different sizes, computed in a bottom-up approach. These are also used in the feature pyramid to predict the feature maps $P_5, ..., P_2$ in a top-down approach. Next, several anchors, of different shapes and sizes are proposed for each feature map. An objectness score is assessed for each anchor and, if an object is detected, i.e., the objectness score is above a given threshold, it sends the current anchor to the RoI Align module, along with a RoI adjustment. The RoI Align block outputs several feature map regions where objects have been detected, concluding the region proposal stage.

The proposed regions now enter the second stage, the network head. This, in turn, is divided into 2 additional branches. One branch is in charge with computing the object mask by computing an individual mask for each type of object

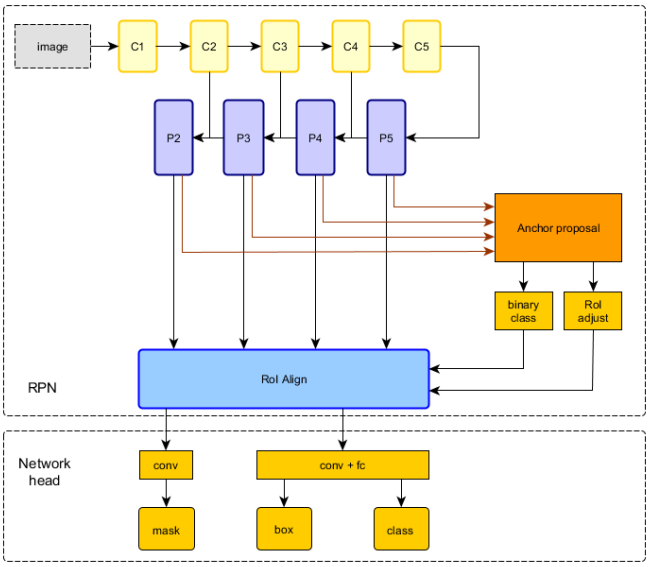

Fig. 1. Mask R-CNN architecture.

that the system is trained to detect. In parallel, the network head also performs classification and bounding box regression for the proposed region. After the class has been predicted, the network head selects the mask corresponding to the given class and it assigns it to the detected object.

As mentioned before, we used this pipeline to extract object features with the help of the RPN, which were used to drive all the 3 stages of our system. In addition, the masks, bounding boxes and class predictions from the network head were also used to completely describe the detected objects and offer a better visualization of the results.

IV. Implementation Details

Our system uses the Mask R-CNN architecture, trained on MS-COCO 2017 dataset [6], containing approximately 330k images, out of which 200k are labeled, spanning a total of 80 different object classes. Out of these classes, we selected only those that impact our system directly and discard the rest. We will now describe in detail the setup that we used for each of the 3 modules and how they interact.

A. Unattended Baggage Detection

The unattended baggage detection algorithm makes use of Mask R-CNN's object detection mechanism. We aim to detect only a subset of classes: 'person', 'backpack', 'handbag' and 'suitcase'. The last 3 classes have been grouped under a general class regarded as baggage. We labeled an object as unattended when there is no person in its immediate vicinity, meaning that the object's bounding box does not intersect any detected person's bounding box. Empirically, this proved to be a good compromise for the problem at hand.

Since our system is designed for security purposes, it is critical that no unattended object is missed by the automatic detection mechanism and we, therefore, favor a very low false

negative rate by setting a lower detection threshold of 0.5. We keep a standard threshold for non-maximum suppression of 0.7. Inference time is also important for our system given that its purpose is to assist first responders.

B. Suspect Detection

We then use the feature vector that Mask R-CNN computed and search for the exact abandoned baggage through all of the available images and rank these images in descending order of feature vector similarity, under the condition that there exists in the image a person whose bounding box intersects the baggage's bounding box. We test for similarity by using a simple Euclidean distance $d(\mathbf{f_q}, \mathbf{f_x}) = \sqrt{\sum_{i=1}^{N} (\mathbf{f_q}(i) - \mathbf{f_x}(i))^2}$. where $\mathbf{f_q}$ is the feature vector of the queried baggage, $\mathbf{f_x}$ is the feature vector of a baggage whose bounding box intersects that of a person's and N is the undimensional feature vector's size. Afterwards, we run a ranking of these distances and display the images where it is most likely that the abandoned baggage was detected in the presence of a person and deem this person to be a suspect.

C. Suspect Re-identification

In the next step, our system starts from the suspect that was just detected and searches for him in the dataset. We used the same procedure to search for the person as we used for the baggage. This time, however, we performed a per-camera ranking and obtain for each camera a set of images where the suspect was detected. This is motivated by the fact that we want to track this person's path throughout the surveilled perimeter and it is more helpful to see the most similar image of the subject that each camera managed to record, rather than the best matching images from the combined cameras which might end up to come from the same camera.

A very important aspect of our approach is that we are able to compare objects of different sizes. As seen in Figure 2 a person can have a very small size (30×60 pixels) on one camera and a very large size (210×500 pixels) on another camera, depending on the distance between them and the recording device. This dimension difference becomes irrelevant since we are comparing the object feature vectors, which are of fixed length.

Additionally, we created a small dataset for demonstration purposes. We gathered 1 hour of images recorded by our research center's CCTV system. We restricted the observed area to the basement, ground floor and the exterior of the building. The motivation is that we wanted to capture all possible entrances, the surrounding perimeter and some additional indoor information. We established a scenario where one person would abandon a backpack on the hallway and leave. Several other people carrying backpacks were captured in this dataset. We downsampled the recordings to only one frame/s and, since the cameras are equipped with motion sensors and only record when they detect movement, our gathered dataset is very small (120 images). In addition, we created a demo graphical user interface to aid a human operator.

TABLE I
PERFORMANCE OF DIFFERENT OBJECT DETECTION MODELS

Backbone	Bbox AP@IoU=0.75	Inference time (s/image)
R50-C4_1x	35.7	0.392
R50-DC5_1x	37.3	0.408
R50-FPN_1x	37.9	0.228
R50-C4_3x	38.4	0.398
R50-DC5_3x	39.0	0.396
R50-FPN_3x	**40.2**	**0.231**
R101-C4_3x	41.1	0.482
R101-DC5_3x	40.6	0.474
R101-FPN_3x	42.0	0.308
X101-FPN_3x	43.0	0.591

V. RESULTS

We performed tests on several backbone architectures and report average precision for Intersection over Union (IoU) score higher than 0.75[1] along with the time it takes for each architecture to process one image in Table I. The backbone architectures should be read as follows:

- R/X: means that ResNet or ResNeXt, respectively has been used as an underlaying architecture;
- 50/101: the architecture consists of 50 or 101 layers;
- C4/DC5/FPN: 3 different backbone combinations as follows. *C4*: uses ResNet conv4 backbone with conv5 network head, the same as in Faster R-CNN. *DC5*: uses ResNet conv5 backbone with dilated convolutions in conv5 layer. *FPN*: uses ResNet + FPN backbone with standard convolutions.
- 1x/3x: models trained with a different number of COCO epochs (~12 or ~37, respectively). 1x models have significantly lower performance than their 3x counterparts as it can be seen in Table I.

We obtained the presented inference times while performing the detection on a single NVIDIA QUADRO M4000 GPU. We consider that in the given circumstances it is better to opt for a model which sacrifices a part of the detection accuracy in favor of a faster inference time. The detection accuracy loss can be overcome by setting a lower detection threshold to force additional proposals and decrease the false negative rate. Decreasing inference time is, however, far more difficult. In our use-case a fast response is a critical aspect of the system. Therefore, we select the R50-FPN_3x as our go-to model in the proposed system.

The person re-identification component was tested on the CUHK03 dataset and obtained a top-1 accuracy of 70.8%. The same technique was used by Xiao et al [18]. In addition to the ResNet50 architecture they also tested an Inception version [19], but the higher number of parameters also made the inference slower, which was detrimental for our system.

During our demonstration we managed to capture all events that are of interest: we could successfully detect the abandoned baggage, the person that left it there and then detect that

[1]Detection performance from Facebook Research Object Detection MSCOCO baseline: https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md.

Fig. 2. From left to right: abandoned object that triggered an alarm, detected suspect leaving the baggage, suspect leaving the building, suspect interacting with another person. The person in the third image is 210×500 pixels in dimension, whereas in the fourth image it is of 30×60 pixels.

person's presence on individual cameras. Results can be seen in Figure 2. Furthermore, we managed to extract important moments such as when the person entered the building while carrying on the backpack and when the person left the building, without the backpack. This proves very useful in narrowing down the time interval during which the suspected person was inside the building. Moreover, we can see the people that the suspected person interacted with, which could increase the ramification of our scenario even further.

VI. Conclusions

In this paper we presented an unattended object detector that can be deployed on CCTV systems. Our approach is composed of 3 modules, each designed to perform a different task: unattended object detection, detect the object's owner and find that person's presence on the CCTV cameras. We used Mask R-CNN to perform the detection and computed similarities on the extracted feature vectors that the network extracted. We gathered a small dataset for our experiment simulating one of the practical use-cases, built a user interface for operators and proved that the system works as intended in the proposed use-case. We evaluated the system and chose the architecture that offered the best trade-off between performance and inference time.

VII. Acknowledgements

This work has been funded by the Ministry of Innovation and Research, UEFISCDI, project SPIA-VA, agreement 2SOL/2017, grant PN-III-P2-2.1-SOL-2016-02-0002, and by the Operational Programme Human Capital of the Ministry of Europe Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125.

References

[1] R. K. Tripathi, A. S. Jalal, and S. C. Agrawal, "Abandoned or removed object detection from visual surveillance: a review," *Multimedia Tools and Applications*, vol. 78, no. 6, pp. 7585–7620, 2019.

[2] S. Foucher, M. Lalonde, and L. Gagnon, "A system for airport surveillance: Detection of people running, abandoned objects and pointing gestures," vol. 8056, 05 2011.

[3] Q. Fan, P. Gabbur, and S. Pankanti, "Relative attributes for large-scale abandoned object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2736–2743, 2013.

[4] J. Ferryman, D. Hogg, J. Sochman, A. Behera, J. A. Rodriguez-Serrano, S. Worgan, L. Li, V. Leung, M. Evans, P. Cornic, *et al.*, "Robust abandoned object detection integrating wide area visual surveillance and social context," *Pattern Recognition Letters*, vol. 34, no. 7, pp. 789–798, 2013.

[5] P. Foggia, A. Greco, A. Saggese, and M. Vento, "A method for detecting long term left baggage based on heat map.," in *VISAPP (2)*, pp. 385–391, 2015.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[7] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 152–159, 2014.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[10] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[17] A. I. D. N. N. I. A. Builders, "Unattended baggage detection using deep neural networks in intel® architecture," tech. rep., July 2014.

[18] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3415–3424, 2017.

[19] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1249–1258, 2016.