Deep Learning for Automatic Detection of Handguns in Video Sequences

Youssef Elmir[1,2][0000−0003−3499−507X], Sid Ahmed Laouar[1], and Larbi Hamdaoui[1]

[1] Department of Mathematics and Computer Science, Tahri Mohammed University, Béchar 08000, Algeria http://www.univ-bechar.dz

[2] Smart Grid and Renewable Energies Lab (SGRE), Tahri Mohammed University, Béchar 08000, Algeria elmir.youssef@yahoo.fr

Abstract. Computer vision is a branch of artificial intelligence (AI) whose purpose is giving machine the ability to understand what it ”sees” when it is connected to one or more cameras, the fact that make computer vision used for pattern recognition. Current monitoring and control systems still require human monitoring and intervention and their performance depends relatively to human attention. This work presents a system of automatic detection of handguns in videos, suitable for both surveillance and control. We re-formulate this problem of detection in the problem of minimizing false positive detection and solving it using deep convolutional neural network (CNN). This project consists of a study of different online handgun detection methods. The method based on supervised deep learning has proved its performance and the results obtained are very encouraging regarding related works despite the technical difficulties (material) encountered during the realization of the experiments.

Keywords: Deep learning · Moving object detection · Computer vision.

1 Introduction

We live in a digital world, where information is stored, processed, indexed and searched using computers, the fact that makes recovery a quick and cheap task. In recent years, considerable progress has been made in the area of object detection. This progress is due the extensive work in this area and the availability of international image databases for machine learning that have allowed researchers to credibly report the simulation of their approaches in this area.

The increase in the use of digital images, motion analysis and object detection in videos has proved that is an indispensable tool for applications as well as video surveillance.

Public safety is a major concern in today's modern society. Weapons creates serious threats to the safety and security of ordinary people, even in the most public places. So dangerous situations in major events may not be avoided by human operators. The question is how can artificial intelligence based machines be used for total security solution?

The use of online automatic handguns detection can enhance the performance of surveillance method with a promising application of deep learning. Recently, this machine learning technique has achieved good performance to classical techniques such as the naive bayes, decision tree and even with other types of deep learning algorithms like recurrent neural network in image classification, detection and segmentation [9][6][11][2][14][13]. Furthermore, similar work has been proposed in [10] in which, the best detector provides satisfactory results as automatic alarm system. But, a proper training of deep CNNs, which contains millions of parameters, requires very large datasets, in the order of millions of samples, as well as High Performance Computing (HPC) resources, e.g., multi-processor systems accelerated with graphics processor unit (GPUs).

To overcome these constraints. We proposed to investigate other types of deep learning model. It consists of using Mobile dedicated model [7]. We aim to develop an online handgun detector in videos using soft deep CNNs.

The purpose of this paper is to present a system that allows fast and reliable processing of high quality video data and can thus detect and react to the presence of a handgun.

In the rest of this paper, the second section describes and studies motion detection methods and CNNs based model is proposed as well as their interest in the field of handguns detection. In section three, experimental part of this work is presented with discussion of different obtained results. The last section gives a general conclusion about the proposed work.

2 Proposed model

Handgun detection consists in object recognition and finding its position in static image or in video sequence. In the context of this work, a basic modeling of CNNs presented in Fig. 1, is proposed and evaluated using the "Handgun Dataset for the sliding window approach" and "Handgun Dataset for the region proposals approach" [10] databases for handguns detection.

The first step is the acquisition of images. It takes place in a minimum of two stages, taking initial images and taking subsequent images. Motion detection can not be done without the subsequent capture of images. The care taken in taking pictures is crucial for the success of the registration because the quality of the final results depends on it.

2.1 Motion Detection

Unlike in static images, motion detection is an essential process for handguns detection in a video sequence. As it is a very expensive process in terms of computation. To avoid unnecessary triggering of handguns detection process in case of no motion in the video, a few simple image processing operations are launched to detect the moving object if applicable. [5]

Differential images [12] are the result of the subtraction of two images:

$$Gdif(x,y) = g1(x,y) - g2(x,y) \tag{1}$$

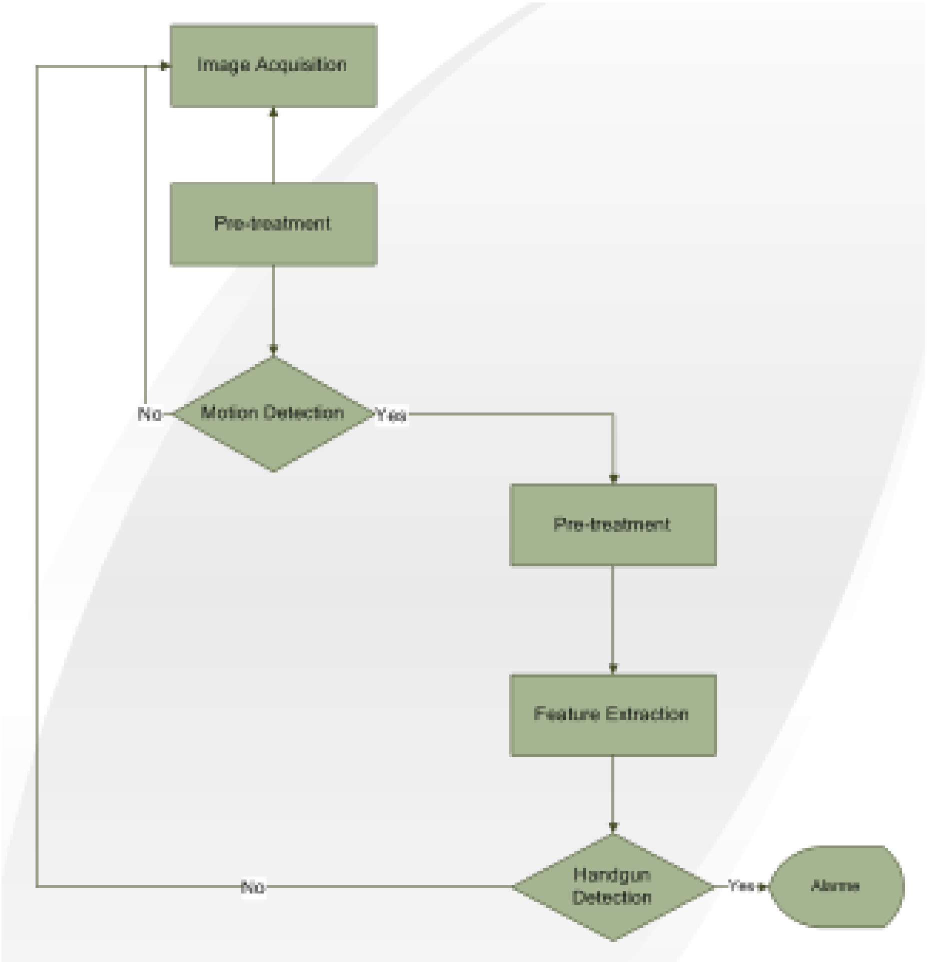

Fig. 1. General block-diagram of the proposed model.

A differential image shows the difference between two images and makes the movement visible. A differential image is calculated from three consecutive images I_{t-1}, I_t and I_{t+1}. The advantage of this method, is that non-useful background is removed from the result (static information):

$$\Delta I_1 = I_{t+1} - I_t, \Delta I_2 = I_t - I_{t-1}, \Delta I = \Delta I_1 \wedge \Delta I_2 \tag{2}$$

Practically, three images are captured at times t-1, t, t+1 to calculate the differences ΔI_1 and ΔI_2. ΔI_1 returns the absolute difference between the last two images, while ΔI_2 returns the absolute difference between the first two images. Finally, the ΔI is calculated between the bits of ΔI_1 and ΔI_2.

Gray level conversion and noise elimination . Before doing any operation with captured images, it is necessary to convert them to gray level. It is less complex and more optimal to work with this type of images. On the other hand, it is necessary to minimize the noise caused by the camera itself and by the lighting. This is done by averaging each pixel with its neighbors.

Threshold application . In this part of the process, the goal is to convert the image to binary, that is, to have two possible values. All pixels that exceed the threshold will be considered as white pixels and other pixels will be considered as black pixels. This will help to locate the moving object [1].

Contour detection . Once the region of interest (ROI) in the image is obtained, the outlines must be detected. At the end of this image, if there is no contour detected, it is considered that there is no movement and the process returns to the acquisition step, otherwise it launches the process of handguns detection [4].

2.2 Handguns Detection

This process is based on a computer vision module that uses a CNN based model [8] to detect handguns.

The CNN is defined as a collection of nodes, where a tensor is given as input and another tensor returned at the output of the last nodes. The input tensor is the input image and the output tensor will be the binary classification label for detection or non detection.

The detection of handguns addresses several solutions to the problem of detecting handguns in real time, but the biggest difficulty is at hardware level. This forced us to do the experiments according to three models:

1. CNN based model.
2. Fast R-CNN based model[3].
3. MobileNet CNN based model[7].

2.3 First Model Architecture

The first model presented in fig. 2. is composed of five convolution layers, two pooling layers with maximum output value and three fully connected layers.

The input image is 32 x 32, firstly, the image goes to the first convolution layer. This layer is composed of 32 filters of size 3 * 3, each of the convolutional layers is followed by a rectified linear unit (ReLU) as function of activation, this function forces the neurons to return positive values, after this convolution 32 features of size 32 * 32 will be created.

The previously obtained features are sent to the input of the second convolution layer which is also composed of 32 filters, a RELU activation function is applied on the convolution layer, then a pooling is applied to reduce the size. of the image as well as the amount of parameters and calculation. At the exit of this layer, we will have 32 features of size 16 * 16. The same thing is repeated with three, four and five convolution layers, these layers are composed of 64 filters, the ReLU activation function is always applied on each convolution.

A pooling layer is applied after the five convolutional layer. At the exit of this layer, we will have 64 features of size 8 * 8. The feature vector resulting from the convolutions has a dimension of 4096.

After these five convolutional layers, we use a neural network composed of three fully connected layers. The first two layers each have 1024 neurons where the activation function used is the ReLU, and the third layer is a normalized exponential function that calculates the probability distribution of the 100 classes (number of classes in Handgun Dataset for the sliding window approach).

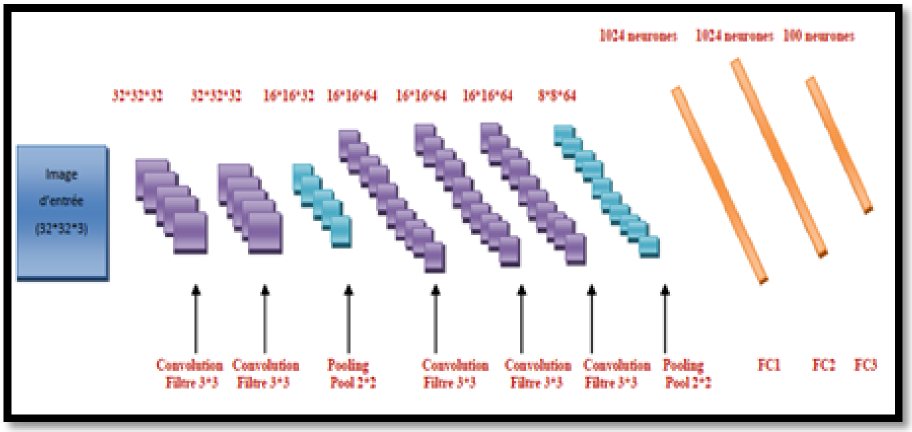

Fig. 2. First Model Architecture.

2.4 Second Model Architecture

The second model presented in fig. 3. receives proposals from regions from an external system (selective search). These proposals will be sent to a pooling layer of the ROI that will resize all regions with their data at a fixed size. This step is necessary because the fully connected layer expects all vectors to be the same size.

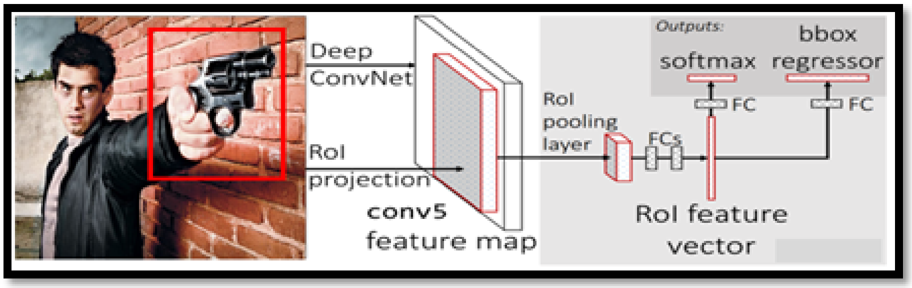

Fig. 3. Second Model Architecture.

2.5 Third Model Architecture

This model has the same architecture as the second model but with some adjustment in the configuration of learning parameters.

3 Experiments

To evaluate the three proposed models, two databases are used for learning phase.

Handgun Dataset for the sliding window approach. The training data set, suitable for classification task, consists of 102 classes with a total of 9261 images. Handgun class at 200.

Fig. 4. Samples from Handgun Dataset for the sliding window approach.

Handgun Dataset for the region. The training dataset, suitable for detection task, contains 3000 handguns images with a rich context.

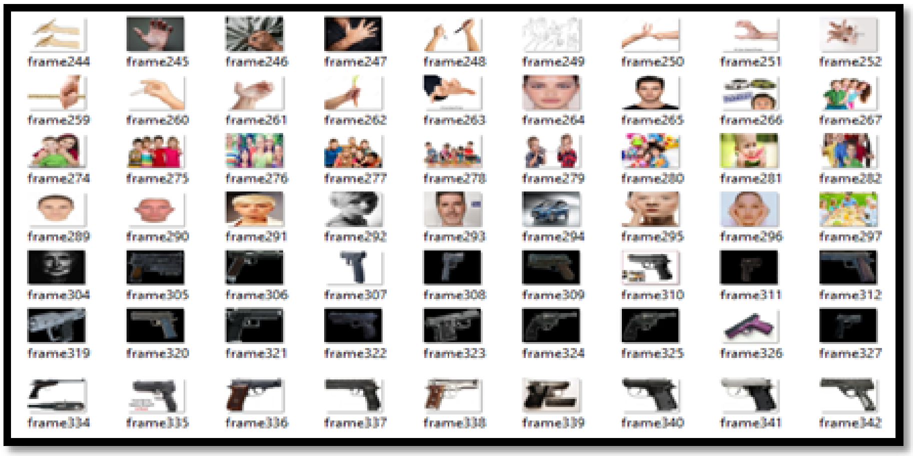

Fig. 5. Samples from Handgun Dataset for the region proposals approach.

Test dataset for classification and detection. A total of 608 images including 304 are images of handguns.

Learning of models is done using a sample of 420 images from the Handgun Dataset database for the region proposals approach. The first model is tested with 608 images (304 handgun images, 304 non-handgun images), the third model with 200 images and the third model with 420 images.

4 Obtained Results

The results obtained are improved as much as the network learns the database by increasing the number of learning epochs. Learning database is also a determining element in deep learning, large learning database could help to achieve better results. After analyzing the results obtained, we notice the following:

From Table 1 , Learning error and validation decreases in correspondence with the number of epochs.

For the first model, we notice that all the misclassified images are 274 images, which generate an error rate of 45% and the totality of the well classified images is 334 with an accuracy rate of 55%. This unsupervised classification model does not give a good result because it needs more learning epochs and the available hardware configuration does not allow this option. On the other hand, for the second model, all the misclassified images are 40 images, an error rate of 20% and the totality of the well classified images is 160 with an accuracy rate of 80%. This model gives a good result on the static images but for real time test (using webcam), it also needs a very powerful hardware configuration (Computation by GPU, ... etc.). The third model has all misclassified images of 42 images, an error rate of 10% and the totality of the well classified images is 378, an accuracy rate of 90% for the static images and with a good performance for real time test (see

Fig. 6). Regarding the time needed for learning, it is clear that all models needs to perform the learning phase using very powerful hardware, but, for execution (test) the third model can perform better than the other two models.

Table 1. Comparison of the three models.

Model	Number of Images	Number of Well Classified Images	Number of Misclassified Images	Learning Time (Hours)	Accuracy (%)	Error Rate (%)
CNN	608	334	274	5	55	45
Fast R-CNN	200	160	40	24	80	20
Mobile-SDD	420	378	42	192	90	10

Table 1 shows the different results obtained by the three models.

For implementation of the two fast R-CNN and MobileNet-CNN models, the learning is relaunched with the Handgun Dataset base for the region's proposals (3000 images) and the test is carried out with 608 images including 304 images of handgun and 304 non-handgun images. The learning period was 48 hours. Both models give a good result on static images with good performance in real time.

Table 2. Obtained Results.

Model	TP	FN	TN	FP	P	R	F1
Fast R-CNN	232	72	248	56	80,76%	76,31%	78,37%
MobileNet-CNN	156	54	168	42	78,78%	74,28%	76,46%
[10]	304	0	247	57	84,21%	100%	91,43%

$$precision = \frac{(TruePositives)}{(TruePositives + FalsePositives)} \tag{3}$$

$$recall = \frac{(TruePositives)}{(TruePositives + FalseNegatives)} \tag{4}$$

$$F1measure = 2 * \frac{(precision * recall)}{(precision + recall)} \tag{5}$$

• True Positives (TP): This is the number of images where the process detects a handgun among 304 images that contains a handgun.

• True Negative (TN): This is the number of images where the process does not detect a handgun among 304 images that contains a handgun.

• False Positive (FP): This is the number of images where the process does not detect a handgun among 304 images that contains a handgun.

• False negatives (FN): This is the number of images where the process detects a handgun among 304 images that does not contain a handgun

• Accuracy (P): This is the percentage of handgun detection in 304 images that contains handgun.

• Recall (R): This is the percentage of handgun detection in the entire 608 images in the test dataset.

• F1 measure.

The results obtained are acceptable regarding the results found in [10].

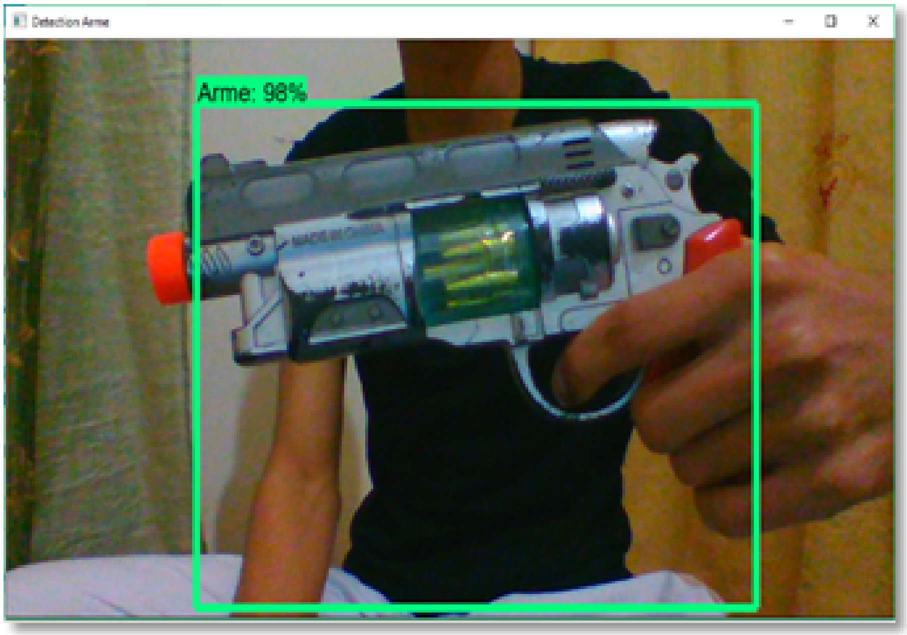

Fig. 6. The real-time detection of the handgun in the image captured by the webcam.

5 Conclusion

According to this study we can conclude that it is possible to use AI as an effective solution in the field of surveillance to ensure total security in major events in real time. This work is divided into two large parts; motion detection and handguns detection in real time. The use of deep learning for the detection of handguns is very complicated because the convolutional neural networks require a large capacity in terms of computation, the fact that requires a supercomputer equipped with a graphics processor (GPU). This work has encountered some difficulties in the experiments due the use of CPU, the fact that increased the learning time, but, it proves at the same time that it is possible to use soft deep learning models for handguns detection even if it still needs very important time for learning. As a future work, we propose the evaluation of the proposed models on powerful machine and the study of the influence of the number of learning epochs on detection performance.

References

1. Corthésy, R., Leite, M.H., (Québec), I.: Détection des mouvements de blocs rocheux par imagerie numérique. Montréal: Institut de recherche Robert-Sauvé en santé et en sécurité du travail (2006)
2. Ghazi, M.M., Yanikoglu, B., Aptoula, E.: Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing **235**, 228–235 (2017)
3. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
4. Gouaillier, V., FLEURANT, A.: La vidéosurveillance intelligente: promesses et défis. rapport de veille technologique et commerciale. Rapport technique, CRIM and Technopôle Défense et Sécurité (2009)
5. Hachemi, F.: La détection et suivi des objets en mouvement dans une scène vidéo en utilisant la bibliothèque OpenCV. Ph.D. thesis, 02/01/2017
6. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al.: Deep neural networks for acoustic modeling in speech recognition. IEEE Signal processing magazine **29** (2012)
7. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
8. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology **160**(1), 106–154 (1962)
9. Le, Q.V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G.S., Dean, J., Ng, A.Y.: Building high-level features using large scale unsupervised learning. arXiv preprint arXiv:1112.6209 (2011)
10. Olmos, R., Tabik, S., Herrera, F.: Automatic handgun detection alarm in videos using deep learning. Neurocomputing **275**, 66–72 (2018)
11. Sainath, T.N., Mohamed, A.r., Kingsbury, B., Ramabhadran, B.: Deep convolutional neural networks for lvcsr. In: 2013 IEEE international conference on acoustics, speech and signal processing. pp. 8614–8618. IEEE (2013)
12. Seol, S.W., Jang, J.H., Kim, H.S., Lee, C.H., Nam, K.G.: An automatic detection and tracking system of moving objects using double difference based motion estimation. In: ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications. pp. 260–263 (2003)
13. Shu, X., Cai, Y., Yang, L., Zhang, L., Tang, J.: Computational face reader based on facial attribute estimation. Neurocomputing **236**, 153–163 (2017)
14. Yu, W., Yang, K., Yao, H., Sun, X., Xu, P.: Exploiting the complementary strengths of multi-layer cnn features for image retrieval. Neurocomputing **237**, 235–241 (2017)