Neural Computing and Applications (2022) 34:1239–1250
https://doi.org/10.1007/s00521-021-06411-x

ORIGINAL ARTICLE

Tensor pooling-driven instance segmentation framework for baggage threat recognition

Taimur Hassan[1] · Samet Akçay[2] · Mohammed Bennamoun[3] · Salman Khan[4] · Naoufel Werghi[1]

Received: 14 February 2021 / Accepted: 17 August 2021 / Published online: 5 September 2021

Abstract
Automated systems designed for screening contraband items from the X-ray imagery are still facing difficulties with high clutter, concealment, and extreme occlusion. In this paper, we addressed this challenge using a novel multi-scale contour instance segmentation framework that effectively identifies the cluttered contraband data within the baggage X-ray scans. Unlike standard models that employ region-based or keypoint-based techniques to generate multiple boxes around objects, we propose to derive proposals according to the hierarchy of the regions defined by the contours. The proposed framework is rigorously validated on three public datasets, dubbed GDXray, SIXray, and OPIXray, where it outperforms the state-of-the-art methods by achieving the mean average precision score of 0.9779, 0.9614, and 0.8396, respectively. Furthermore, to the best of our knowledge, this is the first contour instance segmentation framework that leverages multi-scale information to recognize cluttered and concealed contraband data from the colored and grayscale security X-ray imagery.

Keywords Aviation security · Structure tensors · Instance segmentation · Baggage X-ray scans

1 Introduction

X-ray imagery is a widely used modality for nondestructive testing [1], especially for screening illegal and smuggled items at airports, cargoes, and malls. Manual baggage inspection is a tiring task and susceptible to errors caused due to exhausting work routines and less experienced personnel. Initial systems proposed to address these problems employed conventional machine learning [2]. Driven by hand-engineered features, these methods are only applicable to limited data and confined environmental settings [3]. Recently, attention has turned to deep learning methods, which gave a neat boost in accuracy and generalization capacity toward screening prohibited baggage items [4, 5]. However, deep learning methods are also prone to clutter, and occlusion [6]. This limitation emanates from the proposal generation strategies which have been designed for the color images [7]. Unlike RGB scans, X-ray imagery lacks texture and exhibits low-intensity variations between cluttered objects. This intrinsic difference makes the region-based or anchor-based proposal generation methods such as Mask R-CNN [8], Faster R-CNN [9], RetinaNet [10], and YOLO [11] less robust for detecting the cluttered contraband data [6]. Moreover, the problem is further accentuated by the class imbalance nature of the contraband items in the real world [7]. Despite the considerate strategies proposed to alleviate the occlusion and the imbalance nature [12, 13], recognizing threatening objects in highly cluttered and concealed scenarios is still an open problem [14].

1.1 Contributions

In this paper, we propose a novel multi-scale contour instance segmentation framework for identifying suspicious items using X-ray scans. Unlike standard models that

✉ Taimur Hassan
taimur.hassan@ku.ac.ae

[1] Center for Cyber-Physical Systems (C2PS), Department of Electrical Engineering and Computer Sciences, Khalifa University, Abu Dhabi, United Arab Emirates

[2] Department of Computer Sciences, Durham University, Durham, UK

[3] Department of Computer Science and Software Engineering, The University of Western Australia, Perth, Australia

[4] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates

employ region-based or keypoint-based techniques to generate multiple boxes around objects [5, 6, 15], we propose to derive proposals according to the hierarchy of the regions defined by the contours. The insight driving this approach is that contours are the most reliable cue in the X-ray scans due to the lack of surface texture. For example, the occluded items exhibit different transitional patterns based upon their orientation, contrast, and intensity. We try to amplify and exploit this information through the multi-scale scan decomposition, which boosts the proposed framework's capacity for detecting the underlying contraband data in the presence of clutter. Furthermore, we are also motivated by the fact that organic material's suspicious items show only their outlines in the X-ray scans [16]. To summarize, the main features of this paper are:

- Detection of overlapping suspicious items by analyzing their predominant orientations across multiple scales within the candidate scan. Unlike [16–18], we propose a novel tensor pooling strategy to decompose the scan across various scales and fuse them via a single multi-scale tensor. This scheme results in more salient contour maps (see Fig. 1), boosting our framework's capacity for handling dulled, concealed, and overlapping items.
- A thorough validation on three publicly available large-scale baggage X-ray datasets, including the OPIXray [12], which is the only dataset allowing a quantitative measure of the level of occlusion.
- Unlike state-of-the-art methods such as CST [16], TST [17], and DTS [18], the performance of the proposed framework to detect occluded items has been quantitatively evaluated on OPIXray [12] dataset. Please see Table 4 for more details.

2 Related work

Many researchers have developed computer-aided screening systems to identify potential baggage threats [19]. While a majority of these frameworks are based on conventional machine learning [20], the recent works also employ supervised [5] and unsupervised [21] deep learning, and these methods outperform conventional approaches both in terms of performance, and efficiency [6]. In this section, we discuss some of the major baggage threat detection works. We refer the readers to [14, 22] for an exhaustive survey.

Fig. 1 **a** An exemplar X-ray scan from the GDXray dataset [39], **b** contour map obtained through the modified structure tensors in [16, 17], **c** contour map obtained through proposed tensor pooling strategy

2.1 Traditional methods

The early baggage screening systems were driven via classification [3], segmentation [23], and detection [24] approaches to identify potential threats and smuggled items. Here, the work of Bastan et al. [2] is appreciable, which identifies the suspicious and illegal items within the multi-view X-ray imagery through fused SIFT- and SPIN-driven SVM model. Similarly, SURF [23] and FAST-SURF [25] have also been used with the bag of words [26] to identify threatening items from the security X-ray imagery. Moreover, approaches like adapted implicit shape model [27] and adaptive sparse representation [19] were also commendable for screening suspicious objects from the X-ray scans.

2.2 Deep learning frameworks

The deep learning-based baggage screening frameworks have been broadly categorized into supervised and unsupervised learning schemes.

2.2.1 Supervised methods

The initial deep learning approaches involved scan-level classification to identify the suspicious baggage content [5, 28–31]. However, with the recent advancements in object detection, researchers also employed sophisticated detectors [32] like RetinaNet [10], YOLO [33, 34], Inception [35], and Faster R-CNN [9] to not only recognize the contraband items from the baggage X-ray scans but also to localize them via bounding boxes [6]. Moreover, researchers also proposed semantic segmentation [36] and instance segmentation [17] models to recognize threatening and smuggled items from the grayscale and colored X-ray imagery. Apart from this, Xiao et al. [37] presented an efficient implementation of Faster R-CNN [9] to detect suspicious data from the TeraHertz imagery. Dhiraj et al. [38] used Faster R-CNN [9], YOLOv2 [34], and Tiny YOLO [34] to screen baggage threats contained within the scans of a publicly available GDXray dataset [39]. Gaus et al. [7] utilized RetinaNet [10], Faster R-CNN [9], Mask R-CNN [8] (driven through ResNets [40], VGG-16 [41], and SqueezeNet [42]) to detect prohibited baggage items. In another approach [15], they analyzed the transferability

of these models on a similarly styled X-ray imagery contained within their local dataset as well as the SIXray10 subset of the publicly available SIXray dataset [13]. Similarly, Akçay et al. [6] compared Faster R-CNN [9], YOLOv2 [34], R-FCN [43], and sliding-window CNN with the AlexNet [44]-driven SVM model to recognize occluded contraband items from the X-ray imagery. Miao et al. [13] explored the imbalanced nature of the contraband items in the real world by developing a class-balanced hierarchical refinement (CHR) framework. Furthermore, they extensively tested their framework (backboned through different classification models) on their publicly released SIXray [13] dataset. Wei et al. [12] presented a plug-and-play module dubbed De-occlusion Attention Module (DOAM) that can be coupled with any object detector to enhance its capacity toward screening occluded contraband items. DOAM was validated on the publicly available OPIXray [12] dataset, which is the first of its kind in providing quantitative assessments of baggage screening frameworks under low, partial, and full occlusion [12]. Apart from this, Hassan et al. [16] also addressed the imbalanced nature of the contraband data by developing the cascaded structure tensors (CST)-based baggage threat detector. CST [16] generates a balanced set of contour-based proposals, which are then utilized in training the backbone model to screen the normal and abnormal baggage items within the candidate scan [16]. Similarly, to overcome the need to train the threat detection systems on large-scale and well-annotated data, Hassan et al. [18] introduced meta-transfer learning-based dual tensor-shot (DTS) detector. DTS [18] analyzes the scan's saliency to produce low- and high-density contour maps from which the suspicious contraband items are identified effectively with few-shot training [18]. In another approach, Hassan et al. [17] developed an instance segmentation-based threat detection framework that filters the contours of the suspicious items from the regular content via trainable structure tensors (TST) [17] to identify them accurately within the security X-ray imagery.

2.2.2 Unsupervised methods

While most baggage screening frameworks involved supervised learning, researchers have also explored adversarial learning to screen contraband data as anomalies. Akçay et al. [45], among others, laid the foundation of unsupervised baggage threat detection by proposing GANomaly [45], an encoder–decoder–encoder network trained in an adversarial manner to recognize prohibited items within baggage X-ray scans. In another work, they proposed Skip-GANomaly [21] which employs skip connections in an encoder–decoder topology that not only gives better latent representations for detecting baggage threats but also reduces the overall computational complexity of GANomaly [45].

The rest of the paper is organized as follows: Sect. 3 presents the proposed framework. Section 4 describes the experimental setup. Section 5 discusses the results obtained with three public baggage X-ray datasets. Section 6 concludes the paper and enlists future directions.

3 Proposed approach

The block diagram of the proposed framework is depicted in Fig. 2. The input scan is fed to the tensor pooling module (block A) to generate a multi-scale tensor representation, revealing the baggage content's transitional patterns at multiple predominant orientations and across various scales. Afterward, the multi-scale tensor is passed to the encoder–decoder backbone (block B), implementing the newly proposed contour maps-based instance segmentation. This block extracts the contours of the prohibited data while eliminating the irrelevant scan content. In the third stage (block C), each extracted contour, reflecting the contraband item instance, is utilized in generating the respective mask and the bounding box for localization. In the subsequent sections, we present a detailed description of each module within the proposed framework.

3.1 Tensor pooling module

The tensor pooling module decomposes the input scan into n levels of a pyramid. From each level of the pyramid, the baggage content's transitional patterns are generated by analyzing their distribution of orientations within the associated image gradients. In the proposed tensor pooling scheme, we highlight the transitional patterns in N image gradients (corresponding to N directions) by computing the following $N \times N$ block-structured symmetric matrix [16, 17]:

$$\begin{bmatrix} \phi * (\nabla^0.\nabla^0) & \phi * (\nabla^1.\nabla^0) & \cdots & \phi * (\nabla^{N-1}.\nabla^0) \\ \phi * (\nabla^0.\nabla^1) & \phi * (\nabla^1.\nabla^1) & \cdots & \phi * (\nabla^{N-1}.\nabla^1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi * (\nabla^0.\nabla^{N-1}) & \phi * (\nabla^1.\nabla^{N-1}) & \cdots & \phi * (\nabla^{N-1}.\nabla^{N-1}) \end{bmatrix}, \tag{1}$$

Each tensor $(\phi * (\nabla^k.\nabla^m))$ in the above block matrix is an outer product of two image gradients and a smoothing filter ϕ. Moreover, the orientation (θ), of the image gradient ∇^j, is computed through: $\theta = \frac{2\pi j}{N}$, where j ranges from 0 to $N-1$. Since the block-structured matrix in Eq. 1 is symmetric, we obtain $\frac{N(N-1)}{2}$ unique tensors. From this group, we derive the coherent tensor, reflecting the baggage items'

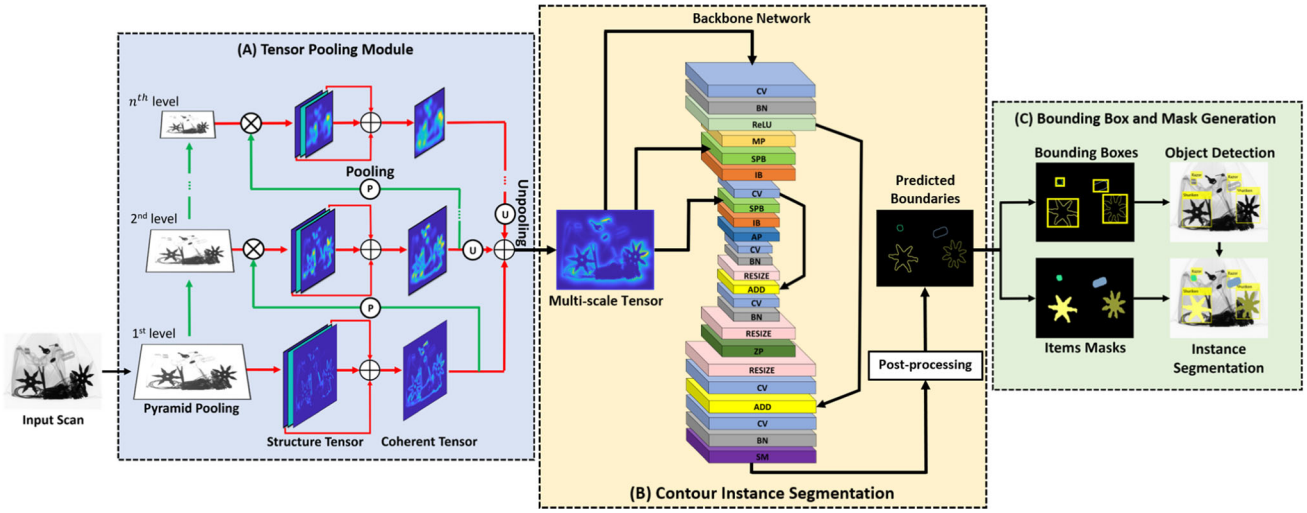

Fig. 2 Block diagram of the proposed framework. The input scan is passed to the tensor pooling module to extract the tensor representations encoding the baggage items' contours at different orientations. These representations are fused into a single multi-scale tensor and passed afterward to an asymmetric encoder–decoder backbone that segments and recognizes the contraband item's contours while suppressing the rest of the baggage content. For each detected contour, the corresponding bounding box and mask are generated to localize the detected contraband items. *CV* Convolution, *BN* batch normalization, *SPB* shape-preserving block, *IB* identity block, *MP* Max pooling, *AP* average pooling, *ZP* zero padding, *SM* softmax

predominant orientations. The coherent tensor is a single tensor representation generated by adding the most useful tensors out of the $\frac{N(N+1)}{2}$ unique tensor set. Here, it should be noted that these useful tensors are selected by ranking all the $\frac{N(N+1)}{2}$ unique tensors according to their norm.

Moreover, the coherent tensor also reveals the variations in the intensity of the cluttered baggage items, aiding in generating individual contours for each item. However, this scheme analyzes only the intensity variations of the baggage items at a single scale, limiting the extraction of the objects having lower transitions with the background [16, 17]. To address this limitation, we propose a multi-scale tensor fusing the X-ray scan transitions from coarsest to finest levels so that each item, even having a low-intensity difference with the background, can be adequately highlighted. For example, see the boundaries of: the *razor* in a multi-scale tensor representation in Fig. 1c, the *straight knife* in Fig. 3g, the two *knives* and a *gun* in Fig. 3h, and the two *guns* and a *knife* in Fig. 3i.

Algorithm 1: Tensor Pooling Module

Input: X-ray scan (I), Scaling Factor (n), Number of Orientations (N)
Output: Multi-scale Tensor (M_t)
$[r, c]$= size(I)
Initialize M_t (of size $r \times c$) with zeros
Set $\eta = 2$ // pyramid pooling factor
for $i = 0$ *to* $n - 1$ **do**
  if i *is 0* **then**
    $\Im$ = ComputeTensors(I, N) // $\Im$: Tensors
    $\Im_c$ = GetCoherentTensor($\Im$)
    $M_t = M_t + \Im_c$
  else
    $[s, t]$= size(I)
    if $(min(s, t)\ \%\ \eta) \neq 0$ *or* $min(s, t) < \eta$ **then**
      break
    end
    I = Pool(I, η)
    $\Im_c$ = Pool($\Im_c$, η)
    $I = I \times \Im_c$
    $\Im$ = ComputeTensors(I, N)
    $\Im_c$ = GetCoherentTensor($\Im$)
    $M_t = M_t$ + Unpool($\Im_c$, η^i)
  end
end

As mentioned earlier, the multi-scale tensors are computed through pyramid pooling (up to *n*th level). At any *l*th level (such that $2 \leq l \leq n$), we multiply, pixel-wise, the decomposed image with the transitions obtained at the previous $(l - 1)$ levels. In so doing, we ensure that the edges of the contraband items (procured earlier) are retained across each scale. The full procedure of the

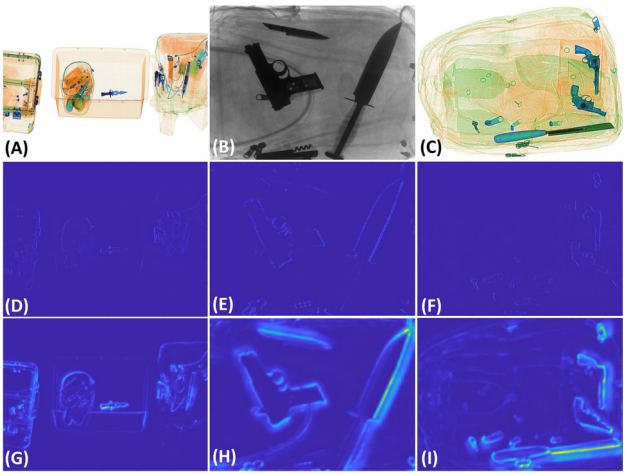

Fig. 3 Difference between conventional structure tensors (used in [16, 17]), and proposed multi-scale tensor approach. First row shows the original scans from OPIXray [12], GDXray [39], and SIXray [13] dataset. The second row shows the output for the conventional structure tensors [16, 17]. The third row shows the output for the proposed tensor pooling module

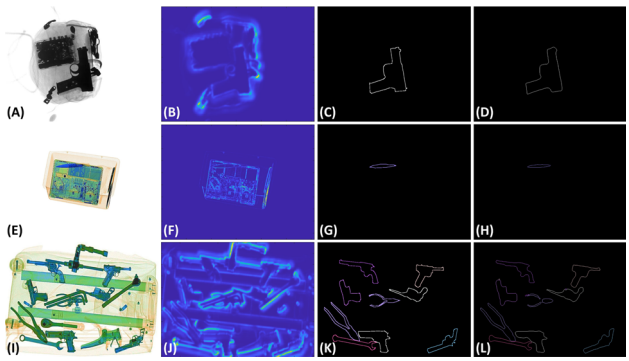

Fig. 4 Contour instance segmentation from multi-scale tensors. The first column shows the original scans, the second column shows the multi-scale tensor representations, the third column shows the ground truths, and the fourth column shows the extracted contours of the contraband items

proposed tensor pooling module is depicted in Algorithm 1 and also shown in Fig. 2.

The multi-scale tensor is then passed to the proposed encoder–decoder model to extract the contours of the individual suspicious items. A detailed discussion about contour instance segmentation is presented in the subsequent section (and Fig. 4).

3.2 Contour instance segmentation

The contour instance segmentation is performed through the proposed asymmetric encoder–decoder network, which assigns the pixels in the multi-scale tensors to one of the

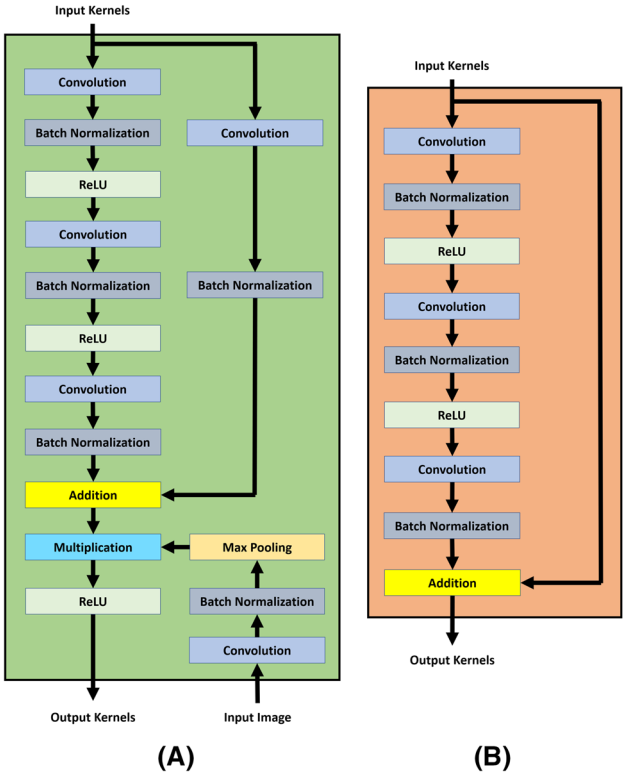

Fig. 5 **a** Shape-preserving block (SPB), **b** identity block (IB)

following categories $\mathcal{C}_{k=1:\mathcal{M}+1}$ where $\mathcal{M}$ denotes the number of prohibited items' instances to which we add the class *background* which include background and irrelevant pixels (i.e., pixels belonging to a non-suspicious baggage content).

Furthermore, to differentiate between the contours of the normal and suspicious items, the custom shape-preserving (SPB) and identity blocks (IB) have been added within the encoder topology. The SPB, as depicted in Figs. 2 and 5a, integrates the multi-scale tensor map (after scaling) in the feature map extraction to enforce further the attention on prohibited items' outlines. The IB (Fig. 5a), inspired by ResNet architecture [40], acts as a residual block to emphasize the feature maps of the previous layer.

Apart from this, the whole network encompasses one input, one zero padding, 22 convolution, 20 batch normalization, 12 activation, four pooling, two multiply, six addition, three lambda (that implements the custom functions), and one reshape layer. Moreover, we use skip connections (via addition) within the encoder–decoder to refine the extracted items' boundaries. The number of parameters within the network is 1,308,160, from which around 6912 parameters are non-trainable. The detailed summary of the proposed model (including the

architectural details of the SPB and IB blocks) is available in the source code repository.[1]

3.3 Bounding box and mask generation

After segmenting the contours, we perform morphological post-processing to remove tiny and isolated fragments. The obtained outlines contain both open and closed contours of the underlying suspicious items. The closed contours can directly lead toward generating the corresponding item's mask. For open contours, we join their endpoints and then derive their masks through morphological reconstruction. Afterward, we generate the items' bounding boxes from the masks as shown in Fig. 2c.

4 Experimental setup

This section presents the details about the experimental protocols, datasets, and evaluation metrics which were in order to assess the proposed system's performance and compare it with state-of-the-art methods.

4.1 Datasets

We validated the proposed framework on three different publicly available baggage X-ray datasets, namely GDXray [39], SIXray [13], and OPIXray [12]. The detailed description of these datasets is presented below.

4.1.1 GDXray

GDXray [39] was first introduced in 2015, and it contains 19,407 high-resolution grayscale X-ray scans. The dataset is primarily designed for the nondestructive testing purposes, and the scans within GDXray [39] are arranged into five categories, i.e., *welds*, *baggage*, *casting*, *settings*, and *nature*. But *baggage* is the only relevant group for this study, and it contains 8150 grayscale X-ray scans. Moreover, the dataset also presents the detailed annotations for the prohibited items such as *shuriken*, *knives*, *guns*, and *razors*. As per the dataset standard, 400 scans from GDXray [39] were used for training purposes, while the remaining scans were used for testing purposes.

4.1.2 SIXray

SIXray [13] is a recently introduced large-scale security inspection X-ray dataset. It contains a total of 1,059,231 colored X-ray scans from which 8929 scans are positive (containing prohibited items such as *knives*, *wrenches*, *guns*, *pliers*, *hammer,* and *scissors* along with their ground truths), and 1,050,302 are negative (containing only the normal items). To validate the performance against class imbalance, the authors of the dataset presented three subset schemes of the dataset, namely SIXray10, SIXray100, and SIXray1000 [13]. Moreover, SIXray [13] is also the largest and most challenging dataset (to date) designed to assess threat detection frameworks' performance toward screening extremely cluttered and highly imbalanced contraband data [13, 18]. As per the SIXray [13] dataset standard, we used 80% scans for the training and the rest of 20% for testing.

4.1.3 OPIXray

OPIXray [12] is the most recent baggage X-ray dataset (released publicly for the research community in 2020). It contains 8885 colored X-ray scans. As per the dataset standard, out of these 8885 scans, 7109 are to be utilized for the training purposes, while the remaining 1776 are to be used for testing purposes, to detect *scissor*, *straight knife*, *multi-tool knife*, *folding knife*, and *utility knife*. Moreover, the dataset authors also quantified occlusion within the test scans into three levels, i.e., OP1, OP2, and OP3. OP1 indicates that the contraband items within the candidate scan contain no or slight occlusion, and OP2 depicts a partial occlusion, while OP3 represents severe or full occlusion cases.

We also want to highlight here that the resolution of the scans within each dataset varies significantly (except for OPIXray [12]). For example, on GDXray [39], the scan resolution varies as 2688×2208, 900×1430, 850×850, and 601×1241. Similarly, on SIXray [13], the scan resolution varies as $681 \times 549 \times 3$, $801 \times 482 \times 3$, $649 \times 571 \times 3$, $1024 \times 640 \times 3$, and $675 \times 382 \times 3$. But on OPIXray [12], the resolution of all the scans is 1225x954x3. In order to process all the scans with the proposed framework, we have re-sized them to the common resolution of $576 \times 768 \times 3$, which is extensively used in the recently published frameworks [16–18].

4.2 Training and implementation details

The proposed framework was developed using Python 3.7.4 with TensorFlow 2.2.0 and Keras APIs on a machine having Intel Core i9-10940X@3.30 GHz CPU, 128 GB RAM and an NVIDIA Quadro RTX 6000 with cuDNN v7.5, and a CUDA Toolkit 10.1.243. Some utility functions are also implemented using MATLAB R2021a. Apart from this, the training on each dataset was conducted for a maximum of 50 epochs using ADADELTA [46] as an

[1] The source code of the proposed framework along with its complete documentation is available at https://github.com/taimurhassan/tensorpooling.

optimizer (with the default learning and decay rate configurations) and a batch size of 4. Moreover, 10% of the training samples from each dataset were used for the validation (after each epoch). For the loss function, we used the focal loss [10] expressed below:

$$l_f = -\frac{1}{b_s}\sum_{i=0}^{b_s-1}\sum_{j=0}^{c-1}\alpha(1-p(l_{i,j}))^{\gamma}t_{i,j}\log(p(l_{i,j})) \quad (2)$$

where c represents the total number of classes, and b_s denotes the batch size. $p(l_{i,j})$ denotes the predicted probability of the logit $l_{i,j}$ generated from ith training sample for the jth class, $t_{i,j}$ tells if the ith training sample actually belongs to the jth class or not, and the term $\alpha(1-p(l_{i,j}))^{\gamma}$ represents the scaling factor that gives more weight to the imbalanced classes. (In other words, it penalizes the network to give emphasize to the classes for which the network obtains low prediction scores.) Through rigorous experiments, we empirically selected the optimal value of α and γ as 0.25 and 2, respectively, as they result in faster learning for each dataset while simultaneously showing good resistance to the imbalanced data. Apart from this, architecturally, the kernel sizes within the proposed encoder–decoder backbone vary as 3×3 and 7×7, whereas the number of kernels varies as 64, 128, 256, 512, 1024, and 2048. Moreover, the pooling size within the network remained 2×2 across various network depths to perform the feature decomposition (at each depth) by the factor of 2. For more architectural and implementation details of the proposed framework, we refer the reader to the source code, which we have released publicly for the research community on GitHub[1].

4.3 Evaluation metrics

In order to assess the proposed approach and compare it with the existing works, we used the following evaluation metrics:

4.3.1 Intersection over Union

Intersection over Union (IoU) tells how accurately the suspicious items have been extracted, and it is measured by checking the pixel-level overlap between the predictions and the ground truths. Mathematically, IoU is defined as:

$$\text{IoU} = \frac{T_p}{T_p + F_p + F_n}, \quad (3)$$

where T_p are true positives (indicating that the pixels of the contraband items are correctly predicted w.r.t the ground truth), F_p represents false positives (indicating that the background pixels are incorrectly classified as positives), and F_n represents false negatives (meaning that the pixels of the contraband items are misclassified as background). Furthermore, we also calculated the mean IoU (μIoU) by taking an average of the IoU score for each contraband item class.

4.3.2 Dice coefficient

Apart from IoU scores, we also computed the dice coefficient (DC) scores to assess the proposed system's performance for extracting the contraband items. DC is calculated through:

$$\text{DC} = \frac{2T_p}{2T_p + F_p + F_n}, \quad (4)$$

Compared to IoU, DC gives more weightage to the true positives (as evident from Eq. 4). Moreover, the mean DC (μDC) is calculated by averaging DC scores for each category.

4.3.3 Mean average precision

The mean average precision (mAP) (in the proposed study) is computed by taking the mean of average precision (AP) score calculated for each contraband item class for the IoU threshold $\geq$ 0.5. Mathematically, mAP is expressed below:

$$\text{mAP} = \sum_{i=0}^{n_c-1} AP(i), \quad (5)$$

where n_c denotes the number of contraband items in each dataset. Here, we want to highlight that to achieve fair comparison with the state of the art, we have used the original bounding box ground truths of each dataset for measuring the proposed framework's performance toward extracting the suspicious and illegal items.

5 Results

In this section, we present the detailed results obtained with GDXray [39], SIXray [13], and OPIXray [12] datasets. Before going into the experimental results, we present detailed ablation studies to determine the proposed framework's hyper-parameters. We also report a detailed comparison of the proposed encoder–decoder network with the popular segmentation models.

5.1 Ablation studies

The ablation studies in this paper aim to determine the optimal values for (1) the number of orientations and scaling levels within the tensor pooling module and (2) the

choice of the backbone model for performing the contour instance segmentation.

5.1.1 Number of orientations and the scaling levels

The tensor pooling module highlights the baggage content transitions in the image gradients oriented in N directions and up to n scaling levels. Increasing these parameters helps generate the best contour representation leading toward a more robust detection, but also incurs additional computational cost. As depicted in Fig. 6a, we can see that for GDXray dataset [39] with $N = 2$, $n = 2$, we obtain an mAP score of 0.82. With the combination $N = 5$, $n = 5$, we get 16.54% improvements in the detection performance but at the expense of a 97.71% increase in computational time (see Fig. 6d). Similarly, on the SIXray dataset [13], we obtain 18.36% improvements in the detection performance (by increasing N and n) at the expense of 95.88% in the computational time (see Fig. 6b, e). The same behavior is also noticed for OPIXray dataset [12] in Fig. 6c, f. Considering all the combinations depicted in Fig. 6, we found that $N = 4$ and $n = 3$ provide the best trade-off between the detection and run-time performance across all three datasets.

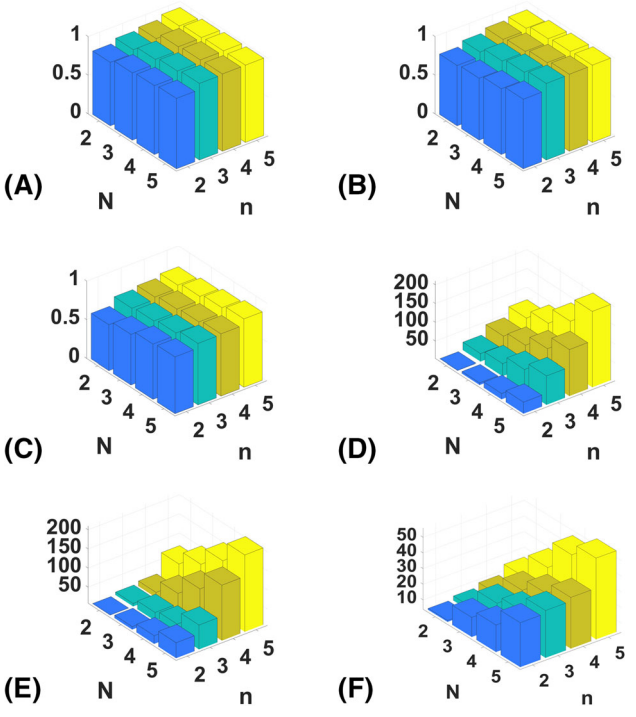

Fig. 6 Detection performance of the proposed system in terms of mAP (**a**–**c**) and computational time in terms of seconds (**d**–**f**) obtained for GDXray [39], SIXray [13], and OPIXray [12] datasets, respectively

5.1.2 Choice of a backbone model

The proposed backbone model has been specifically designed to segment the suspicious items' contours while discarding the normal baggage content. In this series of experiments, we compared the proposed asymmetric encoder–decoder model's performance with popular encoder–decoder, scene parsing, and fully convolutional networks. In terms of μDC and μIoU, we report the performance results in Table 1. We can observe that the proposed framework achieves the best extraction performance on OPIXray [12] and SIXray [13] dataset, leading the second-best UNet [47] by 2.34% and 3.72%. On the GDXray [39], however, it lags from the FCN-8 [48] and PSPNet [49] by 6.54% and 5.91%, respectively. But as our model outperforms all the other architectures on the large-scale SIXray [13] and OPIXray [12] datasets, we chose it as a backbone for the rest of the experimentation.

5.2 Evaluation on GDXray dataset

The performance of the proposed framework and that of the state-of-the-art methods on the GDXray [39] dataset are reported in Table 2. We can observe here that the proposed framework outperforms the CST [16] and the TST framework [17] by 4.98% and 1.07%, respectively. Furthermore, we wanted to highlight the fact that CST [16] is only an object detection scheme, i.e., it can only localize the detected items but cannot generate their masks. Masks are very important for the human observers in cross-verifying the baggage screening results (and identifying the false positives), especially from the cluttered and challenging grayscale scans. In Fig. 7, we report some of the cluttered and challenging cases showcasing the effectiveness of the proposed framework in extracting the overlapping

Table 1 Performance comparison of the proposed backbone network with PSPNet [49], UNet [47], and FCN-8 [48] for recognizing the boundaries of the contraband items

Met	Data	Proposed	PSPNet	UNet	FCN-8
μIoU	GDX	0.4994	0.5585	0.4921	**0.5648**
	SIX	**0.7072**	0.5659	0.6700	0.6613
	OPI	**0.7393**	0.5645	0.7159	0.5543
μDC	GDX	0.6661	0.7167	0.6596	**0.7219**
	SIX	**0.8285**	0.7227	0.8024	0.7961
	OPI	**0.8501**	0.7217	0.8344	0.7132

The best and second-best performances are in bold and underline, respectively

Met Metric, *Data* dataset, *GDX* GDXray [39], *SIX* SIXray [13], *OPI* OPIXray [12]

Table 2 Performance comparison between state-of-the-art baggage threat detection frameworks on GDXray (GDX), SIXray (SIX), and OPIXray (OPI) dataset in terms of mAP scores

Data	Items	PF	CST	TST	FD
GDX	Gun	**0.9872**	0.9101	0.9761	–
	Razor	**0.9691**	0.8826	0.9453	–
	Shuriken	0.9735	**0.9917**	0.9847	–
	Knife	0.9820	**0.9945**	0.9632	–
	mAP	**0.9779**	0.9281	0.9672	–
SIX	Gun	0.9863	**0.9911**	0.9734	–
	Knife	**0.9811**	0.9347	0.9681	–
	Wrench	0.9882	**0.9915**	0.9421	–
	Scissor	0.9341	**0.9938**	0.9348	–
	Pliers	0.9619	0.9267	**0.9573**	–
	Hammer	0.9172	0.9189	**0.9342**	–
	mAP	**0.9614**	0.9595	0.9516	–
OPI	Folding	0.8528	–	0.8024	**0.8671**
	Straight	**0.7649**	–	0.5613	0.6858
	Scissor	0.8803	–	0.8934	**0.9023**
	Multi	**0.8941**	–	0.7802	0.8767
	Utility	**0.8062**	–	0.7289	0.7884
	mAP	**0.8396**	–	0.7532	0.8241

'–' The respective score is not computed

Data Dataset, *GDX* GDXray [39], *SIX* SIXray [13], *OPI* OPIXray [12], *PF* proposed framework, *FD* FCOS [50] + DOAM [12]

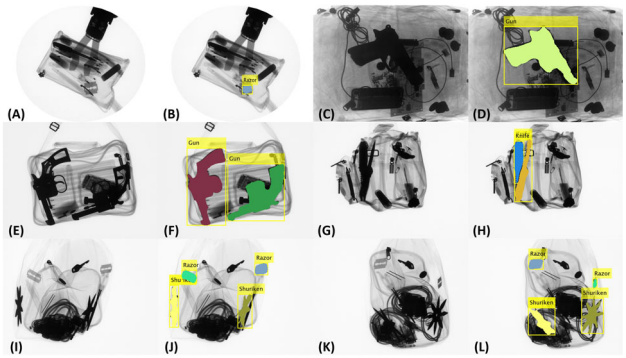

Fig. 7 Qualitative evaluations of the proposed framework on GDXray [39] dataset. Please zoom-in for best visualization

contraband items. For example, see the extraction of merged *knife* instances in (H), and the cluttered *shuriken* in (J, L). We can also appreciate how accurately the *razors* have been extracted in (J, L). Extracting such low contrast objects in the competitive CST framework requires suppressing first all the sharp transitions in an iterative fashion [16].

5.3 Evaluations on SIXray dataset

The proposed framework has been evaluated on the whole SIXray dataset [13] (containing 1,050,302 negative scans and 8,929 positive scans) and also on each of its subsets [13]. In Table 2, we can observe that the proposed framework achieves an overall performance gain of 0.190% and 0.980% over CST [16] and TST [17] framework, respectively. In Table 3, we report the results obtained with each subset of the SIXray dataset [13], reflecting different imbalanced normal and prohibited item categories. The results further confirm the superiority of the proposed framework against other state-of-the-art solutions, especially w.r.t the CHR [13], and [15]. In addition to this, in an extremely challenging SIXray1000 subset, we notice that the proposed framework leads the second-best TST framework [17] by 3.22%, and CHR [13] by 44.36%.

Apart from this, Fig. 8 depicts the qualitative evaluations of the proposed framework on the SIXray [13] dataset. In this figure, the first row shows examples containing one instance of the suspicious item, whereas the second and third rows show scans containing two or more instances of the suspicious items. Here, we can appreciate how accurately the proposed scheme has picked the cluttered *knife* in (B). Moreover, we can also observe the extracted *chopper* (*knife*) in (D) despite having similar contrast with the background. More examples such as F, H, and J demonstrate the proposed framework's capacity in picking the cluttered items from the SIXray dataset [13].

5.4 Evaluations on OPIXray dataset

The performance evaluation of the proposed framework on OPIXray dataset [12] is reported in Table 2. We can observe here that the proposed system achieves an overall mAP score of 0.8396, outperforming the second-best DOAM framework [12] (driven via FCOS [50]) by 1.55%. Here, although the performance of both frameworks is

Table 3 Performance comparison of proposed framework with state-of-the-art solutions on SIXray subsets

Subset	PF	DTS	CHR	[15]	TST
SIX-10	**0.9793**	0.8053	0.7794	0.8600	0.9601
SIX-100	**0.8951**	0.6791	0.5787	–	0.8749
SIX-1k	**0.8136**	0.4527	0.3700	–	0.7814

For fair comparison, all models are evaluated using ResNet-50 [40] as a backbone

SIX-10 SIXray10 [13], *SIX-100* SIXray100 [13], *SIX-1k* SIXray1000 [13], *PF* proposed framework

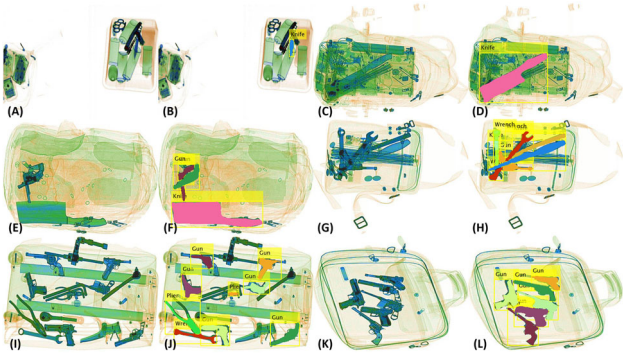

Fig. 8 Qualitative evaluations of the proposed framework on SIXray [13] dataset. Please zoom-in for a best visualization

Table 4 Performance comparison of proposed framework with DOAM [12] (backboned through SSD [51]) on different occlusion levels of OPIXray [12] dataset

Method	OP1	OP2	OP3
Proposed	**0.7946**	**0.7382**	**0.7291**
DOAM + SSD [12]	0.7787	0.7245	0.7078
SSD [51]	0.7545	0.6954	0.6630

identical, we still achieve a significant lead of 7.91% over the DOAM [12] for extracting the *straight knives*.

Concerning the level of occlusion (as aforementioned, OPIXray [12] splits the test data into three subsets, OP1, OP2, and OP3, according to the level of occlusion), we can see in Table 4 that the proposed framework achieves the best performance at each occlusion level as compared to the second-best DOAM [12] framework driven by the single-shot detector (SSD) [51].

Figure 9 reports some qualitative evaluation, where we can appreciate the recognition of the cluttered *scissor* (e.g., see B and F), and overlapping *straight knife* (in H). We can

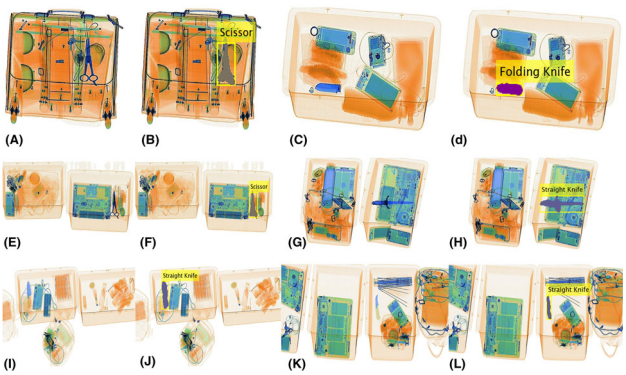

Fig. 9 Qualitative evaluations of the proposed framework on OPIXray [12] dataset. Please zoom-in for a best visualization

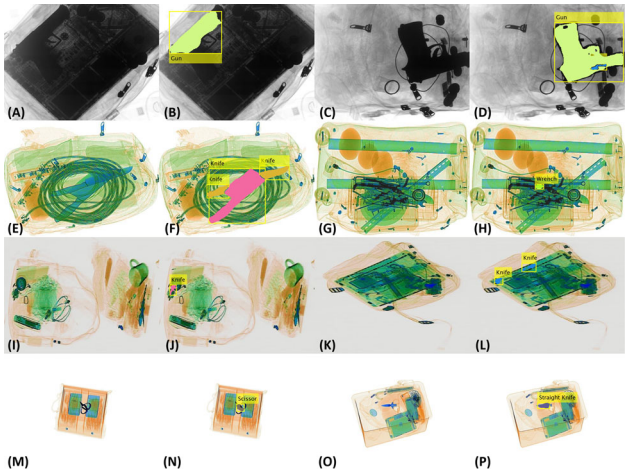

Fig. 10 Failure cases from GDXray [39], SIXray [13], and OPIXray [12] dataset

also notice the detection of the partially occluded *folding and straight knife* in (D) and (J).

5.5 Failure cases

In Fig. 10, we report examples of failure cases encountered during the testing. In cases (B, H, N, and P), we can see that the proposed framework could not pick-up the whole regions of the contraband items, even though the items were detected correctly. However, such cases are observed in highly occluded scans such as (A and G), where it is difficult, even for a human observer, to distinguish the items' regions properly. The second type of failure corresponds to the pixels misclassification as shown in (D) where some of the *gun*'s pixels have been misclassified as *knife*. We can address these scenarios through post-processing steps like blob removal and region filling. The third failure case relates to the proposed framework's inability to generate a single bounding box for the same item. Such a case is depicted in (F), where two bounding boxes were generated for the single orange *knife* item. One possible remedy here is to generate the bounding boxes based upon the minimum and maximum mask value in both image dimensions for each label. Another type of failure is shown in (J) and (L). Here, the scans contain only normal baggage content, but some pixels occupying tiny regions have been misclassified as false positive (i.e., *knife*). We can also address this kind of failure through blob removal scheme.

Examining the failure cases' statistical distributions, we found a majority of 86.09% cases belonging to the curable categories (i.e., second, third, and fourth), meaning that the proposed framework's performance can be further improved using the post-processing techniques mentioned above.

6 Conclusion

In this work, we proposed a novel contour-driven approach for detecting cluttered and occluded contraband items (and their instances) within the baggage X-ray scans, hypothesizing that contours are the most robust cues given the lack of texture in the X-ray imagery. We concretized this original approach through a tensor pooling module, producing multi-scale tensor maps highlighting the items' outlines within the X-ray scans and an instance segmentation model acting on this representation. We validated our approach on three publicly available datasets encompassing gray-level and colored scans and showcased its overall superiority over competitive frameworks in various aspects. For instance, the proposed framework outperforms the state-of-the-art methods [12, 16–18] by 1.07%, 0.190%, and 1.55% on GDXray [39], SIXray [13], and OPIXray [12] dataset, respectively. Furthermore, on each SIXray subsets (i.e., SIXray10, SIXray100, SIXray1000) [13], the proposed framework leads the state of the art by 1.92%, 2.02%, and 3.22%, respectively.

In future, we aim to apply the proposed framework to recognize 3D-printed contraband items from the X-ray scans. Such items exhibit poor visibility in the X-ray scans because of their organic material, making them an enticing and challenging case to investigate and address.

Author contributions TH formulated the idea, wrote the manuscript, and performed the experiments. SA improved the initial design of the framework and contributed to manuscript writing. MB co-supervised the whole research and reviewed the manuscript and experiments. SK reviewed the manuscript and experiments and improved the manuscript writing. NW supervised the whole research, contributed to manuscript writing, and reviewed the experimentation.

Funding information This work is supported by a research fund from ADEK (Grant Number: AARE19-156) and Khalifa University (Grant Number: CIRA-2019-047).

Availability of data and materials All the datasets that have been used in this article are publicly available.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to this article.

Financial and non-financial interests All the authors declare that they have no financial or non-financial interests to disclose for this article.

Employment The authors conducted this research during their employment in the following institutes: (1) T. Hassan (Khalifa University, UAE), (2) S. Akçay (Durham University, UK), (3) M. Bennamoun (The University of Western Australia, Australia), (4) S. Khan (Mohamed bin Zayed University of Artificial Intelligence, UAE), and (5) N. Werghi (Khalifa University, UAE).

Ethical approval All the authors declare that no prior ethical approval was required from their institutes to conduct this research.

Consent for participate and publication All the authors declare that no prior consent was needed to disseminate this article as there were no human (or animal) participants involved in this research.

Code availability The source code of the proposed framework is released publicly on GitHub[1].

References

1. Tang Z, Tian E, Wang Y, Wang L, Yang T (2020) Nondestructive defect detection in castings by using spatial attention bilinear convolutional neural network. IEEE Trans Ind Inf 17:82–89
2. Bastan M, Byeon W, Breuel T (2013) Object recognition in multi-view dual energy X-ray images. In: British machine vision conference
3. Turcsany D, Mouton A, Breckon TP (2013) Improving feature-based object recognition for X-ray baggage security screening using primed visual words. In: 2013 IEEE international conference on industrial technology (ICIT). IEEE, pp 1140–1145
4. Hu B, Zhang C, Wang L, Zhang Q, Liu Y (2020) Multi-label X-ray imagery classification via bottom-up attention and meta fusion. In: Asian conference on computer vision (ACCV)
5. Akçay S et al (2016) Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery. In: IEEE ICIP, pp 1057–1061
6. Akcay S et al (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
7. Gaus YFA et al (2019) Evaluation of a dual convolutional neural network architecture for object-wise anomaly detection in cluttered X-ray security imagery. In: 2019 international joint conference on neural networks (IJCNN), pp 1–8
8. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: IEEE international conference on computer vision (ICCV), pp 2961–2969
9. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell
10. Lin TY et al (2017) Focal Loss for Dense Object Detection. In: IEEE international conference on computer vision and pattern recognition (CVPR)
11. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement
12. Wei Y et al (2020) Occluded prohibited items detection: an X-ray security inspection benchmark and de-occlusion attention module
13. Miao C et al (2019) SIXray: a large-scale security inspection X-ray benchmark for prohibited item discovery in overlapping images. In: IEEE international conference on computer vision and pattern recognition (CVPR), pp 2119–2128
14. Akçay S, Breckon T (2020) Towards automatic threat detection: a survey of advances of deep learning within X-ray security imaging. Preprint arXiv:2001.01293
15. Gaus YFA et al (2019) Evaluating the transferability and adversarial discrimination of convolutional neural networks for threat object detection and classification within X-ray security imagery. arXiv preprint arXiv:1911.08966
16. Hassan T et al (2020) Detecting prohibited items in X-ray images: a contour proposal learning approach. In: Accepted in 27th IEEE international conference on image processing (ICIP)

17. Hassan T, Werghi N (2020) Trainable structure tensors for autonomous baggage threat detection under extreme occlusion. In: Asian conference on computer vision (ACCV), September
18. Hassan T, Shafay M, Akçay S, Khan S, Bennamoun M, Damiani E, Werghi N (2020) Meta-transfer learning driven tensor-shot detector for the autonomous localization and recognition of concealed baggage threats. In: MDPI sensors, November
19. Mery D, Svec E, Arias M (2016) Object recognition in baggage inspection using adaptive sparse representations of X-ray images. In: Pacific-Rim Symposium on image and video technology, pp 709–720
20. Bastan M et al (2013) Object recognition in multi-view dual energy X-ray images. In: BMVC, vol 1, p 11
21. Akçay S, Atapour-Abarghouei A, Breckon TP (2019) Skip-GANomaly: skip connected and adversarially trained encoder-decoder anomaly detection. In: International joint conference on neural networks (IJCNN)
22. 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
23. Heitz G, Chechik G (2010) Object separation in X-ray image sets. In: IEEE international conference on computer vision and pattern recognition (CVPR), pp 2093–2100
24. Bastan M (2015) Multi-view object detection in dual-energy X-ray images. Mach Vis Appl 25:1045–1060
25. Kundegorski ME et al (2016) On using feature descriptors as visual words for object detection within X-ray baggage security screening. In: IEEE international conference on imaging for crime detection and prevention (ICDP)
26. Bastan M, Yousefi MR, Breuel TM (2011) Visual words on baggage X-ray images. In: 14th international conference on computer analysis of images and patterns
27. 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
28. Liu Z, Li J, Shu Y, Zhang D (2018) Detection and recognition of security detection object based on YOLO9000. In: 2018 5th international conference on systems and informatics (ICSAI). IEEE, pp 278–282
29. Xu M et al (2018) Prohibited item detection in airport X-ray security images via attention mechanism based CNN. In: Chinese conference on pattern recognition and computer vision, pp 429–439
30. Jaccard N et al (2017) Detection of concealed cars in complex cargo X-ray imagery using deep learning. J X-ray Sci Technol 25:323–339
31. Griffin LD, Caldwell M, Andrews JTA, Bohler H (2019) Unexpected item in the bagging area: anomaly detection in X-ray security images. IEEE Trans Inf Forensics Secur 14:1539–1553
32. Zou L, Yusuke T, Hitoshi I (2018) Dangerous objects detection of X-ray images using convolution neural network. In: Security with intelligent computing and big-data services
33. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE international conference on computer vision and pattern recognition (CVPR)
34. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: IEEE international conference on computer vision and pattern recognition (CVPR)
35. Szegedy C et al (2015) Going deeper with convolutions. In: IEEE international conference on computer vision and pattern recognition (CVPR)
36. An J, Zhang H, Zhu Y, Yang J (2019) Semantic segmentation for prohibited items in baggage inspection. In: International conference on intelligence science and big data engineering. Visual data engineering, pp 495–505
37. Xiao H et al (2018) R-PCNN method to rapidly detect objects on THz images in human body security checks In: IEEE martworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications. Cloud & big data computing, internet of people and smart city innovation
38. Dhiraj KD (2019) An evaluation of deep learning based object detection strategies for threat object detection in aggage security imagery. Pattern Recognit Lett 120:112–119
39. Mery D et al (2015) GDXray: the database of X-ray images for nondestructive testing. J Nondestruct Eval 34(4):42
40. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE international conference on computer vision and pattern recognition (CVPR)
41. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
42. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv preprint arXiv:1602.07360
43. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387
44. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems
45. Akçay S, Atapour-Abarghouei A, Breckon TP (2018) GANomaly: semi-supervised anomaly detection via adversarial training. In: Asian conference on computer vision. Springer, pp 622–637
46. Zeiler MD (2012) ADADELTA: an adaptive learning rate method. arXiv:1212.5701
47. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. arXiv:1505.04597
48. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE international conference on computer vision and pattern recognition (CVPR), pp 3431–3440
49. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: IEEE international conference on computer vision and pattern recognition (CVPR), pp 2881–2890
50. Tian Z, Shen C, Chen H, He T (2019) FCOS: fully convolutional one-stage object detection. In: IEEE international conference on computer vision (CVPR)
51. Liu W et al (2016) SSD: single shot multibox detector. In: European conference on computer vision