Improving a Low Cost Surveillance System

Carlos Sebastián Castañeda, María José Abásolo[12]

[1] III-LIDI, Facultad de Informática, Universidad Nacional de La Plata, Argentina
[2] CICPBA, Buenos Aires,, Argentina
mjabasolo@lidi.info.unlp.edu.ar

Abstract. The purpose of this work is improving the functionality and usability of a low cost commercial surveillance system. The original system provides simple motion detection and sends alert messages by means of FTP or email. The modified system adds a software layer to the original system for implementing desirable image processing features. Particularly, people detection functionality was implemented by means of OpenCV Oriented Gradient Histograms. Also the modified system adds the use of Telegram messaging service for sending alerts. We compare the performance of both original and modified systems regarding the intruder detection.

Palabras clave: computer vision, motion detection, people detection, video surveillance systems, computer vision

1 Introduction

Monitoring and control of different areas, stories such as public places, banks, shopping malls, shops, airports has become a growing need in these times. With technological advances in the area of security and the availability of low-cost hardware, the use of security systems has become something of regular use in homes and small businesses. In general, the installation of security cameras does not imply a great economic cost, but the hiring of people who can monitor these cameras 24 hours a day does. One of the main goals is having automated surveillance systems that require little human interaction for their operation and detection of potential threats.
It is possible to acquire fixed security cameras at low cost, but a problem that these devices can present is that the embedded motion detection algorithm [1] is very sensitive to changes in the environment, such as changes in lighting conditions and shadows projections. In general, cameras implement motion detection based on background subtraction techniques [2-7] and difference between frames [8-10].
In the background subtraction technique, the difference between the current frame and the frame representing the image background is used to detect moving objects. In this method, it is necessary to maintain a model that represents the background of the image. A background model could be obtained, for example, from the average image over a certain training period. Moving objects are easy to detect using this technique, but the background model must be updated regularly. In its simplest version, this technique is simple to implement but has the disadvantage of being very sensitive to changes in the environment, making it difficult to isolate the interference of the real

movement of objects in the image. Bottom subtraction is the most commonly used technique in fixed security cameras. The statistical model based on background subtraction is flexible and fast but the camera must be stationary.
In the algorithms based on difference between frames, the presence of moving objects is determined through the difference between consecutive frames of the image. The absolute difference in each pixel is calculated, between two to three consecutive frames of the video sequence, and a threshold is applied to obtain the moving objects. This method is highly adaptive to dynamic and computationally less complex environments, but is generally not accurate in extracting the full form of certain types of moving objects. In particular, the main causes of incorrect detections are due to ghost images, close-up apertures with objects that have a homogeneous color
On the other hand, there are different techniques or algorithms in the bibliography to detect people in images or video streams. Viola et al. in [11] he proposes to use cascade classifiers, a method widely used in real-time applications. This method has been extended to employ different types of characteristics and techniques, but fundamentally the cascade concept has been used to achieve real-time detection. One of the most popular features used for human detection are Oriented Gradient Histograms (OGH), developed by Dalal et al. [12].
This work aims to improve a low-cost camera-based surveillance system that implements motion detection based on background subtraction. It is proposed to improve the user interface and add post-processing of images related to alerts, in particular to detect people, being able to filter false positives or increase the level of the alert sent to the user. The rest of the article is organized as follows: in section 2 the original system tests are described. In section 3 the proposed system for improving the original one is explained. Section 4 shows the tests carried out to evaluate the intruder detection alerts that were incorporated. Finally, section 5 presents the conclusions and future work.

2 Low cost commercial surveillance system

2.1 System description

We analyze a TP-LINK model NC220 camera[1] which has the following features: low cost, wireless connectivity, Wi-Fi signal amplifier, night vision, motion and sound detection that can be enabled and disabled, sending alert notifications by email or FTP, application for live video streaming, for Android, IOS, Windows platforms or through the TP-LINK cloud platform[2] .
The camera's embedded motion detection system can be configured with three levels of sensitivity. The execution of motion detection algorithms embedded in cameras and digital video recordings depends on the preselected level of sensitivity which works for all possible environmental conditions. For this reason these systems have an error rate that includes false alarms (false positives) and the omission of true alarms (false negatives). A low sensitivity level can cause more false negatives, while a high sensitivity level can cause more false positives. It is also possible to configure the camera vision in day, night and automatic mode. In daylight mode, the camera shoots

[1] http://www.tp-link.es/products/details/cat-19_NC220.html
[2] https://www.tplinkcloud.com/

in colors, but good lighting conditions are necessary. In night mode the camera shoots in gray scale and is capable of recording video with very little or directly without light in the environment. In automatic mode the camera switches between day and night modes depending on the amount of light in the environment in every instant. In an environment where lighting conditions are not constant, the best option will be to configure the camera in automatic mode.
Figure 1 shows the operating scheme of the original system. The camera is connected to a Wi-Fi internal network through which an integrated application is accessed that allows enabling/disabling alerts. Motion detection alerts are sent to an email or to an FTP server which can be accessed directly to inspect the corresponding images.

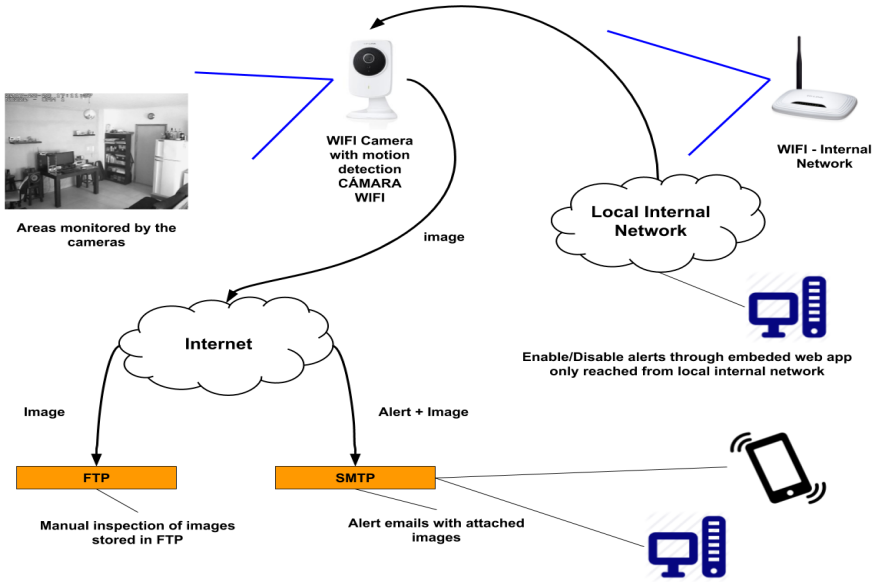


Figure 1. Commercial surveillance system with TP-LINK nc220 cameras (own source)

2.2 Test description

For the tests a camera was installed in an apartment, directing the lens to the main entrance. The tests were carried out for 185 days, collecting the images sent by the camera when the department was empty and the system activated, and also forcing the activation of motion detection by entering the department without disabling.. During this period 1683 images were collected. For each motion detection event the camera sends a sequence of several consecutive screenshots. Each sequence was considered as a single sample, thus reducing the 1683 images to 309 sequences or samples.

2.3 Motion detection

The first test carried out consisted of counting the number of true and false positives registered by the camera. We consider positive samples to those sequences where people are observed, or there was a real movement of some object (for example: the opening of the apartment door without the entrance of any people), or also there was a

sudden change in luminosity due to turning the light on. In the last type of sequences the images are completely burned by light. The negative samples are those where there was no real movement in the scene, and the alerts were caused by light changes in the environment light, brightness and reflections of the objects in the monitored scene. The result of the classification of the sequences are shown in Table 1. The camera reported 35.60% false positives, with the remaining 64.40% reporting true positives, where 88.44% of them correspond to people detection.

Table 1: Summary of the data obtained in the evaluation test of the embedded movement detection algorithm in the camera

Total number of sequences	**309**	**100%**
Positive sequences (true positives)	**199**	**64.40%**
People	176	88.44%
real object motion (door open)	11	5.52%
high light change (light were turned on manually)	12	6.03%
Negative sequences (false positives)	**110**	**35.60%**
day (natural light change)	86	78.18%
night (object reflection or shine)	14	12.72%
night (alert reason not clear in the image)	10	9.09%

3 Proposed system

The objective is to improve the original commercial system described regarding to the following:
- Adding the processing of the images resulting from the motion detection provided by the camera to emphasize alerts;
- Replacing the email based notification system by a messaging notification system.
The new software architecture adds a web application to the original system, and can be seen in figure 2. The original system sends to an FTP server the corresponding image when a motion detection event occurs. The images are processed with an image processing module. Particularly we implement an intruder detection algorithm to confirm the presence of people. The alert is then classified according to the results of the processing (false or true positive), and the image is sent jointly with a text notification to the instant messaging program, particularly Telegram. These alerts can be read from almost any current mobile device. From the instant messaging program it is also possible to send a command to enable/disable the reception of notifications.

3.1 Web Application

All the modules of the system are contained in a web application implemented with DJANGO[3] which is a framework for the development of web applications in Python, which provides a large number of utilities for developing in an organized and agile way. The implemented Django project consists of four modules:

- Django Admin: this utility is provided by the framework, and makes it

[3] https://www.djangoproject.com

possible with a minimum code configuration to register the models defined in the different applications of the project and generate an Administrative Web Application that allows the basic ABM of the entities to be carried out.

- FTP server module
- Image processing module
- Notifications module

The code of our implementation can be downloaded from GitHub[4].

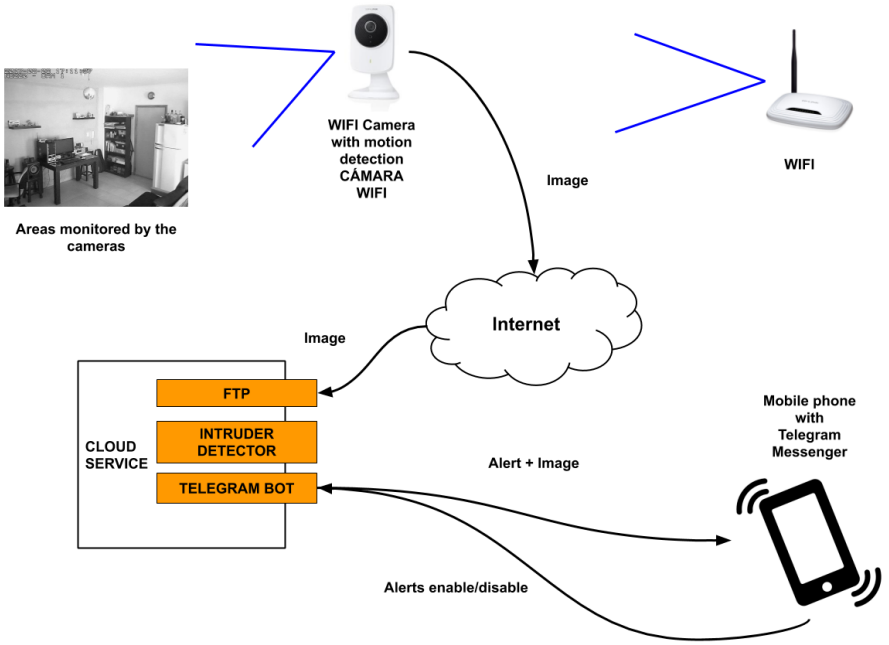


Figure 2. Proposed architecture system that interacts with the TP-LINK nc220 security camera

3.2 FTP server

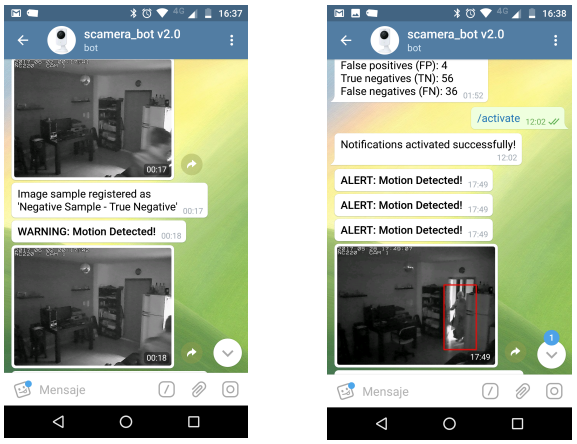
For our purpose the option of sending the screenshots to an FTP server was set in the original system. This module implements the FTP server that can be run through a command. It defines the model that represents the server users and interacts with the framework's authentication and authorization engine to validate incoming connections. A mechanism is needed to trigger an event, upon reception of a new image sent by the camera. The pyftpdlib library[5] was used, which allows customizable implementation of an FTP server. This library is developed entirely in Python and allows redefining hook methods that represent the different significant events that can occur on an FTP server.

3.3 Notifications module

This module implements the attention to commands from users and sending alerts

[4] https://github.com/seba3c/scamera
[5] https://github.com/giampaolo/pyftpdlib

when an image is received on the FTP server. Telegram is used as the instant messaging application. It has similar features and benefits to WhatsApp but it has the advantage of being open source software that provides a set of APIs to implement, among other things what is known as chatbots[6]. A chatbot is basically a program that responds to users' petitions. It can be implemented with different levels of complexity, either from one capable of processing text in natural language and simulating speaking with another person, to the simplest chatbot that can respond to a set of predefined commands. The notifications module allows receiving commands from Telegram users registered in the system and sending notification after an image is processed. Notifications range from warnings to alerts according the results of the image processing (figure 3).



a) b)

Figure 3. Notifications received in the Telegram chatbot. a) Warning notification is received when people are not detected; b) Alert notification is received when intruder was detected and it is framed in red in the image (own source)

3.4 Image processing module

The image processing module receives the images sent by the camera by FTP, processes each image with some desirable image processing algorithm, and enriches the information to be sent to the notification system. The code was structured in such a way that it is possible to add new detection algorithms. Particularly, our objective was to implement an intruder detector. There are different techniques or algorithms in the bibliography to detect people in images or video streams, among them is the OGH algorithm proposed by Dalal and Triggs [12] in conjunction with Least Square Vector Machina (LSVM). In this algorithm, image descriptors are extracted and a supervised classifier is trained to determine if it contains people. There are available implementations of this algorithm in OpenCV, such as that of Rosebrock [13] using the Python programming language.

When the algorithm result is positive a message of greater relevance is sent to the notification module, together with the enlarged image in which the detected people

[6] https://core.telegram.org/bots/api

are highlighted.
At first it was necessary to calibrate parameters of the algorithm, conducting a set of tests adjusting the value of each parameter until reaching the best balance between detection accuracy and processing speed. In this way, the parameters of the intruder detector were configured with the initial values suggested by Rosebrock [14], and the algorithm was iteratively executed on the smallest set of samples, gradually varying the values of parameters such as scale, padding, winStrinde and non-maxima suppression threshold. In each case it is observed if the accuracy and the image execution time increase or decrease. For those tests where the best relationship between accuracy and execution time was obtained, the algorithm was run again on a large set of samples, in order to confirm the results obtained.

4 Intruder detection effectiveness

4.1 Tests and Metrics

To measure the performance of the algorithm, the metrics of a binary classifier, presented by Fawcett [15], since the algorithm has two possible outputs or categories: the presence or absence of intruders (see Table 2).

Table 2: Metrics of the intruder classification

Metric	Formula	Value
Accuracy (ACC)	(TP + TN) / S	0.76
Sensitivity or True Positive Rate (TPR)	TP / P	0.71
Specificity or True Negative Rate (TNR)	TN / N	0.83
Balanced accuracy (BACC)	(TPR + TNR) / 2	0.77
Precision or Positive Predictive Value (PPV)	TP / (TP + FP)	0.87
Negative Predictive Value (NPV)	TN / (TN + FN)	0.65
Fall-out or False Positive Rate (FPR)	1 - TNR	0.17
False Discovery Rate (FDR)	1 - PPV	0.13
Miss rate or False Negative Rate (FNR)	1 - TPR	0.29
TP: True Positives or hits to detect people		559
TN: True Negatives or correct rejections		420
FP: False Positives or false alarm		83
FN: False Negatives or missed alarm		227
P = TP + FN, all positive samples N = FP + TN, all negative samples S = P + N, all samples		

The parameters of the intruder detector were configured with the initial values suggested by Rosebrock [15], and the algorithm was iteratively executed on a small set of samples, gradually varying the values for the parameters. The test samples consist of 1289 images, which 786 are positives and 503 are negatives. Table 2 shows the values of the metrics. Accuracy value means that 76% of the total samples were correctly classified. There are 6% of false alarms (FP) and 18% of missed alarms (FN) on the total samples. An image classified as positive generates an alert

notification, 87% of which corresponds to true alerts as indicated by PPV, and 13% are false alerts that should be just warnings as indicated by FDR. As indicated by sensitivity (TPR) 71% of the positive samples are correctly classified while its counterpart miss rate (FNR) indicates that 29% of the positive samples are not. An image classified as negative generates a warning notification, 65% of which corresponds to true warnings as indicated by NPV, and 35% are warnings that should be alerts. Specificity (TNR) value indicates that 83% of the negative samples are correctly classified while its counterpart FPR indicates that 17% are not.

4.4 Real Time testing

The next test consists of activating the intrusion detection module and evaluating it in conjunction with the rest of the system. In order to measure the performance of the detection algorithm in real time, the possibility of manually indicating the result of each alert received was added to the Telegram chatbot. Every time the chatbot receives an image, a keypad is displayed with the following options:

- PS-TP (Positive Sample- True Positive)
- PS-FN (Positive Sample - False Negative)
- NS-FP (Negative Sample - False Positive)
- NS-TN (Negative Sample - True Negative)
- DISCARD: this option has been added for certain cases where the sample cannot be evaluated and it is not desirable to affect the metrics. This situation occurs, for example, when the camera sensor cannot adapt to a sudden change in light and the recorded image is completely burned, making it impossible even to visually distinguish people or objects.

Figure 4 shows four types of images that were found in this test, excluding the type of discarded images.

a) b) c) d)

Figure 4. a) True Positive sample; b) False Negative sample; c) False Positive sample; d) True Negative sample

Two different tests were carried out using night and automatic camera mode respectively. A total of 320 images were collected, 161 images in night mode and 159 images in automatic mode. Table 3 summarizes the results obtained from both tests. The accuracy value of 0,76 is very similar to that obtained in the parameter adjustment tests. The results of both samples, with night and automatic camera mode, are very similar. There are 2% of false alarms (FP) on total alarms and 21% of the alarms were missed (FN). An image classified as positive generates an alert notification, 95% of which corresponds to true alerts (PPV) and 5% are false alerts (FDR). As indicated by sensitivity (TPR) 65% of the positive samples are correctly classified while its counterpart miss rate (FNR) indicates that 35% of the positive

samples are not. An image classified as negative generates a warning notification, 62% of which corresponds to true warnings (NPV) and 38% are warnings that should be alerts. Specificity (TNR) value indicates that 95% of the negative samples are correctly classified while its counterpart FPR indicates that 5% are not.

Table 3: Image classification and metric values obtained in the tests of night and automatic camera mode

	Night	Automatic	Total
Image classification			
TP	61	66	127
TN	60	56	116
FP	4	3	7
FN	36	34	70
Discarded	0	18	18
Total	161	159	320
Metrics			
Accuracy ACC	0,75	0,76	0,76
Sensitivity (TPR)	0,63	0,66	0,65
Specificity (TNR)	0,94	0,97	0,95
Balanced accuracy BACC	0,78	0,81	0,80
Precision (PPV)	0,94	0,96	0.,95
Negative predictive value (NPV)	0,62	0,61	0,62
Fall out (FPR)	0,06	0,04	0,05
False discovery rate (FDR)	0,06	0,04	0,05
Miss rate (FNR)	0,37	0,34	0,35

5. Conclusions and future work

In this work, we present how to improve a low-cost commercial security camera system which can be useful for monitoring homes and businesses. The characteristics of the equipment were analyzed and some problems were identified in the experience of interaction with the system, in particular: the mechanism to activate and deactivate alerts, and the effectiveness of intruder detection.

The implementation of a software system was proposed, which interacts with the commercial system, adding both a level of image processing and a Telegram based communication between the user and the system. The Python code is open source and it was shared to download. As a level of image processing, we particularly test the detection of intruders using a people recognition algorithm based on OGH, using an OpenCV implementation. The proposed system improved the alerts reported by the camera by means of the post processing algorithm.

As potential future work, it is proposed to add implementations of other algorithms for image processing. Particularly, methods of detecting people based on neural networks or methods based on machine learning are being incorporated..

References

1. R. S. Shirbhate, N. D. Mishra, and R. Pande, “Video Surveillance System Using Motion Detection: A Survey,” *Advanced Networking and Applications*, 2012
2. Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” in *Proceedings of the 17th International Conference on Pattern Recognition*, Cambridge, UK, 26 de agosto, 2004.
3. Z. Zivkovic and van der H. F., “Efficient adaptive density estimation per image pixel for the task of background subtraction,” *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, May 2006.
4. P. Kaewtrakulpong and R. Bowden, “An Improved Adaptive Background Mixture Model for Real Time Tracking with Shadow Detection,” in *Proceedings 2nd European Workshop on Advanced Video Based Surveillance Systems*, 2001.
5. Sobral and A. Vacavant, “A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos,” *Computer Vision and Image Understanding*, no. 122, pp. 4–21, 2014.
6. J. Dou, Q. Qin, and Z. Tu, “Background subtraction based on circulant matrix,” *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, vol. 11, no. 3, pp. 1–8, 2017.
7. S. Brutzer, B. Hoferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1937–1944.
8. N. Singla, “Motion Detection Based on Frame Difference Method,” *International Journal of Information & Computation Technology*, 2014
9. S. S. Sengar and S. Mukhopadhyay, “A novel method for moving object detection based on block based frame differencing,” in *In 3rd International Conference on Recent Advances in Information Technology*, 2016, pp. 462–472.
10. M. Fei, J. Li, and H. Liu, “Visual tracking based on improved foreground detection and perceptual hashing,” *Neurocomputing*, vol. 152, no. C, pp. 413–428, Marzo, 2015.
11. Viola, Jones, and Snow, “Detecting pedestrians using patterns of motion and appearance,” in *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice, France, 2005
12. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005.
13. A. Rosebrock, “Pedestrian Detection OpenCV”, 2015.http://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv
14. A. Rosebrock, “HOG detectMultiScale parameters explained,” 16 de Noviembre, 2015. http://www.pyimagesearch.com/2015/11/16/hog-detectmultiscale-parameters-explained/.
15. T. Fawcett, “Introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.