MOVING-CAMERA VIDEO SURVEILLANCE IN CLUTTERED ENVIRONMENTS USING DEEP FEATURES

Bruno M. Afonso, Lucas P. Cinelli, Lucas A. Thomaz, Allan F. da Silva, Eduardo A. B. da Silva, Sergio L. Netto

Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
Email: {bruno.afonso, lucas.thomaz, allan.silva, lucas.cinelli, eduardo, sergioln}@smt.ufrj.br

ABSTRACT

This paper deals with the challenging problem of visual anomaly detection in a cluttered environment using videos acquired with a moving camera. The anomalies considered are abandoned objects. A new method is proposed for comparing two videos (an anomaly-free reference video and a target one possibly with anomalies) by using convolutional neural networks as feature extractors for a subsequent anomaly-detection stage using a classifier. Two classifier strategies are considered, namely a fully-connected neural network and a random forest algorithm. Results for a comprehensive abandoned object database acquired with a moving camera in a cluttered environment indicate that the proposed architecture can match even the state-of-the-art algorithms in terms of object-detection performance, with a reduction in processing time of 80%.

Index Terms— Video Surveillance, Deep Neural Networks, Background Subtraction, Image Classifiers.

1. INTRODUCTION

Video surveillance systems are becoming commonplace. With each passing day we become increasingly aware of the amount of public places that use such systems primarily for security purposes. Most of them have been designed to operate nonstop, and therefore they generate huge amounts of data that would be virtually impossible for humans to keep constant watch and analyze properly. To help cope with this issue, multiple solutions have been designed to process the data automatically from the recorded videos and extract essential information regarding the detection of anomalies [1][2].

The most common surveillance setups involve installing multiple fixed cameras scattered on the environment. Sometimes, however, this approach might not be ideal due to the increased cost of the equipment. One of the possible ways to circumvent this issue is to install a camera on a moving platform that scans the whole surveilled area. Throughout the years many solutions have been proposed to solve the problem of detecting abandoned objects and other video anomalies in videos acquired with moving cameras in different scenarios. For example, in [3] a camera is mounted on a drone to detect moving objects in real-time, while [4] employs multiple videos featuring different scenarios where a moving camera is used together with computer vision methods to detect movement patterns.

Although there are algorithms based on image processing techniques that are successful in achieving a good detection performance, as discussed in [5] most of them are not capable of processing huge amounts of data in real-time. Also, as shown in [3], some solutions require probabilistic models that are based on both spatial and temporal consistency to achieve their results, leading not only to a large processing time but also to possible classification mistakes in case of abrupt changes to the observed environment.

In this paper we propose a new architecture based on deep-learning features to detect moving-camera video anomalies in a cluttered industrial environment. In this new system a convolutional neural network is used to extract deep features of (anomaly-free) reference and target video frames which are then fed to a classifier for detecting possible abandoned objects.

To describe the proposed algorithm, the remainder of this paper is organized as follows: Section 2 introduces the proposed deep-feature based architecture for anomaly detection in surveillance videos. In Section 3 the experimental results are presented and compared with the ones from state-of-the-art image processing algorithms. Finally in Section 4 the author's conclusions are presented.

2. PROPOSED ANOMALY DETECTOR BASED ON DEEP FEATURES

Recent developments in machine learning have shown through multiple experiments that the weights of a convolutional neural network (CNN) can be adjusted to detect particular image patterns depending on the network size (number of layers and number of neurons in each layer), the proper training algorithm, and available database [6]. In these systems the initial (or shallower) layers tend to detect more generic image patterns, such as color blobs, edges, etc., while at the deeper network layers the weights tend to detect more object-specific patterns, such as a cat's face or the shape of its head. Still according to [6], the first few layers of the network might not detect most patterns related to the main points of interest on the image, represented by salient objects in the frame, but they may be still able to detect some features that are correlated to the image background. This particular trait is relevant to the addressed problem of detecting unknown abandoned objects. In fact, since the surveilled environment is constantly recorded by the moving cameras, one can deduce that it is easier to detect features related to the environment and attribute the foreign features as anomalies, in this case abandoned objects.

To help distinguish among these foreign features, the concept of target ($\mathcal{T}$) and reference ($\mathcal{R}$) frames is introduced. A $\mathcal{T}$ represents frames that may or may not have anomalies and will be analyzed to reveal the presence of a possible abandoned object. An $\mathcal{R}$ represents frames containing only the background, considered to be the normal state of the environment. After obtaining the features for both images, one can separate the features of $\mathcal{T}$ that do not belong to $\mathcal{R}$.

This paper introduces an algorithm that extracts the features gen-

Thanks to CAPES, CNPq and FAPERJ for funding. We also acknowledge funding from CAPES PRO-DEFESA: N^{o} 23038.009094/2013-83 and NVIDIA for providing the hardware used in this work.

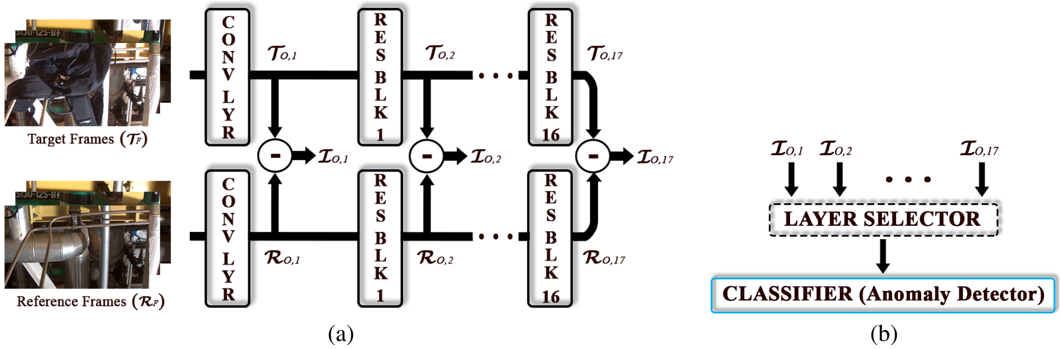

Fig. 1. Block diagram of deep-feature extraction for the proposed anomaly-detector architecture: (a) The pre-trained ResNet50 network analyzes a pair of reference-target video frames, and the deep-feature outputs for 17 distinct ResNet 50 layers (16 from residual blocks (RES BLK) and 1 from the first convolutional layer (CONV LYR)) are extracted and then subtracted. (b) The "LAYER SELECTOR" block selects which subtracted deep-feature output($\mathcal{I}_{O,i}$) is used as the input for the classifier (anomaly-detector).

erated from a pair of target and reference frames in multiple convolutional layers from a deep CNN and trains classifiers to detect the presence of possible abandoned objects in the target frame, as illustrated in Figure 1. The CNN architecture employed here is a pre-trained ResNet50, as described in [7], which includes a large number of convolutional weights and implements the residual blocks with bottlenecks, allowing the algorithm to extract more varied features through its layers [8]. Thanks to its ability to identify 1000 different classes of objects, the ResNet50 network provides a robust filter selection to detect as many features as possible for several object types and the background, as discussed in [6]. It is important to note that transfer learning is justified here, since a complete training procedure for the deep-learning network would require huge amounts of annotated data which are quite hard to obtain due to the cluttered-environment constraint imposed in our application.

The ResNet50 can generate multiple different feature outputs depending on which layer one wants to extract them from. In our case we chose to get the features from the entry of each convolutional block in the network, generating 17 sets of features. The proposed algorithm gathers all the feature outputs from one of the residual blocks (the only exception being the last set of features that is the input of the average pooling layer or, equivalently, the output of the last residual block) and feed them into a classifier. An average pooling layer is inserted at the end of each layer to reduce the amount of parameters that the classifier analyzes [8]. The array A_P containing the average polling kernel size for each CNN layer is given below:

$$A_P = \{21{,}28{,}28{,}28{,}21{,}21{,}21{,}21{,}14{,}14{,}14{,}14{,}14{,}14{,}7{,}7{,}7\}. \qquad (1)$$

Let $\mathcal{T}_F$ and $\mathcal{R}_F$ be two sets of matching target and reference frames, respectively. The algorithm does a forward-pass through the CNN resulting, for layer i, $i = 1, \ldots, 17$, in a pair of feature output arrays represented by $\mathcal{T}_{O,i}$ and and $\mathcal{R}_{O,i}$. A feature discrepancy is then determined as

$$\mathcal{I}_{O,i} = \mathcal{T}_{O,i} - \mathcal{R}_{O,i} \qquad (2)$$

and can be used as the classifier input to determine the presence or not of an abandoned object.

The architecture of the detection algorithm is made with two classifiers in mind: First, a multilayer perceptron (MLP) network is proposed with 2 hidden layers, the first with 50 neurons and the second one with 1600. The ReLU [9] activation is used after each one of the hidden layers, and for the output layer the sigmoid [10] activation is employed. The choice of the latter is due to the loss function used in this network, which is the binary cross-entropy, a special case for the log (cross-entropy) loss [11] using only one class on the output to determine if the input belongs to that class or not. In our experiments we empirically set an MLP learning rate of 2×10^{-3} and use the AdaMax optimizer function, which is a variation for the standard Adam optimizer [12], we also applied an L2 regularization of 5×10^{-3} which is standard in the deep learning literature. The second classifier considered here uses the random forest algorithm (RFA) [13]. This algorithm is more capable of properly classifying an input which contains more randomized or non-correlated features. We compare the performance of this alternative classifier to the one of a standard neural network. In our experiments, the RFA has a maximum of 100 trees. Even though one could consider this algorithm to be more complex than a neural network, it is easier to implement and train on a computer with just a modern CPU, in contrast with the MLP that requires the use of GPU processing to be trained in a viable time depending on the data volume.

The specific parameter values used in the configuration of the classifiers were obtained by testing several combinations of configurations for these classifiers. The MLP configurations were tested with 0, 1 or 2 hidden layers, where each layer contained a random number of neurons between 50 and 3000. The range of testing for the learning rate was from 10^{-2} to 10^{-5}. As for the RFA classifier, its configurations were tested with their numbers of trees ranging from 10 to 300 trees.

Let $\mathcal{K}$ be an array that cointains the labels for $\mathcal{I}_{O,i}$, such as:

$$\mathcal{K}_i = \begin{cases} 1, & \text{if } \mathcal{T}_i \text{ has an abandoned object} \\ 0, & \text{otherwise} \end{cases}. \qquad (3)$$

The output generated by both classifiers is an array Λ with predictions in the form of a probability value between 0 and 1 for each frame $\mathcal{T}_i$. To determine if the value from the output represents the presence of an abandoned object, a threshold *Th* is applied to Λ, creating the classification array $\Lambda_T$such that

$$\Lambda_{Ti} = \begin{cases} 1, & \text{if } \Lambda_i > \mathit{Th} \\ 0, & \text{otherwise} \end{cases}. \qquad (4)$$

This procedure is carried out first on a validation frame set, whose labels $\mathcal{K}$ are known beforehand. *Th* is a threshold between 0 and 1 that generates a Λ_T with the least amount of errors when compared to $\mathcal{K}$, as determined by the validation set during the training phase of the algorithm. This chosen *Th* is then used to generate a classification array Λ_T for the test set of frames. This array determines if each frame $\mathcal{T}_i$ contains an abandoned object or not.

3. EXPERIMENTAL RESULTS

3.1. Video Database

The proposed algorithm was tested using the VDAO (Video Database of Abandoned Objects), described in [14] and available at [15]. The database consists of multiple videos recorded in a cluttered industrial environment with a camera attached to a moving platform guided by a fixed rail, which makes it move according to a back and forth pattern. Several different objects are placed along the same path covered by the field of view of the camera and for every object combination there are two different lighting conditions. Every video that features at least one abandoned object is labeled as a target video, and each one of those has a correspondent video labeled as a reference video, which features no abandoned object.

A subset of the VDAO database composed of 59 pairs of target/reference videos was chosen to evaluate the performance of the proposed algorithm by comparing it to other methods. The aforementioned videos are the same ones used in [16] and may be accessed at [17]. Due to the size constraint of the VDAO, the 59-video set was split onto nine disjoint test subsets, each one containing all pairs of target/reference videos of a given object. We trained nine networks using one video group for test, and the eight groups for training, while separating 10% of the frames for parameter validation. In this way all the videos were used for testing and an object is never used simultaneously for training and testing. The videos were previously synchronized using the algorithm described in [18].

Although the VDAO videos were recorded at a 1280x720 resolution, the frames of both training and test batches were spatially downsampled to a 640x480 resolution and downsampled in time by 4. The mini-batch size for both training and testing was 32 frames. In the dataset, 79% of the frames contain an abandoned object, while the remaining 21% are random background frames without abandoned objects.

3.2. Performance Metrics

The performance for the classifiers and the other methods was measured with the following metrics: (i) True Positive (TP) detection rate, where a TP occurs when an object is detected by the classifier in a frame from the test batch that features an abandoned object; (ii) False Positive (FP) detection rate, where an FP occurs when an object is detected by the classifier in a frame from the test batch which does not feature an abandoned object. Also, since the metrics described in [5] are different from the ones used on this paper, for a fair comparison we have executed in our database the software from the methods whose results are given in [5] (STC-MC [16], DAOMC [19], MCBS [20] and mcRoSuRe-A [5]) to find TP and FP rates. By using the whole frame as a bounding box for the abandoned object in the set of frames that contain anomalies, a TP occurs if at least one pixel is detected in a frame that contains an object. In addition, if a frame does not contain any abandoned object, an FP occurs if there is at least one pixel detected in the frame. Finally, we consider the DIS parameter that is defined as:

$$\mathrm{DIS} = \sqrt{(1 - \mathrm{TP})^2 + \mathrm{FP}^2}, \tag{5}$$

which can be interpreted as the minimum distance of an operating point to the point of ideal behaviour (TP = 1 and FP = 0).

3.3. Detection Results

The detection results for the proposed architecture are shown in Table 1 for the several different CNN layer depths that provided the best DIS results when input to the classifier (see Figure 1). In that table, the TP and FP rates represent the mean of the TP and FP rates for all the 59 videos. According to these results, we can see that most of the features that deliver the best DIS results are extracted from the initial layers (first four) of the CNN. This suggests that, when using pre-trained deep features for the application at hand, it is not necessary to have a very deep convolutional architecture. The best results for both classifiers illustrate well that statement, since they have their best abandoned object detection results when they are fed by the output of the third and fourth convolutional layers.

Table 1. Mean and standard deviation for the FP and TP and the resulting DIS metric of the features extracted from the six first layers for both proposed architectures (MLP and RF classifiers). "Layer" indicates which CNN layer output features are input to the classifier.

	CNN + RF			CNN + MLP		
Layer	FP	TP	DIS	FP	TP	DIS
1	0.32	0.68	0.45	0.39	0.70	0.49
2	0.30	0.73	0.40	0.36	0.70	0.47
3	0.24	0.69	0.39	0.34	0.71	0.44
4	0.25	0.74	0.36	0.28	0.66	0.44
5	0.27	0.68	0.42	0.40	0.76	0.47
6	0.27	0.68	0.42	0.39	0.73	0.47

By inspecting these results one can notice that the CNN + RF method presents better results than those of CNN + MLP. One can also observe that the MLP and RF classifiers have their best results with the same group of features. This suggests that these two classifiers deal similarly with features output from the CNN.

Since the best results come from features extracted by early layers of the CNN it seems that the high level features extracted by the CNN are not very suitable for this application. This might be explained by the fact that the early layers of this pre-trained network extract general geometric features from the image, while the deeper layers extract more class specific features that are linked with the original classification task of the ResNet.

Table 2. Average detection comparison of proposed methods with that of STC-MC [16], DAOMC [19], MCBS [20], and mcRoSuRe-A [5] methods for the 59 single-object videos in [17].

Method	TP	FP	DIS
STC-mc [16]	0.48	0.41	0.66
DAOMC [19]	0.89	0.46	0.47
MCBS [20]	**0.99**	0.98	0.98
mcRoSuRe-A [5]	0.95	0.37	0.37
CNN + MLP	0.66	0.28	0.44
CNN + RF	0.74	**0.25**	**0.36**

In the assessment of the detection performance of the proposed methods relative to the state-of-the-art for the application at hand, we used the best configuration for both classifiers in Table 1. The state-of-the-art methods used in this comparison are: spatio-temporal composition for moving-camera detection (STC-mc) [16], detec-

Table 3. Time (in seconds) comparison between the digital image processing methods and the machine learning methods when analyzing seven out of the 59 videos from the VDAO database.

Object	STC-mc	DAOMC	MCBS	mcRoSuRe-A	CNN + MLP (worst case)	CNN + RF (worst case)	CNN + MLP (best case)	CNN + RF (best case)
Dark Blue Box 1	433	265	50924	52	9.4	9.5	6.2	6.3
Towel	345	280	50403	38	9.6	9.7	6.5	6.5
Shoe	542	293	50427	38	9.8	9.9	6.6	6.8
Pink Bottle	415	280	50170	38	9.3	9.4	6.2	6.3
Camera Box	448	299	50238	45	9.8	9.9	6.7	6.8
Dark Blue Box 2	221	289	51740	38	9.7	9.8	6.7	6.8
White Jar	248	282	49901	36	9.4	9.6	6.3	6.4
Average	379	284	50543	41	9.6	9.7	6.4	6.5

tion of abandoned objects with a moving camera (DAOMC) [19], moving-camera background subtraction (MCBS) [20], and moving camera robust subspace recovery - accelerated (mcRoSuRe-A) [5]. The results in Table 2 show that the CNN + RF method is competitive with the state-of-the-art methods when one considers the DIS metric. Even though the proposed methods are based on deep features, they might not achieve TP values as high as those of the other methods, but on the other hand, one of their key advantages is their low FP value. This means that the algorithm is less prone to making false evaluations when it comes to frames without an abandoned object.

In the proposed method only a simple temporal consistency was used to improve the results. We applied a median filter with window size five to the output detection of each video, this way consecutive frames on a video tend to present the same positive or negative detection.

3.4. Execution-Time Results

Table 3 exibits the execution times for both the state-of-the-art and the proposed methods. The videos used for this benchmark are the first seven videos from [17]. All the training for the proposed algorithms, as well as the performance tests were carried out on a computer with the following specifications: a CPU Intel Core i7-4790K @ 4.00GHz, with 32GB of RAM and a discrete NVIDIA Geforce GTX Titan Xp with 12GB of VRAM, running MATLAB©2015a and Tensorflow v.1.1.0 API.

For the proposed methods, the execution time is measured as the elapsed time between a forward-pass through the CNN from a test batch input and the creation of the Λ_T output. The training time for the classifiers is not taken into account in these results, since the training in such applications is done offline. Regarding the depth of the CNN, Table 3 shows two different results for each type of classifier. According to Table 1, layer 8 will be used as the worst case scenario for a time performance and layer 1 as the best case scenario. From these values, it is clear that the CNN + MLP method is the fastest one among the proposed methods, with the CNN + RF being marginally slower (due to the CPU implementation used to evaluate the test sets). The results show that the CNN + MLP method is at least three times faster than the mcRoSuRe-A even for the worst case scenario for the CNN.

4. CONCLUSION

This paper introduces an algorithm that detects abandoned objects in a cluttered environment using a moving camera. The proposed method is a combination of different machine-learning algorithms that extract image features from multiple intermediate layers of a deep residual neural network [8] using pre-trained weights. This strategy avoids the need of a huge image database to train a deep convolutional neural network, so that the training can be focused on smaller classifiers.

The proposed method presents competitive overall results with the state-of-the-art in abandoned object detection in a cluttered environment using a moving camera, with the advantage of having smaller processing times. Results obtained using the VDAO database show that the RF classifier method is able to attain similar performance compared to state-of-the-art image processing techniques while using 80% less processing time on average. The algorithm presented the closest result to the ideal scenario on average for all test videos, achieving 0.74 overall true positive and 0.24 of false positive detections.

Two different classifier algorithms were considered, namely the multilayer perceptron (MLP) and the random forest algorithm (RF). We verified that the RF classifier provides comparable results with state-of-the-art using simple features from the shallower layers in the network. The other experimented classifier, namely the MLP, attained slightly worse results, yet having similar results to the state-of-the-art. This might be an indication that the features obtained from deeper layers of the CNNs pretrained on a general classification task, are not adequate for the application at hand. However, the good results obtained using the RF classifier, which can be easily trained on CPU processors in opposition to the GPU processors needed to train the MLP, and the few first convolutional layers highlight its capability as a good classifier and suggest that the proposed architectures should be further investigated. A possible path to improve those results should be using features extracted from a network trained with data that is more suitable to our application.

5. REFERENCES

[1] L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, S. L. Netto, X. Bian, and H. Krim, "Abandoned object detection using operator-space pursuit," in *IEEE International Conference on Image Processing*, Quebec, Canada, Sept. 2015, pp. 1980–1984.

[2] C. Cuevas, R. Martínez, and N. García, "Detection of stationary foreground objects: A survey," *Computer Vision and Image Understanding*, vol. 152, pp. 41 – 57, Nov. 2016.

[3] C. Huang, P. Chen, X. Yang, and K.-T. T. Cheng, "REDBEE: A visual-inertial drone system for real-time moving object detection," in *IEEE/RSJ International Conference on In-*

telligent Robots and Systems, Vancouver, Canada, Sept. 2017, pp. 1725–1731.

[4] M. Narayana, A. Hanson, and E. Learned-Miller, "Coherent motion segmentation in moving camera videos using optical flow orientations," in *IEEE International Conference on Computer Vision*, Sydney, Australia, Dec. 2013, pp. 1577–1584.

[5] L. A. Thomaz, E. Jardim, A. F. da Silva, E. A. B. da Silva, S. L. Netto, and H. Krim, "Anomaly detection in moving-camera video sequences using principal subspace analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, pp. 1003–1015, March 2018.

[6] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*. 2014, pp. 818–833, Springer.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 2016, pp. 770–778.

[9] H. K. Vydana and A. K. Vuppala, "Investigative study of various activation functions for speech recognition," in *National Conference on Communications*, Chennai, India, Mar. 2017, pp. 1–5.

[10] C. H. Tsai, Y. T. Chih, W. H. Wong, and C. Y. Lee, "A hardware-efficient sigmoid function with adjustable precision for a neural network system," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 11, pp. 1073–1077, Nov. 2015.

[11] L. Liu and H. Qi, "Learning effective binary descriptors via cross entropy," in *IEEE Winter Conference on Applications of Computer Vision*, Santa Rosa, USA, Mar. 2017, pp. 1251–1258.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations*, San Diego, USA, 2015, pp. 1–13.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[14] A. F. da Silva, L. A. Thomaz, G. Carvalho, M. T. Nakahata, and E. Jardim, "An annotated video database for abandoned-object detection in a cluttered environment," in *International Telecommunications Symposium*, Sao Paulo, Brazil, Aug. 2014, pp. 1–5.

[15] *VDAO - Video Database of Abandoned Objects in a Cluttered Industrial Environment*, Available: http://www02.smt.ufrj.br/~tvdigital/database/objects/.

[16] M. T. Nakahta, L. A. Thomaz, A. F. da Silva, E. A. B. da Silva, and S. L. Netto, "Anomaly detection with a moving camera using spatio-temporal codebooks," *Multidimensional Systems and Signal Processing*, pp. 1–30, Mar. 2017.

[17] *200-Frame Excerpts From VDAO Database. [Online]*, Available: http://www02.smt.ufrj.br/~tvdigital/database/research/.

[18] A. F. da Silva, L. A. Thomaz, S. L. Netto, and E. A. B. da Silva, "Online video-based sequence synchronization for moving camera object detection," in *IEEE International Workshop on Multimedia Signal Processing*, Luton, UK, Oct. 2017, pp. 1–6.

[19] H. Kong, J.-Y. Audibert, and J. Ponce, "Detecting abandoned objects with a moving camera," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2201–2210, Aug. 2010.

[20] H. Mukojima, D. Deguchi, Y. Kawanishi, I. Ide, H. Murase, M. Ukai, N. Nagamine, and R. Nakasone, "Moving camera background-subtraction for obstacle detection on railway tracks," in *IEEE International Conference on Image Processing*, Phoenix, USA, Sept. 2016, pp. 3967–3971.