MOVING OBJECT DETECTION IN NOISY VIDEO SEQUENCES USING DEEP CONVOLUTIONAL DISENTANGLED REPRESENTATIONS

Jorge García-González, Rafael M. Luque-Baena,
Juan M. Ortiz-de-Lazcano-Lobato, Ezequiel López-Rubio

Department of Computer Languages and Computer Science. University of Málaga
Biomedic Research Institute of Málaga (IBIMA)

ABSTRACT

Noise robustness is crucial when approaching a moving detection problem since image noise is easily mistaken for movement. In order to deal with the noise, deep denoising autoencoders are commonly proposed to be applied on image patches with an inherent disadvantage with respect to the segmentation resolution. In this work, a fully convolutional autoencoder-based moving detection model is proposed in order to deal with noise with no patch extraction required. Different autoencoder structures and training strategies are also tested to get insights into the best network design approach.

***Index Terms*—** Moving Object Detection, Foreground Segmentation, Autoencoders

1. INTRODUCTION

Foreground segmentation can be referred to as identifying genuine motion within a sequence. Unlike object detection, foreground segmentation (also known as background subtraction) is based on analyzing changes in a video sequence over time. It does not make sense for a single static image. Motion detection methods are therefore usually based on the analysis of the change in image regions, either pixel-level ([1]), or patch-level ([2]). They can also rely on deep learning-based object detection and tracking ([3, 4]). Still, these methods rely on detection models with a limited number of classes defined at training time and are currently unable to adapt to identifying the motion of an object not included in these classes. Either way, noise robustness is a key feature.

When applying deep learning to foreground segmentation, a common strategy is to use autoencoders ([5]). An autoencoder is a non-supervised neural network trained to return as output its input ([6]). An autoencoder includes two modules: an encoder (the layers to get a coded representation of the input data) and a decoder (layers to turn the coded representation into the input data again). An autoencoder can also be trained to return a denoised version of the input as output, and then they are known as denoising autoencoders. To train autoencoders in order to use the encoder to get a denoised and dramatically smaller version of image patches is a well-known strategy in foreground segmentation ([2, 7]). This approach needs to divide images into patches with enough size to contain useful information (i.e. 16×16), therefore the segmentation resolution is minimal. To use a tiling strategy to increase resolution is an approach to deal with that problem ([8, 4] but the computational requirement rises with the segmentation resolution. The present paper proposes a foreground segmentation model based on deep convolutional autoencoders in noisy sequences in order to apply the autoencoder strategy with no patch extraction, thus the segmentation resolution is increased due to avoiding the tiling. A comparison between different structures and training strategies is also included in order to obtain insights about the best autoencoder definition approach.

The remaining document is divided as follows: a methodology section 2 with the new proposal, an experiment section 1 with implementation and experimentation details and results, and finally a conclusions sections 4 is included.

This work is partially supported by the Ministry of Science, Innovation and Universities of Spain under grant RTI2018-094645-B-I00, project name Automated detection with low-cost hardware of unusual activities in video sequences. It is also partially supported by the Autonomous Government of Andalusia (Spain) under project UMA18-FEDERJA-084, project name Detection of anomalous behavior agents by deep learning in low-cost video surveillance intelligent systems. It is also partially supported by the Autonomous Government of Andalusia (Spain) under project UMA20-FEDERJA-108, project name Detection, characterization and prognosis value of the non-obstructive coronary disease with deep learning. All of them include funds from the European Regional Development Fund (ERDF). It is also partially supported by the University of Malaga (Spain) under grants B1-2019_01, project name Anomaly detection on roads by moving cameras, and B1-2019_02, project name Self-Organizing Neural Systems for Non-Stationary Environments. 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. The authors also thankfully acknowledge the grant of the Universidad de Málaga and the Instituto de Investigación Biomédica de Málaga - IBIMA.

2. METHODOLOGY

In this section, we propose a probabilistic model to determine which pixels of the video frame belong to the background, as opposed to the foreground. The first step consists on learning a compressed representation of the incoming video data by a Deep Convolutional Autoencoder (DA from now on). In what follows we will assume that tristimulus pixel color values are employed. The DA takes the whole frame $\mathbf{X}$ of size $G \times H \times 3$ as input, where the G is the number of pixel rows and H is the number of pixel columns, and returns the reconstructed frame $\tilde{\mathbf{X}}$, also of size $G \times H \times 3$ as output. The innermost layer of the DA is defined by a convolutional block of size $M \times N \times L$, where $M < G$, $N < H$ and L is the number of channels. The operation of the DA is given by:

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

$$f \ : \ \mathbb{R}^{G \times H \times 3} \rightarrow \mathbb{R}^{M \times N \times L} \tag{2}$$

$$g \ : \ \mathbb{R}^{M \times N \times L} \rightarrow \mathbb{R}^{G \times H \times 3} \tag{3}$$

where f is the encoder contained in the autoencoder, and g is the decoder contained in the autoencoder. The autoencoder transforms the input of dimension $G \times H \times 3$ to a disentangled set of significant features of dimension $M \times N \times L$, and then decodes this disentangled data to a reconstructed frame.

For the second step, we have $M \times N$ probabilistic mixture models, so that each model is provided with its associated feature vector $\mathbf{v} \in \mathbb{R}^L$ for each input video frame. Each probabilistic mixture model estimates the probability of its associated frame region to belong to the foreground. Such mixture models are updated as the video sequence progresses.

For each probabilistic mixture model, the background modeling is performed by approximating the L-dimensional probabilistic distribution of $\mathbf{v}$. A Gaussian mixture component K models the background, while the foreground is captured by a uniform mixture component U. It is assumed that the distribution of each feature vector can be approximated by this model:

$$p(\mathbf{v}) = \pi_{Back} p(\mathbf{v}|Back) + \pi_{Fore} p(\mathbf{v}|Fore) \tag{4}$$

where π_{Back} and π_{Fore} are the a priori probabilities of the background and foreground, respectively. Also:

$$p(\mathbf{v}|Back) = K(\mathbf{v}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{5}$$

$$p(\mathbf{v}|Fore) = U(\mathbf{v}) \tag{6}$$

The Gaussian and uniform mixture components are given by:

$$\begin{aligned} K(\mathbf{v}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= (2\pi)^{-L/2}\text{det}(\boldsymbol{\Sigma})^{-1/2} \\ &\text{exp}(-\frac{1}{2}(\mathbf{v} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{v} - \boldsymbol{\mu})) \end{aligned} \tag{7}$$

$$U(\mathbf{v}) = \begin{cases} 1/Vol(H) & \text{iff } \mathbf{v} \in H \\ 0 & \text{iff } \mathbf{v} \notin H \end{cases} \tag{8}$$

where H is the support of the uniform probability density function.

The L-dimensional volume of H is noted $Vol(H)$. Let d_h and e_h, $1 \leq h \leq L$, be the minimum and maximum values in the domain of encoding function f, respectively. Then we have:

$$H = [d_1, e_1] \times [d_2, e_2] \times ... \times [d_L, e_L] \tag{9}$$

$$Vol(H) = \prod_{h=1}^{L} (e_h - d_h) \tag{10}$$

The mean vector and the vector of standard deviations of the Gaussian are defined as follows:

$$\boldsymbol{\mu} = E[\mathbf{v}|Back] \tag{11}$$

$$\boldsymbol{\sigma} = \left(\sqrt{E\left[\left(v_j - \mu_j\right)^2 |Back \right]} \right)_{j=1,...,L} \tag{12}$$

The covariance matrix of the Gaussian is assumed to be diagonal:

$$\boldsymbol{\Sigma} = E[\left(\mathbf{v} - \boldsymbol{\mu}\right)\left(\mathbf{v} - \boldsymbol{\mu}\right)^T |Back] = \begin{pmatrix} \sigma_1^2 & 0 & . & . & 0 \\ 0 & \sigma_2^2 & . & . & 0 \\ . & . & . & . & . \\ 0 & 0 & . & . & \sigma_L^2 \end{pmatrix} \tag{13}$$

The Robbins-Monro stochastic approximation algorithm is employed to estimate the mean vector $\boldsymbol{\mu}$ and the variance vector $\boldsymbol{\psi}$ of each component of $\mathbf{v}$:

$$\boldsymbol{\psi} = \left(\sigma_j^2\right)_{j=1,...,L} = \left(E\left[\left(v_j - \mu_j\right)^2 |Back \right] \right)_{j=1,...,L} \tag{14}$$

Welford's online algorithm ([9]) is employed to obtain the first approximations of $\boldsymbol{\mu}$ and $\boldsymbol{\psi}$ for each probabilistic model during training phase.

The class probabilities given the feature vector $\mathbf{v}_n$ at time n are given by the Bayes theorem:

$$\forall i \in \{Back, Fore\}, R_{n,i} = P(i|\mathbf{v}_n) =$$

$$\frac{\pi_i p(\mathbf{v}_n|i)}{\pi_{Back} p(\mathbf{v}_n|Back) + \pi_{Fore} p(\mathbf{v}_n|Fore)} \tag{15}$$

So, given α as update step size, to update the background model $\boldsymbol{\mu}_n$ and $\boldsymbol{\psi}_n$ for feature vector $\mathbf{v}_n$ with background probability $R_{n,Back}$ the following equations are used:

$$\boldsymbol{\mu}_{n+1} = (1 - \alpha R_{n,Back})\boldsymbol{\mu}_n + \alpha R_{n,Back}\mathbf{v}_n \tag{16}$$

$$\boldsymbol{\psi}_{n+1} = (1 - \alpha R_{n,Back})\boldsymbol{\psi}_n + \alpha R_{n,Back}[\mathbf{v}_n - \boldsymbol{\mu}_n]^2 \tag{17}$$

Also, in order to update the a priori probabilities $\pi_{n,Back}$ and $\pi_{n,Fore}$ given $R_{n,Back}$ and $R_{n,Fore}$ the following equations can be applied:

$$\begin{gathered} \forall i \in \{Back, Fore\}, \\ \pi_{n+1,i} = (1 - \alpha)\pi_{n,i} + \alpha R_{n,i} \end{gathered} \tag{18}$$

It must be noted that $\boldsymbol{\Sigma}$ is a diagonal matrix with the elements of the variance vector $\boldsymbol{\psi}$ at the main diagonal. Therefore:

$$\det(\boldsymbol{\Sigma}) = \prod_{j=1}^{L} \sigma_j^2 \tag{19}$$

$$(\mathbf{v} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{v} - \boldsymbol{\mu}) = \sum_{j=1}^{L} \frac{(v_j - \mu_j)^2}{\sigma_j^2} \tag{20}$$

In order to keep the model simple, we set all prior probabilities $\pi_{Fore} = \pi_{Back} = 0.5$ and we do not update them.

3. EXPERIMENTS

Sequences and evaluation: In order to test our proposal, video sequences from categories baseline (4 sequences) and dynamic background (6 sequences) from changedetection.net [10] are used. Four kinds of noise are added: low Gaussian noise ($\mu = 0$ and $\sigma = 0.1$), medium Gaussian noise ($\mu = 0$ and $\sigma = 0.2$), high Gaussian noise ($\mu = 0$ and $\sigma = 0.3$) and Uniform noise from -0.5 to 0.5. Therefore, the final number of sequences is 40. *F-measure* metric which provides a value in the interval $[0, 1]$ has been selected to obtain a fair comparison.

Autoencoders training: Our proposal can be adapted to the sequence mainly in two ways: through the probabilistic model and by means of the specialisation of the DA. Other proposals ([11, 12]) only use a generic DA trained with generic images to encode the images from all sequences. In this work, in addition to the strategy of using the generic DA, two more strategies are tested in which a specific DA is trained for each sequence:

- Generic: a single DA trained with 400,000 64x64 images extracted from Imagenet2017 [13] for 20 epochs adding medium Gaussian noise ($\mu = 0$ and $\sigma = 0.2$) to the input.
- Specific Realistic: a DA for each sequence trained with 40.000 64x64 images obtained from the altered changedetection sequence for 20 epochs with no noise added to the input.
- Specific Ideal: a DA for each sequence trained with 40.000 64x64 images obtained from the original changedetection for 20 epochs adding the same noise the video was modified with.

Three DA structures with a different number of layers and convolutional block depth L are used. DA arquitecture features are shown in Table 1. Since 40 different sequences are tested with different training strategies, the total number of trained models is $3 \times (1 + 40 + 40) = 243$

Reference Methods: Results of three state-of-the-art methods have been included as reference: LOBSTER [14], PAWCS [15] and SuBSENSE [16]. The BGS library ([17]) is used to apply these methods.

Results: Table 2 shows quantitative results. Contrary to expectations, the generic training strategy is shown to be the most effective in 5 of the 8 comparisons, although the other strategies do not show particularly worse results. It should not be forgotten that the generic training strategy requires a single training. In contrast, the other two require sequence-specific training so that, even with similar performance, it is much more efficient and requires less computational power to get a single model. Regarding the structures and size of L, D_2 obtains the best scores in 4 out of 8 comparisons.

Implementation: Python[1] and Tensorflow 2 [18] have been used to implement the proposal.

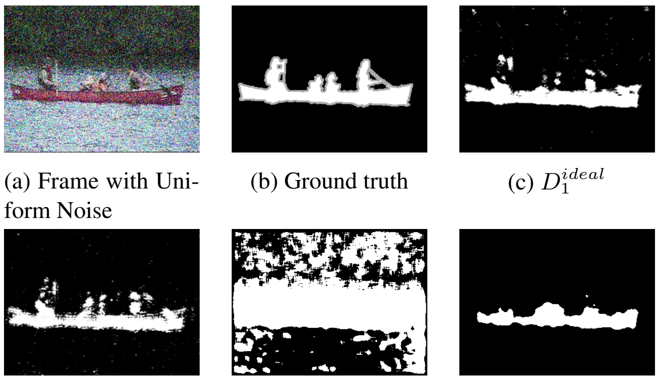
4. CONCLUSIONS

This work proposes a fully convolutional autoencoder-based method to identify the pixels belonging to moving objects in noisy video sequences. Nine versions of the proposal are tested and compared with three other state-of-the-art methods considered as references in the field of foreground segmentation. According to the results it seems that the proposed model presents a high resilience to noise. Besides, it does not depend on the common patch extraction preprocess related to autoencoder-based foreground segmentation methods and thus it avoids their segmentation resolution problems. Three training strategies are proposed and tested in order to measure the convenience of using generic autoencoders or scene-specific ones. Contrary to the expected, the generic autoencoder seems the best approach. Not only is it based on

[1]https://www.python.org/

D_1	GxHx3 - (G-2)x(H-2)x64 - (G-4)x(H-4)x32 - (G-6)x(H-6)x16 - **(G-8)x(H-8)x8** - (G-6)x(H-6)x16 - (G-4)x(H-4)x32 - (G-2)x(H-2)x64 - **GxHx3**
D_2	GxHx3 - (G-2)x(H-2)x64 - (G-4)x(H-4)x32 - (G-6)x(H-6)x16 - (G-8)x(H-8)x8 - **(G-10)x(H-10)x4** - (G-8)x(H-8)x8 - (G-6)x(H-6)x16 - (G-4)x(H-4)x32 - (G-2)x(H-2)x64 - **GxHx3**
D_3	GxHx3 - (G-2)x(H-2)x64 - (G-4)x(H-4)x32 - (G-6)x(H-6)x16 - (G-8)x(H-8)x8 - (G-10)x(H-10)x4 - **(G-12)x(H-12)x2** - (G-10)x(H-10)x4 - (G-8)x(H-8)x8 - (G-6)x(H-6)x16 - (G-4)x(H-4)x32 - (G-2)x(H-2)x64 - **GxHx3**

Table 1: DA structures. Left column represents DA structure identification. Right column represents DA structure as a succession of data shapes. Orange represents the input **X** shape, blue represents the output after a convolutional layer, red represents the shape after a deconvolutional layer. Bold blue represents the **innermost layer** $f(\mathbf{X})$ and bold red represents the **output** $g(f(\mathbf{X}))$ **shape**. $L = 8$ for D_1, $L = 4$ for D_2 and $L = 2$ for D_3. Al layers use ReLU as activation function except the bold ones which use sigmoid, filters are 3×3 and no extra padding is added.



(a) Frame with Uniform Noise (b) Ground truth (c) D_1^{ideal}

(d) $D_2^{generic}$ (e) LOBSTER (f) PAWCS

Fig. 1: Qualitative examples. (a) shows the frame 959 from canoe sequence with high Gaussian noise. (b) shows its ground-truth. (c) shows our segmentation using D_1 structure trained under ideal circumstances specifically for this sequence (d) shows our segmentation using D_2 structure trained in a generic way. (e) and (f) shows segmentations created by LOBSTER and PAWCS method respectively.

the necessity of training an autoencoder only once but also it achieves better performance. Given the proposal results, the application of morphological operators could be a future line to explore in order to increase the segmentation quality.

5. REFERENCES

[1] Thierry Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 11-12, pp. 31–66, 2014.

Method	Baseline	D. Background
Low Gaussian ($\mu = 0$ and $\sigma = 0.1$)		
$D_1^{generic}$	0.779358	0.626676
D_1^{ideal}	0.826057	0.660197
$D_1^{realistic}$	0.806041	**0.681824**
$D_2^{generic}$	0.814077	0.626943
D_2^{ideal}	0.783720	0.616150
$D_2^{realistic}$	0.807642	0.591761
$D_3^{generic}$	0.780034	0.616604
D_3^{ideal}	0.738924	0.562192
$D_3^{realistic}$	0.772673	0.553164
LOBSTER	0.818204	0.303046
PAWCS	0.792160	0.638228
SuBSENSE	**0.850084**	0.619803
Medium Gaussian ($\mu = 0$ and $\sigma = 0.2$)		
$D_1^{generic}$	0.749738	0.603604
D_1^{ideal}	0.787216	0.605378
$D_1^{realistic}$	0.668165	0.570674
$D_2^{generic}$	**0.800593**	**0.623706**
D_2^{ideal}	0.791945	0.596775
$D_2^{realistic}$	0.737218	0.547748
$D_3^{generic}$	0.772652	0.615213
D_3^{ideal}	0.724262	0.518304
$D_3^{realistic}$	0.734519	0.558834
LOBSTER	0.382661	0.071825
PAWCS	0.544240	0.465313
SuBSENSE	0.542270	0.219326
High Gaussian ($\mu = 0$ and $\sigma = 0.3$)		
$D_1^{generic}$	0.598292	0.464894
D_1^{ideal}	**0.769419**	0.578944
$D_1^{realistic}$	0.557288	0.406536
$D_2^{generic}$	0.758937	**0.602011**
D_2^{ideal}	0.701751	0.537704
$D_2^{realistic}$	0.595141	0.471129
$D_3^{generic}$	0.749747	0.601661
D_3^{ideal}	0.682415	0.440006
$D_3^{realistic}$	0.686032	0.469861
LOBSTER	0.103146	0.037094
PAWCS	0.369273	0.466141
SuBSENSE	0.305885	0.084691
Uniform from -0.5 to 0.5		
$D_1^{generic}$	0.580646	0.453299
D_1^{ideal}	0.749891	0.605946
$D_1^{realistic}$	0.554663	0.416700
$D_2^{generic}$	**0.749928**	0.600678
D_2^{ideal}	0.715505	0.473488
$D_2^{realistic}$	0.578980	0.513192
$D_3^{generic}$	0.745047	**0.606113**
D_3^{ideal}	0.668892	0.533003
$D_3^{realistic}$	0.702139	0.508050
LOBSTER	0.097819	0.035357
PAWCS	0.366042	0.466313
SuBSENSE	0.349744	0.103754

Table 2: Average *F-measure* value (the higher, the better) for each category for the application of our probabilistic model with each DA and noise. The **best result** in each column is bold and the second best result is colored in gray.

[2] Yaquing Zhang, Xi Li, Zhongfei Zhang, Fei Wu, and Liming Zhao, “Deep learning driven blockwise moving object detection with binary scene modeling,” *Neurocomputing*, vol. 168, pp. 454 – 463, 2015.

[3] Marc Braham, Sébastien Piérard, and Marc Van Droogenbroeck, “Semantic Background Subtraction,” in *IEEE International Conference on Image Processing (ICIP)*, Beijing, China, sep 2017, pp. 4552–4556.

[4] Jorge García-González, Juan Miguel Ortiz-de Lazcano-Lobato, Rafael M Luque-Baena, and Ezequiel López-Rubio, “Foreground Detection by Probabilistic Mixture Models Using Semantic Information from Deep Networks,” in *24th European Conference on Artificial Intelligence, ECAI*, 2020, vol. 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 2696–2703.

[5] Thierry Bouwmans, Sajid Javed, Maryam Sultana, and Soon Ki Jung, “Deep neural network concepts for background subtraction:A systematic review and comparative evaluation,” *Neural Networks*, vol. 117, pp. 8 – 66, 2019.

[6] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion,” *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[7] Jorge García-González, Juan Miguel Ortiz-de Lazcano-Lobato, Rafael Marcos Luque-Baena, Miguel Ángel Molina-Cabello, and Ezequiel López-Rubio, “Foreground detection by probabilistic modeling of the features discovered by stacked denoising autoencoders in noisy video sequences,” *Pattern Recognition Letters*, vol. 125, pp. 481–487, 2019.

[8] Jorge García-González, Juan Miguel Ortiz-de Lazcano-Lobato, Rafael Marcos Luque-Baena, and Ezequiel López-Rubio, “Background Modeling by Shifted Tilings of Stacked Denoising Autoencoders,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11487 LNCS, pp. 307–316.

[9] B P Welford, “Note on a Method for Calculating Corrected Sums of Squares and Products,” *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962.

[10] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Prakash Ishwar, and Others, “Changedetection. net: A new change detection benchmark dataset.,” in *CVPR Workshops*, 2012, number 2012, pp. 1–8.

[11] Jorge García-González, Miguel Ángel Molina-Cabello, Rafael Marcos Luque-Baena, Juan Miguel Ortiz-de Lazcano-Lobato, and Ezequiel Lopez-Rubio, “Deep Autoencoder Architectures For Foreground Object Detection In Video Sequences Based On Probabilistic Mixture Models,” in *ICIP 2020*. oct 2020, IEEE.

[12] Jorge García-González, Juan Miguel Ortiz-de Lazcano-Lobato, Rafael Marcos Luque-Baena, and Ezequiel López-Rubio, “Background subtraction by probabilistic modeling of patch features learned by deep autoencoders,” *Integrated Computer-Aided Engineering*, vol. 27, no. 3, pp. 253–265, 2020.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, may 2017.

[14] Pierre-Luc St-Charles and Guillaume-Alexandre Bilodeau, “Improving background subtraction using Local Binary Similarity Patterns,” in *IEEE Winter Conference on Applications of Computer Vision*, mar 2014, pp. 509–515.

[15] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin, “Universal Background Subtraction Using Word Consensus Models,” *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4768–4781, oct 2016.

[16] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin, “SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity,” *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, 2015.

[17] Andrews Sobral and Thierry Bouwmans, “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.

[18] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015.