Background modeling for video sequences by stacked denoising autoencoders

Jorge García-González, Juan M. Ortiz-de-Lazcano-Lobato, Rafael M. Luque-Baena, Miguel A. Molina-Cabello, and Ezequiel López-Rubio

Department of Computer Languages and Computer Science. University of Málaga. Bulevar Louis Pasteur, 35. 29071 Málaga. Spain.
{jorgegarcia,jmortiz,rmluque,miguelangel,ezeqlr}@lcc.uma.es

Abstract. Nowadays, the analysis and extraction of relevant information in visual data flows is of paramount importance. These images sequences can last for hours, which implies that the model must adapt to all kinds of circumstances so that the performance of the system does not decay over time. In this paper we propose a methodology for background modeling and foreground detection, whose main characteristic is its robustness against stationary noise. Thus, stacked denoising autoencoders are applied to generate a set of robust characteristics for each region or patch of the image, which will be the input of a probabilistic model to determine if that region is background or foreground. The evaluation of a set of heterogeneous sequences results in that, although our proposal is similar to the classical methods existing in the literature, the inclusion of noise in these sequences causes drastic performance drops in the competing methods, while in our case the performance stays or falls slightly.

Keywords: Background modeling · deep learning · autoencoders

1 Introduction

In today's society, the fact that it is necessary to automate the exploitation of visual information that we capture in the most reliable and efficient possible way is ever more present. In the field of artificial vision, video surveillance is still a very active field at research level, since not all the open fronts have been satisfactorily addressed in recent years. One of the main areas to improve resides in the background modeling, which consists of determining which pixels of the image correspond to the movement objects in the scene and which ones are part of the background of the scene.

Foreground detection algorithms must work 24 hours a day, with robustness against background variations. This variability can be observed in outdoor environments where the weather can change and generate rain, hail or snow, or in indoor environments where changes in lighting compromise the reliability of detection. Not only is it necessary for a detection algorithm to behave correctly for a few hundred frames, but it is necessary to ensure that changes in the external

conditions of the scene will not cause a drop in performance. These requirements are difficult to achieve in many of the works already published, in which changes in the scene could cause that the system stops working properly.

Most of the foreground detection algorithms work at pixel level, that is, modeling the intensity of each pixel and determining the probability of belonging to one of the two possible classes: foreground or background. Thus, there are many highly referenced proposals that achieve more than satisfactory results. The main differences between them reside in the underlying model that represents the intensity of color of each pixel over time. Wren et al. [11] uses a Gaussian distribution as a basis for the modeling of each pixel, while the GMM model [7] uses K distributions to manage multimodal funds. Zivkovic [13] uses an intermediate strategy, considering as many Gaussians as necessary up to a maximum value (K). On the other hand, Elgammal et al. [2] make use of kernel distributions, less restrictive than the previous ones statistically but more complex to update. Other more complex models go through modeling each pixel through self-organized maps, a type of unsupervised neural network. Both SOBS [4] and FSOM [3] models are based on the previous algorithm, in addition to combining the output of each pixel (probability of belonging to the background or foreground) with the output of their neighbors, which provides robustness to the model and makes it less sensitive to false positives.

The use of deep learning networks is not alien to this field. In this work we will use autoencoders, as an unsupervised learning technique, to minimize the impact of noise backgrounds in the modeling of the scene. Each image will be divided into patches whose noise will be eliminated by a previously trained autoencoder. Subsequently, and using the autoencoder information after the coding phase, a N dimensional Gaussian model will be used to estimate the probability of belonging to the background of each patch.

Autoencoders are well suited to information representation. Single layer linear autoencoders are proved to span the same subspace as a Principal Components Analysis (PCA) does when they attempt to learn an undercomplete representation of the input data, i.e., the number of neurons in the hidden layer is less than or equal to the input data dimension [1]. Therefore, the features that retain most of the input data variance will be kept. In stacked linear autoencoders, subsequent layers of the autoencoder will be used to condense that information gradually to the desired dimension of the reduced representation space. On the other hand, sparse autoencoders or autoencoders with layers made up of non-linear units will also obtain relevant features which can be expected to be easier to interpret and used by a classifier, though they will likely differ from those provided by the PCA technique, as it is discussed in [9].

The paper is divided in the following sections: Section 2 presents the object detection methodology based on the analysis of image patches to obtain a foreground mask from an input frame; section 3 reports the experimental results over several public surveillance sequences and Section 4 concludes the article.

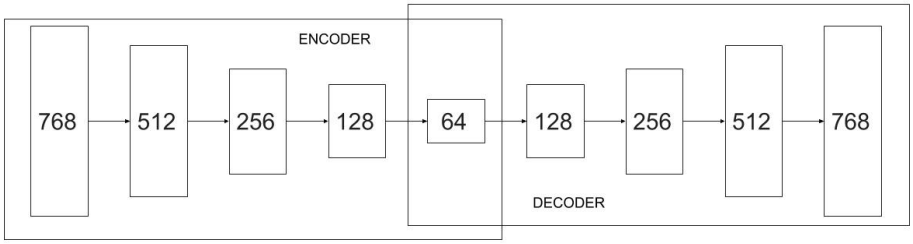


Fig. 1. Complete autoencoder structure with layers sizes.

2 Methodology

Most previous approaches to background modeling in video sequences model each pixel of the video frame separately. Our model intends to model small patches of size $N \times N$ pixels, so that for each incoming video frame an estimation is made in order to know whether each patch belongs to the background of the scene. The process is divided in two stages: firstly, a condensed representation of the patch, composed of significant features, is obtained by means of previously trained Stacked Denoising Autoencoder (SDA) [9]; secondly, a probabilistic model classifies the patch according to their computed set of relevant features.

2.1 Patch feature extraction

Let $\mathbf{X} \in \mathbb{R}^H$ be a patch of size H, where tristimulus pixel color values are assumed. The patch is processed by a stacked denoising autoencoder:

$$\tilde{\mathbf{X}} = g\left(f\left(\mathbf{X}\right)\right) \tag{1}$$

$$f \ : \ \mathbb{R}^H \rightarrow \mathbb{R}^L \qquad g \ : \ \mathbb{R}^L \rightarrow \mathbb{R}^H \tag{2}$$

where $\tilde{\mathbf{X}} \in \mathbb{R}^H$ is the reconstructed version of the input patch $\mathbf{X}$, f is the encoding part of the autoencoder, g is the decoding part of the autoencoder, and L is the number of neurons of the innermost layer of the neural architecture, i.e. the size of the last layer of the encoding part and the first layer of the decoding part (see Fig. 1). The goal of the autoencoder is to reduce the high dimensional input of size H to a a low dimensional set of features of size L with $L < H$.

An autoencoder is usually trained to minimize the reconstruction error $\mathcal{E}$:

$$\mathcal{E} = \sum_{i=1}^{R} \left\| \mathbf{X} - \tilde{\mathbf{X}} \right\|^2 \tag{3}$$

where R is the number of patches in the training set.

However denoising autoencoders are trained with corrupted input samples $\hat{\mathbf{X}}$ instead of the input samples themselves $\mathbf{X}$.

$$\mathcal{E} = \sum_{i=1}^{R} \left\| \mathbf{X} - g\left(f\left(\hat{\mathbf{X}}\right)\right) \right\|^2 \tag{4}$$

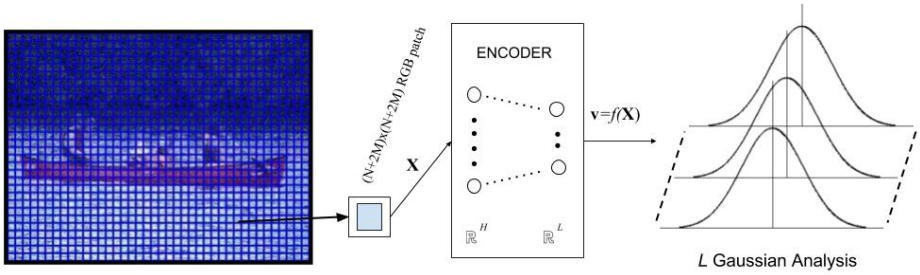


Fig. 2. Method overwiev scheme.

Denoising autoencoders try to learn a robust representation made up of more general features which prevents from overtraining and diminishes the influence of scene factors such as illumination and local variation. In an attempt to enforce the invariance of the autoencoder to the diverse scene conditions, several authors [10][12] have used a training set that comprises not patches extracted from the frames corresponding to the video to process but a huge amount of generic natural image patches that may be corrupted. This approach is followed in our proposal, where the training set for our single autoencoder is generated from the Tiny Images dataset [8].

It turns out that stacked denoising autoencoders might find difficulties in modeling too small patches. Here we propose to overcome this limitation by augmenting the $N \times N$ pixel patch by M pixels in each direction (up, down, left and right), so that an augmented patch of size $(N + 2M) \times (N + 2M)$ is supplied to the autoencoder, while the estimation about the pertenence to the background only affects to the central $N \times N$ pixel section of the augmented patch. In this way, the augmented patches overlap with their neighbors, while the small patches do not. Therefore, the dimension of the samples the autoencoder processes is $H = 3\,(N + 2M)^2$.

2.2 Patch classification

As the video sequence progresses, the features which are discovered by the autoencoder are extracted, and a probabilistic model is learned for them. (Figure 2) This model aims to capture the main characteristics of the probability distribution of the feature vector $\mathbf{v} \in \mathbb{R}^L$:

$$\mathbf{v} = f\,(\mathbf{X}) \tag{5}$$

To this end, the mean $\mu_j = E\,[v_j]$ and the variance $\sigma_j^2 = E\left[(v_j - \mu_j)^2\right]$ of each component of $\mathbf{v}$ are approximated by the Robbins-Monro stochastic approximation algorithm [5]:

$$\mu_{j,t+1} = (1 - \alpha)\,\mu_{j,t} + \alpha v_{j,t} \tag{6}$$

$$\sigma_{j,t+1}^2 = (1-\alpha)\,\sigma_{j,t}^2 + \alpha\,(v_{j,t} - \mu_{j,t})^2 \tag{7}$$

where t is the time instant (the frame index) and α is the step size.

Each small patch is declared to belong to the foreground whenever the number of components of the feature vector which are far from its estimated mean, as measured with respect to the estimated variance, is higher than a given threshold:

$$C < \sum_{j=1}^{L} \mathbb{I}\left(|v_{j,t} - \mu_{j,t}| > K\sigma_{j,t}\right) \tag{8}$$

where $\mathbb{I}$ stands for the indicator function, C is a tunable parameter which specifies the number of components which must be far from its estimated mean to declare that the small patch belongs to the foreground, and K is another tunable parameter which specifies how many standard deviations an observation must depart from its estimated mean to be considered to be far away.

3 Experimental Results

3.1 Methods

Five methods have been selected in order to make a performance comparison with our proposal: WrenGA [11], ZivkovicGMM [13], MaddalenaSOBS [4], ElgammalKDE [2] and Lopez-RubioFSOM [3].

Four of this methods are available on BGS library [6] [1]. The version 1.3.0 of the BGS library has been implemented by using the C++ language and version 2.4.8 of the OpenCV[2] library. On the other hand, Lopez-RubioFSOM is written in Matlab, with MEX files written in C++ for the most time-consuming parts and Matlab scripts for the rest. The employed parameter values are those indicated as default by the authors.

Finally, our proposed approach has been implemented using Python version 2.7. For neural network implementation we have used TensorFlow[3] version 1.5.0 by means of Keras[4] version 2.1.3 as high-level API. All evaluation has been made using MATLAB R2017B.

Our autoencoder implementation has been trained and tested using 100,000 random images from Tiny Images dataset [8][5]. Since each image has 32x32 pixels, we have divided each one to obtain four 16x16 images.

In order to be as fair as possible a random seed has been used to generate the input Gaussian noise for each noise level. The same videos with the applied noise are used in all the studied methods. We do not use any additional post processing in any of the methods.

[1] https://github.com/andrewssobral/bgslibrary
[2] http://opencv.org/
[3] https://www.tensorflow.org/
[4] https://keras.io/
[5] http://groups.csail.mit.edu/vision/TinyImages/

Table 1. Considered values for each parameter.

Parameter	Values
C	{3,6,9,12,15}
K	{2,3,4,5,6,7,8}
α	{0.001,0.005,0.01,0.05}

Table 2. Final parameter selection for each video an noise level.

Video	$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.2$
canoe	$C = 3$, $K = 5$, $\alpha = 0.001$	$C = 6$, $K = 4$, $\alpha = 0.001$	$C = 3$, $K = 5$, $\alpha = 0.001$
boats	$C = 3$, $K = 4$, $\alpha = 0.001$	$C = 3$, $K = 4$, $\alpha = 0.001$	$C = 3$, $K = 4$, $\alpha = 0.001$
fountain02	$C = 15$, $K = 6$, $\alpha = 0.001$	$C = 12$, $K = 5$, $\alpha = 0.001$	$C = 12$, $K = 5$, $\alpha = 0.001$
overpass	$C = 3$, $K = 6$, $\alpha = 0.001$	$C = 3$, $K = 6$, $\alpha = 0.001$	$C = 3$, $K = 6$, $\alpha = 0.001$
port_0_17fps	$C = 15$, $K = 5$, $\alpha = 0.05$	$C = 12$, $K = 5$, $\alpha = 0.05$	$C = 12$, $K = 5$, $\alpha = 0.05$
pedestrians	$C = 15$, $K = 7$, $\alpha = 0.001$	$C = 15$, $K = 6$, $\alpha = 0.001$	$C = 12$, $K = 7$, $\alpha = 0.001$

3.2 Sequences

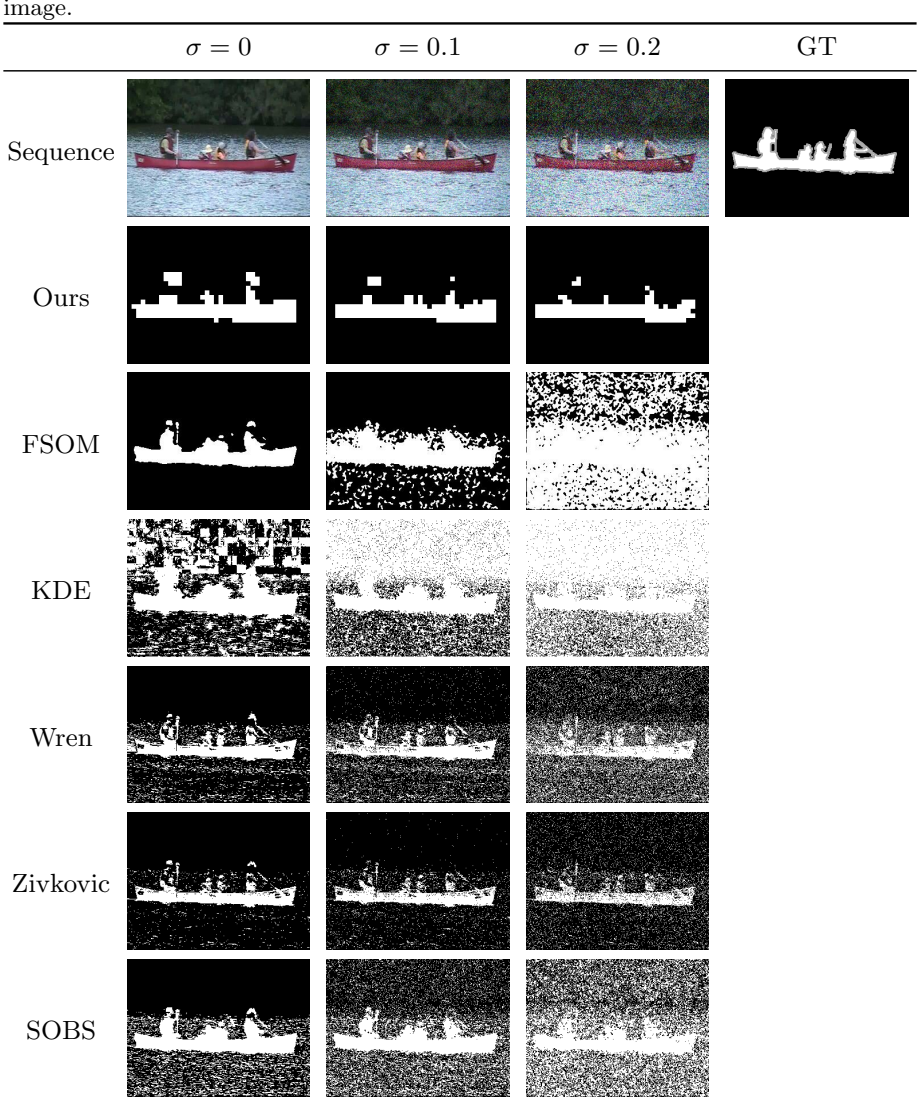
A set of video sequences have been selected from the 2014 dataset of the ChangeDetection.net website[6]. Four of the selected scenes are from Dynamic Background category, one from Low Frame Rate category and another one from the Baseline one. *Canoe* shows a river with water and forest background where a canoe goes across (320x240 pixels and 1189 frames). *Fountain02* shows a road behind a fountain that spits water out (432x288 pixels and 1499 frames). *Boats* shows a river next to a road. Two boats cross through the river while various vehicles move on the road (320x240 pixels and 7999 frames). *Overpass* shows a bridge traversed by a man with a river, forest and a road behind (320x240 pixels and 3000 frames). *Port_0_17fps* is a low frame rate video that shows a little dock with a lot of boats constantly moving, water and clouds as background and some persons and boats crossing from time to time as foreground (640x480 pixels and 3000 frames). *Pedestrians* is a baseline video where several people walk over a pavement next to grass with sun and shadows (360x240 pixels and 1099 frames).

3.3 Parameter selection

Our method needs three parameters to be selected (C, K and α). To get a good combination of parameter values, we have carried out and analyzed each possible combination from values in table 1 on page 6 for images without noise (140 combinations). We have used the parameter configuration that achieves

[6] http://changedetection.net/

Fig. 3. Qualitative results for frame 960 from canoe dataset. From left to right: images with different amount of Gaussian noise with mean 0. First row is original dataset input image with different amounts of Gaussian noise and ground-truth. Other rows correspond to foreground segmentation performed for each method and each input image.



best performance for each video without noise. The top 3 combinations have been tested for videos with Gaussian noise in order to select the combination which best performs. Table 2 on page 6 shows final parameter selection for each video and noise level.

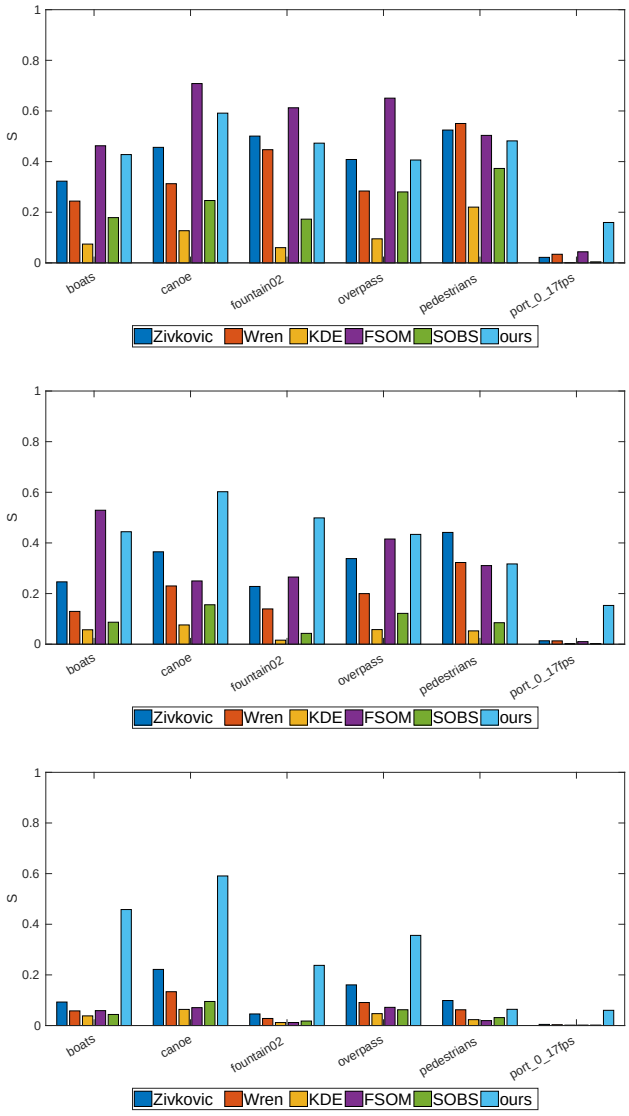


Fig. 4. Comparison between methods for each video with $\sigma = 0$, $\sigma = 0.1$ and $\sigma = 0.2$ Gaussian noises.

3.4 Results

A well-known measure has been selected in order to compare the performance from a quantitative point of view. In this work we have considered the spatial

accuracy S. This measure provides values in the interval $[0, 1]$, where higher is better, and represent the percentage of hits of the system.

The definition of this measure can be described as follows:

$$S = \frac{TP}{TP + FN + FP} \tag{9}$$

where TP refers to the foreground patches classified correctly(true positives) whereas FN and FP are the type II (false negatives) and type I (false positive) errors respectively.

S has been calculated for each binarized frame in Region of Interest (specified by ChangeDetection.net) generated using each previously mentioned method and we have obtained the mean for all frames with TP pixels in ground-truth.

Figure 4 on page 8 shows comparison between each method result for videos with different noises. We can observe our proposed method is able to deal with low level noise and even improve a bit its performance for some videos (canoe and boats). It is interesting to point that adding noise to other methods causes a lot more FP pixels while our method deals with it by increasing FN pixels as can be observed on figure 3 on page 7.

4 Conclusions

In this work we have proposed a methodology for the background modeling in video sequences, that uses autoencoders to filter the possible noise in the background and a multidimensional probabilistic model to determine the probability of belonging to one of the following two classes, background or foreground. Although our proposal works at region level, the comparative results with other techniques at pixel level where they take advantage of more information, leave us in good place for all the experiments carried out. To simulate more heterogeneous scenes, we have added Gaussian noise to each sequence, being our method much more robust than competitors to this increase in variability in the scene. In fact, the improvement is significant, being the best method on average in all the scenes analyzed. The greater the background noise, the greater the fall in performance of the method used. The results indicate that after introducing slight noise, the fall of our method is 4% on average, while the rest of the techniques have drops of over 30%. If the noise is magnified our performance goes down near 30% of its original value, but the performance of the rest falls 70% on average. These data corroborate the idea of robustness of our proposal, in addition to its usefulness for the processing and analysis of continuous data during uninterrupted periods of time.

Acknowledgements

This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grant TIN2014-53465-R, project name Video surveillance by active search of anomalous events, besides for the projects with codes

TIN2016-75097-P and PPIT.UMA.B1.2017. It is also partially supported by the Autonomous Government of Andalusia (Spain) under grant TIC-657, project name Self-organizing systems and robust estimators for video surveillance. All of them include funds from the European Regional Development Fund (ERDF). The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They also gratefully acknowledge the support of NVIDIA Corporation with the donation of two Titan X GPUs used for this research. The authors would like to thank the grant of the Universidad de Malaga.

References

1. Baldi, P., Hornik, K.: Neural networks and principal component analysis: Learning from examples without local minima. Neural Networks **2**(1), 53–58 (1989)
2. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: Computer Vision (ECCV). pp. 751–767. Springer (2000)
3. López-Rubio, E., Luque-Baena, R., Domínguez, E.: Foreground detection in video sequences with probabilistic self-organizing maps. International Journal of Neural Systems **21**(3), 225–246 (2011)
4. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. IEEE Transactions on Image Processing **17**(7), 1168–1177 (2008)
5. Robbins, H., Monro, S.: A stochastic approximation method. The Annals of Mathematical Statistics **22**(3), 400–407 (1951)
6. Sobral, A., Bouwmans, T.: Bgs library: A library framework for algorithm's evaluation in foreground/background segmentation. In: Background Modeling and Foreground Detection for Video Surveillance. CRC Press, Taylor and Francis (2014)
7. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. vol. 2, pp. 246–252 (1999)
8. Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence **30**(11), 1958–1970 (2008)
9. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research **11**, 3371–3408 (2010)
10. Wang, N., Yeung, D.: Learning a deep compact image representation for visual tracking. In: Advances in Neural Inform. Processing Systems 26, pp. 809–817 (2013)
11. Wren, C., Azarbayejani, A., Darrell, T., Pentl, A.: Pfinder: Real-time tracking of the human body. IEEE Trans. on Pattern Analysis and Machine Intelligence **19**(7), 780–785 (1997)
12. Zhang, Y., Li, X., Zhang, Z., Wu, F., Zhao, L.: Deep learning driven blockwise moving object detection with binary scene modeling. Neurocomputing **168**, 454 – 463 (2015)
13. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters **27**(7), 773–780 (2006)