Foreground object detection enhancement by adaptive super resolution for video surveillance

Miguel A. Molina-Cabello[1]
miguelangel@lcc.uma.es
David A. Elizondo[2]
elizondo@dmu.ac.uk
Rafael Marcos Luque-Baena[1]
rmluque@lcc.uma.es
Ezequiel López-Rubio[1]
ezeqlr@lcc.uma.es

[1] Department of Computer Languages and Computer Science
University of Málaga
Málaga, Spain

[2] Department of Computer Technology
De Montfort University
Leicester, United Kingdom

Abstract

Foreground object detection is a fundamental low level task in current video surveillance systems. It is usually accomplished by keeping a model of the background at each frame pixel. Many background learning algorithms have difficulties to attain real time operation when applied directly to the output of state of the art high resolution surveillance cameras, due to the large number of pixels. Here we propose a strategy to address this problem which consists in maintaining a low resolution model of the background which is upscaled by adaptive super resolution in order to produce a foreground detection mask of the same size as the original input frame. Extensive experimental results demonstrate the suitability of our proposal, in terms of reduction of the computational load and foreground detection accuracy.

1 Introduction

Detection of foreground objects which are active in a video surveillance scene is one of the most important low level tasks to be carried out in an automated video surveillance system [2, 3, 4, 6] . The clustering of pixels of an image or video sequence into two classes, foreground and background, has always aroused great interest in the scientific community. Therefore, this issue has been addressed from different areas, both in the image segmentation field [19] and in the motion detection one [18]. Difficulties like complex backgrounds [8] and moving cast shadows [14] among many others complicate this endeavour. Recently, there has been a great increase in the use of deep learning techniques for foreground detection [5, 15, 21], obtaining, after a training period, surprising results in terms of the success rate. However, despite the efforts to design efficient algorithms for this task, the advent of inexpensive high resolution surveillance cameras implies that many of those algorithms are not able to attain real time operation for high resolution videos.

Super resolution from a single image allows increasing the spatial resolution of an image. It is a standard technique to implement state of the art video surveillance systems [22, 23]. A typical application of super resolution is face recognition from surveillance videos. Face images are acquired with a low spatial resolution within the surveillance video frame, so that super resolution can help enhance the accuracy of the face recognition procedures [13]. A similar problem arises for license plate recognition, in particular if the camera produces low quality footage [20]. Super resolution has also been employed to increase the quality of the inputs for object detection in video surveillance based on infrared cameras [24].

In order to reduce the computational load of background model learning, it is possible to downscale the incoming video frames so that the number of pixels for which a background model must be learned is reduced. This reduction of the model helps to alleviate the memory needs posed by these algorithms and that are especially relevant when the foreground detection is deployed in low cost devices [1, 17]. After that, an upscaling must be carried out to restore the original spatial resolution of the incoming video. The critical step of this scheme is the upscaling phase, where the estimations coming from a downsized background model are interpolated to a finer grid in order to recover the original frame size. In [16] a straighforward bicubic interpolation is employed. In this work we propose to use an adaptive approach based on the Median Filter Transform [9]. This way we aim to improve the foreground object detection performance of the system by enhancing the quality of the upscaled foreground object detection masks.

The structure of this paper is as follows. First, our proposed method is described in Section 2. Then the experiments that we have carried out are reported in Section 3. Finally, Section 4 is devoted to conclusions.

2 Methodology

Next we define our proposed method for background modeling. Let us consider a frame size of $N \times M$ pixels, so that the foreground object detection is meant to be carried out at such resolution. In order to reduce the computational complexity of the background model learning, we propose to maintain a model with a smaller spatial resolution. To this end, let us consider a feature function which maps each point in the input frame to a feature vector of size D:

$$\psi : [1,N] \times [1,M] \rightarrow \mathbb{R}^D \tag{1}$$

$$\mathbf{z} = \psi(\mathbf{x}) \tag{2}$$

where the values ψ are only known at the points with integer pixel coordinates, $\mathbf{x} \in \{1,...,N\} \times \{1,...,M\}$.

In order to reduce the spatial resolution a downsampling procedure must be carried out. This means that a background model is learned only at the following coordinates:

$$\mathcal{H} = \left\{ \left(1 + i\frac{N-1}{n-1}, 1 + j\frac{M-1}{m-1} \right) \mid i \in \{0,...,n-1\}, j \in \{0,...,m-1\} \right\} \tag{3}$$

so that the background model contains $n \times m$ points, with $n < N$, $m < M$.

In order to estimate the feature vector at non integer pixel coordinates, an interpolation procedure is required. The simplest one is the nearest neighbor approach:

$$\psi_{NN}(\mathbf{x}) = \psi(\text{round}(\mathbf{x})) \tag{4}$$

where round is the rounding function, applied componentwise to the coordinate vector $\mathbf{x}$.

Another option is window averaging over blocks of size $W \times W$ pixels:

$$\psi_{AVG}(\mathbf{x}) = \frac{1}{W^2} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \psi(\mathbf{y}) \tag{5}$$

where $\mathcal{N}(\mathbf{x})$ stands for the block of size $W \times W$ pixels which point $\mathbf{x}$ belongs to.

Other options include bilinear and bicubic interpolation:

$$\psi_{LIN}(\mathbf{x}) = \sum_{p,q \in \{0,1\}} a_{pq} x_1^p x_2^q \tag{6}$$

$$\psi_{CUB}(\mathbf{x}) = \sum_{p,q \in \{0,1,2,3\}} b_{pq} x_1^p x_2^q \tag{7}$$

where a_{pq} and b_{pq} are suitable bilinear and bicubic interpolation coefficients, respectively.

The estimate feature vectors at the points in $\mathcal{H}$ are then used to learn the background model at those points. The background model outputs the probabilities to belong to the background at those points:

$$\rho : [1,N] \times [1,M] \rightarrow [0,1] \tag{8}$$

$$\rho(\mathbf{x}) = P(Back \mid \mathbf{x}) \tag{9}$$

where $\rho(\mathbf{x})$ stands for the probability to belong to the background at point $\mathbf{x}$, which is known for $\mathbf{x} \in \mathcal{H}$.

Finally, an upsampling procedure is carried out to estimate the values of $\rho(\mathbf{x})$ for integer pixel coordinates, $\mathbf{x} \in \{1,...,N\} \times \{1,...,M\}$. For this purpose, bicubic interpolation can be used, since it yields the most accurate results:

$$\rho_{CUB}(\mathbf{x}) = \sum_{p,q \in \{0,1,2,3\}} c_{pq} x_1^p x_2^q \tag{10}$$

where c_{pq} are suitable bicubic interpolation coefficients.

We have also considered the option of applying the Median Filter Transform (MFT) for the upsampling procedure [9]. The MFT yields the following estimation of the foreground mask:

$$\rho_{MFT}(\mathbf{x}) = \text{median}(\{\varphi(\mathbf{x}, \mathbf{A}_1, \mathbf{b}_1), ..., \varphi(\mathbf{x}, \mathbf{A}_H, \mathbf{b}_H)\}) \tag{11}$$

$$\forall i \in \{1,...,H\},\ \varphi(\mathbf{x}, \mathbf{A}_i, \mathbf{b}_i) = \text{median}(\zeta(\mathbf{x}, \mathbf{A}_i, \mathbf{b}_i)) \tag{12}$$

where:

- H is a parameter which specifies how many tilings will be considered.
- $\mathbf{A}_i$ are real valued matrices of size 2×2 which are randomly drawn from a probability distribution $p(\mathbf{A})$.
- $\mathbf{b}_i$ are real valued vectors of size 2×1 which are randomly drawn from the probability distribution $p(\mathbf{b})$.
- $\zeta(\mathbf{x}, \mathbf{A}_i, \mathbf{b}_i)$ is a set of real numbers which correspond to some pixel values of the downsized image, as follows:

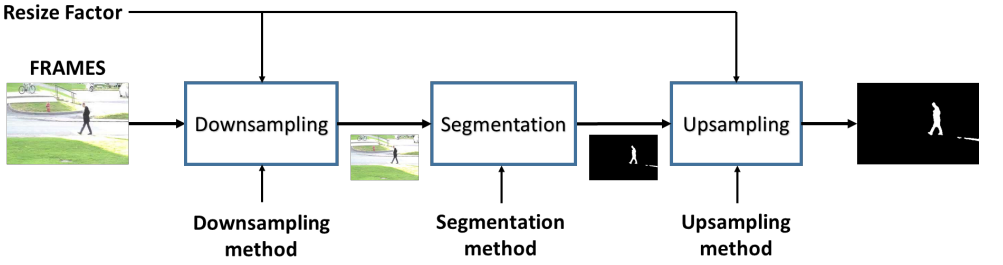


Figure 1: Operation of the proposed methodology. First of all, a sequence of frames is provided as input. The model also requires the selected resize factor parameter. Then, the downsampling process is carried out in order to reduce the frame size. After that, the segmentation method is applied to the resized frames to obtain the binary masks with the detected foreground pixels (white pixels) and the background (black pixels). These masks have the same frame size than the resized frames. The last step is the upsampling process, where the binary masks are resized to obtain a masks with the original frame size.

$$\zeta(\mathbf{x},\mathbf{A},\mathbf{b}) = \{\psi(\mathbf{w}) \mid \mathbf{w} \in \mathcal{H}, \text{round}(\mathbf{A}\mathbf{w}+\mathbf{b}) = \text{round}(\mathbf{A}\mathbf{x}+\mathbf{b})\} \tag{13}$$

The set $\zeta(\mathbf{x},\mathbf{A},\mathbf{b})$ comprises all the pixels of the downsized image which belong to the parallelogram where pixel $\mathbf{x}$ lies, according to the plane tiling described by $\mathbf{A}$ and $\mathbf{b}$. For more information about the MFT please refer to [9], where the detailed specifications of $p(\mathbf{A})$ and $p(\mathbf{b})$ can be found.

Please note that our proposed procedure can be applied to any background model learning algorithm. Therefore, a different foreground detection algorithm is obtained by applying our proposal to each possible background model learning method.

In order to preserve the proportion of both dimensions of the frame after the application of the downsampling and upsampling methods, we have consider a resize factor parameter μ. As we commented before, let us consider an input frame size of $N \times M$ pixels and a frame size of $n \times m$ after the downsampling procedure is carried out, where $n < N$ and $m < M$. Thus, the resize factor parameter μ fulfills $\mu = \frac{n}{N} = \frac{m}{M}$ and $\mu \in [0,1]$.

Figure 1 describes the operation of our proposed methodology. It must be highlighted that the traditional approach is composed only by the segmentation method, which has a sequence of frames as input and produces a sequence of binary frames as output.

3 Experimental results

The experiments that have been carried out are described in this section. First of all, Subsection 3.1 details the software and the hardware resources employed in the experiments. Then, the image dataset used to test our proposal is described in Subsection 3.2. After that, the parameter selection of our approach is specified in Subsection 3.3. Finally, the obtained results are reported in Subsection 3.4.

3.1 Methods

According to the proposed approach, some downsampling methods are considered, namely: Nearest neighbor (NN), Bicubic interpolation (CUB), Bilinear interpolation (LIN) and Block-wise average (AVG). In the case of the upsampling process, we have considered Bicubic

Parameter		Values
Downsampling method	=	{NN, CUB, LIN, AVG}
Upsampling method	=	{CUB, SR}
Resize factor, μ	=	$\{0.5, 0.25, 0.125\}$
Segmentation method	=	{MFBM, CL-VID, FSOM}

Table 1: Considered parameter values for the experiments, forming the set of tuned configurations.

interpolation (CUB) and Superresolution (SR). The SR method employed in the proposal is based on the MFT algorithm [9].

Several well-known reference segmentation methods have been considered for the comparisons with the aim of assessing the suitability of our approach. Three different segmentation methods have been selected: MFBM [12], CL-VID [11] and FSOM [10].

All of these methods are implemented in Matlab, with MEX files written in C++ for the most time-consuming parts and Matlab scripts for the rest.

The reported experiments have been carried out on a 64-bit Personal Computer with two Intel E5-2670 CPU with eight cores, 2.60 GHz per core, 32 GB RAM and standard hardware. The implementation of our method does not use any GPU resources, so it does not require any specific graphics hardware.

3.2 Dataset

We have used a large amount of videos to test the performance of the compared methods. Different typical challenging situations for the foreground detection problem, such as intermittent shades, lighting changes or dynamic background motions, are presented in the selected videos. The dataset chosen to carry out our experiments is the 2012 Dataset of the ChangeDetection.net web site[1] [7], which is formed by 31 videos organised into 6 categories.

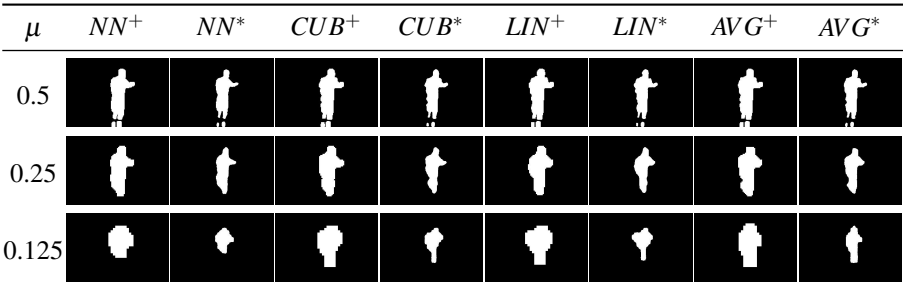
3.3 Parameter selection

The selected values of the parameters of the competing methods have been set to the recommended values from their original papers. For the proposed approach, additionally to the considered downsampling, segmentation and upsampling methods, we have also consider a range of values for the resize factor parameter. Table 1 summarises the tuned configurations.

In order to denote the possible configurations of our proposal in an easy way, we will note them with the name of the employed downsampling method followed by a symbol: + for those configurations where the upsampling method is CUB and * for those configurations where the upsampling method is SR. For example, in the case where the downsampling method selected is NN, we will have NN^+ when the upsampling method is CUB and NN^* when the upsampling method is SR. On the other hand, the configurations where our proposed downsampling-upsampling process is not carried out, so that, we are applying the traditional schema with only the segmentation method, we will note that as the original frame size method (ORIG).

[1] http://changedetection.net/

(a)



(b)

Figure 2: Qualitative results for a benchmark scene corresponding to the frame 1000 of the video Office by applying the FSOM segmentation method. Subfigure (a) shows the raw frame, the Ground Truth (GT) mask and the output of the segmentation method by applying the traditional schema (ORIG result), respectively. Subfigure (b) exhibit the results of the approach considering different tuned configurations for the downsampling and upsampling methods and the resize factor parameter μ. First column represents the selected value of the resize factor parameter, while remaining columns show the chosen downsampling-upsampling method.

3.4 Results

Our aim is to determine the influence of the analysed compression methods on the foreground mask produced by the object detection method, its execution time and the memory used.

In this subsection the results of the experiments are shown. The goal is to establish how the analysed tuned configurations of downsampling and upsampling methods and the resize of the frames affect to the foreground mask provided by the segmentation method.

First of all, we compare the obtained result from a qualitative point of view. Figure 2 exhibits some results. It can be observed how the result is worse when the resize factor decreases: the foreground detected objects adopt squared forms and they are not detailed as well as the ORIG result. As it is shown, the person that appears in the scenario is detected as a rectangular form with the lowest tested value of the resize factor parameter. Nevertheless, it is interesting to observe how the application of the proposal can reduce the rate of false positives. While the ORIG result presents this kind of error around the person, our proposal removes these false positive pixels.

Additionally, we have compared the results provided by the tuned configurations in a quantitative way. The performance measure selected in order to compare the output mask is the F-measure (F-m), which is a well-known measure with a value between 0 and 1, where higher is better. The execution time is also studied, where the frames per second rate (fps) is used as a measure. It is a positive value where higher is better. And the last selected measure is the memory used (in KBytes) which is a positive number and lower is better.

Figure 3 reports the mean performance achieved by all the tested configurations in the

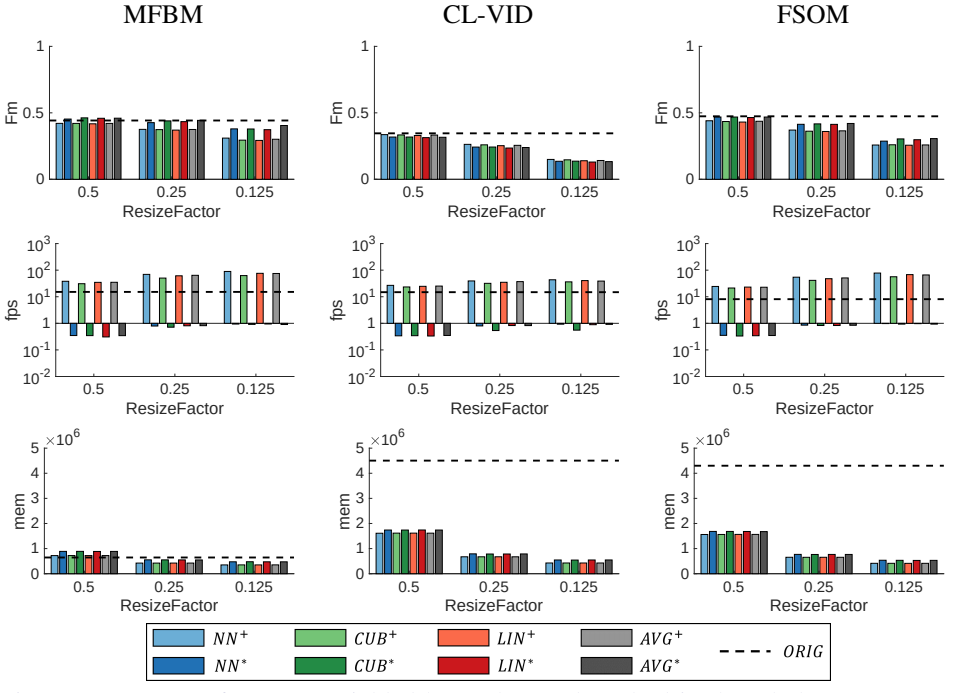


Figure 3: Mean performances yielded by each tested method in the whole CDNET 2012 dataset. The different studied measures are shown from up to bottom: F-measure (Fm), Frames per second rate (fps) and Memory (mem); while the tested method are listed from left to right: MFBM, CL-VID and FSOM. Inside each figure, each bar corresponds to the mean performance achieved by the different tuned configurations of resize factor and downsampling and upsampling methods. The mean performance yielded by the ORIG method is shown with a dashed line. Note that the fps figures are in logarithmic scale in order to show them in a better way.

whole dataset. It must be highlighted that the lower the resize factor the lower the performance, the faster the execution time and the lower the memory used. In particular, in the first row, the detection performance of our approach is, in general, lower than the ORIG performance. Nevertheless, the use of our strategy attains a faster execution time (second row) and requires a lower amount of memory (third row).

It is interesting to observe how the SR upsampling method achieves better results than CUB in two of the three tested segmentation methods (MFBM and FSOM, while CL-VID offers worse performance). However, in terms of the execution time and memory, the CUB upsampling method is faster (the difference is quite significant) and consumes lower memory than SR. All the downsampling methods exhibit a similar performance.

If we focus on the performance achieved in the different videos that compose the tested dataset, it can be observed how our proposal achieves better performances in those videos which present a dynamic background. In particular, Table 2 reports the F-measure yielded for the tuned configurations which uses the FSOM segmentation method and a resize factor equal to 0.25. In general, the traditional schema yields better results than our proposal. However, our proposal outperforms the traditional schema in the videos which belong to the dynamic background category.

Video	ORIG	NN^+	NN^*	CUB^+	CUB^*	LIN^+	LIN^*	AVG^+	AVG^*
highway	**0.938**	0.720	0.853	0.710	0.827	0.706	0.813	0.709	0.827
office	**0.737**	0.604	0.623	0.591	0.616	0.591	0.613	0.598	0.627
pedestrians	**0.743**	0.439	0.679	0.403	0.683	0.394	0.675	0.415	0.695
PETS2006	**0.824**	0.701	0.799	0.691	0.811	0.687	0.802	0.692	0.817
badminton	0.670	0.720	0.857	0.713	0.860	0.717	0.856	0.722	**0.874**
boulevard	0.416	0.400	**0.440**	0.393	0.437	0.389	0.429	0.393	0.436
sidewalk	**0.578**	0.143	0.153	0.134	0.141	0.134	0.140	0.134	0.141
traffic	0.443	0.487	**0.555**	0.477	0.548	0.471	0.544	0.476	0.546
boats	0.137	0.108	0.090	0.125	0.144	0.125	**0.146**	0.124	0.144
canoe	0.571	0.472	0.562	0.469	0.576	0.468	0.576	0.472	**0.581**
fall	0.176	0.221	0.253	0.226	0.266	0.228	**0.270**	0.227	0.265
fountain01	0.060	0.090	0.111	0.076	0.105	0.074	0.100	0.080	**0.113**
fountain02	0.170	0.131	0.172	0.127	0.176	0.126	0.174	0.128	**0.177**
overpass	**0.294**	0.190	0.228	0.190	0.245	0.187	0.243	0.188	0.234
abandonedBox	**0.342**	0.226	0.295	0.207	0.279	0.203	0.271	0.217	0.278
parking	**0.323**	0.180	0.188	0.099	0.118	0.097	0.108	0.123	0.136
sofa	**0.580**	0.470	0.443	0.460	0.451	0.453	0.445	0.468	0.459
streetLight	**0.419**	0.128	0.146	0.147	0.208	0.147	0.212	0.133	0.184
tramstop	**0.247**	0.228	0.227	0.227	0.233	0.226	0.232	0.228	0.232
winterDriveway	0.311	0.216	0.372	0.191	0.366	0.184	0.358	0.198	**0.392**
backdoor	**0.323**	0.271	0.299	0.267	0.303	0.266	0.302	0.268	0.307
bungalows	**0.349**	0.281	0.342	0.275	0.342	0.274	0.341	0.273	0.341
busStation	**0.758**	0.652	0.711	0.628	0.711	0.618	0.695	0.627	0.718
copyMachine	**0.678**	0.611	0.627	0.614	0.630	0.614	0.630	0.613	0.632
cubicle	**0.311**	0.232	0.251	0.230	0.260	0.230	0.259	0.229	0.262
peopleInShade	**0.559**	0.489	0.550	0.485	0.551	0.484	0.549	0.488	0.553
corridor	**0.508**	0.335	0.352	0.332	0.360	0.329	0.360	0.336	0.366
diningRoom	**0.746**	0.720	0.674	0.721	0.684	0.719	0.682	0.725	0.686
lakeSide	**0.450**	0.125	0.084	0.133	0.095	0.134	0.096	0.133	0.096
library	**0.427**	0.379	0.323	0.382	0.328	0.381	0.327	0.382	0.328
park	**0.603**	0.512	0.567	0.494	0.568	0.500	0.562	0.500	0.573
Average	**0.474**	0.370	0.414	0.362	0.417	0.360	0.413	0.365	0.420

Table 2: F-measure yielded by the FSOM method in the whole CDNET 2012 dataset with a resize factor $\mu = 0.25$. From left to right, first column represents the tested video, second column exhibits the performance of the method ORIG and the remaining columns show the different tuned configurations of the studied downsampling and upsampling methods. The performance of the tested videos for each configuration are listed from up to bottom, where the last row reports the average performance in the whole dataset. Best results are highlighted in **bold**.

Additionally, we have studied the performance in the videos with the largest frame size of the tested dataset. These videos (PETS2006, badminton, fall and copyMachine) have a frame size higher than 100,000 pixels. The performance of the tuned configurations is reported in Figure 4. In this case it must be highlighted how the proposal increases the fps rate and decreases the memory used, while the performance is similar to ORIG.

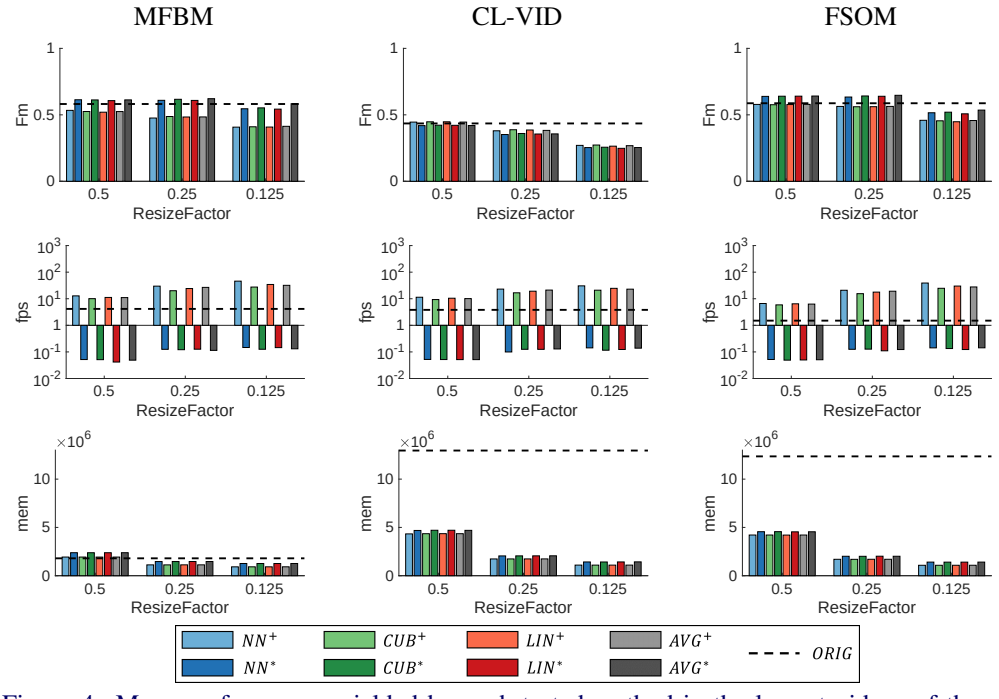


Figure 4: Mean performances yielded by each tested method in the largest videos of the CDNET 2012 dataset. The different studied measures are shown from up to bottom: F-measure (Fm), Frames per second rate (fps) and Memory (mem); while the tested methods are listed from left to right: MFBM, CL-VID and FSOM. Inside each figure, each bar corresponds to the mean performance achieved by the different tuned configurations of resize factor and downsampling and upsampling methods. The mean performance yielded by the ORIG method is shown with a dashed line. Note that the fps figures are in logarithmic scale in order to show them in a clearer way.

4 Conclusions

This work proposes a methodology to enhance the detection of foreground objects in video sequences, especially when the input is of high quality (e.g. 4K) or there are certain limitations in the computing capacity of the hardware device. It is aimed to reduce the computational load required to process high resolution videos by learning the background model at each pixel of a low resolution version of the original input video frame. The predictions obtained from those models are subsequently interpolated to a high resolution grid by super resolution, so that a high quality foreground detection mask of the same size as the original frame is generated. The experimental results demonstrate that our approach with the bicubic interpolation (CUB) as upsampling method, reduces the CPU time in all cases. In addition, the superresolution (SR) manages to increase even the performance of the original proposal in one of the methods studied (FSOM), which is interesting if an improvement of the model performance is required, at the expense of longer processing time. The memory reduction is especially relevant for the deployment of this type of systems in low cost devices.

Acknowledgments

This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grants TIN2016-75097-P and PPIT.UMA.B1.2017. It is also partially supported by the Ministry of Science, Innovation and Universities of Spain [grant number 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 P12-TIC-657, project name Self-organizing systems and robust estimators for video surveillance. It is also partially supported by the Autonomous Government of Andalusia (Spain) under project UMA18-FEDERJA-084, project name Anomalous behaviour agent detection by deep learning in low cost video surveillance intelligent systems. 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 acknowledge the funding from the Universidad de Málaga.

References

[1] J. Benito-Picazo, E. López-Rubio, J.M. Ortiz-De-lazcano lobato, E. Domínguez, and E.J. Palomo. Motion detection by microcontroller for panning cameras. *Lecture Notes in Computer Science*, 10338 LNCS:279–288, 2017. doi: 10.1007/978-3-319-59773-7_29.

[2] T. Bouwmans. Recent advanced statistical background modeling for foreground detection - a systematic survey. *Recent Patents on Computer Science*, 4(3):147–176, 2011.

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

[4] T. Bouwmans and E.H. Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34, 2014.

[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*, 117:8 – 66, 2019. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.04.024.

[6] H. Fu, X. Cao, and Z. Tu. Cluster-based co-saliency detection. *IEEE Transactions on Image Processing*, 22(10):3766–3778, 2013.

[7] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. Changedetection. net: A new change detection benchmark dataset. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 1–8. IEEE, 2012.

[8] L. Li, W. Huang, I.Y.-H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13 (11):1459–1472, 2004.

[9] Ezequiel López-Rubio. Superresolution from a single noisy image by the median filter transform. *SIAM Journal on Imaging Sciences*, 9(1):82–115, 2016.

[10] Ezequiel López-Rubio, Rafael Marcos Luque-Baena, and Enrique Domínguez. Foreground detection in video sequences with probabilistic self-organizing maps. *International Journal of Neural Systems*, 21(3):225–246, 2011.

[11] Ezequiel López-Rubio, Miguel A Molina-Cabello, Rafael Marcos Luque-Baena, and Enrique Domínguez. Foreground detection by competitive learning for varying input distributions. *International Journal of Neural Systems*, 28(05):1750056, 2018.

[12] Francisco Javier López-Rubio and Ezequiel López-Rubio. Features for stochastic approximation based foreground detection. *Computer Vision and Image Understanding*, 133:30–50, 2015.

[13] S. Mandal, S. Thavalengal, and A.K. Sao. Explicit and implicit employment of edge-related information in super-resolving distant faces for recognition. *Pattern Analysis and Applications*, 19(3):867–884, 2016.

[14] N. Martel-Brisson and A. Zaccarin. Learning and removing cast shadows through a multidistribution approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1133–1146, 2007.

[15] Tsubasa Minematsu, Atsushi Shimada, Hideaki Uchiyama, and Rin-ichiro Taniguchi. Analytics of deep neural network-based background subtraction. *Journal of Imaging*, 4(6), 2018. ISSN 2313-433X. doi: 10.3390/jimaging4060078.

[16] Miguel A. Molina-Cabello, Ezequiel López-Rubio, Rafael Marcos Luque-Baena, Esteban J. Palomo, and Enrique Domínguez. Frame size reduction for foreground detection in video sequences. In Oscar Luaces, José A. Gámez, Edurne Barrenechea, Alicia Troncoso, Mikel Galar, Héctor Quintián, and Emilio Corchado, editors, *Advances in Artificial Intelligence*, pages 3–12, Cham, 2016. Springer International Publishing.

[17] F. Ortega-Zamorano, M.A. Molina-Cabello, E. LÃČÂşpez-Rubio, and E.J. Palomo. Smart motion detection sensor based on video processing using self-organizing maps. *Expert Systems with Applications*, 64:476–489, 2016. doi: 10.1016/j.eswa.2016.08.010.

[18] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp. Urban surveillance systems: from the laboratory to the commercial world. *Proceedings of the IEEE*, 89(10):1478–1497, 2001. doi: 10.1109/5.959342.

[19] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015720.

[20] H. Seibel, S. Goldenstein, and A. Rocha. Eyes on the target: Super-resolution and license-plate recognition in low-quality surveillance videos. *IEEE Access*, 5:20020–20035, 2017.

[21] Maryam Sultana, Arif Mahmood, Sajid Javed, and Soon Ki Jung. Unsupervised deep context prediction for background estimation and foreground segmentation. *Machine Vision and Applications*, 30(3):375–395, 2019. ISSN 1432-1769. doi: 10.1007/s00138-018-0993-0.

[22] X. Sun, X.-G. Li, J.-F. Li, and L. Zhuo. Review on deep learning based image super-resolution restoration algorithms. *Zidonghua Xuebao/Acta Automatica Sinica*, 43(5): 697–709, 2017.

[23] S. Xi, C. Wu, and L. Jiang. Super resolution reconstruction algorithm of video image based on deep self encoding learning. *Multimedia Tools and Applications*, 78(4):4545–4562, 2019.

[24] H. Zhang, C. Luo, Q. Wang, M. Kitchin, A. Parmley, J. Monge-Alvarez, and P. Casaseca-de-la Higuera. A novel infrared video surveillance system using deep learning based techniques. *Multimedia Tools and Applications*, 77(20):26657–26676, 2018.