# Real-Time Deep Learning Method for Abandoned Luggage Detection in Video

Sorina Smeureanu*‡, Radu Tudor Ionescu*‡
*University of Bucharest, 14 Academiei, Bucharest, Romania
‡SecurifAI, 24 Mircea Vodă, Bucharest, Romania
E-mails: sorinasmeureanu@gmail.com, raducu.ionescu@gmail.com

***Abstract*—Recent terrorist attacks in major cities around the world have brought many casualties among innocent citizens. One potential threat is represented by abandoned luggage items (that could contain bombs or biological warfare) in public areas. In this paper, we describe an approach for real-time automatic detection of abandoned luggage in video captured by surveillance cameras. The approach is comprised of two stages: $(i)$ static object detection based on background subtraction and motion estimation and $(ii)$ abandoned luggage recognition based on a cascade of convolutional neural networks (CNN). To train our neural networks we provide two types of examples: images collected from the Internet and realistic examples generated by imposing various suitcases and bags over the scene's background. We present empirical results demonstrating that our approach yields better performance than a strong CNN baseline method.**

## 1. Introduction

Recent terrorist attacks in major cities around the world have brought many casualties among innocent citizens[1]. In this context, we need systems able to provide real-time alerts about potential threats, so that people can enjoy life in their own cities without being afraid of terrorist attacks. In this paper, we deal with the problem of abandoned luggage detection. When an object, usually a suitcase or a bag, is left unattended in a public area, e.g. airport terminal or train station, it represents a security threat because the abandoned object may contain dangerous items such as explosives or biological warfare. For this reason, the abandoned object must be removed immediately from the public area by authorized personnel. To this end, we propose a system that is able to automatically detect abandoned luggage items in real-time by analyzing the video from surveillance cameras. Our approach is comprised of two stages. In the first stage, we detect static objects based on background subtraction and motion estimation, as most related works [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. In the second stage, we apply a cascade of *convolutional neural networks* (CNN) based on the GoogLeNet [16] architecture to distinguish between abandoned luggage items and other objects, e.g. persons standing still. To obtain robust CNN models, we provide two types of examples: images collected from the Internet and realistic examples generated by imposing various suitcases and bags over the scene's background. To our best knowledge, we are the first to train a cascade of convolutional neural networks for abandoned luggage recognition.

The paper is organized as follows. Related work on abandoned luggage detection is presented in Section 2. Our learning framework is described in Section 3. The abandoned luggage detection experiments are presented in Section 4. Finally, we draw our conclusions in Section 5.

1. https://en.wikipedia.org/wiki/Boston_Marathon_bombing

## 2. Related Work

Most works for abandoned objects detection use background subtraction as a low-level preliminary step [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] to detect foreground regions or objects, although Smith et al. [17] start directly by tracking multiple objects in the scene using trans-dimensional Markov Chain Monte Carlo (MCMC). After background subtraction, some works aim to reduce false positive detections using object tracking and classification methods [1], [2], [7], [8], [9], [10], [11], while other approaches employ edge detection frameworks [14], [18] or generative models [6]. Some frameworks model the abandoned objects detection problem as finite state automata [9], [12], [13], while others use temporal logic-based inference as an alternative solution [1], [5], [19]. Different from all other works, Kong et al. [20] try to detect abandoned objects on the road using a moving camera. Ferryman et al. [19] present a threat assessment algorithm that combines the concept of ownership with automatic understanding of social relations in order to infer abandonment of objects. Similar to Porikli et al. [3], Lin et al. [12], [13] combine short-term and long-term background models to extract foreground objects. Szwoch [18] describes an algorithm for the detection of stable regions by comparing these regions with the contours of moving objects. Dahi et al. [14] present a method based on static edge detection and classification. Pham et al. [15] propose a two-stage method for unattended object detection. The first stage tries to detect all possible abandoned objects, preventing false negatives. In the second stage, their method reduces false alarms by using similarity matching between first-stage candidates and the background model. Different from all previous methods, we use a cascade of convolutional neural networks in the second

ISBN 978-90-827970-1-5 

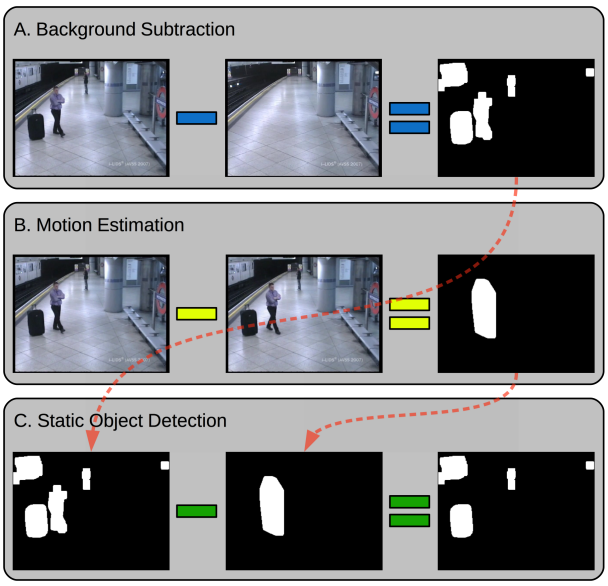



Figure 1. Static object detection (SOD) pipeline used in the first stage of our approach for abandoned luggage detection. The pipeline is comprised of three steps. Step A: foreground estimation based on background subtraction. Step B: motion estimation based on subtracting temporally-close video frames. Step C: static object detection based on subtracting the motion mask from the foreground mask. In this particular example, the static objects are a subway train, a suitcase and two persons. Best viewed in color.

stage to recognize abandoned versus attended luggage items or other objects. For a complete review of recent works on abandoned object detection, the reader is referred to the survey of Cuevas et al. [21].

## 3. Method

We propose a two-stage approach for abandoned luggage detection. In the first stage, we aim to detect all static objects in the scene. Among these objects however, there are many false positive detections, i.e. objects that do not represent abandoned luggage, e.g. persons standing still or even attended luggage items. Therefore, a second stage is necessary to distinguish between abandoned luggage items and other objects. We next describe the first stage in Section 3.1 and the second stage in Section 3.2.

### 3.1. Static Object Detection

Our *static object detection* (SOD) approach is comprised of two components, namely background subtraction and motion estimation. The entire pipeline for the first stage is illustrated in Figure 1. The pipeline is based on three sequential steps: A, B and C. On each video frame, we first apply a standard method for background subtraction (step A) that simply subtracts the estimated background from each video frame. The resulted foreground mask is further processed by applying erosion and dilation filters. In the end, the foreground mask contains both static and moving objects. In order to find the moving objects in the scene, we estimate the motion (step B) by subtracting frames that are $5$ frames apart from each other. This will provide a mask representing the contour of moving objects. To fill the objects we apply erosion and dilation filters on the motion mask. We then apply a standard algorithm to find the connected components. For each connected component, we compute the convex hull. In the end, the motion mask contains the moving objects as convex connected components. To single out the static objects in the scene, we subtract the motion mask from the foreground mask and we obtain the static pixels mask (step C). On the static pixels mask, we compute the connected components. The resulted connected components represent the static objects in the scene. Finally, we determine the bounding box for each static object and extract the corresponding sub-image. The resulted images represent individual static objects. We track the static objects over multiple frames. We consider that two bounding boxes belong to the same track if the *intersection over union* (IoU) is greater than $0.5$. Finally, the static object tracks are further processed in the second stage to determine if the objects are indeed abandoned luggage items.

### 3.2. Abandoned Luggage Recognition

In the second stage, we employ a *cascade of convolutional neural networks* (CCNN) to recognize abandoned luggage in the object tracks resulted from the first stage. We employ the GoogLeNet [16] architecture for both convolutional neural networks. We start from a GoogLeNet model that is pre-trained on the ILSVRC benchmark [22], and train only the last layer.

The first neural network is trained to recognize images containing luggage items, e.g. suitcases, hand bags, backpacks and so on. However, some of the test samples labeled as positive by the CNN may contain luggage items that are not abandoned. For instance, a person that stands still next to its own luggage (a common situation when one waits in a train station or airport) is likely to be detected as a static object in the first stage, and the neural network might activate if there is a luggage item in the corresponding image. Hence, this kind of situation is likely to be detected as an abandoned luggage item, until this point. Nonetheless, the second neural network in our cascade is specifically trained to solve such undesired situations. The network is given positive examples with abandoned luggage items and negative examples with attended luggage items, i.e. luggage items with people standing by. The second network is only applied on the image samples that are labeled as positive by the first network. In literature, this kind of pipeline is known as a *cascade of classifiers* [23].

An important remark is that we need to extend the bounding box around the luggage item to detect if there are persons standing nearby with the second CNN. Let $h \times w$ be the size of a bounding box. For each image sample labeled as positive by the first CNN, we extend the bounding box to a size of $2h \times 3w$, and extract the corresponding larger image. To process the static object tracks in real-time, we apply the CCNN at every $10$ frames. The predicted classification scores for an object track are

ISBN 978-90-827970-1-5 © EURASIP 2018

a) Positive generated samples for the two CNN models.

b) Negative generated samples for the first CNN.

c) Negative generated samples for the second CNN.

Figure 2. Realistic image samples generated by randomly sampling sub-images from the estimated background image (second row) and by superimposing luggage items (first row) or people standing by their luggage items (third row). These samples are used to fine-tune our convolutional neural networks for a particular scene. Best viewed in color.

temporally smoothed using a Gaussian filter of 25 frames (1 second). The sign transfer function is then applied to transform the scores for an object track into class labels. For each object track, we apply a majority voting scheme to determine the final class label for the respective object track.

To train both CNN models, we provide two types of examples. On one hand, we use images collected from the Internet in order to improve the generalization capacity of our neural networks. On the other hand, we want our models to be adapted to each individual scene in order to provide the best possible results in the respective scene. However, obtaining real image samples from each scene in order to fine-tune the networks is not a viable approach, as collecting these samples requires a large amount of time. For faster deployment, we propose to generate realistic image samples instead of collecting them. To fine-tune the first CNN model, we superimpose various template luggage items at random locations over the estimated background of each individual scene to obtain additional positive samples. We also select random sub-images from the background as negative examples. In a similar manner, we generate samples to train the second CNN model in our cascade. We use the same positive samples as for the first CNN model, but we generate the negative samples by superimposing people carrying or standing by their luggage items at various random locations over the estimated background. Figure 2 illustrates a few image samples generated for one video scene considered in the experiments presented in Section 4.

# 4. Experiments

## 4.1. Data Sets

We test our method on four datasets: AVSS 2007 [24], PETS 2006 [25], PETS 2007 [26] and TCD [27]. In the i-LIDS bag and vehicle detection challenge (AVSS 2007) abandoned baggage data set, the detection areas are divided into near, middle and far, and there is one video per area. The PETS 2006 data set consists of seven scenarios captured from four different viewpoints (cameras). It contains multiple types of luggage: briefcase, suitcase, 25 litre rucksack, 70 litre backpack and sky gear carrier. We used the videos from a single viewpoint (third camera). In a similar fashion, we used the abandoned luggage scenarios (S7 and S8) from PETS 2007, using a single viewpoint (third camera). The Trinity College Dublin (TCD) data set contains two videos showing objects being abandoned and later collected. In total, there are 14 test videos with a total of 42869 frames.

TABLE 1. IMAGE CLASSIFICATION RESULTS FOR OUR FIRST CNN TRAINED TO DISTINGUISH ABANDONED LUGGAGE ITEMS FROM OTHER OBJECTS, AND OUR SECOND CNN TRAINED TO DISTINGUISH ABANDONED LUGGAGE ITEMS FROM ATTENDED LUGGAGE ITEMS.

| Method | Precision | Recall | Accuracy |
|---|---|---|---|
| First CNN | 97.31% | 82.12% | 96.37% |
| Second CNN | 96.96% | 94.11% | 95.36% |

## 4.2. Evaluation

We use three frame-level and pixel-level metrics for evaluation: precision, recall and $F_1$ score. *Precision* is defined as the number of true-positive detections divided by the total number of detections, and *recall* as the number of true-positive detections divided by the number of ground-truth instances. The $F_1$ measure (also known as the $F_1$ score) is the harmonic mean of precision and recall. We define the frame-level and pixel-level metrics as in other works for abnormal event detection [28], [29], [30], [31]. At the pixel-level, the corresponding luggage item is considered as being correctly detected if the IoU between the detected bounding box and the ground-truth bounding box is greater than 0.2. At the frame-level, a frame is considered a correct detection if it contains at least one abandoned luggage item (there is no constraint regarding the overlap between the detected bounding box and the ground-truth bounding box). In order to calculate these metrics, we manually annotated all videos with ground-truth bounding boxes, since the data sets do not provide such annotations.

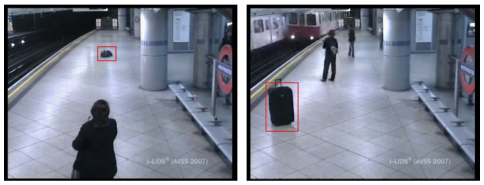
## 4.3. Baseline and Models

We consider as baseline a stripped-down version of our approach. The baseline is also based on two stages, with the first stage for static object detection identical to our own approach. In the second stage, we replace the cascade of convolutional neural networks with a single CNN. The baseline CNN is identical to the first CNN in our cascade. In the experiments, we consider two versions for the CCNN. The first version (SOD+CCNN) is trained using only the samples collected from the Internet. The second version (SOD+CCNN+Generated Samples) includes the generated samples in the training process. We present results with and without generated samples, to show the performance gain provided by the addition of generated samples.

ISBN 978-90-827970-1-5 © EURASIP 2018

TABLE 2. FRAME-LEVEL AND PIXEL-LEVEL METRICS FOR OUR APPROACH BASED ON STATIC OBJECT DETECTION (SOD) AND A CASCADE OF CONVOLUTIONAL NEURAL NETWORKS (CCNN) TRAINED WITH AND WITHOUT GENERATED SAMPLES VERSUS A BASELINE APPROACH BASED ON CNN. THE METHODS ARE EVALUATED ON FOUR DATA SETS: AVSS 2007, PETS 2006, PETS 2007 AND TCD (TRINITY COLLEGE OF DUBLIN). THE BEST SCORE FOR EACH METRIC ON EACH DATA SET IS HIGHLIGHTED IN BOLD.

| Data Set | Method | Frame-level | | | Pixel-level | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | $F_1$ score | Precision | Recall | $F_1$ score |
| AVSS 2007 | SOD + CNN (baseline) | 60.17% | 60.87% | 60.52% | 41.19% | 53.03% | 46.37% |
| | SOD + CCNN | **97.77%** | 51.89% | 67.80% | **97.77%** | 51.89% | 67.80% |
| | SOD + CCNN + Generated Samples | 97.48% | **66.59%** | **79.13%** | 97.47% | **65.70%** | **78.49%** |
| PETS 2006 | SOD + CNN (baseline) | 68.01% | 69.54% | 68.77% | 68.00% | 69.54% | 68.76% |
| | SOD + CCNN | 83.25% | 69.54% | 75.78% | 83.25% | 69.54% | 75.78% |
| | SOD + CCNN + Generated Samples | **95.67%** | **83.74%** | **89.31%** | **95.67%** | **83.74%** | **89.31%** |
| PETS 2007 | SOD + CNN (baseline) | 65.35% | **99.61%** | 78.92% | 65.17% | **99.61%** | 78.79% |
| | SOD + CCNN | 69.13% | **99.61%** | 81.62% | 68.99% | **99.61%** | 81.52% |
| | SOD + CCNN + Generated Samples | **97.47%** | **99.61%** | **98.53%** | **97.46%** | **99.61%** | **98.52%** |
| TCD | SOD + CNN (baseline) | **98.62%** | **100%** | **99.31%** | **98.62%** | **100%** | **99.31%** |
| | SOD + CCNN | **98.62%** | **100%** | **99.31%** | **98.62%** | **100%** | **99.31%** |
| | SOD + CCNN + Generated Samples | **98.62%** | **100%** | **99.31%** | **98.62%** | **100%** | **99.31%** |
| Overall Average | SOD + CNN (baseline) | 70.32% | 76.33% | 73.20% | 66.22% | 74.65% | 70.18% |
| | SOD + CCNN | 86.54% | 74.41% | 80.02% | 86.52% | 74.40% | 80.00% |
| | SOD + CCNN + Generated Samples | **96.74%** | **84.65%** | **90.29%** | **96.73%** | **84.46%** | **90.18%** |



a) Examples of detections on AVSS 2007.

b) Examples of detections on PETS 2006.

c) Examples of detections on PETS 2007.

d) Examples of detections on TCD.

Figure 3. Examples of abandoned luggage items detected by our approach based on SOD and CCNN trained with generated samples. Best viewed in color.

### 4.4. Results and Discussion

**Preliminary classification results.** We first train and test our convolutional neural networks on a collection of images collected from the Internet. In our data set, there are 2207 images with abandoned luggage items, 2000 with attended luggage items, and another 8035 samples with other objects such as people, cars, buses, trains, and so on. The data set for the first CNN is formed of the images with abandoned luggage items as positive examples and the images with other objects as negative examples. The data set for the second CNN is formed of the images with abandoned luggage items as positive examples and the images with people attending luggage items as negative examples. Both data sets are split into 80% for training and 20% for testing. The training sets are augmented using flipped and blurred versions of the original images. The classification results presented in Table 1 indicate that both methods attain precision levels of around 97% and accuracy rates higher than 95%. The first CNN in our cascade attains a lower recall of almost 82%. Overall, the classification results indicate that both networks are well trained and ready to be used in practice.

**Results for abandoned luggage detection in video.** Table 2 shows the results of our approach trained with and without generated samples against a strong CNN baseline. On the AVSS 2007 data set, both CCNN approaches yield considerably better frame-level and pixel-level precisions compared to the baseline CNN. The best $F_1$ scores (79.13% at the frame-level and 78.49% at the pixel-level) on AVSS 2007 are obtained by the CCNN version that is trained with generated samples. The same CCNN version also attains the best $F_1$ scores on PETS 2006, with an improvement higher than 20% over the baseline. On PETS 2007, all methods obtain the same recall scores (above 99%). However, the CCNN version trained with generated samples reaches much better precision levels (97.47% at the frame-level and 97.46% at the pixel-level) than the baseline CNN. The TCD data set seems to be quite easy, since all the evaluated methods obtain scores higher than 98% for all metrics. Table 2 also includes the average for each metric computed on the entire 14 videos from all four data sets.

ISBN 978-90-827970-1-5 © EURASIP 2018

We can observe that on average, the CCNN approach trained without generated samples yields average $F_1$ scores that are almost $10\%$ higher than the baseline CNN. However, the CCNN approach trained with generated samples attains even higher average $F_1$ scores. With an $F_1$ score of $90.29\%$ at the frame-level and an $F_1$ score of $90.18\%$ at the pixel-level, our best approach is roughly $20\%$ better than the baseline. We note that our framework runs at nearly $40$ frames per second on a machine with Intel Xeon Processor E5 1.7 GHz CPU and 32 GB of RAM, without using parallel threading.

## 5. Conclusion

In this paper, we have presented a novel approach for abandoned luggage detection in video that works in real-time. We compared our approach with a strong CNN baseline. The empirical results indicate that employing a cascade of two CNN models trained on collected as well as generated image samples provides improvements above $8\%$ for all evaluation metrics.

## Acknowledgments

The work of Radu Tudor Ionescu is supported through project grant PN-III-P2-2.1-PED-2016-1842. The work of Sorina Smeureanu is supported by SecurifAI through Project P/38/185 funded under POC-A1-A1.1.1-C-2015.

## References

[1] M. Bhargava, C.-C. Chen, M. S. Ryoo, and J. K. Aggarwal, "Detection of abandoned objects in crowded environments," in *Proceedings of AVSS*, 2007, pp. 271–276.

[2] H.-H. Liao, J.-Y. Chang, and L.-G. Chen, "A Localized Approach to Abandoned Luggage Detection with Foreground-Mask Sampling," in *Proceedings of AVSS*, 2008, pp. 132–139.

[3] F. Porikli, Y. Ivanov, and T. Haga, "Robust abandoned object detection using dual foregrounds," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, p. 30, 2008.

[4] Y.-L. Tian, R. Feris, and A. Hampapur, "Real-Time Detection of Abandoned and Removed Objects in Complex Environments," in *Proceedings of VS Workshop*, 2008.

[5] M. Bhargava, C.-C. Chen, M. S. Ryoo, and J. K. Aggarwal, "Detection of object abandonment using temporal logic," *Machine Vision and Applications*, vol. 20, no. 5, pp. 271–281, 2009.

[6] J. Wen, H. Gong, X. Zhang, and W. Hu, "Generative model for abandoned object detection," in *Proceedings of ICIP*, 2009, pp. 853–856.

[7] J.-Y. Chang, H.-H. Liao, and L.-G. Chen, "Localized Detection of Abandoned Luggage," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, 2010.

[8] P. Forczmański and M. Seweryn, "Surveillance Video Stream Analysis Using Adaptive Background Model and Object Recognition," in *Proceedings of ICCVG*, 2010, pp. 114–121.

[9] S. Kwak, G. Bae, and H. Byun, "Abandoned luggage detection using a finite state automaton in surveillance video," *Optical Engineering*, vol. 49, no. 2, pp. 027 007–1–027 007–10, 2010.

[10] G. Szwoch, P. Dalka, and A. Czyżewski, *A Framework for Automatic Detection of Abandoned Luggage in Airport Terminal*. Springer, 2010, pp. 13–22.

[11] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M. T. Sun, "Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 41, no. 5, pp. 565–576, 2011.

[12] K. Lin, S. Chen, C. Chen, D. Lin, and Y. Hung, "Left-Luggage Detection from Finite-State-Machine Analysis in Static-Camera Videos," in *Proceedings of ICPR*, 2014, pp. 4600–4605.

[13] K. Lin, S. C. Chen, C. S. Chen, D. T. Lin, and Y. P. Hung, "Abandoned Object Detection via Temporal Consistency Modeling and Back-Tracing Verification for Visual Surveillance," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1359–1370, 2015.

[14] I. Dahi, M. Chikr el Mezouar, N. Taleb, and M. Elbahri, "An Edge-based Method for Effective Abandoned Luggage Detection in Complex Surveillance Videos," *Computer Vision and Image Understanding*, vol. 158, no. C, pp. 141–151, 2017.

[15] N. T. Pham, K. Leman, J. Zhang, and I. Pek, "Two-stage unattended object detection method with proposals," in *Proceedings of ICSIP*, 2017, pp. 1–4.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper With Convolutions," *Proceedings of CVPR*, June 2015.

[17] K. C. Smith, P. Quelhas, and D. Gatica-Perez, "Detecting Abandoned Luggage Items in a Public Space," in *Proceedings of PETS Workshop*, 2006.

[18] G. Szwoch, "Extraction of stable foreground image regions for unattended luggage detection," *Multimedia Tools and Applications*, vol. 75, no. 2, pp. 761–786, 2016.

[19] 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.

[20] 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, 2010.

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

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, K. A., A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, 2015.

[23] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[24] i LIDS Team, "Imagery Library for Intelligent Detection Systems (i-LIDS); A Standard for Testing Video Based Detection Systems," in *Proceedings of CCST*, 2006, pp. 75–80.

[25] J. Ferryman and D. Tweed, "An Overview of the PETS2006 Dataset," in *Proceedings of PETS Workshop*, 2006.

[26] ——, "An Overview of the PETS2007 Dataset," in *Proceedings of PETS Workshop*, 2007.

[27] K. Dawson-Howe, *A Practical Introduction to Computer Vision with OpenCV*, 1st ed. Wiley Publishing, 2014.

[28] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proceedings of CVPR*, 2011, pp. 3449–3456.

[29] A. Del Giorno, J. Bagnell, and M. Hebert, "A Discriminative Framework for Anomaly Detection in Large Videos," in *Proceedings of ECCV*, October 2016, pp. 334–349.

[30] V. Mahadevan, W.-X. LI, V. Bhalodia, and N. Vasconcelos, "Anomaly Detection in Crowded Scenes," in *Proceedings of CVPR*, 2010, pp. 1975–1981.

[31] R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Unmasking the abnormal events in video," in *Proceedings of ICCV*, 2017, pp. 2895–2903.

ISBN 978-90-827970-1-5 © EURASIP 2018