Neural Computing and Applications (2021) 33:5885–5895
https://doi.org/10.1007/s00521-020-05365-w

ORIGINAL ARTICLE

Deep autoencoder for false positive reduction in handgun detection

Noelia Vallez[1] · Alberto Velasco-Mata[1] · Oscar Deniz[1]

Received: 24 September 2019 / Accepted: 11 September 2020 / Published online: 25 September 2020

Abstract
In an object detection system, the main objective during training is to maintain the detection and false positive rates under acceptable levels when the model is run over the test set. However, this typically translates into an unacceptable rate of false alarms when the system is deployed in a real surveillance scenario. To deal with this situation, which often leads to system shutdown, we propose to add a filter step to discard part of the new false positive detections that are typical of the new scenario. This step consists of a deep autoencoder trained with the false alarm detections generated after running the detector over a period of time in the new scenario. Therefore, this step will be in charge of determining whether the detection is a typical false alarm of that scenario or whether it is something anomalous for the autoencoder and, therefore, a true detection. In order to decide whether a detection must be filtered, three different approaches have been tested. The first one uses the autoencoder reconstruction error measured with the mean squared error to make the decision. The other two use the *k*-NN (*k*-nearest neighbors) and one-class SVMs (support vector machines) classifiers trained with the autoencoder vector representation. In addition, a synthetic scenario has been generated with Unreal Engine 4 to test the proposed methods in addition to a dataset with real images. The results obtained show a reduction in the number of false positives between 22.5% and 87.2% and an increase in the system's precision of 1.2%−47% when the autoencoder is applied.

Keywords Handgun detection · False positive reduction · Autoencoder · One-class classification

1 Introduction

Weapons, among other threats, need to be detected as soon as possible to eliminate or mitigate the danger they could cause [1]. Traditionally, the surveillance of public scenarios has been accomplished by the human supervision of the images captured by closed-circuit television (CCTV) systems. However, even an experienced guard may miss a dangerous event due to fatigue or loss of attention [2]. To help with this situation, the creation of automated surveillance systems (AVSs) able to locate potentially threatening objects (or other events) in video has been studied during the last decades [3].

Similarly to other areas, with the introduction of the new deep learning methods these frameworks have obtained promising results and are closer to be used in real scenarios [4, 5]. Nevertheless, although those detectors have high detection (D) and low false positive (FP) rates, when they are used in a different scenario from the one used for training, the false positive ratio increases [6]. This fact represents a major problem since even an increase of a 0.1% of the false positive ratio may cause 90 false alarms per hour with a video input of 25 fps. Therefore, when running the surveillance system in a real scenario, the outcome is usually an unsatisfactory number of false alarms. In most cases, this may lead to the guard switching off the system.

In this context, we propose to include an extra step that models the false alerts that are specific of the new scenario while approximately maintaining the capability of identifying the objects it was trained for. As an specific application, this work focuses on detecting handguns in video surveillance. After running the detector in a new scenario, it is possible to collect all the detector alarms. Practically, all of these alarms are false positives since the incidence of the true event (a handgun in the scene) is very low.

✉ Noelia Vallez
Noelia.Vallez@uclm.es

[1] VISILAB, University of Castilla La-Mancha, ETSI Industriales, Av. Camilo Jose Cela SN, 13071 Ciudad Real, Spain

Therefore, all of these detections can be stored and used to model the new scenario.

The new step will act as a filter able to recognize typical FPs of the detector in the particular scenario. Therefore, this problem can be seen as an anomaly detection problem where the anomalies are those detections that are not similar to the FPs modeled by the filter [7, 8]. In fact, the anomalies detected on this step will be the real alarms.

To detect abnormal and extreme data, one-class classifiers have been widely used in the literature [9]. More concretely, autoencoders have proven to be the most suitable of the techniques, obtaining good results even where other methods fail [10]. In order to use the autoencoder as a filter and decide whether a detection is an anomaly or not, we have tested different approaches: using the autoencoder reconstruction error as a threshold and using the central vector representation to train a nearest neighbor (NN) and a one-class support vector machine (SVM) classifiers. Although autoencoders have been applied in anomaly detection problems, to the best of our knowledge, this is the first time they have been applied to reduce false positive detections when the detector runs in a new scenario from which it is not possible to obtain labeled data.

For the purpose of testing our idea, we have generated an entirely synthetic dataset from the frames captured from a realistic 3D scenario. The synthetic scenario resembles a school hallway from the point of view of a surveillance camera. This allow us to generate as much data as needed with and without handguns to train and test the autoencoder.

The rest of the paper is organized as follows. Section 2 performs an overview of the advances in handgun detection. Section 3 shows the handgun detector used as base detector of the proposed false positive reduction method. Section 4 describes the datasets used including the synthetic dataset that has been generated. Section 5 provides detailed information about the proposed autoencoder-based filtering step. Finally, Sect. 6 shows the results and Sect. 7 summarizes the main conclusions.

2 Related work

In addition to automatic CCTV video surveillance, several approaches have been proposed to deal with concealed handguns in X-ray or millimetric wave images. These types of image are commonly used in airports, train stations or the entrance of some public buildings. In 2008, Nercessian et al. presented a system for handgun detection using X-ray luggage scan images [11]. The approach was based on the Gaussian mixture expectation maximization (EM) method to perform image segmentation prior to the obtention of the edge-based feature vectors. Gesick et al. compared three different approaches for the detection of handguns inside luggage [12]. The first method employs edge detection combined with pattern matching with reliable results. However, both the computational time and the number of false positives were high. The second method uses Daubechies wavelet transforms with inconclusive results as the authors commented. The third algorithm proposed in that work was based on the scale-invariant feature transform (SIFT). Later, in 2010, Harmer et al. used a completely different approach based on the modeling of the complex natural resonances of handguns and compared them with those of other objects [13]. In addition, the work of Flitton et al. concluded that using simpler 3D feature, descriptors outperform even complex RIFT/SIFT solutions with an accuracy of more than 95% [14]. In [15], Xiao et al. employed an extension of the Haar-like features with an AdaBoost-based cascade classifier to detect handguns in passive millimeter wave (PMMW) images. Following a similar approach, the study of Kundegorski et al. combines bag of visual words (BoVW) based on feature point descriptors and support vector machines (SVMs) and random forest classifiers [16].

While there are numerous methods and devices that can detect concealed weapons, unfortunately, the incidence of mass shootings requires the use of RGB surveillance images. Tiwari and Verma proposed a framework that applies color-based segmentation and *k*-means clustering to remove irrelevant objects and then uses Harris interest point detector and Fast Retina Keypoint (FREAK) to locate the handguns [17]. This resulted in high robustness when detecting the desired object at different scales and rotations. In addition, Halima and Hosam worked on a detector that combined SIFT features, *k*-means clustering, a word vocabulary histogram, and SVM [18].

The recent advances in deep learning have also been applied to the handgun detection problem using CCTV images. The first contribution in this area came in the work of Olmos et al. where two different approaches were used [4]. The first one uses a classification CNN to detect handguns with the *sliding window* method, whereas the second one is based on the Faster R-CNN detection architecture. The latter obtained the best results when tested in a dataset composed of several YouTube videos. On the other hand, Gelana et al. followed a more traditional approach using edge detection and a classification CNN with the sliding window method [19]. In addition, Romero and Salamea trained a YOLO object detection and localization system to detect firearms with the particularity of running the detector only in areas where there are people [20]. Another study of Olmos et al. proposed using a symmetric dual camera system to increase the performance of the detection model in low quality surveillance videos improving both the false positive and the detection rates.

To model outliers, discordant objects or simply data that has a different behavior or pattern, anomaly detection techniques have been used [7]. Anomaly detection has a wide range of applications. For example, it can be used to detect anomalies in stock prices and time series [21, 22], abnormal medical images of findings [23–25], abnormal events in video [5, 26, 27], intrusion detection [28], or disaster areas from radar images [29].

A simple method to model anomalies is to use neighbor-based methods such as the *k*-nearest neighbor [30–32] where anomalies are identified as those points in the data space that differ from the surrounding data points. The advantage of these methods is the independence of the data distribution. However, their performance relies on the values of the parameters selected such as the number of neighbors.

An alternative to use neighbor-based methods is to detect anomalies taken into account that they are grouped in a zone of the data space. Thus, the anomaly detection problem is solved as a subspace learning problem [33–36]. Although this method work well in some cases, finding the number of subspaces in which the anomalies are distributed is not trivial.

As in classification and detection tasks, CNNs have demonstrated to improve the performance in anomaly detection problems [26]. More concretely, convolutional autoencoders have been used to model input data and reduce data space dimensionality [37]. Their use has reduced the need of reprocessing input data and compute handcrafted features from it [10]. Following this approach, Mabu et al. proposed to use a convolutional autoencoder followed by a one-class SVM to model normal areas in satellite images and detect abnormal areas caused by natural disasters in Japan [29]. Lu and Xu demonstrated the potential of using variational autoencoders to detect anomalies in skin disease images [23]. The authors recommend to use them instead of GANs (generative adversarial networks) due to their training stability and interpretable results. Sugimoto et al. use an autoencoder followed by a *k*-NN classifier to detect myocardial infarction.

Another approach is the one followed by Gutoski et al. in which autoencoders and stacked denoising autoencoders are used for clustering [38]. With the clustering, representation in possible to define whether a new sample is an anomaly or not according to its distance to the clusters. Gutoski et al. also followed this approach for one-class classification [38].

In some cases, there are more than one group of abnormalities as in the work carried out by Mirsky et al. in [28]. The authors proposed to use an ensemble of autoencoders instead of one to detect online network intrusions. The decision of what is an anomaly or not is based on the RMSE (root-mean-squared errors) score output by the autoencoders.

For video input, Singh and Mohan use deep stacked autoencoders to obtain a deep representation of spatiotemporal video volumes to detect road accidents [37]. The anomaly score is obtained with a one-class SVM as in other works.

Finally, non-symmetric autoencoders have also been used to learn space representations. An example of this is the work carried out by Tran and Hogg where the autoencoder representation is used for detecting anomalies in video [39]. In addition, recurrent autoencoders with LSTM (long short-term memory) layers have also been applied for anomaly detection in video in the work carried out by Yan et al. in [40].

3 Handgun detector

Before addressing the false positive rate reduction through the use of the autoencoder, we needed to train and test a handgun detector. As shown in Sect. 2, there are several approaches that can be selected, from the use of classification CNNs with the *sliding window* approach to the most modern CNN detection architectures. While the former examines every subregion of the image, the latter uses region proposal algorithms to reduce the number of examined windows or process the full image in one pass [41]. The advantage of the new architectures is the ability to detect objects in different locations of the image without being restricted to a certain aspect ratio. Moreover, the number of regions to be examined is drastically reduced in comparison with other methods. The most representative architectures for object detection that follow a region proposals approach are R-CNN, Fast R-CNN, and Faster R-CNN [42].

Two well-known architectures are YOLO (You Only Look Once) and SSD (single-shot detector). YOLO addresses object detection as a regression problem with spatially separated bounding boxes and their corresponding class probabilities [43]. SSD is able to predict, with only one pass over the entire image, the bounding boxes and the class probabilities for them [44].

In addition to all the above, there is a recently developed CNN-based detector called RetinaNet [45]. RetinaNet was designed to solve the problem of having extreme foreground–background class imbalanced problems and has been also applied to X-ray images [46].

For the particular problem of weapon (handgun and knife) detection, [47] reviews recent work and shows that Faster R-CNN has been the prevalent method. For that reason, we have selected the Faster R-CNN architecture to train a handgun detector with a dataset provided by the

University of Seville [1]. The dataset is composed of 871 images that contain 177 annotated handguns. Those images were extracted from the video captured by 2 CCTV cameras located in two different college hallways.

4 Datasets

The collection and labeling of the data necessary to train deep learning models are tasks that require significant time and effort. This is even more complicated in detection or segmentation problems in which someone has to select the area of the image in which the object is located, or the exact contour of the object, in addition to the category. A possible solution to this problem is the use of public datasets, but, depending on the problem, it is not always possible to have one available. The use of synthetic images facilitates the work required to obtain large datasets. For this work, a completely synthetic dataset has been generated with Unreal Engine 4 [48], rendering a scenario that represents a high-school hallway where people are walking. There are other popular alternatives such as Unity [49] and Lumberyard/CryEngine [50] that can also be used for the same purpose. While some of the people on the scenario carry everyday objects in their hands, such as mobile phones, others carry guns or nothing (see Fig. 1).

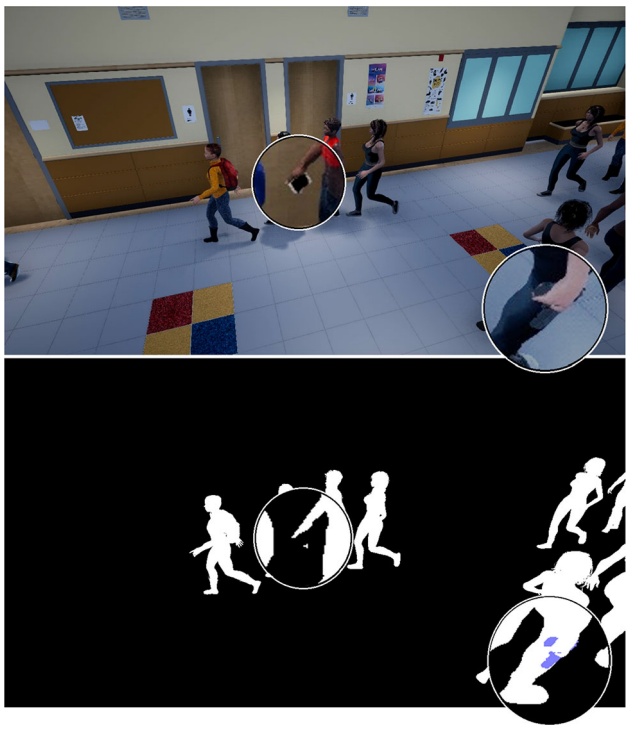

Fig. 1 Synthetic scenario with a zoom on the elements of interest (in this case, a mobile phone and a handgun)

Another advantage of having the data generation fully controlled by the researcher is that it is also possible to automatically generate a mask image with the desired objects for each frame. In this case, each generated image contains the people in white, the background covered in black, and each handgun filled with a different color to help extract the information about its location. Once all masks are obtained, the coordinates of the bounding boxes that contain the weapons are extracted storing the annotations in XML files with the format defined by the *Pascal VOC 2012 Challenge* [51].

A total of 4000 images were generated with this method with a resolution of 1280×720. From these, 3000 frames were used to train and adjust the proposed autoencoder filter, containing 5437 annotated handguns. The remaining 1000 frames were used to evaluate and compare the detector and detector + autoencoder systems.

In addition to the synthetic dataset, the Gun Movies Database [52] has also been used to ensure the differences are caused by the proposed method and not by changes in the texture by the origin of the data. This dataset contains

Fig. 2 Sample frame from the Gun Movies dataset

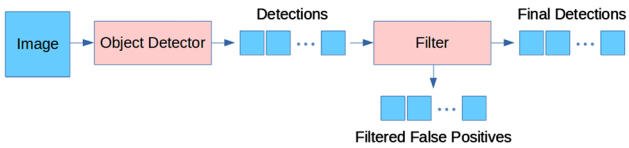

Fig. 3 Proposed system

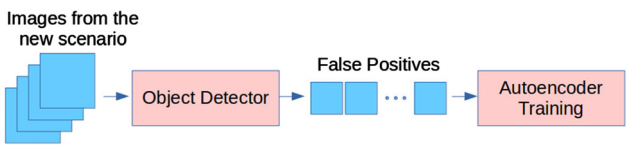

Fig. 4 Autoencoder training phase

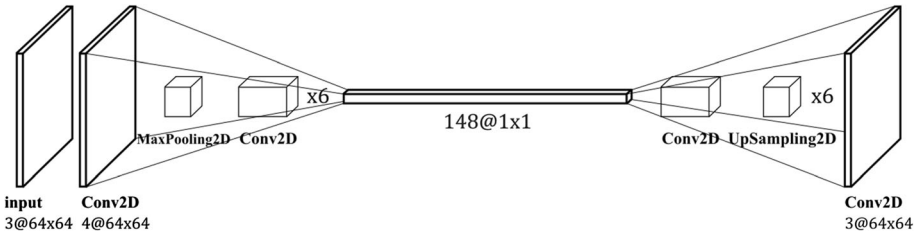

Fig. 5 Autoencoder architecture used

Table 1 Detailed description of the autoencoder architecture used. The output of the conv2d_7 layer (in bold) is used as the FP intermediate representation

Layer (type)	Output shape	Param #
input_1 (InputLayer)	(None, 64, 64, 3)	0
conv2d_1 (Conv2D)	(None, 64, 64, 4)	112
max_pooling2d_1 (MaxPooling2)	(None, 32, 32, 4)	0
conv2d_2 (Conv2D)	(None, 32, 32, 28)	1036
max_pooling2d_2 (MaxPooling2)	(None, 16, 16, 28)	0
conv2d_3 (Conv2D)	(None, 16, 16, 52)	13156
max_pooling2d_3 (MaxPooling2)	(None, 8, 8, 52)	0
conv2d_4 (Conv2D)	(None, 8, 8, 76)	35644
max_pooling2d_4 (MaxPooling2)	(None, 4, 4, 76)	0
conv2d_5 (Conv2D)	(None, 4, 4, 100)	68500
max_pooling2d_5 (MaxPooling2)	(None, 2, 2, 100)	0
conv2d_6 (Conv2D)	(None, 2, 2, 124)	111724
max_pooling2d_6 (MaxPooling2)	(None, 1, 1, 124)	0
conv2d_7 (Conv2D)	**(None, 1, 1, 148)**	**165316**
conv2d_8 (Conv2D)	(None, 1, 1, 148)	197284
up_sampling2d_1 (UpSampling2)	(None, 2, 2, 148)	0
conv2d_9 (Conv2D)	(None, 2, 2, 124)	165292
up_sampling2d_2 (UpSampling2)	(None, 4, 4, 124)	0
conv2d_10 (Conv2D)	(None, 4, 4, 100)	111700
up_sampling2d_3 (UpSampling2)	(None, 8, 8, 100)	0
conv2d_11 (Conv2D)	(None, 8, 8, 76)	68476
up_sampling2d_4 (UpSampling2)	(None, 16, 16, 76)	0
conv2d_12 (Conv2D)	(None, 16, 16, 52)	35620
up_sampling2d_5 (UpSampling2)	(None, 32, 32, 52)	0
conv2d_13 (Conv2D)	(None, 32, 32, 28)	13132
up_sampling2d_6 (UpSampling2)	(None, 64, 64, 28)	0
conv2d_14 (Conv2D)	(None, 64, 64, 3)	759

images of size 640×480 pixels from 7 laboratory-shot movies with a total of 817 frames and 686 annotated handguns (Fig. 2).

For this second dataset, a total of 817 images were used. From these, 571 frames were used to train and adjust the proposed autoencoder filter and the remaining 246 frames were used to evaluate and compare the detector and detector + autoencoder systems.

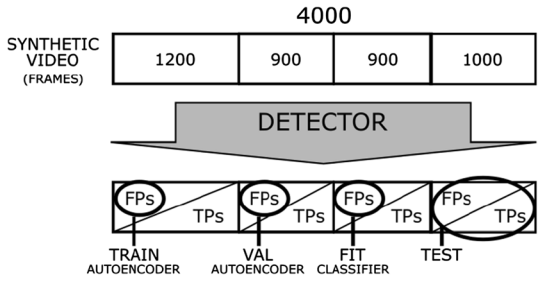

Fig. 6 Subsets of the synthetic dataset used. The Gun Movies dataset is similarly split

5 Proposed method

As introduced above, when a detector runs in a new scenario, the false positive rate increases due to its particularities that were not seen in the training data. To deal with this problem, we propose to add a filtering step after the detector inference (see Fig. 3). The filter is used to discard the FP detections of the object detector produced by the particularities of the new scenario.

The filter can be considered as a one-class classifier that learns how to identify a certain type of samples. Thus, the rest of the samples can be considered as anomalies. This problem has been addressed in the literature through the use of the one-class versions of the SVM, *k*-nearest neighbor (*k*-NN), random forests classifiers, and more recently with deep autoencoders [9, 38, 53]. In our case, an autoencoder is trained to model the class of the typical FP detections.

In order to collect the training samples for the autoencoder, the detector is run in the particular scenario for a certain period of time, storing all the FP detections (Fig. 4). Initially, all detections can be considered as FPs in a real scenario since the incidence of handguns is very low.

Deep autoencoders learn the input data distribution using an intermediate representation. They are able to compress the data into a small vector and then reconstruct the input from it with accurate results. If new input data come from a different distribution, the reconstruction error will be higher.

Finally, according to the autoencoder structure, we define and compare 3 different methods to check whether the output of the detector is a typical false positive. The

Fig. 7 Typical false positives of the handgun detector in the synthetic scenario (enlarged)

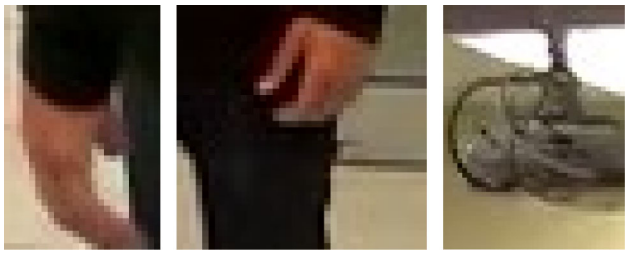

Fig. 8 Typical false positives of the handgun detector in the Gun Movies dataset (enlarged)

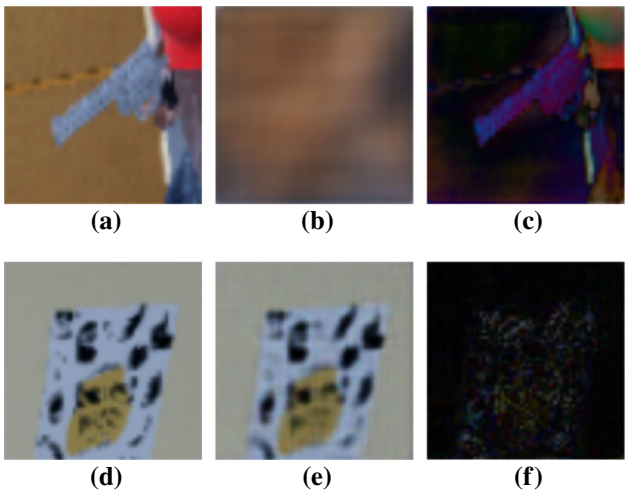

Fig. 9 Autoencoder reconstruction of: **a** TP and d) FP of the detector from the synthetic dataset. **b** and **e** are the reconstructed images, and **c** and **f** are the absolute difference between the reconstructions and their corresponding original images

Table 2 Percentage of FPs that are filtered by method and dataset

	MSE	kNN	SVM
Synthetic	26.4%	30%	22.5%
Gun Movies	87.2%	74.1%	49%

Table 3 Increase in the detector precision when the autoencoder is applied by method and dataset

	MSE	kNN	SVM
Synthetic	1.46%	1.77%	1.2%
	th = 0.0057	th = 0.34	th = 14600
Gun Movies	47%	20%	8%
	th = 0.047	th = 1.92	th = 175

simplest one is to establish a threshold for the reconstruction error. Therefore, detections with low reconstruction error will be discarded as typical FPs of the scenario. The other two methods are based on the use of the central vector as a compact representation of the images and then train a one-class classifier with it. For that, SVM and k-NN with $k = 1$ were used, and the thresholds were selected according to the scores and the distance to the closest neighbor, respectively.

5.1 Autoencoder architecture

The structure of an autoencoder consists of an encoder path that ignores the noise and reduces the dimensionality and a decoder path that makes the reconstruction. The compressive path of the autoencoder used consists of a set of 6 convolutional and max-pooling layers (Fig. 5). Similarly, the reconstruction path has also 6 convolutional and upsampling layers. The input is a 3-channel image of size 64×64, and the central vector has 148 elements (conv2d_7 layer). A more detailed description of the architecture can be seen in Table 1.

6 Results

The Faster R-CNN model trained with the dataset from the University of Seville obtained an mAP of 0.7933. Training took 2 days and executed 62 epochs. An Ubuntu 14.04 LTS

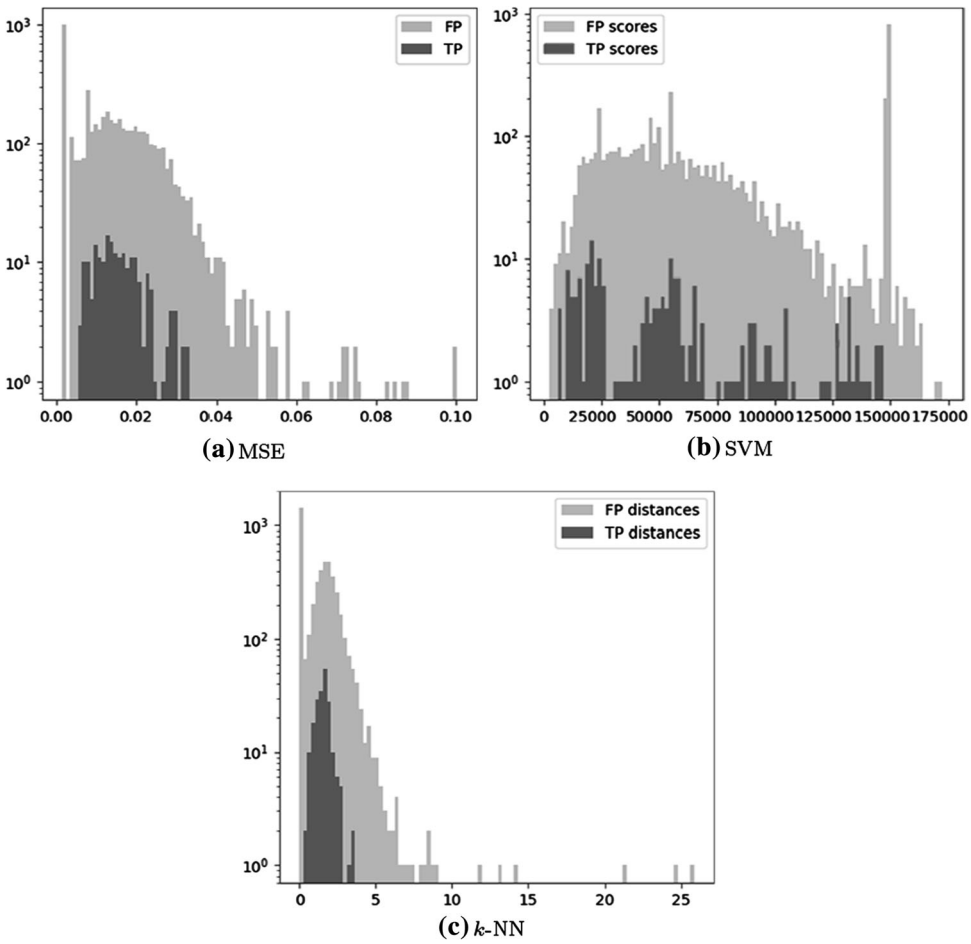

Fig. 10 Synthetic dataset. Histograms of the MSE reconstruction error, SVM score, and K-NN distance (y-axis uses logarithmic scale)

machine with 2 nVIDIA Quadro M4000 cards, Keras with TensorFlow backend and CUDA 8.0 were used to perform the training.

After obtaining this base detector, both datasets were divided into 4 parts to (1) train and (2) validate the autoencoder, (3) fit the k-NN and SVM classifiers, and (4) test the system variants (Fig. 6). The detector was then run on each of the subsets, and the FP and TP patches were stored. Although only the FP detections are used to train and validate the autoencoder and fit the classifiers, the correct detections were also generated and stored for the test subset to check that the detection rate is minimally affected.

Overall, for the autoencoder training and validation, two sets with 4913 and 3607 FPs, respectively, were obtained for the synthetic dataset and 586 and 405 FPs for the Gun Movies dataset. Another set composed of 3712 FP regions for the synthetic dataset and 95 FP regions for the Gun Movies dataset was used to fit the classifiers used to perform the decision. Finally, a set of 4832 regions (4632 FPs and 200 TPs) for the synthetic dataset and 499 regions (359 FPs and 40 TPs) for the Gun Movies dataset was reserved for testing. Figure 7 shows some examples of the typical FP detections obtained in the synthetic scenario.

The autoencoders were then trained and validated with the stored FP detections of the training and validation subsets. This process took only about an hour for each dataset to complete 500 epochs in a Windows 10 PC with an nVIDIA GTX 1060 MaxQ card using Keras with TensorFlow backend and CUDA 9.0. At this point, if the autoencoder is used with some test images from TP and FP detections, the ability to effectively reconstruct FPs is evidenced (see Fig. 9).

The stored FP detections from the fit subset of each dataset were used to feed each of the autoencoders and get intermediate vectors to train the SVM and k-NN one-class classifiers. The SVM selected uses a linear kernel. On the other hand, $k = 1$ was selected for the k-NN algorithm.

To illustrate the performance of both the detector and detector + autoencoder approaches on the two datasets, they were tested with the 1000 images from the fourth subset of the synthetic scenario and the 246 frames of the Gun Movies dataset. The histograms of the reconstruction error for the MSE thresholding-based method, the probability score for the SVM one-class classifier, and the

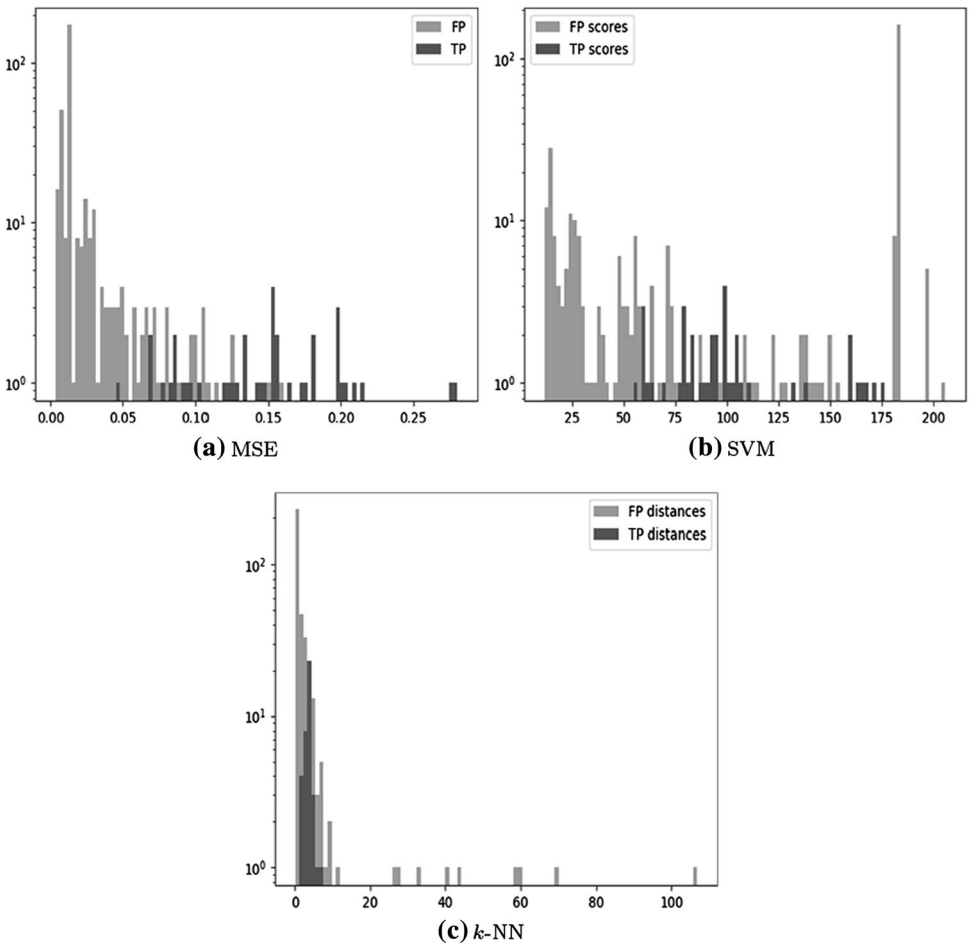

Fig. 11 Gun movies dataset. Histograms of the MSE reconstruction error, SVM score, and *K*-NN distance (*y*-axis uses logarithmic scale)

distance to the nearest neighbor for the *k*-NN were obtained (Figs. 10 and 11). Although TPs and FPs are overlapped in all cases, the first part of the MSE and *k*-NN histograms and the last part of the SVM histogram do not contain TPs. Therefore, the FPs that lie on those parts of the histograms can be potentially filtered selecting the value of the first bin (or the last for the SVM) in the histogram that contains TPs as threshold. For the synthetic dataset, this shows that, without affecting the detection rate, up to 26.4% of all the FPs can be filtered using the MSE reconstruction error, 22.5% using the one-class SVM, and 30% with the distance to the nearest neighbor of the *k*-NN (2). On the other hand, in the histograms obtained from the Gun Movies dataset TPs and FPs are less overlapped making it possible to remove up to 87.2% of all the FPs using the MSE reconstruction error, 49% using the one-class SVM, and 74.1% with the distance to the nearest neighbor of the *k*-NN without affecting the detection rate.

In addition, the precision–recall curves corresponding to the detector and the autoencoder with the three proposed decision methods were obtained (see Figs. 12 and 13). The experimental results show a reduction in the number of false positives while roughly maintaining the detection capabilities [54]. Since the precision–recall curve is calculated by varying the detector output threshold and the autoencoder has another threshold that can be varied too, each curve was obtained under a specific value for the autoencoder and varying the threshold of the detector (3). For the synthetic dataset, comparing all the curves for a specific autoencoder thresholding method, they show a maximum increase in the precision of 1.46% at the same recall values when the autoencoder and the MSE are used (threshold = 0.0057), of 1.2% using the autoencoder and the SVM classifier (threshold = 14600), and of 1.77% in case of the autoencoder and *K*-NN (threshold = 0.34). For the Gun Movies dataset, results show a maximum increase in the precision of 47% at the same recall values when the autoencoder and the MSE are used (threshold = 0.047), of 8% using the autoencoder and the SVM classifier (threshold = 175), and of 20% in case of the autoencoder and *K*-NN (threshold = 1.92).

Overall, the results show that the autoencoder is able to filter part of the new FPs in all cases without affecting the detection rate of the original system. All thresholding

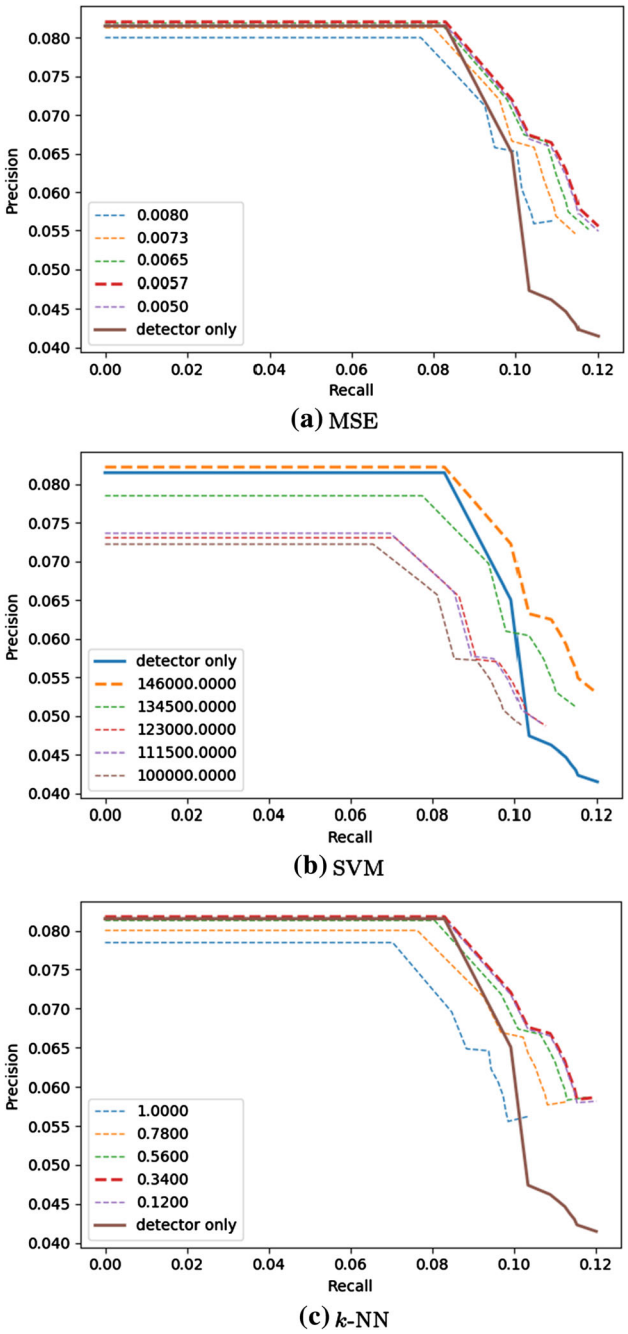

Fig. 12 Precision–recall curves for the synthetic dataset. Best viewed in color (color figure online)

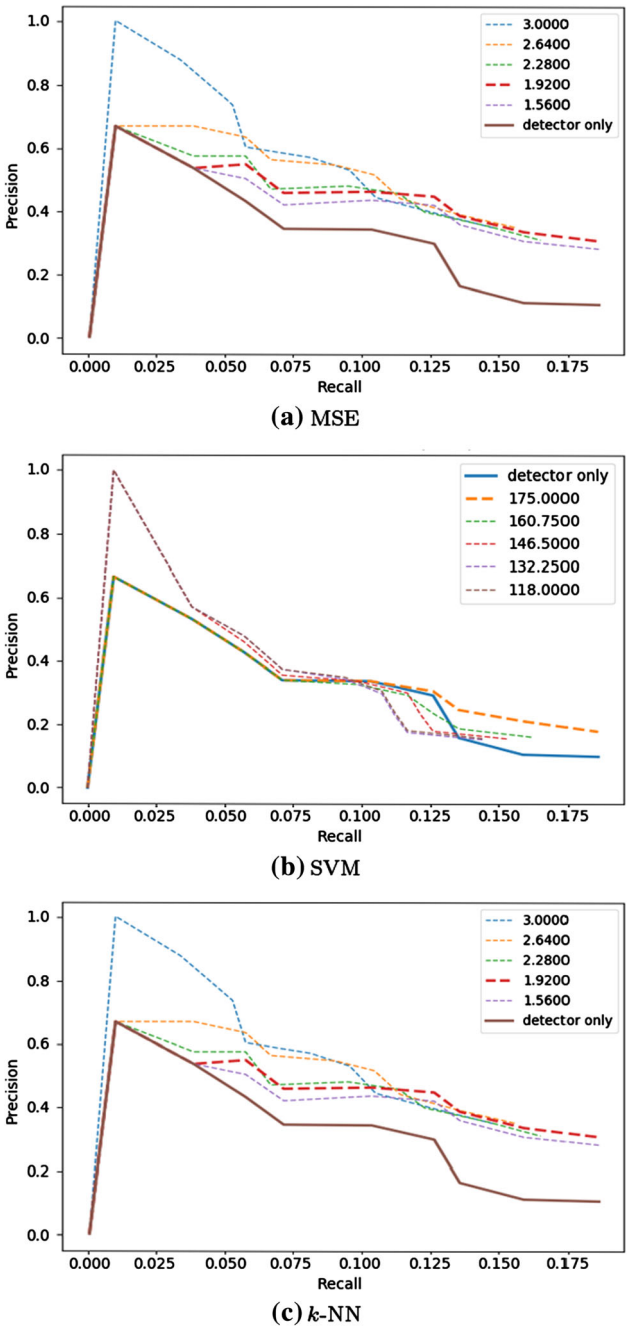

Fig. 13 Precision–recall curves for the Gun Movies dataset. Best viewed in color (color figure online)

methods are able to reduce the number of FPS to some degree.

7 Conclusions

In this work, a step to filter the false positive detections that appear when a pre-trained handgun detector is deployed in the final surveillance scenario has been proposed. This step consists of training a deep autoencoder with the false positive regions obtained from the particular scenario. Once the autoencoder is trained, it can be used to decide whether a detection is similar to the already known typical false alarms and can be filtered, or otherwise if an alert should be triggered.

The ability of the autoencoder to reduce the number of FPs has been demonstrated with a potential reduction by up to 30% for the synthetic scenario when it is combined with

a *k*-NN classifier trained with the vector representation of the detector FPs regions and up to 78% of the FPs for the Gun Movies dataset when the autoencoder is combined with the MSE error metric. Furthermore, the handgun detection capability of the system is not compromised by the added filtering step under a wide range of threshold levels.

Although the proposed approach has been only applied to two particular scenarios, it can be extended to more than one since having different perspectives, lighting conditions or background objects will generate different false positives. Thus, during the system's deployment only a generic detector (in this case a handgun detector) is required and one autoencoder will be trained for each camera feed.

Acknowledgements We thank Professor Dr. J.A. Alvarez for the surveillance images provided for training the handgun detector and J. J. Corroto for generating the synthetic dataset. This work was partially funded by projects TIN2017-82113-C2-2-R by the Spanish Ministry of Economy and Business and SBPLY/17/180501/000543 by the Autonomous Government of Castilla-La Mancha and the ERDF.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Enríquez F, Soria LM, Álvarez-García JA, Caparrini FS, Velasco F, Deniz O, Vallez N (2019) Vision and crowdsensing technology for an optimal response in physical-security. Comput Sci (ICCS) 11540:15–26
2. Ashby MPJ (2017) The value of CCTV surveillance cameras as an investigative tool: an empirical analysis. Eur J Crim Policy Res 23(3):441–459
3. Raghunandan A, Mohana M, Raghav P, Aradhya HR (2018) Object detection algorithms for video surveillance applications. In: IEEE-7th international conference on communication and signal processing (ICISPC 2018), pp 563–568
4. Olmos R, Tabik S, Herrera F (2018) Automatic handgun detection alarm in videos using deep learning. Neurocomputing 275:66–72
5. Dan X, Yan Y, Ricci E, Dan N (2017) Detecting anomalous events in videos by learning deep representations of appearance and motion. Comput Vis Image Underst 156:117–127
6. Vallez N, Bueno G, Deniz O (2013) False positive reduction in detector implantation. In: 14th Conference on artificial intelligence in medicine, (AIME)
7. Xiaodan X, Liu H, Yao M (2019) Recent progress of anomaly detection. Complexity 2015:1–11
8. Vallez N, Velasco-Mata A, Corroto JJ, Deniz O (2019) Weapon detection for particular scenarios using deep learning. In: 9th Iberian conference on pattern recognition and image analysis (IbPRIA)
9. Khan SS, Madden MG (2013) One-class classification: taxonomy of study and review of techniques. CoRR, abs/1312.0049
10. Hofer-Schmitz K, Nguyen P-H, Berwanger K (2018) One-class autoencoder approach to classify Raman spectra outliers. In: ESANN
11. Nercessian S, Panetta K, Agaian S (2008) Automatic detection of potential threat objects in X-ray luggage scan images. In: 2008 IEEE conference on technologies for homeland security, pp 504–509
12. Gesick R, Saritac C, Hung C-C (2009) Automatic image analysis process for the detection of concealed weapons. In: Proceedings of the 5th annual workshop on cyber security and information intelligence research: cyber security and information intelligence challenges and strategies, pp 1–20
13. Harmer SW, Andrews DA, Rezgui ND, Bowring NJ (2010) Detection of handguns by their complex natural resonant frequencies. IET Microwaves Antennas Propag 4:1182–1190
14. Flitton G, Breckon TP, Megherbi N (2013) A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery. Pattern Recognit 46(9):2420–2436
15. Xiao Z, Lu X, Yan J, Wu L, Ren L (2015) Automatic detection of concealed pistols using passive millimeter wave imaging. In: 2015 IEEE international conference on imaging systems and techniques (IST), pp 1–4
16. Kundegorski ME, Akcay S, Devereux M, Mouton A, Bath University, Breckon TP (2016) On using feature descriptors as visual words for object detection within X-ray baggage security screening. In: 7th International conference on imaging for crime detection and prevention (ICDP), pp 1–12
17. Tiwari RK, Verma GK (2015) A computer vision based framework for visual gun detection using Harris interest point detector. In: 11th International conference on communication networks, ICCN, 54: 703–712
18. Halima NB, Hosam O (2016) Bag of words based surveillance system using support vector machines. Int J Secur Appl 10(4):331–346
19. Gelana F, Yadav A (2019) Firearm detection from surveillance cameras using image processing and machine learning techniques. In: Smart innovations in communication and computational sciences, pp 25–34
20. Romero David, Salamea Christian (2019) Convolutional models for the detection of firearms in surveillance videos. Appl Sci 9:1–11
21. Gotou Hiroyuki, Suzuki Tomoya (2016) Biased reactions to abnormal stock prices detected by autoencoder. J Signal Process 20(4):157–160
22. Zhang C, Chen Y (2019) Time series anomaly detection with variational autoencoders. CoRR, abs/1907.01702
23. Lu Y, Xu P (2018) Anomaly detection for skin disease images using variational autoencoder. CoRR, abs/1807.01349
24. Sato D, Hanaoka S, Nomura Y, Takenaga T, Miki S, Yoshikawa T, Hayashi N, Abe O (2018) A primitive study on unsupervised anomaly detection with an autoencoder in emergency head CT

volumes. In: Medical imaging 2018: computer-aided diagnosis, p 60
25. Freiman M, Manjeshwar R, Goshen L (2019) Unsupervised abnormality detection through mixed structure regularization (MSR) in deep sparse autoencoders. CoRR, abs/1902.11036
26. Chong YS, Tay YH (2017) Abnormal event detection in videos using spatiotemporal autoencoder. In: Cong F, Leung A, Wei Q (eds) Advances in neural networks - ISNN 2017. Springer International Publishing, Cham, pp 189–196
27. Sabokrou M, Fathy M, Hoseini M (2016) Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder. Electron Lett 52(13):1122–1124
28. Mirsky Y, Doitshman T, Elovici Y, Shabtai A (2018) Kitsune: an ensemble of autoencoders for online network intrusion detection. CoRR, abs/1802.09089
29. Mabu Shingo, Fujita Kohki, Kuremoto Takashi (2019) Disaster area detection from synthetic aperture radar images using convolutional autoencoder and one-class svm. J Robot Netw Artif Life 6:48–51
30. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. 29: 427–438
31. Angiulli F, Pizzuti C (2002) Fast outlier detection in high dimensional spaces. 2431: 15–26
32. Hautamäki V, Karkkainen I (2004) Outlier detection using k-nearest neighbour graph. 3: 430–433
33. Zhang J, Jiang Y, Chang K, Zhang S, Cai J, Hu L (2009) A concept lattice based outlier mining method in low-dimensional subspaces. Pattern Recognit Lett 30:1434–1439
34. Zhang J, Xiaolong Y, Li Y, Zhang S, Xun Y, Qin X (2016) A relevant subspace based contextual outlier mining algorithm. Knowledge-Based Syst 99:02
35. Muller E, Assent I, Steinhausen U, Seidl T (2008) Outrank: ranking outliers in high dimensional data. pp 600–603
36. Pasillas-Diaz J, Ratté S (2016) Bagged subspaces for unsupervised outlier detection: FBSO. Comput Intell 33
37. Singh D, Mohan CK (2019) Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. IEEE Trans Intell Transp Syst 20(3):879–887
38. Gutoski M, Ribeiro M, Romero Aquino NM, Lazzaretti AE, Lopes HS (2017) A clustering-based deep autoencoder for one-class image classification. In: 2017 IEEE Latin American conference on computational intelligence (LA-CCI), pp 1–6
39. Tran H, Hogg DC (2017) Anomaly detection using a convolutional winner-take-all autoencoder. In BMVC
40. Yan S, Smith JS, Lu W, Zhang B (2020) Abnormal event detection from videos using a two-stream recurrent variational autoencoder. IEEE Trans Cognit Dev Syst 12(1):30–42
41. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 580–587
42. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst, pp 91–99
43. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 779–788
44. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: single shot multibox detector. In: European conference on computer vision (ECCV), pp 21–37
45. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision (CVPR), pp 2980–2988
46. Cui Y, Oztan B (2019) Automated firearms detection in cargo x-ray images using RetinaNet. In: Anomaly detection and imaging with X-Rays (ADIX) IV, 10999: 105–115
47. Fernandez-Carrobles MM, Deniz O, Maroto F (2019) Gun and knife detection based on faster R-CNN for video surveillance. In: 9th Iberian conference on pattern recognition and image analysis (IbPRIA)
48. Unreal Engine 4. https://www.unrealengine.com. Accessed 20 Sep 2019
49. Unity. https://unity.com. Accessed 20 Sep 2019
50. Lumberyard. https://aws.amazon.com/es/lumberyard. Accessed 20 Sep 2019
51. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The Pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
52. Grega M, Lach S, Sieradzki R (2013) Automated recognition of firearms in surveillance video. In: 2013 IEEE International multidisciplinary conference on cognitive methods in situation awareness and decision support (CogSIMA), pp 45–50
53. Leng Q, Qi H, Miao J, Zhu W, Guiping S (2015) One-class classification with extreme learning machine. Math Probl Eng 1–11(05):2015
54. Tharwat A (2018) Classification assessment methods. Appl Comput Inf pp 1–13