

# Deep learning-based anomalous object detection system powered by microcontroller for PTZ cameras

Jesús Benito-Picazo, Enrique Domínguez, Esteban J. Palomo,  
Ezequiel López-Rubio, Juan Miguel Ortiz-de-Lazcano-Lobato

*Department of Computer Languages and Computer Science  
University of Málaga*

Bulevar Louis Pasteur, 35. 29010 Málaga. Spain.

{jpicazo,enrique,d,ejpalomo,ezeqlr,jmortiz}@lcc.uma.es

**Abstract**—Automatic video surveillance systems are usually designed to detect anomalous objects being present in a scene or behaving dangerously. In order to perform adequately, they must incorporate models able to achieve accurate pattern recognition in an image, and deep learning neural networks excel at this task. However, exhaustive scan of the full image results in multiple image blocks or windows to analyze, which could make the time performance of the system very poor when implemented on low cost devices. This paper presents a system which attempts to detect abnormal moving objects within an area covered by a PTZ camera while it is panning. The decision about the block of the image to analyze is based on a mixture distribution composed of two components: a uniform probability distribution, which represents a blind random selection, and a mixture of Gaussian probability distributions. Gaussian distributions represent windows in the image where anomalous objects were detected previously and contribute to generate the next window to analyze close to those windows of interest. The system is implemented on a Raspberry Pi microcontroller-based board, which enables the design and implementation of a low-cost monitoring system that is able to perform image processing.

**Index Terms**—Foreground detection, feed forward neural network, PTZ camera, convolutional neural network

## I. INTRODUCTION

Video surveillance systems have become an extremely active research area due to increasing levels of social conflict and public awareness about security issues. This has led to motivation for the development of robust and precise automatic video surveillance systems, which need a source of data to be processed and analysed. Fixed and PTZ cameras have become the essential information source in those greatly demanded systems [1]–[3]. Real time operation is essential for the successful deployment of these systems [4], [5].

Research on computer vision systems based on pan-tilt-zoom (PTZ) cameras has been intense for many years [6]–[10]. Nevertheless, there is a lack of a comprehensive theory which sets the foundations for the development of practical systems. Fragmentary approaches that are limited to some parts of the problem are available, but it is still not clear how to combine them to yield complete and reliable systems that can be deployed in many situations. In the present work we focus on the detection of anomalous objects in the scene while a PTZ camera is panning, tilting or zooming in or out, in order

to cover the entire environment which the video surveillance system has to be aware of.

Nowadays, deep learning have become increasingly popular and widely applied to computer vision systems [11], [12], such as object recognition (e.g. handwriting, face, behavior...) or image classification. Hundreds of papers have been published in the last years providing different types of deep neural networks, but all of them need the help of GPU-accelerated computing techniques in order to achieve their objectives. These applications are normally based on high performance and expensive hardware, which requires a high power consumption.

Microcontroller boards are economic, small and flexible hardware devices. They are frequently employed in motion detection systems due to their low energy consumption and reduced cost [13]–[15]. A flexible Printed Board Circuit (PCB) prototype which integrates a microcontroller has been proposed to estimate motion and proximity [16]. In this prototype, eight photodiodes are used as light sensors. The efficiency of solar energy plants can be improved by low power systems which estimate cloud motion [17]. The approximation of the cloud motion vectors is carried out by an embedded microcontroller, so that the arrangement of the solar panels can be optimized for maximum electricity output. Energy-saving street lighting for smart cities can be accomplished by low power motion detection systems equipped with low consumption microcontrollers and wireless communication devices [18]. This way, the street lamps are switched on when people are present in their surroundings. Recently, a motion detection algorithm based on Self-Organizing Maps (SOMs) was developed in an Arduino DUE board [19]. The implementation of the SOM algorithm was employed as a motion detector for static cameras in a video surveillance system.

In this paper, a low energy and low cost system for moving anomalous object detection is proposed. The proposal is based on deep learning algorithms and powered for microcontrollers using PTZ cameras. Initial experiments carried out show a very promising proposal which delivers competitive results.

The rest of the paper is organized as follows: Section II is reserved for the formulation of the mathematical model this system relies on. In Section III both the software architecture

and the hardware architecture this system is supposed to be deployed in, are described. Section IV is intended to illustrate the results of the experiments performed with the objective of determining the system performance when deployed in a raspberry Pi 3 model B microcontroller-based board. Last section will be reserved for some conclusions based on the results obtained from the experiments performed.

## II. METHODOLOGY

We consider a certain object as anomalous in a certain environment when its presence should generate an alert in any surveillance system installed in the cited environment.

In this section, our proposed anomalous object detection model is detailed. A set of active detections is to be maintained during the operation of the system. Each detection is a tuple  $(\pi_i, x_1, x_2, x_3)$  where:

- $\pi_i$  is the *a priori* probability or mixing weight
- $x_1$  is the pixel row of the center of the detection in the current video frame coordinate system.
- $x_2$  is the pixel column of the center of the detection in the current video frame coordinate system.
- $x_3$  is the size of the detection, i.e. the length (in pixel) of the side of the square which defines the detection.

In order to translate a detection from the previous frame to the next one, the following equations must be used:

$$\pi_i(t) = \alpha \pi_i(t-1) \quad (1)$$

$$x_1(t) = x_1(t-1) \quad (2)$$

$$x_2(t) = x_2(t-1) + \delta \quad (3)$$

$$x_3(t) = x_3(t-1) \quad (4)$$

where  $\alpha$  is a damping factor,  $0 < \alpha < 1$ , and  $\delta$  is the number of pixels per frame of the apparent panning motion (as estimated in [20]). A detection goes inactive whenever it goes out of the current frame.

Let us define  $\mathbf{x} = (x_1, x_2, x_3)$ . The set of valid values for  $\mathbf{x}$  is:

$$\mathcal{V} = [1, N_{rows}] \times [1, N_{cols}] \times [S_{min}, S_{max}] \subset \mathbb{R}^3 \quad (5)$$

where the video frame size is  $N_{rows} \times N_{cols}$  pixels, and the minimum and maximum side lengths of the detections to be considered are  $S_{min}$  and  $S_{max}$ , respectively.

Let us consider the following probability density function in order to sample potential locations for detections:

$$p(\mathbf{x}) = qU_{\mathcal{V}}(\mathbf{x}) + (1-q) \frac{1}{N_{detections}} \sum_{i=1}^{N_{detections}} \pi_i Gauss(\mathbf{x}_i, \sigma) \quad (6)$$

where  $U_{\mathcal{V}}(\mathbf{x})$  is the uniform distribution on  $\mathcal{V}$ ,  $Gauss(\boldsymbol{\mu}, \sigma)$  stands for the spherical multivariate gaussian distribution with mean  $\boldsymbol{\mu}$  and standard deviation  $\sigma$ ,  $N_{detections}$  is the current number of active detections,  $q \in (0, 1)$  is a tunable parameter and  $\sigma$  is a standard deviation tunable parameter. The aim is to focus on the parts of the video frame where active detections have been found before, while we also pay some attention to all other regions of the video frame to search for previously undetected objects.

The object detection algorithm which is associated with the above defined model is as follows:

- 1) The set of active detections  $\mathcal{A}$  is initialized to the empty set.
- 2) A new video frame is acquired from the camera.
- 3) All active detections are updated by Equations (1)-(4). All updated detections which fall outside  $\mathcal{V}$  are discarded, as they have become inactive.
- 4) A set of  $M$  random samples are drawn from (6). For each random sample, the associated frame window is found and resized to the shape required by a suitable convolutional neural network (CNN). Then the sample is fed to the CNN. If the resulting output reveals that an object has likely been found, then the sample is inserted into  $\mathcal{A}$ , with a weight which is proportional to the likelihood that an object is really there.
- 5) Go to step 2.

In the following section, a possible way to implement the model and its associated algorithm on low power hardware is proposed.

## III. SYSTEM ARCHITECTURE

Because of the big amount of data this task comprises, moving foreground anomalous objects detection and classification from images supplied by a PTZ camera could be considered as a high computing power consuming task. However, it would be desirable to make such a system with the highest autonomy levels and as inexpensive as possible. For these reasons, apart from the mathematical model proposed in Section II a system architecture has been proposed that is capable of accomplishing this tasks with a small fraction of the computing power and electric consumption required for these kinds of systems. The proposed system architecture consists on a convolutional neural network-based object classification and tracking software, specially designed and optimized for its deployment in a microcontroller-based board that will result in a functional yet inexpensive and low power consuming system.

### A. Software architecture

System software architecture can be checked in Figure 2 and consists of a program developed in c++ which accepts a continuous stream of images supplied by a PTZ camera emulator named 'virtual PTZ' [21]. This software simulates the functionality of a real PTZ camera. More precisely, it is capable of emulating the performance of a Sony SNC-RZ50N PTZ camera when supplied with panoramic spherical video footage. In fact, all the video footage employed in this project

was obtained from a *Point Grey Ladybug 3 Spherical camera* (Figure 1) and supplied to the Virtual PTZ software.

The reason for us considering this software as a valid framework for researching is mostly its capability of replacing an actual PTZ camera by providing the operator the possibility of moving the focus all along the spherical scene, just as with a real PTZ camera, in a way that allows total control over the images that are being supplied to the program without having to face the dynamical and electrical issues affecting physical devices.



Fig. 1. 360° spherical images supplied by the *Point Grey Ladybug 3 Spherical camera*.

The second part of the software architecture is a convolutional neural network which will be in charge of detecting and characterizing the objects contained in each one of the windows guessed by the random window generator in each frame. An efficient way for working with CNNs is to select an existing Deep Learning framework that will facilitate the use and implementation of these networks. At the same time, it is very important to have in mind that the computing power of the hardware we are going to use is limited so an optimized library for deploying CNNs in microcontrollers is needed. For all these reasons, we have employed for our project the *Microsoft Cognitive Toolkit* (CNTK) combined with the *Embedded Learning Library* (ELL) by Microsoft. First one is a Deep Learning framework intended for designing and training Convolutional Neural Networks and the second one is a special library to improve the performance of the code by using a series of dlls aiming to properly parallelize every operation so it can take advantage of all the performance of these microcontroller architectures.

The third part of the software architecture described in this document is a c++ program that implements the mathematical model explained in Section II. As can be seen in the diagram proposed in Fig. 2, this program will accept a video frame supplied by the Virtual PTZ software and it will launch a fixed number of windows whose location in the frame will be determined by the result of the gaussian-uniform mixture proposed in the cited mathematical model. Next, the portion of the main frame delimited by every window will be fed to the CNN who will determine whether the guessed window is containing any object appearing in our list of anomalous objects. May that be the case, this new detection is added to the detections set ( $A$ ) and the window will be reflected and upgraded in the main frame as the camera keeps moving.

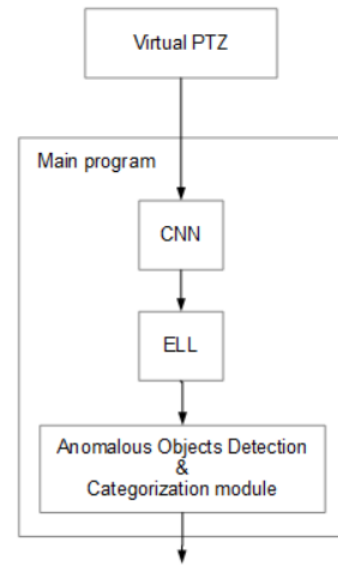


Fig. 2. Overview of the proposed implementation

### B. Hardware architecture

Hardware selection is a very important issue when it comes to microcontroller-powered Deep Learning applications. Generally, projects consisting on object detection and categorization would be supposed to have high power computing needs and a good performance, but at the same time, they should be affordable and low-energy consuming so they may be installed in places where no general power network is accessible. For all these reasons, raspberry Pi microcontroller-based boards have been considered to be used in our project. More precisely a raspberry Pi 3 Model B.

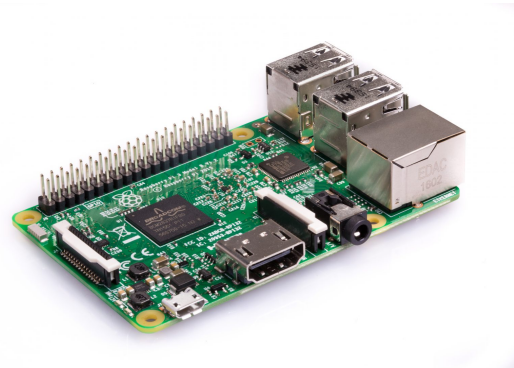


Fig. 3. raspberry Pi 3 model B overview

It presents a Broadcom BCM2837 microcontroller, featuring a CortexV8 Quad Core CPU running at 1200 Mhz from ARM, 1GB of RAM memory and a microSD data storage card. When it comes to electrical features, it can be powered by a 5.1 V power source and its max power consumption is up to 2.5 A approximately at max operating load using USB external devices.

#### IV. EXPERIMENTAL RESULTS

Section II describes our detection system as an algorithm that is capable of detecting the presence of certain number of objects belonging to certain categories considered anomalous for a particular partially controlled environment, by analyzing a video stream of the cited environment captured by a PTZ camera which is performing a panoramic movement. Frames from this video stream are supplied to a random window generation module that will extract a certain number of portions of the frame that will be fed to a convolutional neural network that will be in charge of characterizing the objects framed by the cited windows. This way, the program will be capable of determining whether anomalous objects appear in the delimited image sections and therefore, in the chosen environment at the time the video was recorded.

As was said in Section III, because of its speed and adaptation to the raspberry Pi architecture and its exploitation of parallelism, we have selected a CNN designed and trained using the convolutional neural networks implementation from Microsoft cognitive toolkit (Microsoft CNTK)<sup>1</sup> combined with the Embedded Learning Library, also from Microsoft<sup>2</sup>, which optimizes neural networks for their use in non-GPU micro-controller architectures such as raspberry Pi.

Aiming to increase our control over the experiments, tests have been performed from 16 *fps* panoramic videos supplied by the Virtual PTZ where the camera is performing a panoramic movement to the left at 16°/s angular speed. This speed also determines the  $\delta$  parameter from the detection actualization fixing it to 5 pixels/frame.

All set up, the operating of the main program follows the diagram in Figure 4. First, a new frame is acquired from the virtual PTZ software. Next, the program spreads certain amount of windows that represent potential detections all over the frame, each one localized in the place determined by the  $p$  function expressed in Eq. 6. Then, the frame's sections delimited by each one of those windows are resized and properly transformed so they can be fed to a pre-trained convolutional neural network which will determine whether the frame section delimited by the window contains any of the objects of a previously elaborated anomalous object category list. Should the neural network finds some object belonging to the anomalous objects categories in a certain window, this window's frame will appear drawn in the original input frame, and depending on the prediction confidence degree achieved it will be drawn in different colors using green for a value above 70%, yellow for a value between 40% and 70% and red for a value under 40%.

In order to achieve adequate experiments to test the accuracy of our system and to confront our mixture-based window generator against other random window generator systems, it is important to use a controlled scenario that allows us to test the methods in the same conditions. With this objective in mind, a certain video supplied by the virtual PTZ camera has

been prepared by introducing, using a video edition software, 11 random moving objects to be detected: cat, dog, laptop, hat, bird, banana, soccer ball, sunglasses, wall clock and backpack. Experiments consisted of counting the number of objects detected by the system by performing 10 360° recognition passes to the mentioned 360° modified video for a number of random windows that goes from 1 to 10. These operations were performed for each one of the two potential detections generation methods considered in this document: A gaussian-uniform mixture and a uniform distribution.

When it comes to test execution, it is prudent to constrain the amount of variable values so the amount of cases to try are not unmanageable. With this purpose, according to the mathematical model described in Section II, and leaning on our empirical results, we have fixed some values so the amount of tests that are required is less extent and the results are more significant. Thus, the damping factor  $\alpha$  will be 0.1, the standard deviation  $\sigma$ , will be assigned the value 0.5 and  $q$  will also be fixed to 0.5. For the correct performance of this system, the choice of the neural network that is going to be in charge for the detection and categorization of the anomalous objects is critical, because it has to be as simple as possible as it has to work fast enough in a low computation power microcontroller architecture but at the same time it has to be accurate enough for the system performance to be meaningful. These reasons led us to select a particular CNN designed by the Microsoft Embedded Learning Library team, for the dataset of the Large Scale Visual Recognition Challenge 2012 (ILSVRC2012), which architecture is illustrated in the Table I. It is important to remark that only the classification stage is done online using the raspberry Pi. The training of the network has been achieved offline using a *NVIDIA TITAN X* GPU. The reason for this is that the training would be orders of magnitude slower if done online with the raspberry Pi itself. This would be very inefficient and there is no need for doing it as the network can be periodically retrained offline to detect new objects and sent to the raspberry Pi so it can use it.

Experiments were performed by making the virtual PTZ to do 10 360° scans for each number of potential detections while running on the raspberry Pi. This number of potential detections goes from 1 to 10. And the results provided would be the mean number of anomalous objects detected for each number of potential detections. Once the experiments are properly performed, results for the mean number of objects detected by the two methods can be checked in Figure 5.

This figure illustrates how the gaussian-uniform mixture potential detection generator designed for this investigation achieves a higher mean number of anomalous object detections for a number equal or smaller than 3 while it achieves the same number of detections for 4 windows or higher. This makes the mixture method more accurate and functional, especially for its deployment in a raspberry Pi microcontroller where a number of potential windows higher than 3 reduces drastically the system performance.

Time performance is a critical matter when facing the design of an deep learning-based moving object detection and

<sup>1</sup><https://www.microsoft.com/en-us/cognitive-toolkit/>

<sup>2</sup><https://microsoft.github.io/ELL/>

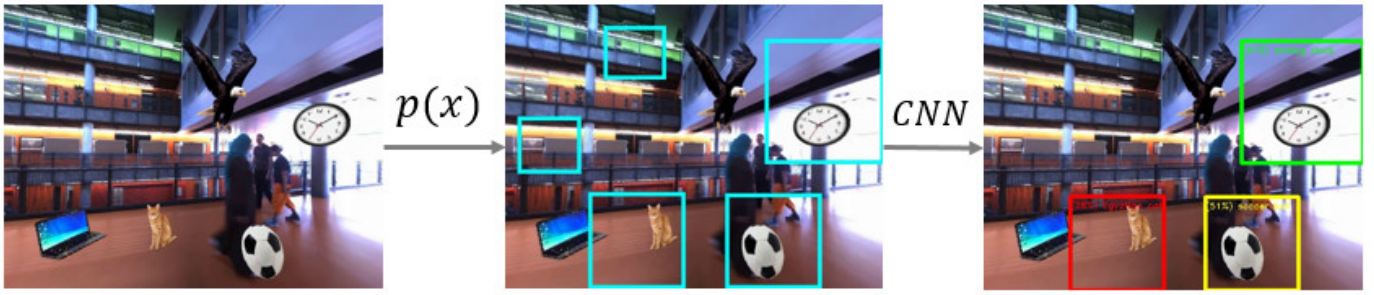


Fig. 4. Working diagram of the algorithm's regular operation mode.

<b>Accuracy ILSVRC2012</b>	80.71% (Top 5)
	57.56% (Top 1)
<b>Input</b>	256 x 256 x {B,G,R}
<b>Architecture</b>	Convolution, 224x224x16, size=3x3, stride=1
	Pooling, 112x112x16 size=2x2, stride=2
	Convolution 112x112x64 size=3x3, stride=1
	Pooling 56x56x64 size=2x2, stride=2
	Convolution 56x56x64 size=3x3, stride=1
	Pooling 28x28x64 size=2x2, stride=2
	Convolution 28x28x128 size=3x3, stride=1
	Pooling 14x14x128 size=2x2, stride=2
	Convolution 14x14x256 size=3x3, stride=1
	Pooling 7x7x256 size=2x2, stride=2
	Convolution 7x7x512 size=3x3, stride=1
	Pooling 4x4x512 size=2x2, stride=2
	Convolution 4x4x1024 size=3x3, stride=1
	Convolution 4x4x1000 size=1x1, stride=1
	Pooling 1x1x1000 size=4x4, stride=1
<b>Output</b>	ILSVRC2012 1000 classes

TABLE I  
ARCHITECTURE OF THE CNN

categorisation software, more specially when this software is going to be deployed in a low computing capability hardware such as the one described in Section III. Hence, several time performance tests have been performed after deploying the described system in the raspberry Pi microcontroller. Just as it was done with the previously illustrated accuracy tests, experiments consisted of counting the number of objects detected by the system by performing 10  $360^\circ$  recognition passes to the mentioned  $360^\circ$  modified video for a number of random windows that goes from 1 to 10. These operations were performed for each one of the gaussian-uniform mixture and the uniform distribution. Results obtained can be checked in the Table II and Fig. 6.

These results illustrate the system performance in frames per second when deployed in a raspberry Pi 3 model B. According to them, it can be observed that even though it is not capable of real-time object detection, is important to

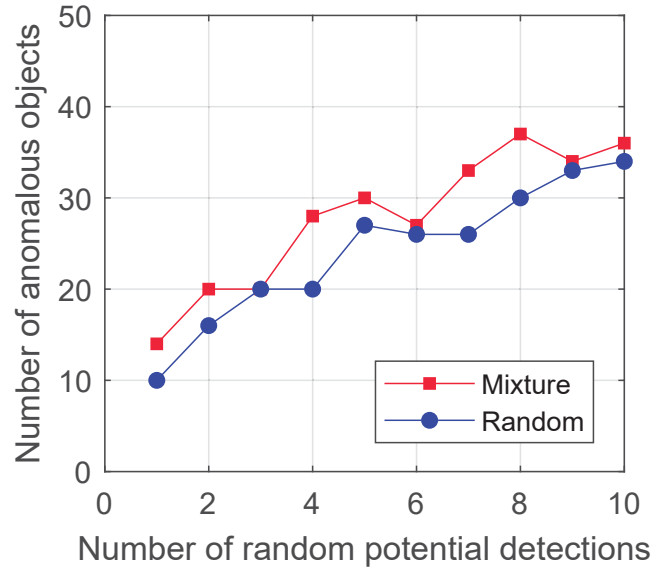


Fig. 5. Number of objects detected from the scene for each method versus number of random windows generated in each frame

remark that the system described in this paper is capable of detecting foreground objects which are in motion in a non-controlled environment in half a second approximately. For this reasons we think our proposal is justified in terms of autonomy and price/performance relation, as it can be deployed in a hardware that costs approximately 25\$. When it comes to time performance comparison, the mixture model performance seems to be similar than the model involving the uniform distribution potential detection generator. Hence, the accuracy improvement illustrated in the figures and tables also justifies the utilisation of the mixture model suggested in this document in terms of time performance.

## V. CONCLUSIONS

In this paper, an anomalous object detection system powered for microcontrollers using PTZ cameras is proposed. This system is based on a Convolutional Neural Network (CNN) for detecting and characterizing objects present in the scene. Moreover, a mathematical model has been proposed to launch a fixed number of windows in the frame, which are used to fed



# Windows	1	2	3	4	5	6	7	8	9	10
Mixture (fps)	0.607903	0.531067	0.38625	0.314169	0.267237	0.224115	0.197433	0.173732	0.161525	0.141844
Uniform (fps)	0.747384	0.519751	0.304633	0.317662	0.261643	0.20008	0.196541	0.172058	0.162285	0.1462

TABLE II

SYSTEM PERFORMANCE EXPRESSED IN MEAN FRAMES PER SECOND VERSUS NUMBER OF POTENTIAL DETECTION GENERATIONS FOR MIXTURE MODEL AND UNIFORM MODEL.

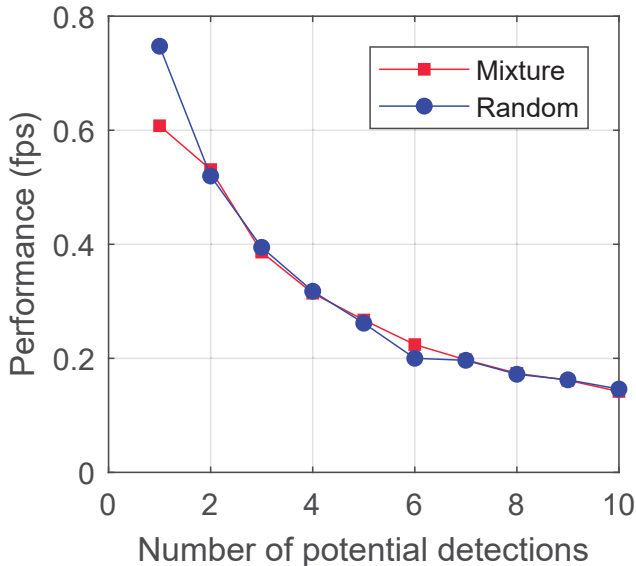


Fig. 6. System performance in fps for both random window generation models versus number of potential detection number

the CNN and to track anomalous objects. On the other hand, images supplied by a PTZ camera can be high computing power consuming. However, our proposal is a low energy and low cost system since it has been implemented in a raspberry Pi microcontroller. Experimental results confirm the good performance of our low-cost proposal.

#### ACKNOWLEDGMENT

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. It is also partially supported by the Autonomous Government of Andalusia (Spain) under projects TIC-6213, project name Development of Self-Organizing Neural Networks for Information Technologies; and 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.

#### REFERENCES

- [1] C. Chen, S. Li, H. Qin, and A. Hao, "Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis," *Pattern Recognition*, vol. 52, pp. 410 – 432, 2016.
- [2] H. Sajid, S.-C. S. Cheung, and N. Jacobs, "Appearance based background subtraction for PTZ cameras," *Signal Processing: Image Communication*, vol. 47, pp. 417 – 425, 2016.
- [3] J. Huo, Y. Gao, W. Yang, and H. Yin, "Multi-instance dictionary learning for detecting abnormal events in surveillance videos," *International Journal of Neural Systems*, vol. 24, no. 03, p. 1430010, 2014.
- [4] R. Mesquita and C. Mello, "Object recognition using saliency guided searching," *Integrated Computer-Aided Engineering*, vol. 23, no. 4, pp. 385–400, 2016.
- [5] B. Lacabex, A. Cuesta-Infante, A. S. Montemayor, and J. J. Pantrigo, "Lightweight tracking-by-detection system for multiple pedestrian targets," *Integrated Computer-Aided Engineering*, vol. 23, no. 3, pp. 299–311, 2016.
- [6] T. Boulton, X. Gao, R. Micheals, and M. Eckmann, "Omni-directional visual surveillance," *Image and Vision Computing*, vol. 22, no. 7, pp. 515–534, 2004.
- [7] K.-T. Song and J.-C. Tai, "Dynamic calibration of pan-tilt-zoom cameras for traffic monitoring," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 5, pp. 1091–1103, 2006.
- [8] C. Micheloni, B. Rinner, and G. Foresti, "Video analysis in pan-tilt-zoom camera networks," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 78–90, 2010.
- [9] C. Ding, B. Song, A. Morye, J. Farrell, and A. Roy-Chowdhury, "Collaborative sensing in a distributed PTZ camera network," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3282–3295, 2012.
- [10] C. Ding, J. H. Bappy, J. A. Farrell, and A. K. Roy-Chowdhury, "Opportunistic image acquisition of individual and group activities in a distributed camera network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 664–672, March 2017.
- [11] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, no. November 2016, pp. 11–26, 2017.
- [12] M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, Nov 2017.
- [13] L. Tong, F. Dai, D. Zhang, D. Wang, and Y. Zhang, "Encoder combined video moving object detection," *Neurocomput.*, vol. 139, pp. 150–162, Sep. 2014.
- [14] P. Angelov, P. Sadeghi-Tehran, and C. Clarke, "AURORA: Autonomous real-time on-board video analytics," *Neural Comput. Appl.*, vol. 28, no. 5, pp. 855–865, May 2017.
- [15] A. Dziri, M. Duranton, and R. Chapuis, "Real-time multiple objects tracking on raspberry-pi-based smart embedded camera," *Journal of Electronic Imaging*, vol. 25, p. 041005, 04 2016.
- [16] M. K. Dobrzynski, R. Pericet-Camara, and D. Floreano, "Vision tape-a flexible compound vision sensor for motion detection and proximity estimation," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1131–1139, May 2012.
- [17] V. Fung, J. L. Bosch, S. W. Roberts, and J. Kleissl, "Cloud shadow speed sensor," *Atmospheric Measurement Techniques*, vol. 7, no. 6, pp. 1693–1700, 2014.
- [18] L. Adnan, Y. Yusoff, H. Johar, and S. Baki, "Energy-saving street lighting system based on the waspmote mote," *Jurnal Teknologi*, vol. 76, no. 4, pp. 55–58, 2015.

- [19] F. Ortega-Zamorano, M. A. Molina-Cabello, E. López-Rubio, and E. J. Palomo, "Smart motion detection sensor based on video processing using self-organizing maps," *Expert Systems with Applications*, vol. 64, pp. 476 – 489, 2016.
- [20] 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*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2017, vol. 10338 LNCS. [Online]. Available: [www.scopus.com](http://www.scopus.com)
- [21] G. Chen, P. St-Charles, W. Bouachir, G. Bilodeau, and R. Bergevin, "Reproducible evaluation of pan-tilt-zoom tracking," in *Proceedings - International Conference on Image Processing (ICIP)*, 2015, pp. 2055–2059.