Neural Computing and Applications (2021) 33:7803–7819
https://doi.org/10.1007/s00521-020-05521-2

ORIGINAL ARTICLE

Detection of threat objects in baggage inspection with X-ray images using deep learning

Daniel Saavedra[1] · Sandipan Banerjee[2] · Domingo Mery[1]

Received: 4 July 2020 / Accepted: 6 November 2020 / Published online: 19 November 2020

Abstract

In the field of security, baggage-screening with X-rays is used as nondestructive testing for threat object detection. This is a common protocol when inspecting passenger baggage particularly at airports. Unfortunately, the accuracy of such human inspection is around 80–90%, under optimal operator conditions. For this reason, it is quite necessary to assist human inspectors with the aid of computer vision algorithms. This work proposes a deep learning-based methodology designed to detect threat objects in (single spectrum) X-ray baggage scan images. For this purpose, our proposed framework simulates a large number of X-ray images, using a combination of PGGAN (Karras et al. in International conference on learning representations, 2018. https://openreview.net/forum?id=Hk99zCeAb) and superimposition (Mery and Katsaggelos in 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), 2017.https://doi.org/10.1109/CVPRW.2017.37) strategies, that are used to train state-of-the-art detection models such as YOLO (Redmon et al. in You only look once: unified, real-time object detection. CoRR abs/1506.02640, 2015. http://arxiv.org/ abs/1506.02640), SSD (Liu et al. in SSD: single shot multibox detector. CoRR abs/1512.02325, 2015. http://arxiv. org/abs/1512.02325) and RetinaNet (Lin et al. in Focal loss for dense object detection. CoRR abs/1708.02002, 2017. http://arxiv.org/abs/1708.02002). Our method has been tested on real X-ray images in the detection of four categories of threat objects: guns, knives, razor blades and shuriken (ninja stars). In our experiments, YOLOv3 (Redmon and Farhadi in Yolov3: An incremental improvement. CoRR abs/1804.02767, 2018. http://arxiv.org/abs/1804.02767) obtained the best mean average precision (mAP) with 96.3% for guns, 76.2% for knives, 86.9% for razor blades and 93.7% for shuriken, while the average mAP for all threat objects was 80.0%. We believe the effectiveness of our method in the detection of threat objects makes its use in checkpoints possible. Moreover, our methodology is scalable and can be easily extended to detect other categories automatically.

Keywords Object detection · Baggage inspection · X-ray testing

1 Introduction

Security in both public and private sectors has been of utmost importance with the aim of preventing crime and spread of pests or diseases, among other reasons [57]. This has mainly focused on the visual detection of threat elements, such as weapons and certain animals or organic elements. In the particular case of airports, security is extremely important for the authorities and passengers, so different technologies have been implemented for this task. The most common of which is the noninvasive analysis of passenger baggage through the use of X-rays, which expedites inspection while avoiding direct contact with the baggage. Unfortunately, X-ray inspection is a complicated process due to various factors, one of which is that the

✉ Domingo Mery
domingo.mery@uc.cl

Daniel Saavedra
dlsaavedra@uc.cl

Sandipan Banerjee
sandipan.banerjee@affectiva.com

[1] Department of Computer Science, Pontificia Universidad Catolica de Chile, Santiago, Chile

[2] Affectiva, Boston, MA, USA

objects sought are in closed places and often hidden by other items in the bag. Them being rotated at different angles does not facilitate detection as well [7]. Another element to consider is human performance in the detection of objects, where inspectors only have a couple of seconds to make decisions. In normal situations, the human brain can quickly detect threat objects; however, this performance can be hampered due to noise, fatigue or simple distraction due to the monotony of the work. The literature indicates that these inspectors only reach a yield between 80 and 90 % on average [35]. Therefore, developing different fast techniques based on computer vision for threat object detection can assist security personnel, especially during higher public influx. In this paper, the state-of-the-art is addressed in Sect. 2.

In order to use deep learning models in the detection of threat objects, it is necessary to train the model with a large number of such object images. Only if the training set is representative enough, the trained model can be effectively used in the testing stage. For training purposes, we use a set of simulated X-ray images [30] by superimposing two X-ray images (see Fig. 1): (i) an X-ray image of a baggage *with no* threat object and (ii) an X-ray image of an isolated threat object. Using a model that observes the absorption's law of X-rays [28], it is possible to obtain a very realistic simulated X-ray image of the baggage *with* the threat object (see gun in Fig. 1). Since threat objects, like guns, are rarely encountered and hence difficult to collect, we use both real and synthetic X-ray images of isolated threat objects, generated by a progressively growing generative adversarial network (PGGAN) [19], for our experiments. In this paper, the details of the simulation of X-rays are presented in Sect. 3. The experiments are conducted on the GDXray dataset [31], which contains thousands of (single spectrum) X-rays that have been widely used in other works [33]. The new dataset, that contains simulated X-ray images from GDXray, is used to train different deep learning-based object detection algorithms such as YOLO [44], SSD [25] and RetinaNet [23], outlined in Sect. 4. In order to validate the proposed methodology, we evaluate the detection performance on real X-ray images using the mean average precision (mAP) metric. We propose an experimental protocol that can be used in the future for clear and fair comparisons. The experiment details, results and analysis can be found in Sect. 5. Finally, concluding remarks and future work are highlighted in Sect 6.

2 Related work

Object detection in X-ray images has been used in different areas of research, such as in medicine [27] for the detection of breast cancer [41, 55] or automatic bone count [36], as well as in the food industry [26]. For detection of threat objects in baggage inspection, there are several research directions that we will separate into the following three groups: (i) methods that detect threat objects using a single view of the baggage, (ii) methods that use multiple views of the baggage, and (iii) finally, methods that use modern deep learning techniques for this task.

Single view methods One of the first works was based on the segmentation of the image along with the extraction of features at the edges of the sample [38]. A more recent study uses 'bag of words' techniques [53] to detect objects of interest [47]. On the other hand, there are studies that use pseudo-color images computed from dual-energy images. From them, it is worthwhile to mention two algorithms with acceptable performance in the detection of guns: (i) the use of 'bag of words' in [54] and (ii) the use of SURF features [6] in key-points detected on corners [15] in [21].

In a recent study [33], modern computer vision techniques with and without deep learning strategies have been compared in the classification of threat objects on the GDXray database [31]. In this study, no deep learning models were trained for this task, but their pre-trained versions were used (like AlexNet [20] and GoogleNet [52] trained on

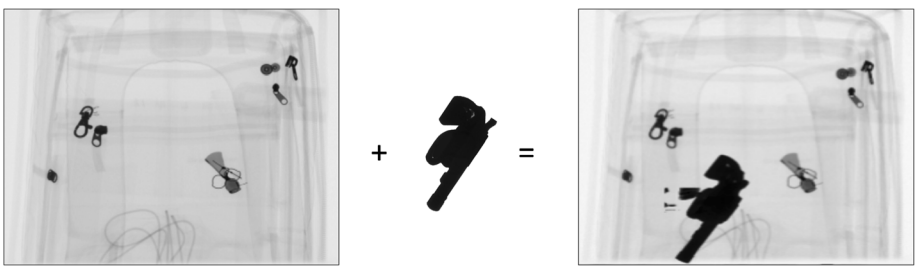

Fig. 1 Simulation by superimposition of two X-ray images: a synthetically generated X-ray image of an isolated threat object using PGGAN [19] (middle) is superimposed onto a real X-ray image of a baggage with no threat objects (left); the result is a simulated X-ray image of a baggage with the threat object (right)

ImageNet [8]). The conclusion was that even pre-trained deep learning models can achieve very good results via transfer learning [56]. Unsurprisingly, the same conclusion has been drawn for other computer vision applications [58].

Multiple view methods In order to improve detection performance, more complex solutions have been developed with the use of multiple view geometry [16]. In the case of single spectrum, researchers have utilized data association between multiple detections [32]. This idea is further improved in [29] by tracking the points of interest across multiple views. On the other hand, dual-energy images and multiple-views are used to train a SVM classifier in [5].

Deep learning After the remarkable success of AlexNet [20] for classification tasks in the ImageNet competition [8], CNN-based methods for solving computer vision problems have increased significantly generating excellent results compared to methods that use classical procedures. For the particular problem of object detection, there are a large number of models that obtain favorable results on the popular MS COCO [24] and Pascal VOC [10] benchmarks. These results are substantially better than those obtained through classical methods, further underpinning the need to implement deep learning-based models in more specific problems, such as the detection of baggage by X-rays.

The first set of approaches that train deep learning models for the detection of threat objects are described in [1, 34], where dual-energy images from private databases are used. There are more recent investigations as well that report experiments on single-spectrum X-ray images such as [9] and [59] that use a private dataset and GDXray database [31], respectively. Unfortunately, experiments that are conducted on private datasets cannot be reproduced and compared with new algorithms. Finally, there is a recent study [51], which uses a CNN of multi-view X-ray images for object detection. The key-idea is to combine 2D CNN-features extracted from different 2D projections into a common 3D feature space. Afterward, 3D features are used by a faster-RCNN [45] to detect prohibited objects. The latter confirms that better results can be obtained by combining the information granted by the different views.

While CNNs can classify objects irrespective of their position within an image (i.e., translation invariant), they are not necessarily rotation or scale invariant as shown in a variety of works [2, 14, 18, 46]. Additionally, objects placed closely together can also lead to occlusion which can further hamper performance. Consequently, researchers have explored the idea of preemptively augmenting training images with rotation and scaling operations [37, 39, 49]. Thus, our idea of synthetically superimposing threat objects within clean images at an arbitrarily chosen position, rotation and scale can lead to superior classifier performance as quantitatively validated by our experiments.

3 Simulation by superimposition

For the simulation of X-ray images, used later to build the training dataset, we used the superimposition strategy presented in [30], in which a simulated X-ray image is created by combining two X-ray images (an X-ray image of the isolated threat object and an X-ray image of a baggage) as illustrated in Fig. 1. In our work, two different approaches are used to obtain the X-ray image of the isolated threat object: (i) real X-ray images [30] and (ii) synthetic X-ray images generated by a progressively growing GAN model (PGGAN) [19].

3.1 Superimposition strategy

The method to be presented here aims to generate simulations of X-ray images with threat objects that can be used both in the training of human inspectors and automatic detection models. These simulations need to be similar in appearance to the real X-ray images in order to dispense with human inspection due to technical difficulties and any possible danger. For this, a limited amount of images of baggage with non-threat elements (laptop, notebook, pencils) and images of isolated threat weapons such as guns, knives, razors and shuriken are used (few representative images of the baggage and threat object classes can be seen in Figs. 5 and 7, respectively).

The central idea of the method is to superimpose the threat objects in the baggage, thus obtaining a baggage with threat elements inside, as presented in [30]. This is based on the use of the absorption's law of X-rays [28] that characterizes the distribution of X-rays through matter. In general, we know that a gray scale X-ray image I is followed by a linear model, defined by

$$I = A \cdot \varphi + B, \tag{1}$$

where A and B are constant parameters of the capture model, and

$$\varphi = \varphi_0 e^{-\mu d} \tag{2}$$

with μ absorption coefficient, d thickness of the irradiated matter, φ_0 incident energy flux density and φ energy flux density after passage through matter with the thickness of d. Using this model, we can define the image of the threat object (I_1) and the image of luggage (I_2) as:

$$I_1 = A \cdot \varphi_1 + B \qquad I_2 = A \cdot \varphi_2 + B \tag{3}$$

with

$$\varphi_1 = \varphi_0 e^{-\mu_1 d_1} \qquad \varphi_2 = \varphi_0 e^{-\mu_2 d_2}, \tag{4}$$

where the absorption coefficient and thickness are (μ_1, d_1) for threat object and (μ_2, d_2) for the luggage.

Thus, to get the combined image I_t modeled as

$$I_t = A \cdot \varphi_t + B = A \cdot \varphi_0 e^{-(u_2 d_2 + u_1 d_1)} + B, \tag{5}$$

we use (3), (4), (5) and define $C = A\varphi_0$ to obtain:

$$I_t = C \cdot J_1 \cdot J_2 + B, \tag{6}$$

where

$$J_1 = \frac{I_1 - B}{C} \qquad J_2 = \frac{I_2 - B}{C}. \tag{7}$$

An example of the method is shown in Fig. 1; it can be seen that it is practically impossible to discriminate that the result is not from a real image.

3.2 GAN strategy

Although our simulation method can seamlessly blend in an isolated threat object, like a gun, into an X-ray image of a baggage, it cannot generate new views of the threat object in question. In order to generate more views of the threat objects, we simulate new threat object samples and strengthen training models using one of the recent GAN [13] architectures. A typical GAN model is composed of a generator and a discriminator network where the generator tries to synthesize new threat object images from random noise vectors while the discriminator is trained to distinguish between real threat object images and such synthetically generated images, as shown in Fig. 2. Mathematically, this training can be represented as:

$$L = -\sum_{i=1}^{N} \log(D(x_i)) - \sum_{i=1}^{N} \log(1 - D(G(z_i)))], \tag{8}$$

where L is the overall loss, z is the input noise vector, x is the corresponding real image, N is the batch size, and G and D are the generator and the discriminator, respectively. We use the original loss formulation with binary cross-entropy [13, 40] for training our model as shown in (8). Once training finishes, the trained generator is used to hallucinate new synthetic X-ray images of isolated threat objects from a set of different noise vectors.

In our experiments, we use the recently proposed progressively growing regime of GANs (PGGAN) [19], that increases the depth of the network by adding convolutional layers after each iteration in the training is completed. More specifically, the model starts training with 4×4 threat object images by downsampling the original gallery, and after training for a number of epochs adds new layers to the generator and the discriminator and trains with 8×8. This iterative multi-resolution training by slowly increasing image resolution and adding additional layers continues till the model is trained for the target image resolution of 128×128. This progressive growing not only improves model convergence but also generates sharper results at higher resolutions. For the creation of new samples of threat objects, separate GAN models were trained for the threat object categories used in our experiments: knife, razor blade, gun, shuriken, where only images of isolated objects were used. To enforce translation, rotation and scale invariance in the final detection models, we augment our gallery by translating the images in X, Y and XY directions, rotating them 45°, 90°, 135° and 180° and flipping the image in horizontal and vertical directions.

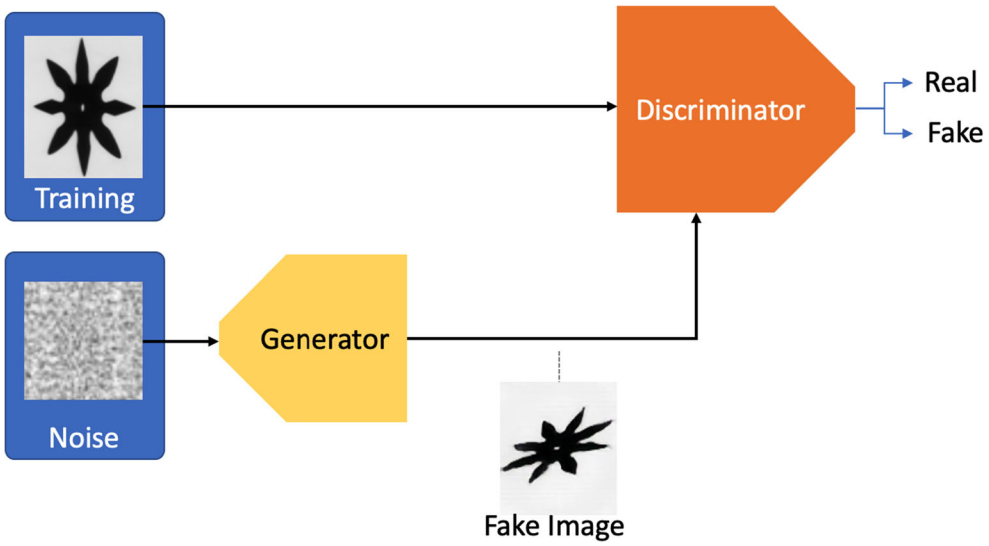

Fig. 2 The training strategy of our GAN: the generator synthesizes new threat objects from random noise, while the discriminator is trained to detect such fake images from real ones. Once training finishes, only the generator is used to generate synthetic X-ray images from random noise

This augmented dataset is then used to train our PGGAN model to synthesize realistic new threat object samples, an example of which can be seen in Fig. 3.

4 Detection methods

In our work, we studied the best performing and most representative deep learning-based object detection models, as stated in [58]: YOLOv2 [42], YOLOv3 [44], SSD [25] and RetinaNet [23]. In this section, we give a brief description of these detection models and their principal differences.

YOLOv2 In [42], a new type of architecture named (You-Only-Look-Once) YOLO was proposed, based on a direct transform of the analyzed image into a map of characteristics. They are used to predict both the bounding box that delimits the figure and the probability of an object of a certain class contained in the box.

The key-idea of this architecture is to subdivide the image into $S \times S$ sub-images or grids, where in each of these grids the algorithm predicts B frames with their respective coordinates x, y, w, h (position, width and height) and probability that contains an object. The improvements proposed in [43] focus on expanding the subdivision of the image, and the use of anchor boxes of different dimensions in each subdivision of the image (as proposed in the faster-RCNN [45] model). These anchor boxes are pre-configured using the 'k-means' algorithm with Euclidean distance in the training set. Then, for each cell of the feature map extracted using the DarkNet-19 model, its anchor boxes are created with predictions for the objects inside [43].

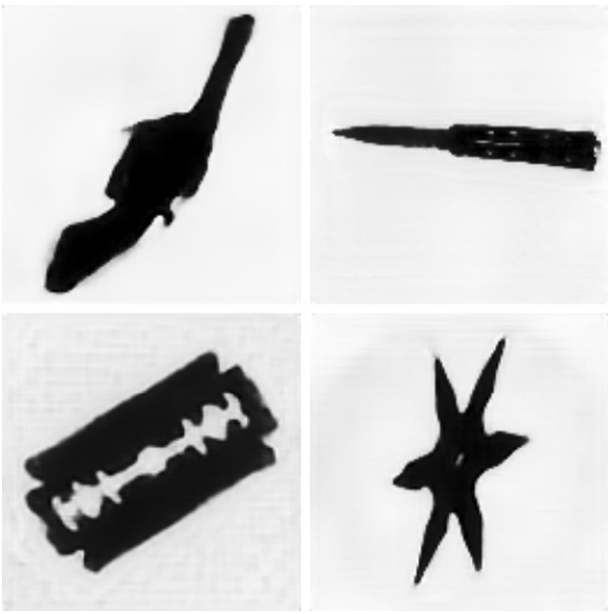

Fig. 3 Sample synthetic X-ray images of the different threat object classes generated by our trained PGGAN model. All images are 128×128 in size

YOLOv3 In comparison with previous versions, YOLOv3 [44] includes two main updates: (i) the use of different scales (3 scales) using a pyramidal architecture that aims to solve the problem of detection of small objects and (ii) the use of a new feature extractor architecture called DarkNet-53 that improves upon DarkNet-19.

SSD Another architecture contemporary to faster-RCNN [45] and YOLO [42] is the SSD (Single Shot Multi-Box Detector) [25]. Using direct image transformations, like YOLO, it predicts the location of the desired objects. The major difference is the use of map features in different depths, in order to obtain the analysis at different scales of the image. SSD combines the use of anchor boxes, like faster-RCNN [45] and YOLOv2 [43], to predict the desired frames and uses a loss function for multi-tasking, as in the aforementioned detectors.

RetinaNet Together with YOLOv3 [44], the RetinaNet architecture [23] is one of the most recent object detection models and combines the pyramidal feature extraction structure [22] with a residual architecture [17] that has obtained promising results in image classification. Another novelty of this structure is the shift from the typical cross-entropy to a 'focal loss'-based objective that reduces the penalty for well-classified classes while punishing misclassifications more aggressively for the rest.

5 Experiments and results

In this section, we provide details of the evaluation protocol, datasets used, experiments we conducted, and the implementation and computation time of our work.

5.1 General evaluation protocol

In this work, we use single-spectrum X-ray images, some of them from group 'Baggage' of GDXray database [31] and the rest correspond to new X-ray images that were acquired for our experiments. In this protocol, the threat objects to be detected are: guns, knives, razor blades and shuriken (ninja stars). The X-ray images that are used in the proposed evaluation protocol consist of backpacks (with and without threat objects), threat objects and other objects. They are organized in the following four groups of objects (see Table 1 and Fig. 4):

G1: Isolated threat objects such as guns, knives, razor blades and shuriken (series of GDXray: B0049,

Table 1 Groups of X-ray images defined by the experimental protocol

Group	Purpose	Series	Database	Guns	Knives	Razor blades	Shuriken	Threat Objects	Others	Backpacks	Images
G1	Train	B0049	1	200	0	0	0	200	0	0	200
G1	Train	B0051	1	0	0	100	0	100	0	0	100
G1	Train	B0052	1	0	0	0	144	144	0	0	144
G1	Train	B0076	1	0	576	0	0	576	0	0	576
G2	Train	B0082	1	0	0	0	0	0	600	0	600
G3	Train	BX100	2	0	0	0	0	0	Many	48	48
G4-1	Test	B0046	1	210	24	78	33	345	Many	200	200
G4-1	Test	B0047	1	69	46	160	210	485	Many	200	200
G4-1	Test	B0048	1	107	5	210	88	410	Many	200	200
G4-2	Test	B0042	1	19	35	0	0	54	Many	19	19

1: GDXray, 2: new images acquired for this research

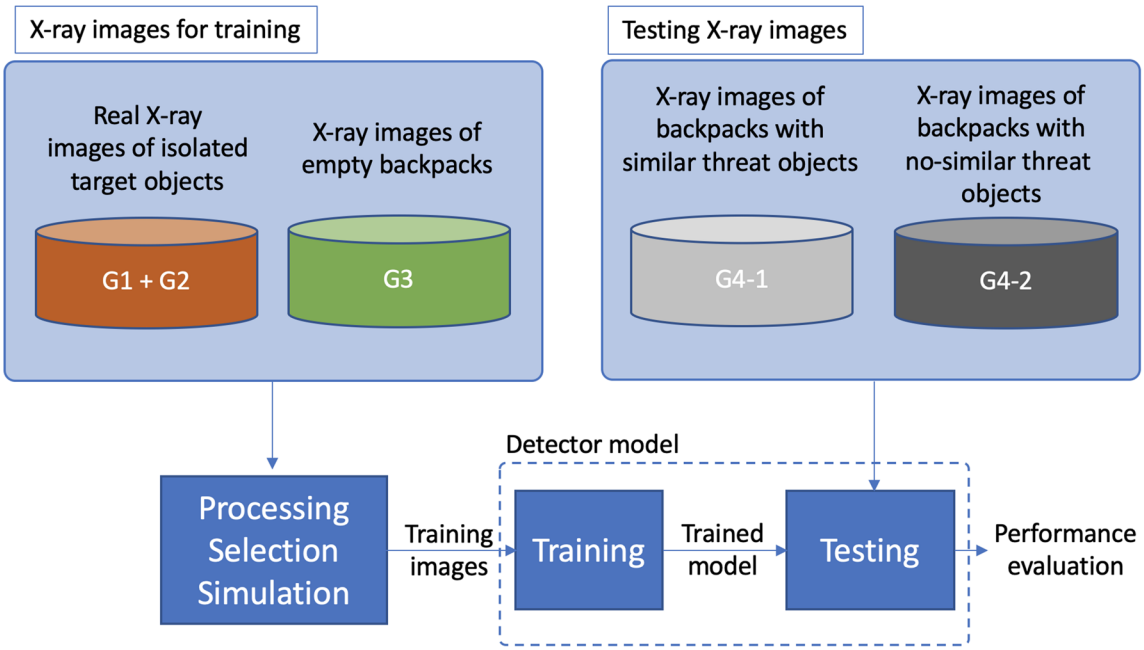

Fig. 4 Proposed experimental protocol: In training stage, the X-ray images for training are used. In testing stage, two testing subsets (1 and 2) are used

B0051, B0052 and B0076). These X-ray images were taken by locating the objects inside a sphere of expanded polystyrene (EPS), that has a low X-ray absorption coefficient (Fig. 5).

G2: Objects that are not classified as threat objects such as clips, notebook banding, among others (series of GDXray: B0082), as shown in Fig. 6.

G3: Backpacks with no threat objects (new series BX100), as can be seen in Fig. 7. Additionally, in this group, a blank image without any element is included.

G4: Backpacks in different poses with one to four threat objects (series of GDXray: B0042, B0046, B0047 and B0048). Sample images are depicted in Fig. 8.

For our experiments, we propose an evaluation protocol (see Fig. 4) that defines the X-ray images that might be used for training and testing purposes as follows:

• *Training details* The X-ray images from the first three groups (G1, G2 and G3) from Table 1 can be used. For the first four classes, the images have a single element of the respective class with a centered and complete view of the element without occlusion. These images have been

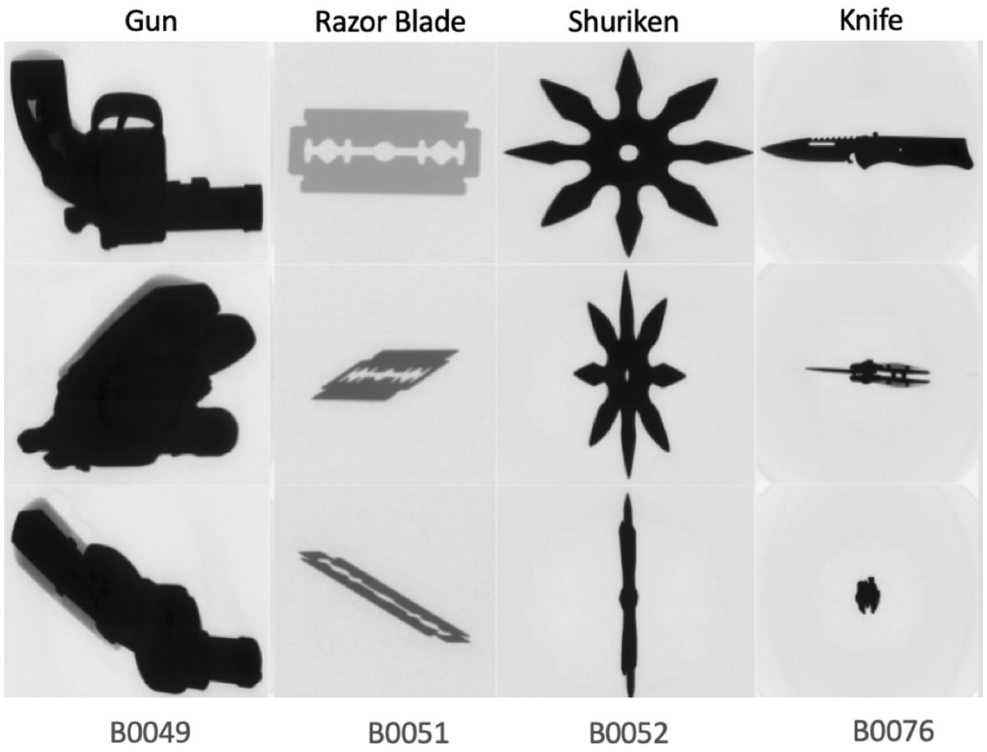

Fig. 5 Group G1: Images of threat objects, with a view from 'Frontal' (top) to 'Non-Frontal' (bottom)

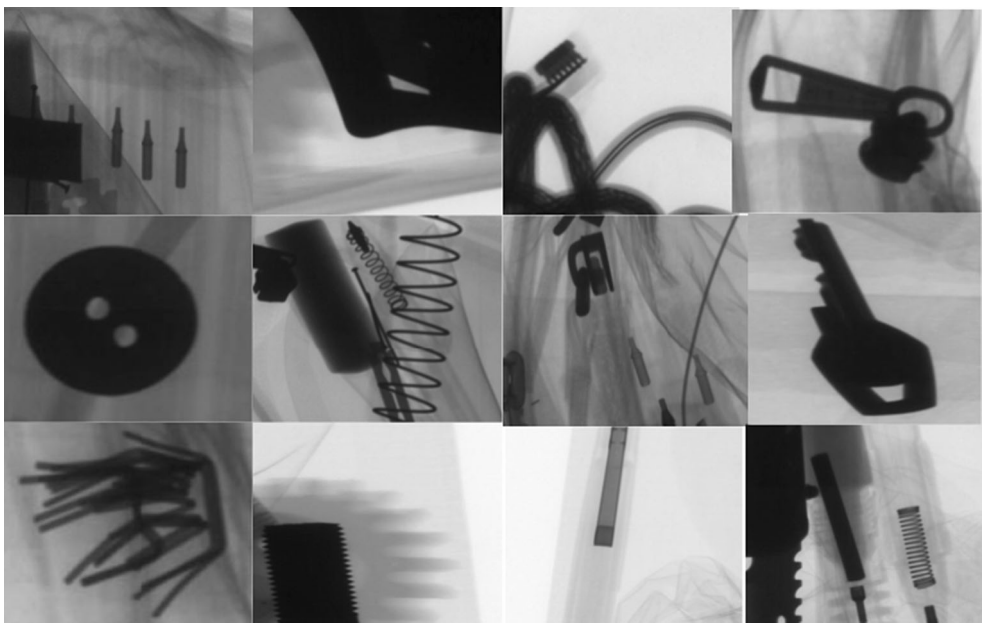

Fig. 6 Group G2: Multiple examples of images of non-threat objects in the B0082 series

captured from different points of view, providing different variations in pose of the element, as shown in Fig. 5. Images of non-threat objects from series B0082 are also used, some of which can be seen in Fig. 6. These are used to obtain a more complex training set, forcing the detector to have greater decision-making capacity. Thus, we define the target objects as the four threat objects (guns, razor blades, shuriken and knives) and the other non-threat objects (clips, cellphones, etc.). It is worthwhile to mention that in our proposed protocol, how these images are used to train the detector model is not defined, and future researchers following this protocol are allowed to use these

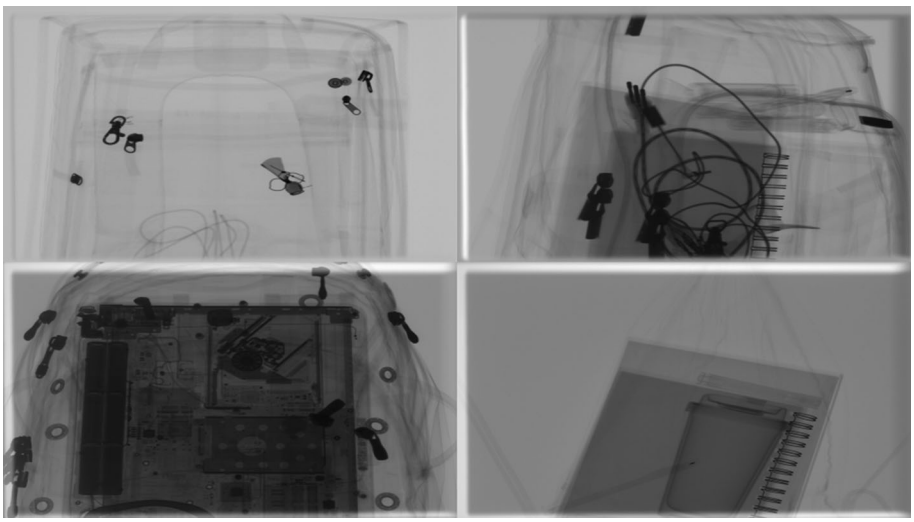

Fig. 7 Group G3: Samples of the BX100 series, which is composed of backpacks without threat objects

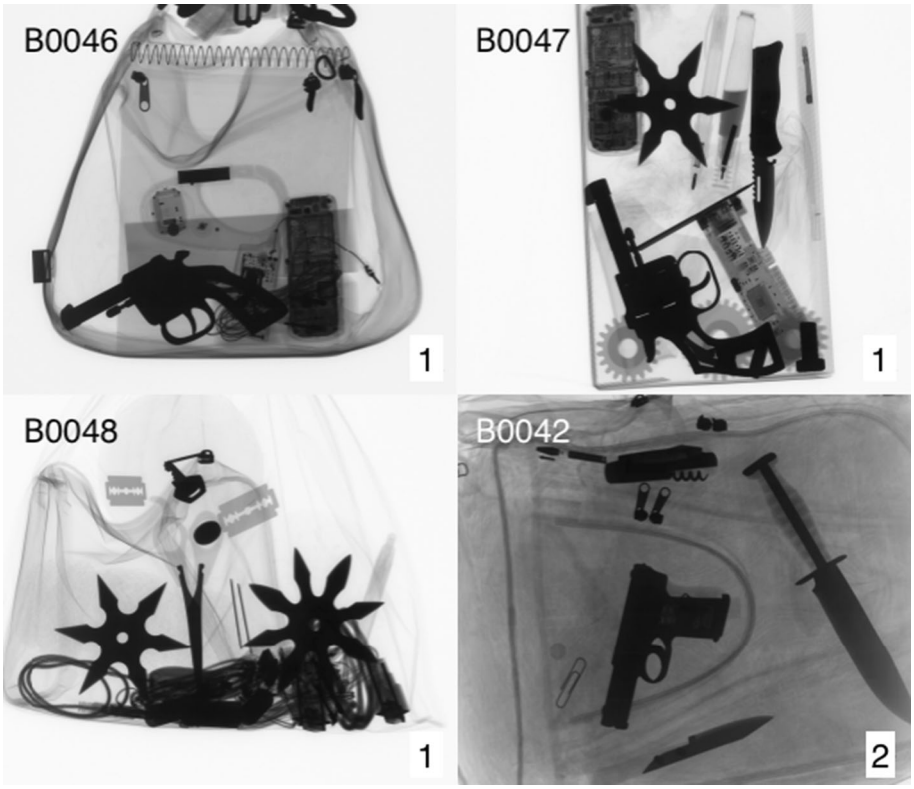

Fig. 8 Group G4: Example of series of X-ray images for testing: Subsets 1 and 2

images at their convenience (*e.g.,* data augmentation, simulation of new images that are generated using the images of these three groups, etc.).

• *Testing details* The last group (G4) is used for testing. We distinguish in this group two subsets (see Table 1 and Fig. 8), in each of them the detection performance must be measured independently:

+ *Testing subset 1* It is defined by Group G4-1 (series B0046, B0047 and B0048) and consists of X-ray images of backpacks with the same threat objects that are used in the training set.

+ *Testing subset 2* It is defined by Group G4-2 (series B0042) and consists of X-ray images of backpacks with different knives and guns[1], i.e., threat objects of the same categories but different from the threat objects used in training dataset. The idea of testing on Subset 2 is to analyze the robustness of the detector against changes in the shape of the threat objects. Thus, the detection of threat objects in Subset 2 is more challenging than Subset 1.

[1] Razor blades and shuriken are not present in Subset 2.

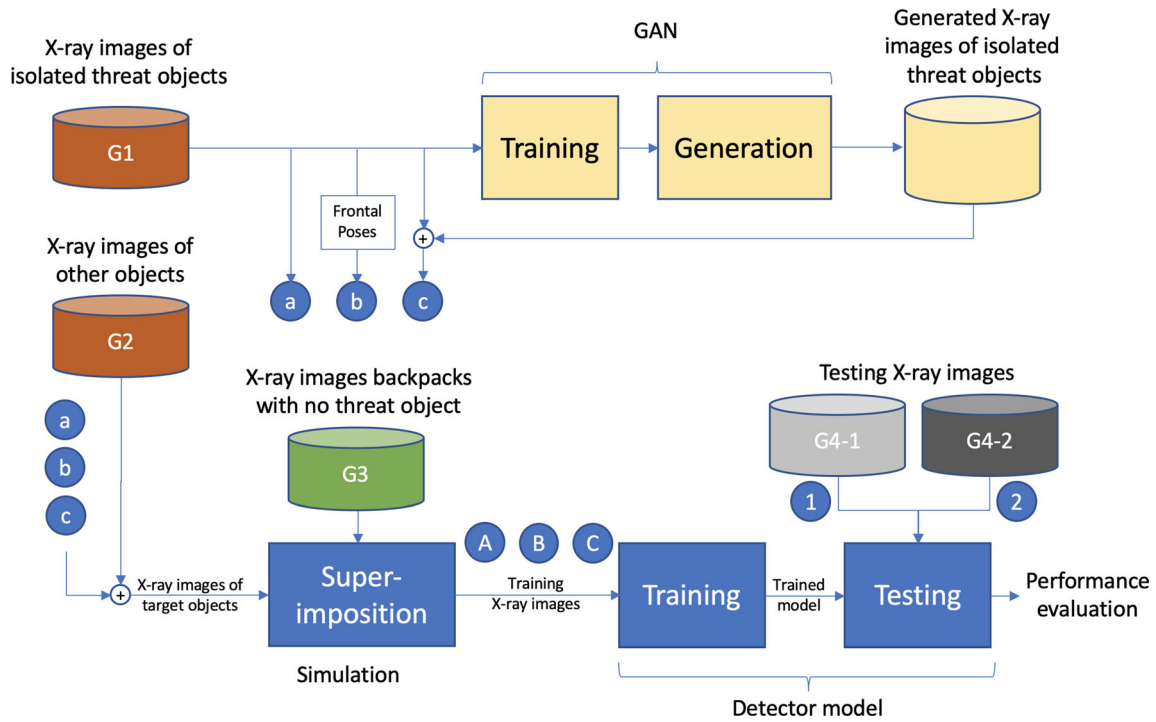

Fig. 9 Block diagram of our three experiments: in Experiments A, B, C, the detector model is trained using Training Subsets A, B, C, respectively. In each experiment, we evaluate the performance of the trained model on 'Testing Subset 1' or 'Testing Subset 2'

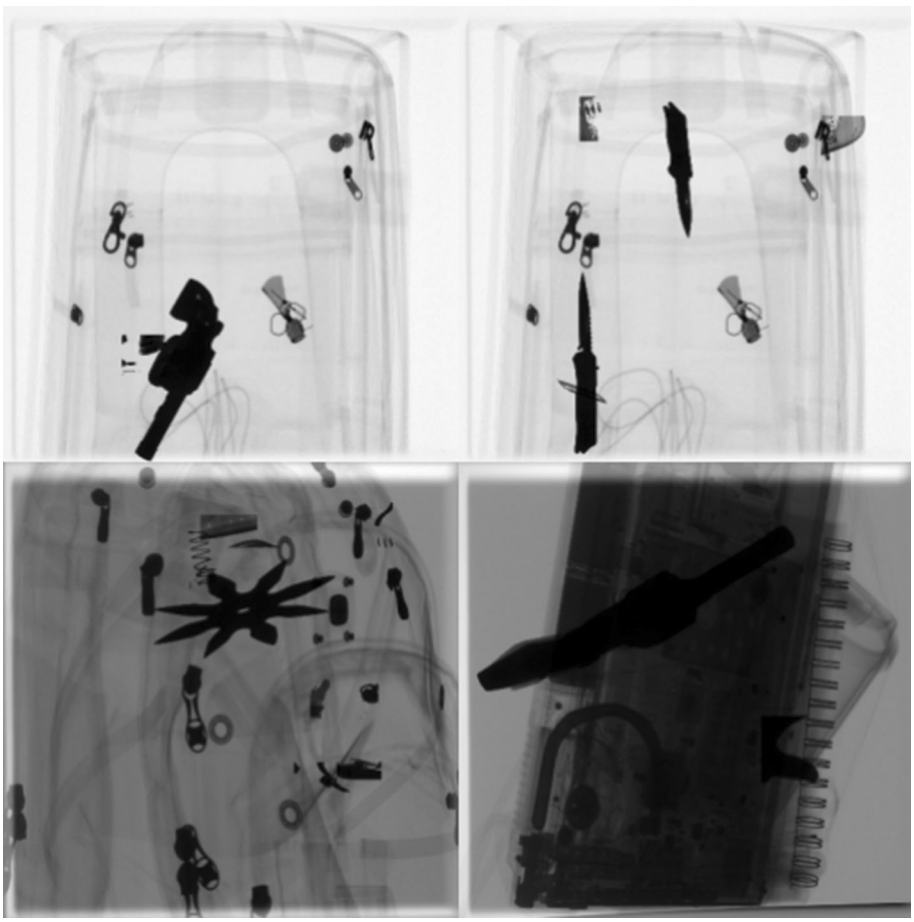

Fig. 10 Images of the training set. Simulated backpacks with threat objects

Table 2 X-ray images of training data using GAN

Class	Real images	GAN images
Razor blade	100	100
Gun	200	200
Knife	576	576
Shuriken	144	144

5.2 Training datasets

The proposed evaluation protocol defines precisely the X-ray images on which the trained detector model must be tested but not the X-ray images for training the detector model. In this section, we describe three different datasets of training images that are built from the set of images for training purposes. In our experiments, we train each model using these three different datasets.

We use the following three subsets of X-ray images of target objects (see Fig. 9):

- Subset of Threat Objects 'a' X-ray images of Group G1.
- *Subset of threat objects* 'b' synthetic X-ray images and subset of Target Objects 'a'. It should be mentioned that the synthetic threat objects are generated by the trained PGGAN [19] model (see Sect. 3.2), where the images of Group G1 are used.
- *Subset of threat objects* 'c' X-ray images of Group G1 in frontal poses only (see frontal and non-frontal poses in Fig. 5).

For all subsets a, b, c, we add the X-ray images of other (non-threat) objects from Group G2. In order to simulate the training images, we use the superimposition method addressed in Sect. 3, which superimposes a target object onto an X-ray image of a backpack. It simulates an X-ray of a backpack that contains a threat element.

Thus, for training purposes, we define Training Subsets A, B and C, as shown in Fig. 9, that are generated by a superimposition of backpack images from group G3 and Subsets of Threat Objects a, b and c, respectively, and other objects from Group G2 as follows:

- Training Subset A It consists of 200 simulated samples for each X-ray image in G3. In each of these simulations, threat and non-threat elements are superimposed at random positions. In simulations of threat objects, there is an overlap of 1 to 4 elements, while in non-threat objects the overlap is of 1 to 9 elements. In turn, we select randomly the type of element and the image. There are 9800 training images with their respective annotations of the position of

Table 3 Average precision(AP) on testing subset 1

Model	Experiment	Gun	Knife	R. Blade	Shuriken	mAP	wmAP
Yolov2	A	0.780	0.054	0.607	0.880	0.580	0.700
Yolov3	A	**0.963**	**0.762**	**0.869**	**0.937**	**0.883**	**0.910**
	B	**0.928**	0.385	0.847	0.784	0.736	0.828
	C	**0.958**	0.916	0.825	0.925	0.906	0.899
SSD300	A	**0.901**	0.011	0.696	0.904	0.630	0.776
	B	0.878	0.012	0.449	0.903	0.561	0.677
	C	0.906	0.064	0.704	0.903	0.644	0.781
Retina	A	**0.981**	0.005	0.819	**0.956**	0.690	**0.857**
Net	B	**0.949**	0.036	0.774	0.934	0.672	0.826
	C	**0.904**	0.001	0.771	0.944	0.655	0.812

Table 4 Average precision (AP) on testing subset 2[1]

Model	Experiment	Gun	Knife	Razor	Shuriken	mAP	wmAP
Yolov2	A	0.026	0.038	–	–	0.032	0.034
Yolov3	A	0.161	0.118	–	–	0.140	0.133
	B	0.318	0.033	–	–	0.175	0.133
	C	0.189	0.005	–	–	0.097	0.070
SSD300	A	**0.514**	0.323	–	–	**0.419**	**0.390**
	B	0.531	0.414	–	–	0.473	0.455
	C	0.571	0.397	–	–	0.484	0.458
Retina	A	**0.723**	0.029	–	–	0.376	0.273
Net	B	**0.757**	0.099	–	–	0.428	0.330
	C	**0.891**	0.126	–	–	0.509	0.395

Best results are given in bold

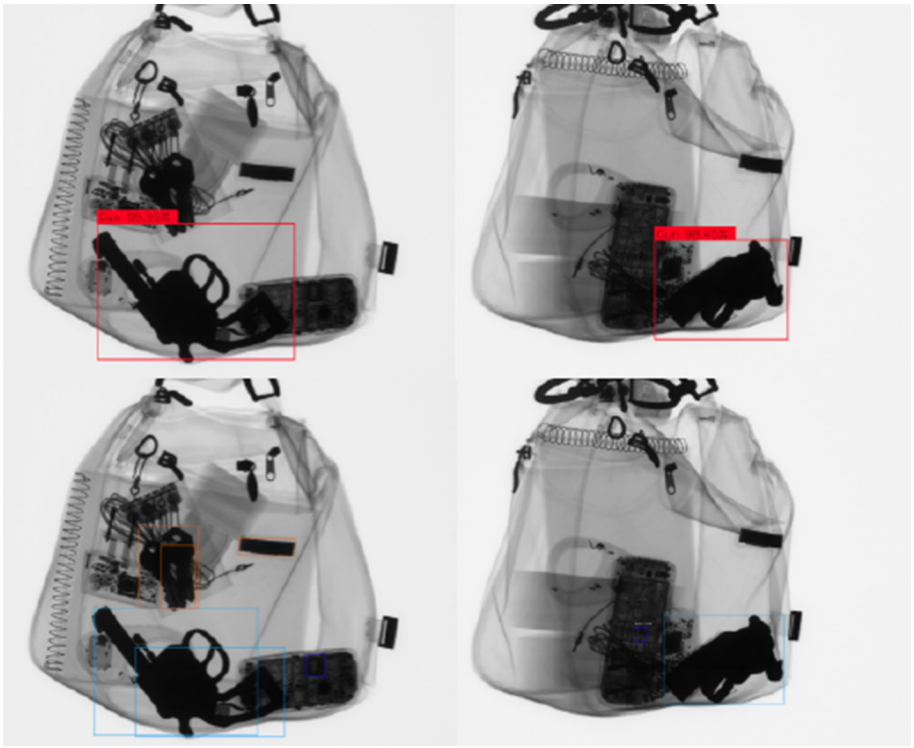

Fig. 11 Sample detection of Gun class objects—the left column is a sample with a favorable view for detection, while in the right column is a perspective difficult to detect. The first row corresponds to the results of the Yolov3 model and the second row to the RetinaNet model

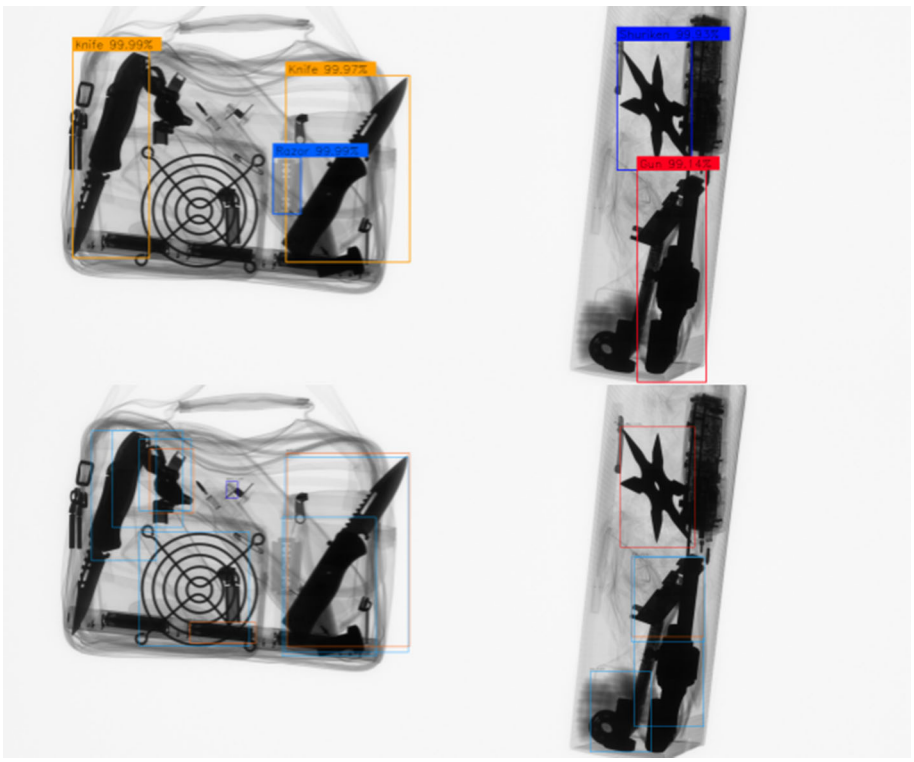

Fig. 12 Sample detection of Knife class objects—the left column is a sample with a favorable view for detection, while in the right column is a perspective difficult to detect. The first row corresponds to the results of the Yolov3 model and the second row to the RetinaNet model

significant objects, with the following number of instances per object: Gun 6118, Knife 6239, Razor 6052 and Shuriken 6111. In addition, the images of the backpacks without elements are added to the training set generating a total of 9849 images for the training phase of the models. Some elements of this training set are shown in Fig. 10. The

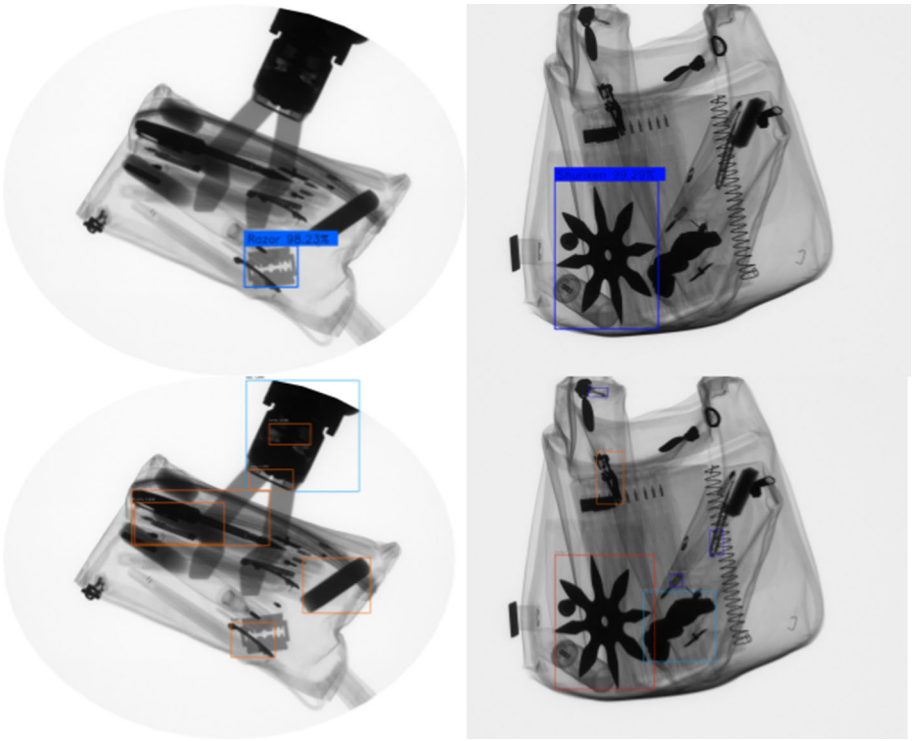

Fig. 13 Sample detection of Razor class objects -the left column is a sample with a favorable view for detection, while in the right column is a perspective difficult to detect. The first row corresponds to the results of the Yolov3 model and the second row to the RetinaNet model

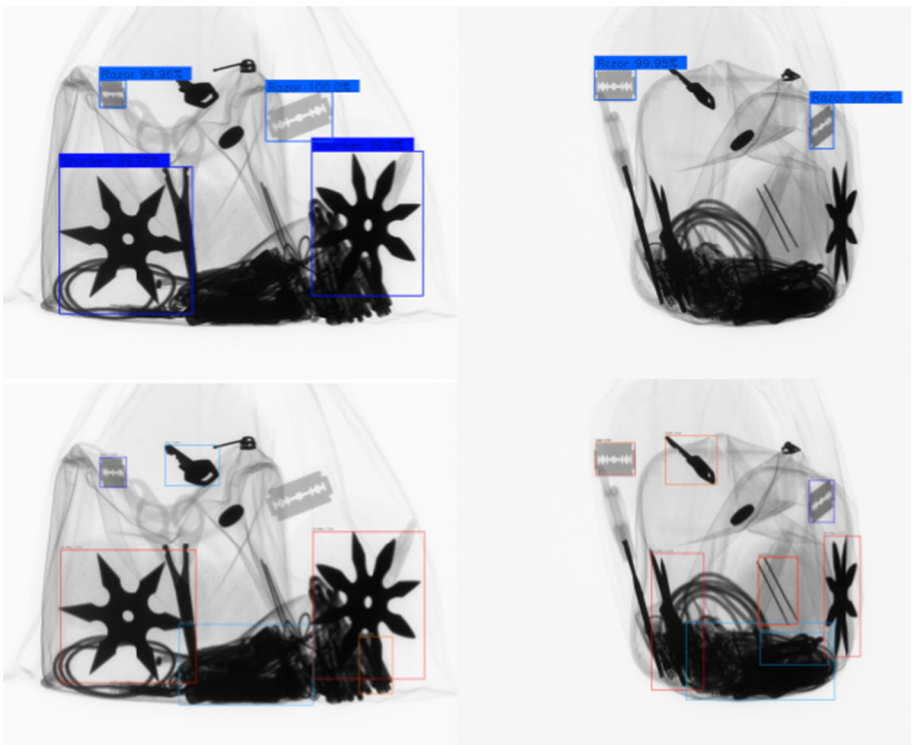

Fig. 14 Sample detection of Shuriken class objects—the left column is a sample with a favorable view for detection, while in the right column is a perspective difficult to detect. The first row corresponds to the results of the Yolov3 model and the second row to the RetinaNet model

validation set was defined as a randomly selected 10% subset of the simulated training data.

• *Training Subset B* The number of simulated elements using the trained PGGAN [19] and the number of real X-ray images of isolated threat objects is the same, i.e., half of instances of threat objects in simulated backpacks comes from Training Subset A. A full breakdown of the quantity of isolated objects is shown in Table 2.

• Training Subset C This subset considers threat objects in a frontal pose only (top rows in Fig. 5). For this, the 100 best samples are selected in the series of the Gun, Knife and Shuriken classes, which are defined as those views where the element can be classified without problems. The number of samples is the same of the other subsets.

5.3 Experiments

We conducted three experiments: A, B and C. In each one, we use the corresponding Training Subsets A, B and C from the previous Section. For each experiment, we test the trained model on Testing Subsets 1 and 2 (see Fig. 9), and we report the average precision (AP), mean average precision (mAP) and the weighted mean average precision (wmAP) metrics. We tested the Yolov2, Yolov3, SSD300 and RetinaNet models trained on our datasets. Since Yolov3 performance is better than Yolov2, the latter was tested only in Experiment A. The results are summarized in Tables 3 and 4.

5.3.1 Experiment A

In Tables 3 and 4 (see rows of Experiment A), we can see that the detectors that obtain the best results are YOLOv3 [44] and RetinaNet [23] in Testing Subset 1. Making a specific analysis for each object, we can show that in the Gun and Shuriken classes they have excellent results, specifically in the RetinaNet, SSD300 and YOLOv3 detectors while the Razor Blade class has acceptable results, under the performance of a human inspector. In turn, the worst results were obtained in the Knife category, where only acceptable results were obtained by detector YOLOv3, while the object was not properly detected at all by the other models. One of the factors that proved to be influential in the detection results was the extended shape of the knife, since its shape does not fit the anchor boxes of the models like the other threat objects as shown in Figs. 11, 12, 13 and 14.

On the other hand, very different results were obtained in Testing Subset 2 compared to Testing Subset 1. The YOLOv2 and YOLOv3 models were not able to generalize to the shape of the objects, that is, they only learn the particular shape of those found in the training set. In turn, the RetinaNet and SSD detectors have unexpected results, since both have a moderately acceptable performance in the Gun class, taking into account the different shapes of the gun. In the Knife class, the SSD model showed a clear superiority with respect to other algorithms, without presenting high index yields.

5.3.2 Experiment B

It consists in verifying whether the hypothesis of adding simulated threat objects through GAN gives more information to the training set, promoting better performance by the detectors.

In Tables 3 and 4, we can see the results of Experiment B (see rows of Experiment B). When examining the results in Testing Subset 1 (and comparing with the results of Experiment A), the slight general reduction in the performance of the detectors in all categories, so it is concluded that the entry of simulated samples similar to the real ones harms the specificity achieved in the detectors. The opposite is the case of the Testing Subset 2, where there is an increase in the performance of the two detectors in the Gun category. In the Knife category, the changes are not conclusive, since a disparate change is observed between the two models. In conclusion, the integration of simulated objects can provide greater generalization by the detector in the case of guns. Unfortunately, this affects the specificity that algorithms obtain when you want to detect similar views of the same object with which you train.

5.3.3 Experiment C

The next hypothesis to study is how the use of images of threat objects with a view that greatly distort the object affects the performance of the detection models, various views of which are shown in top rows of Fig. 5. This especially affects certain perspectives of the Knife and Shuriken class, since in certain views these objects are only a narrow element without depth.

In Tables 3 and 4, we can see the results of Experiment C. In Testing Subset 1, the trend indicates a slight decrease (between 6% and 7%) in the performance of the detectors in the classes less affected by the database change, i.e., Gun and Razor. On the other hand, in the Knife and Shuriken classes, there is an increase of up to 10% in the first class, while the second class there is a subtle decrease in performance that fluctuates between 1% and 3%). In the same way, in Testing Subset 2, a clear increase in the detection of the Gun class can be seen, giving the RetinaNet model a performance of up to 89% AP, so a good behavior of the algorithm in the detection is deduced from guns of another type to the trained. This may indicate that training with images with good perspectives is necessary for types of objects that have a great distortion in some view, such as a knife or shuriken, since it is these planes of the objects that are to be detected and not extreme cases where objects have deformations that make them unrecognizable. Unfortunately, this can adversely affect other classes that generalize better and thus it cannot be fully argued that the

use of this variant benefits the results obtained from the models.

5.4 Implementation

The experiments were carried out on an Ubuntu machine with AMD Ryzen 5 1600 CPU @3.2 GHZ with 16 GB memory and NVIDIA GeForce GTX 1060 (6 GB) GPU. The PGGAN [19] model was trained with a batch size of 64 on a pair of NVIDIA Titan Xp GPUs with each having a memory of 12GB. The model was trained starting from a 4×4 image resolution to the final target size of 128×128. It is to be noted that even higher resolution images (*e.g.*, 1024×1024 as shown in [19]) can be generated using the PGGAN model; however, we restrict the target image size to 128×128 due to constraints in time and computing resources.

For overlapping objects parameter values are used as B = 2 and C = 254 in Eq. (7). The detectors were implemented by third parties, in Python, particularly using the Keras framework on TensorFlow. The source of each library and the feature extractor are detailed in Table 5. Code and datasets are available on a public repository[2].

As usual in this type of problem, a method called transfer learning [56] is used, which consists of initializing the parameters with pre-trained values of the feature extractor network, which were calibrated using the ImageNet [8]. Training is done by adjusting all the parameters of the model, as recommended in [56]. In addition, the following configuration is used in training: optimizer: Adam [48]; epochs: 100; learning rate: 0.0001; input size: 448×448.

5.5 Computational time

The execution time of each detector is shown in Table 6. It is observed that the best performance, in terms of detection speed, is achieved by the SSD model. This result is mainly related to the complexity of the feature extractor network, since VGG16 [50] has a low number of hidden layers. In this way, the computational times obtained in these results could perfectly exceed the speed of real-time videos, which are usually made in 24 frames per second. Nevertheless, for baggage inspection that support human operators is more than enough.

[2] Code and datasets can be downloaded from https://github.com/dlsaavedra/Detector_GDXray.

Table 5 Code used in our experiments

Model	Code developed by	Feature extractor
YOLOv2	[3]	DarkNet-19 [43]
YOLOv3	[4]	DarkNet-53 [44]
SSD 300	[11]	VGG16 [50]
RetinaNet	[12]	ResNet50 [17]

Table 6 Detection time

Model	Time per image [s]	Images per second
YOLOv2	0.056	17.99
YOLOv3	0.182	5.50
SSD 300	0.036	28.07
RetinaNet	0.131	7.61

6 Conclusions

This research is meant to advance the study of detection of threat objects in X-rays. Through the proposed methodology, emphasis is placed on the ease in creating simulated images and their almost complete independence from the test images. In this way, a solid alternative is generated for the training of new deep learning models, without the need to have access to more complex databases and difficult public concession. Together with the proposal of a clear evaluation protocol that can be replicated, and therefore compared to future research that wishes to corroborate its methodology.

The idea of our work is to show how to use known object detection strategies in baggage inspection, rather than to develop new models from scratch. However, as we can see from our experiments, these strategies cannot be implemented in a straightforward manner, because they are not effective when the number of available X-ray images for training is low. Unfortunately, the databases in X-ray testing are rather limited. To overcome this problem, we propose a strategy for deep learning training that is performed with a low number of target-free X-ray images with superimposition of many simulated targets. The simulation is based on absorption's law of X-rays that allows us to superimpose different layers. Using this method, it is very simple to generate additional training data. Our proposed strategy was used to train known object detection models (YOLO, RetinaNet and SSD). The learned models were tested on real X-ray images.

In terms of results, the YOLOv3 [44] model yields around 90% (wmAP) on the GDXray dataset [31], while the second most successful model, RetinaNet [23],

achieves a 80% (wmAP), in the different series of test images that contained similar threat objects as in training. In particular, the Gun and Shuriken classes obtain results that exceed human operators (between 80% and 90%) [35], which is an indicator of their possible application in real environments.

With respect to the variations of Experiment A, we generate simulated elements using a combination of the recently proposed progressively growing GAN model [19] and a superimposition strategy [30], to create more diversity in our training dataset. However, adding such simulated elements resulted in an adverse effect in the detection in 'Testing Subset 1' where the testing set is very similar in appearance to the original training examples, so the simulated threat objects confuse the learning of the models.

However, in 'Testing Subset 2,' where the threat objects in testing are much different from the ones used for training, there was an improvement in performance of the models. Consequently, the introduction of simulated elements gives greater diversity that is favorable in cases where the models have to detect elements dissimilar to those with which they were trained, i.e., better generalization.

On the other hand, in Experiment C threat objects are selected for training, discarding those that have a distorted views. This produced a decrease in the Gun, Razor and Shuriken classes, while in the Knife class the performance was increased. This conflicting result can be explained by the greater sensitivity that the Knife class has in sight where it is captured, producing many views where the element is unidentifiable, in contrast to the other classes where only the profile position produces a greater complication. Therefore, if adequate views for detection of a threat object are present, it is recommended to use this last variant of discrimination.

In our experiments, we show that the proposed solution is simple (the implementation of the training can be done with a few lines of code using open source libraries), effective (average precision was around 0.90) and fast (training was done in a couple of hours, and testing can be performed in 0.2 s per image). We believe that this strategy means a practical solution to the problem of baggage inspection.

Funding This work was supported in part by Fondecyt Grants 1161314 and 1191131 from CONICYT—Chile.

Compliance with ethical standards

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

References

1. Akcay S, Kundegorski ME, Willcocks CG, Breckon TP (2018) Using deep convolutional neural network architectures for object classification and detection within X-ray baggage security imagery. IEEE Trans Inf Forensics Secur 13(9):2203–2215. https://doi.org/10.1109/TIFS.2018.2812196
2. Alcorn MA, Li Q, Gong Z, Wang C, Mai L, Ku WS, Nguyen A (2019) Strike (with) a pose: neural networks are easily fooled by strange poses of familiar objects. In: Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4845–4854)
3. Anh HN (2018) keras-yolo2. https://github.com/experiencor/keras-yolo2
4. Anh HN (2018) keras-yolo3. https://github.com/experiencor/keras-yolo3
5. Baştan M (2015) Multi-view object detection in dual-energy X-ray images. Mach Vis Appl 26(7):1045–1060. https://doi.org/10.1007/s00138-015-0706-x
6. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: Leonardis A, Bischof H, Pinz A (eds) Computer vision—ECCV 2006. Springer, Berlin, pp 404–417
7. Bolfing A, Halbherr T, Schwaninger A (2008) How image based factors and human factors contribute to threat detection performance in X-ray aviation security screening. In: Holzinger A (ed) HCI and usability for education and work. Springer, Berlin, pp 419–438
8. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: CVPR09
9. Dhiraj Jain DK (2019) An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery. Pattern Recognit Lett 120:112–119. https://doi.org/10.1016/j.patrec.2019.01.014
10. Everingham M, Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338. https://doi.org/10.1007/s11263-009-0275-4
11. Ferrari P (2018) ssd keras. https://github.com/pierluigiferrari/ssd_keras
12. Fizyr: keras-retinanet. https://github.com/fizyr/keras-retinanet (2018)
13. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial networks. arXiv e-prints arXiv:1406.2661
14. Goodfellow I, Bengio Y, Courville A (2016) Deep learning, vol 1. MIT Press, Cambridge, p 2
15. Harris C, Stephens M (1988) A combined corner and edge detector. In: In Proc. of fourth Alvey vision conference, pp. 147–151
16. Hartley R, Zisserman A (2003) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge
17. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. CoRR **abs/1512.03385**. http://arxiv.org/abs/1512.03385
18. Kanazawa A (2014) Locally scale-invariant convolutional neural networks. In: NeurIPS workshops
19. Karras T, Aila T, Laine S, Lehtinen J (2018) Progressive growing of GANs for improved quality, stability, and variation. In: international conference on learning representations. https://openreview.net/forum?id=Hk99zCeAb
20. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in neural

information processing systems, vol 25. Curran Associates Inc., New York, pp 1097–1105
21. Lin CC, Shiou FJ (2018) Object recognition based on foreground detection using X-ray imaging. J Chin Inst Eng 41(5):395–402. https://doi.org/10.1080/02533839.2018.1482235.
22. Lin T, Dollár P, Girshick RB, He K, Hariharan B, Belongie SJ (2016) Feature pyramid networks for object detection. CoRR **abs/1612.03144**. http://arxiv.org/abs/1612.03144
23. Lin T, Goyal P, Girshick RB, He K, Dollár P (2017) Focal loss for dense object detection. CoRR **abs/1708.02002**. http://arxiv.org/abs/1708.02002
24. Lin T, Maire M, Belongie SJ, Bourdev LD, Girshick RB, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. CoRR **abs/1405.0312** (2014). http://arxiv.org/abs/1405.0312
25. Liu W, Anguelov D, Erhan D, Szegedy C, Reed SE, Fu C, Berg AC (2015) SSD: single shot multibox detector. CoRR **abs/1512.02325**. http://arxiv.org/abs/1512.02325
26. Mathanker S (2013) X-ray applications in food and agriculture: a review. Trans ASABE (American Society of Agricultural and Biological Engineers) 56:1227–1239. https://doi.org/10.13031/trans.56.9785
27. McBee MP, Awan OA, Colucci AT, Ghobadi CW, Kadom N, Kansagra AP, Tridandapani S, Auffermann WF (2018) Deep learning in radiology. Acad. Radiol 25(11):1472–1480
28. Mery D (2015) Computer vision for X-ray testing. Springer, Berlin
29. Mery D (2015) Inspection of complex objects using multiple-X-ray views. IEEE/ASME Trans Mechatron 20(1):338–347. https://doi.org/10.1109/TMECH.2014.2311032
30. Mery D, Katsaggelos AK (2017) A logarithmic X-ray imaging model for baggage inspection: Simulation and object detection. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp. 251–259 . 10.1109/CVPRW.2017.37
31. Mery D, Riffo V, Zscherpel U, Mondragón G, Lillo I, Zuccar I, Lobel H, Carrasco M (2015) Gdxray: the database of X-ray images for nondestructive testing. J Nondestruct Eval 34:42. https://doi.org/10.1007/s10921-015-0315-7
32. Mery D, Riffo V, Zuccar I, Pieringer C (2013) Automated X-ray object recognition using an efficient search algorithm in multiple views. In: 2013 IEEE conference on computer vision and pattern recognition workshops, pp. 368–374 . 10.1109/CVPRW.2013.62
33. Mery D, Svec E, Arias M, Riffo V, Saavedra JM, Banerjee S (2017) Modern computer vision techniques for X-ray testing in baggage inspection. IEEE Trans Syst Man Cybern Syst 47(4):682–692. https://doi.org/10.1109/TSMC.2016.2628381
34. Miao C, Xie L, Wan F, Su C, Liu H, Jiao J, Ye Q (2019) Sixray : A large-scale security inspection X-ray benchmark for prohibited item discovery in overlapping images. CoRR **abs/1901.00303**. http://arxiv.org/abs/1901.00303
35. Michel S, Koller SM, de Ruiter JC, Moerland R, Hogervorst M, Schwaninger A (2007) Computer-based training increases efficiency in X-ray image interpretation by aviation security screeners. In: 2007 41st Annual IEEE international Carnahan conference on security technology, pp. 201–206. 10.1109/CCST.2007.4373490
36. Mikhaylichenko A, Demyanenko Y, Grushko E (2016) Automatic detection of bone contours in X-ray images. In: AIST (Supplement), pp. 212–223
37. Mikolajczyk A, Grochowski M (2018) Data augmentation for improving deep learning in image classification problem. In 2018 International interdisciplinary PhD workshop (IIPhDW) (pp. 117-122). IEEE
38. 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 . 10.1109/THS.2008.4534504
39. Perez L, Wang J (2017) The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621
40. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: International conference on learning representations (ICLR)
41. Ramani R, Vanitha S, Valarmathy S (2013) The pre-processing techniques for breast cancer detection in mammography images. Int J Image Gr Signal Process 5:47–54. https://doi.org/10.5815/ijigsp.2013.05.06
42. Redmon J, Divvala SK, Girshick RB, Farhadi A (2015) You only look once: unified, real-time object detection. CoRR **abs/1506.02640**. http://arxiv.org/abs/1506.02640
43. Redmon J, Farhadi A (2016) YOLO9000: better, faster, stronger. CoRR **abs/1612.08242**. http://arxiv.org/abs/1612.08242
44. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. CoRR **abs/1804.02767**. http://arxiv.org/abs/1804.02767
45. Ren S, He K, Girshick RB, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. CoRR **abs/1506.01497** . http://arxiv.org/abs/1506.01497
46. RichardWebster B, Anthony SE, Scheirer WJ (2018) Psyphy: a psychophysics driven evaluation framework for visual recognition. IEEE Trans Pattern Anal Mach Intell 41(9):2280–2286
47. Riffo V, Mery D (2016) Automated detection of threat objects using adapted implicit shape model. IEEE Trans Syst Man Cybern Syst 46(4):472–482. https://doi.org/10.1109/TSMC.2015.2439233
48. Ruder S (2016) An overview of gradient descent optimization algorithms. CoRR **abs/1609.04747** . http://arxiv.org/abs/1609.04747
49. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):60
50. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** . http://arxiv.org/abs/1409.1556
51. Steitz JO, Saeedan F, Roth S (2018) Multi-view X-ray R-CNN. In: Proceedings of the German conference on pattern recognition (GCPR), LNCS 11269, pp. 153–158
52. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9
53. Szeliski R (2010) Computer vision: algorithms and applications. Springer, Berlin
54. Turcsany D, Mouton A, Breckon TP (2013) Improving feature-based object recognition for X-ray baggage security screening using primed visualwords. In: 2013 IEEE International conference on industrial technology (ICIT), pp. 1140–1145. IEEE
55. Wu N, Phang J, Park J, Shen Y, Huang Z, Zorin M, Jastrzebski S, Fevry T, Katsnelson J, Kim E, Wolfson S, Parikh U, Gaddam S, Lin LLY, Ho K, Weinstein JD, Reig B, Gao Y, Toth H, Pysarenko K, Lewin A, Lee J, Airola, K, Mema E, Chung S, Hwang E, Samreen N, Kim SG, Heacock L, Moy L, Cho K, Geras KJ (2019) Deep neural networks improve radiologists' performance in breast cancer screening. arXiv preprint arXiv:1903.08297
56. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? CoRR **abs/1411.1792**. http://arxiv.org/abs/1411.1792
57. Zentai G (2008) X-ray imaging for homeland security. In: 2008 IEEE international workshop on imaging systems and techniques, pp. 1–6 . 10.1109/IST.2008.4659929

58. Zhao Z, Zheng P, Xu S, Wu X (2018) Object detection with deep learning: a review. CoRR **abs/1807.05511**. http://arxiv.org/abs/1807.05511
59. Zou L, Yusuke T, Hitoshi I (2020) Dangerous objects detection of X-ray images using convolution neural network. In: Yang CN, Peng SL, Jain LC (eds) Security with intelligent computing and big-data services. Springer, Cham, pp 714–728