Pattern Recognition 153 (2024) 110493

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Incremental convolutional transformer for baggage threat detection

Taimur Hassan [a,*], Bilal Hassan [b], Muhammad Owais [b], Divya Velayudhan [b], Jorge Dias [b], Mohammed Ghazal [a], Naoufel Werghi [b]

[a] Department of Electrical, Computer and Biomedical Engineering, Abu Dhabi University, United Arab Emirates
[b] KUCARS and C2PS, Department of Electrical Engineering and Computer Science, Khalifa University, United Arab Emirates

ARTICLE INFO

Keywords:
Threat detection
Transformers
Knowledge distillation
Incremental learning
Incremental instance segmentation
X-ray imagery
Catastrophic forgetting

ABSTRACT

Detecting cluttered and overlapping contraband items from baggage scans is one of the most challenging tasks, even for human experts. Recently, considerable literature has grown up around the theme of deep learning-based X-ray screening for localizing contraband data. However, the existing threat detection systems are still vulnerable to high occlusion, clutter, and concealment. Furthermore, they require exhaustive training routines on large-scale and well-annotated data in order to produce accurate results. To overcome the above-mentioned limitations, this paper presents a novel convolutional transformer system that recognizes different overlapping instances of prohibited objects in complex baggage X-ray scans via a distillation-driven incremental instance segmentation scheme. Furthermore, unlike its competitors, the proposed framework allows an incremental integration of new item instances while avoiding costly training routines. In addition to this, the proposed framework also outperforms state-of-the-art approaches by achieving a mean average precision score of 0.7896, 0.5974, and 0.7569 on publicly available GDXray, SIXray, and OPIXray datasets for detecting concealed and cluttered baggage threats.

1. Introduction

Recognizing concealed baggage threats is of the utmost importance for transportation security. X-ray scanners are widely used for this purpose at airports, malls, and cargoes to detect illegal activities in order to prevent loss of lives. However, the increased volume of baggage makes the whole process hectic, time-consuming, and vulnerable to human errors. To address this, many researchers have presented automated solutions for detecting baggage threats [1]. Despite their noticeable advances, concealment, occlusion, and extreme clutter in the baggage scenes still present a significant challenge towards threat recognition [2]. Also, due to the inherent difference between the X-ray and the RGB scans, many popular occlusion-aware object detection schemes fail to recognize these threatening items [3–6]. Contrary to this, semantic segmentation, thanks to its ability to produce pixel-wise recognition, seems promising for effectively extracting the occluded objects [7–11] and robustly identifying the baggage threats [12–15]. However, semantic segmentation frameworks have an inherent limitation of differentiating the same object's cluttered instances. For example, see the extraction of *knife* on top of another *knife* in Fig. 1(B) as compared to Fig. 1(C).

1.1. Motivation

In surveillance and inspection environment, and more particularly in the context of baggage threat detection, instance semantic segmentation has several advantages over semantic segmentation, namely: (a) By "searching" for each instance separately, the risk to produce false detection by the autonomous systems is significantly reduced, which eventually reduces the rate of false negatives. The reduction in the false negatives is most critical for security applications, especially related to the aviation; (b) In passenger baggage, which often exhibit clutter, threat items are partially or fully occluded by other objects. Instance semantic segmentation can handle occlusion better than semantic segmentation by extracting each instance individually, meaning that even if a suspicious instance is partially occluded, instance segmentation can still identify and delineate it accurately; (c) Instance semantic segmentation infers learning detailed and fine-grained information about individual objects and suspicious items in particular. In inspection, this level of detail is crucial for identifying specific attributes or characteristics of suspicious instances, for example, the blade of a *knife*, the trigger of a *pistol*, etc. Instance segmentation allows the best capturing of these fine details and their underlying interaction, thus identifying potential threats that may not be discernible with

* Corresponding author.
E-mail address: taimur.hassan@adu.ac.ae (T. Hassan).

https://doi.org/10.1016/j.patcog.2024.110493
Received 20 October 2023; Received in revised form 5 February 2024; Accepted 7 April 2024
Available online 10 April 2024

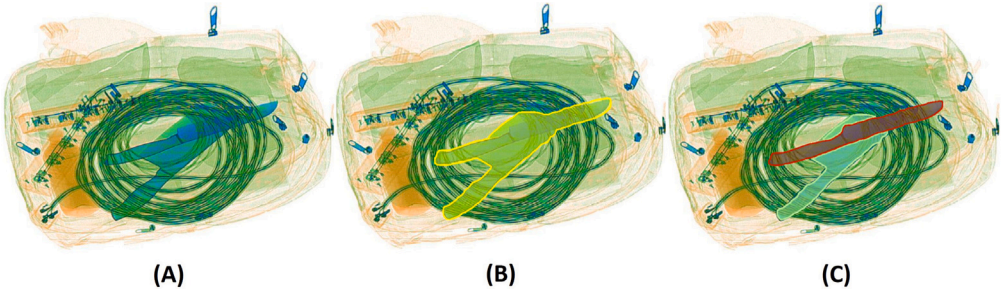

Fig. 1. Exemplar case: (A) original image, (B) extracted suspicious items through semantic segmentation, and (C) extracted items through the proposed instance segmentation approach.

semantic segmentation alone. On another side, autonomous baggage threat detection is a lifelong process where new types of suspicious items or variants of the already learned threat objects (e.g., new *knife* or *pistol* models) can emerge anytime. Retraining the conventional models from scratch in such situations is an unpractical and resource-demanding solution, which limits their applicability in the real world for detecting baggage threats. Also, in semantic segmentation, the average time for manually annotating a single image is in the order of minutes. Therefore, to overcome the limitations discussed above, there is a need to develop models that can incrementally learn new instances of the threat objects using the new training examples without forgetting its prior learned knowledge. For this purpose, we present (i) a novel incremental convolutional transformer, driven via a distillation-based incremental instance segmentation scheme to detect the prohibited item instances individually, even in extreme occlusion. (ii) We propose a new approach to incrementally learn new instances of previously learned contraband items (not the new contraband items), i.e., in each increment, we add a new instance of a given contraband item where each instance is considered a separate class. (iii) The proposed scheme incurs significantly lesser amount of training cost, as compared to the conventional semantic or instance segmentation models, to recognize cluttered and overlapping instances of the contraband items.

1.2. Contributions

The contribution of this work is five-fold:

- We present a first attempt towards designing an incremental convolutional transformer model driven via distillation-based incremental instance segmentation scheme designed explicitly for baggage threat recognition tasks.
- The proposed incremental framework allows accommodating new instances of the contraband items (e.g., new types of a *gun*) without incurring costly full training from scratch, and/or conventional fine-tuning processes.
- The proposed scheme offers significantly higher computational efficiency over state-of-the-art methods during the training phase.
- Unlike other instance segmentation approaches [16,17], the proposed framework's incremental nature eliminates the need for an additional object detector and classification backbone.
- We validate the proposed framework on three public datasets evidencing a best trade-off between computational efficiency and accuracy compared to state-of-the-art threat detection models at the inference (test) phase.

The rest of the paper is organized as follows: Section 2 overviews related works. Section 3 presents the proposed approach. Section 4 describes the datasets and the experimentation protocols. Section 5 presents the evaluation results. Section 6 discusses the proposed framework in detail. Section 7 concludes the paper.

2. Related work

Many researchers have developed autonomous systems for recognizing baggage threats via X-ray imagery. In the subsequent sections, we present an overview of these approaches and also shed light on some of the recent advancements in incremental learning.

2.1. Baggage threat detection schemes

We broadly categorized the baggage threat detection literature into two groups, i.e., machine and deep learning. In both of these groups, we have thoroughly discussed the pioneer works that are designed for recognizing baggage threats using X-ray imagery. But for a detailed survey, we refer the reader to the work of [1,18].

2.2. Machine learning methods

Earlier systems for classifying and detecting baggage threats involved traditional machine learning schemes. The majority of the classification methods utilized feature descriptors such as SURF [19], and SIFT [20], in conjunction with models like Random Forest [21], and K-NN [20]. Bastan et al. [19] leveraged multi-view X-ray imagery for detecting prohibited items based upon SIFT and SURF features coupled with SVM. Apart from this, Mery et al. [20] used structure from motion to generate 3D feature points and detected contraband items through K-NN classifier. The threat detection methods, built using traditional machine learning models, performed satisfactorily under constrained experimental settings. However, when exposed to a generic set of X-ray imagery obtained from different scanners, the performance of such methods deteriorated as they were built using subjective or handcrafted features [22].

2.3. Deep learning methods

To overcome the limitations of traditional methods, researchers employed deep learning for recognizing baggage threats using X-ray scans. The deep learning-based threat detection methods are mainly categorized as classification [22], detection [23], or segmentation [24,25] approaches. Moreover, several unsupervised learning methods have also been proposed which treat baggage threats as anomalies [26]. Nevertheless, the major focus of the previous works lies towards supervised learning, where researchers have used sequential networks as a feature extractor coupled with SVM for object classification [22]. Miao et al. [23] worked on the imbalanced set of prohibited items by proposing a class-balanced hierarchical refinement (CHR) scheme that accurately recognizes and localizes baggage threats from the SIXray [23] dataset. Wei et al. [27] presented a De-occlusion Attention Module (DOAM) integrated with one-staged and two-staged object detectors to identify and localize occluded baggage threats. In addition to this, Hassan et al. presented contour-driven object detectors [28] and instance segmentation frameworks [29] for recognizing baggage threats depicted within grayscale and colored baggage X-ray scans.

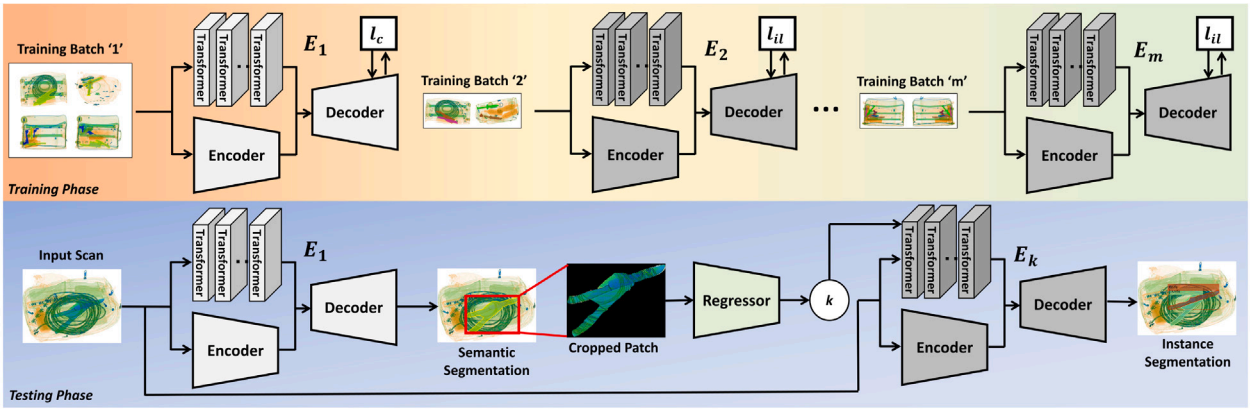

Fig. 2. The overview of the proposed system. In the top sub-figure, model E_1 is trained for semantic segmentation to extract different isolated and overlapping threat objects (e.g., a *knife* overlaid on the *gun*). To differentiate between overlapped instances of the same item (e.g., a *knife* on top of another *knife*), we incrementally update E_1 in each iteration by adding new item instance classes (given a small set of training examples) such that a model trained in jth iteration (E_j) can recognize up to 'j' instances of each item. In the testing phase reported in the bottom sub-figure, the input scan is first fed to the trained E_1 model to obtain masks of the different detected items (whether they are isolated or overlapping). These detected instances (produced by E_1) are passed to the regressor model $\mathcal{R}$ which determines the maximum number of overlapped instances k in the scan. This number, k, is then used to select the appropriate model (E_k) to perform the required k-instance segmentation task.

The deep learning methods can give good performance when exposed to a generic set of X-ray imagery for detecting contraband objects. However, these methods are vulnerable to occlusion and require extensive re-training or fine-tuning routines in order to learn new emerging instances of contraband items [22,28,30–33]. The proposed framework can overcome these limitations due to its incremental nature and its ability to provide an optimal trade-off between detection performance and computational efficiency. Furthermore, the proposed framework can accurately extract occluded and cluttered instances of the contraband objects from the X-ray scans irrespective of their spatial characteristics (please see Section 5 for more details).

3. Proposed framework

The incremental aspect of the proposed approach (shown in Fig. 2) can be summarized as follows: Initially, we employ a trained system that can identify various threat items by performing semantic segmentation, with each threat item represented by a single instance. These threat items may appear isolated or overlapping in the X-ray image. Subsequently, new instances, or variations, of each threat item emerges. To accommodate these changes, the system needs to be updated through incremental training to recognize the distinct instances of threat items, even when they overlap. During this phase, the maximum number of overlapping instances of the same item is two. After some time, another set of new threat instances emerges, and the maximum number of instances per threat increases to three. To handle this expansion, the system undergoes the same incremental training process as before, enabling it to semantically segment up to three instances of the same threat items. This incremental learning process continues as long as new threat instances arise. It should be noted that the proposed system can detect any number of overlapping instances of the contraband items specified either by the number of users or within the dataset specifications. However, the proposed system learns this task incrementally. For example, in the first training iteration, it learns to detect a single instance (or multiple non-overlapping instances) of the contraband items. In the second iteration, the system learns to detect up to two overlapping instances of the contraband items. For the third training iteration, the system learns to segment up to three overlapping instances of the contraband items. The total number of training iterations here are equal to the maximum number of overlapping instances of the threat items specified within the dataset or by the users.

Moreover, to perform the instance segmentation, we propose a novel convolutional transformer model, as shown in Fig. 3. Architecturally, the proposed convolutional transformer consists of an encoder, a transformer, and a decoder block. The reason for fusing transformers with the convolutional blocks is to boost the separation of background content and threat object instances by combining the attentional projections generated by the transformer block with the latent feature representations generated by the encoder end. These fused features are then passed to the decoder end to segment the cluttered contraband instances.

3.1. System overview

The proposed system is shown in Fig. 2. During the training phase, we extend the proposed convolutional transformer architecture to perform instance-aware segmentation by incrementally adding the class of each contraband item instance. In the first iteration, the model dubbed E_1 (isolated objects) performs semantic segmentation to extract isolated and merged contraband items. In the second training iteration, the model E_2 (2 overlapped instances) extends E_1 to recognize at max two cluttered instances of the same object (e.g., a *gun* overlaid on another *gun*). This process is repeated till the mth iteration, thus generating model E_m (m overlapped instances) where m represents the maximum overlapped instances of any threat object contained within the given dataset. To create a threat detection system, we can incorporate a wide range of threat items. However, the current datasets contain only a limited number of categories. Therefore, we determined the hyper-parameter m based on the maximum number of overlapped items in a given dataset. This approach allows the proposed system to be scalable and requires minimal computational power, without having to retrain the entire system from scratch. In addition to this, we also train a regression model $\mathcal{R}$ to recognize the maximum overlapped instances k (within the candidate test scan), and hence select the appropriate model instance E_k for k-instance segmentation. We want to highlight here that the inclusion of $\mathcal{R}$ within the proposed framework is optional, and it can be excluded where the model instance E_m, for the given dataset, can be utilized to perform the incremental instance segmentation. But performing k-instance segmentation with E_m, where $k < m$ can lead to over-segmentation results, and this can be avoided by selecting the appropriate model instance E_k by the regressor ($\mathcal{R}$).

At the testing phase, we pass the candidate X-ray scan to E_1, which performs an initial semantic segmentation. This initial segmentation recognizes overlapped instances of the same item as a single connected blob (e.g., two overlapped instances of the *knife*, see Fig. 1-B). Afterward, we perform connected component analysis (CCA) [34] on the output masks (only in this first iteration) to identify isolated

regions. These regions contain either a single item instance or overlapped instances of the same item. Each region is then cropped into a bounding box (from the original scan), and this cropped patch is passed to $\mathcal{R}$, which has been trained to identify the number of overlapped instances in it. Afterward, the maximum number of detected instances (dubbed k) across all the patches within the scan is used to select the appropriate model instance E_k for the instance-aware segmentation. Note that all the proposed convolutional transformer instances have the same architecture as E_1 (except the last layer), which is updated in each iteration j to recognize up to j overlapping item instances. To summarize, in each iteration j where $j \geq 2$, we perform: (a) update the last layer of E_j to recognize j overlapping instances, and (b) fine-tune the weights of E_j inherited from E_{j-1} such that E_j recognizes new item instance classes without catastrophically forgetting its prior knowledge (learned in iterations 1 to $j-1$). It should also be noted that the selected E_k model does not take a single patch for k-instance segmentation. Rather, it takes a complete test scan for recognizing contraband items regardless of the clutter and concealment (see the testing phase in Fig. 2). For example, if a test scan contains two overlapping *knives* and three overlapping *guns*. The regressor $\mathcal{R}$ would select E_3 for instance-aware segmentation, which can also extract the two overlapping *knives*, and the false positives (if any) are easily removed with morphological post-processing since the proposed model is based on pixel-wise recognition. Moreover, we again emphasize that $\mathcal{R}$ is modular in the proposed framework. But if we exclude $\mathcal{R}$ from the proposed framework and choose only the E_m model for instance-aware segmentation. The system will then produce more false positives, because of the over-segmentation, compared to the scenario of selecting the appropriate model E_k by $\mathcal{R}$.

In general, incremental learning allows a model to learn from new data incrementally over time, without having to retrain the entire model from scratch. While using only the last trained model can save storage space and reduce computational costs, it does have some limitations. For example, if the last model is trained on a small or biased sample of data, it may not generalize well to new or diverse data in the future. Additionally, if the model encounters a catastrophic forgetting problem, where it forgets previously learned information when learning new information, relying on only the last model can lead to significant performance degradation. However, it is true that storing and loading multiple models can be computationally expensive and inefficient. Therefore, it is important to balance efficiency and effectiveness in incremental learning according to the domain application of a deep learning model.

3.2. Incremental training strategy

In this section, we present a detailed discussion on the incremental training of the proposed convolutional transformer network. In the first increment (or iteration), we train the model in a conventional manner to perform semantic segmentation for extracting different contraband items. However, in the subsequent increments, we constrain the network to recognize overlapping instances of the contraband data by stacking their instance classes in the final network layer. A detailed description of the incremental training is presented below:

3.2.1. First increment

The first training increment relates to the semantic segmentation, where the model instance E_1, after training, recognizes the concealed contraband items from the given test scans. In the first training increment, E_1 contains ∼31.5M parameters from which ∼61K are non-trainable, and ∼31.4M are trainable. The number of channels in each layer and the kernel sizes can be seen in the detailed summary report within the source code repository on GitHub.[1]

Apart from this, E_1 is constrained using a cross-entropy objective function, as expressed below:

$$l_c = -\frac{1}{b_s}\sum_{u=0}^{b_s-1}\sum_{v=0}^{c-1} t_{u,v}\log(p(l_{u,v})), \tag{1}$$

where c is the total number of classes, b_s is the batch size, $t_{u,v}$ tells whether or not the uth sample (pixel) corresponds to the vth class, and $p(l_{u,v})$ represents the softmax probability of the uth sample belonging to the vth category. Since E_1 is a typical network for semantic segmentation, it cannot differentiate between multiple cluttered instances of the same object (e.g., a *knife* on top of another *knife*) within the candidate scan. Also, it cannot differentiate between isolated item instances (e.g., two isolated *guns* within a single scan). However, such isolated instances are separated in post-processing using the CCA technique as previously reported.

3.2.2. Subsequent increments

To make the convolutional transformer instance-aware, we use a portion of training examples (containing overlapping item instances). The exact number of training examples used in each increment across each dataset is reported in Table 1. From this table, we can notice that with a significantly lesser amount of data supervision as compared to the dataset standards, the proposed framework can achieve competitive segmentation performance across different datasets, as it will be confirmed in the results.

In each increment j, $j \geq 2$, we derive the model instance E_j from E_{j-1}, so that E_j has the same architecture as E_{j-1} but with the exception in the last layer where we add j classes for each instance of the item. Here, we use the training batch j to update the weights of E_j to recognize up to j item instances. The total number of learnable parameters of each model instance will remain approximately same while the number of instance classes increases for each item instance during each increment j, $j \geq 2$. Moreover, to avoid catastrophic forgetting, we constrain the network E_j to retain its prior knowledge while differentiating the j overlapping item instance information through proposed incremental learning objective function l_{il} (for more details, see Section 3.4). The number of training increments depends on the maximum number of overlapping items contained within the dataset scans. During the testing phase, the regressor $\mathcal{R}$ outputs k, indicating the contraband items' maximum overlapping instances within the candidate scans. Afterward, the appropriate model instance E_k for instance-aware segmentation is selected. Note that the output of each model E_j for $j \geq 1$ is an indexed-mask in which each contraband item class (and its instances) are assigned a unique class number. For example, suppose we have a test scan containing three overlapping *guns*. In that case, the $\mathcal{R}$ will select E_3 for instance-aware segmentation where each extracted instance of *gun* will have its unique class number.

3.3. Proposed convolutional transformer architecture

The detailed architectural overview of the proposed convolutional transformer E is shown in Fig. 3. We can see here that the model is comprised of three blocks, i.e., the encoder, transformer, and decoder. The input image is first passed to the encoder block that produces the latent features to identify different instances of the contraband objects. Also, the scan is decomposed into non-overlapping sequenced patches from which the latent as well as the flattened projections are generated. The flattened projections are obtained via positional embeddings of the sequenced patches, whereas the latent projections are yielded using linear embeddings. These projections are then added together and are passed to the t-layered transformer, which computes the contextual multi-head self-attention distributions. These feature distributions are convolved with the latent space representations of the encoder block, and the resultant distribution is passed to the decoder end, which extracts the prohibited items through rescaling and un-pooling blocks. The detailed description of each block within E is presented in the subsequent sections:

[1] Link to source code and demo videos: https://github.com/taimurhassan/convolutionalTransformer.

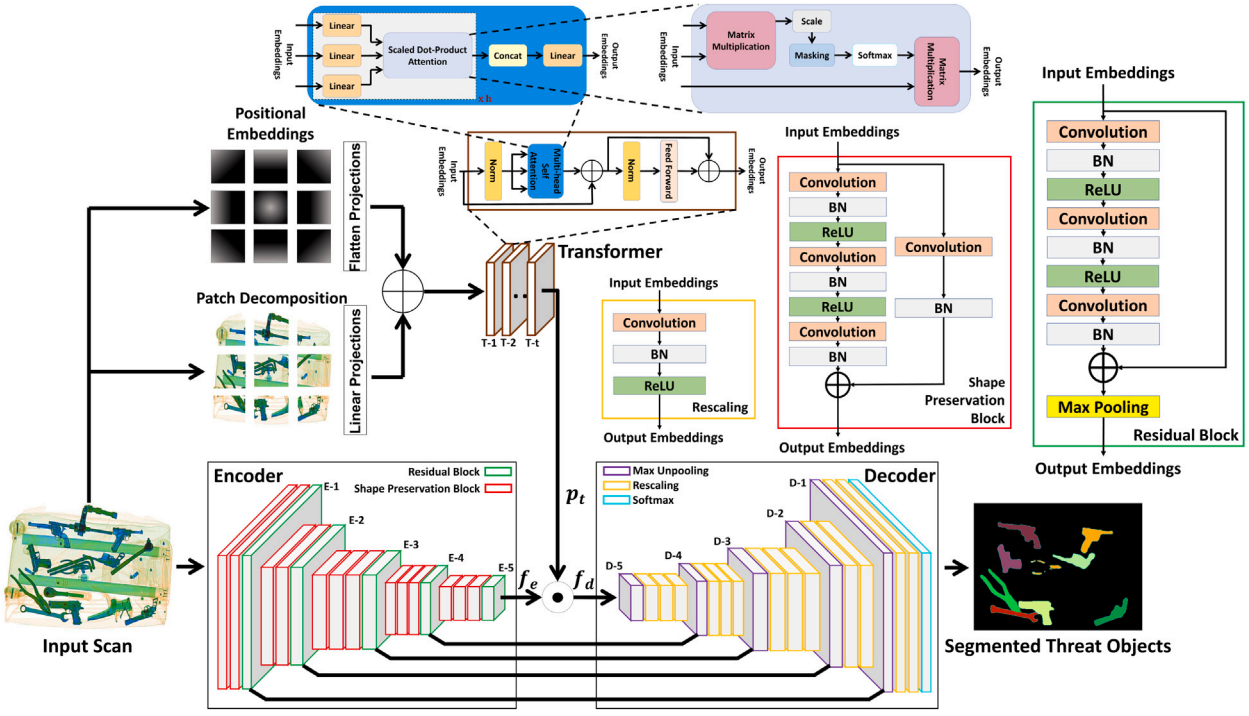

Fig. 3. Detailed architecture of the proposed convolutional transformer. The proposed model is comprised of three blocks, i.e., the encoder, transformer, and decoder. The input image is first passed to the encoder block that produces the latent features to identify different instances of the contraband objects. Also, the scan is decomposed into non-overlapping sequenced patches from which the latent as well as the flattened projections are generated. The flattened projections are obtained via positional embeddings of the sequenced patches, whereas the latent projections are yielded using linear embeddings. These projections are then added together and are passed to the t-layered transformer, which computes the contextual multi-head self-attention distributions. These feature distributions are convolved with the latent space representations of the encoder block, and the resultant distribution is passed to the decoder end, which extracts the prohibited items through rescaling and un-pooling blocks.

3.3.1. Encoder block

The encoder block within E is responsible for generating the distribution of the latent features $f_e(x)$ from the baggage X-ray scans $x \in \mathbb{R}^{R\times C\times C_h}$, where R represents the rows, C corresponds to columns, and C_h represents the channels. Unlike the conventional models, the encoder within E consists of five levels (E-1 to E-5), where each level contains three to four shape preservation and residual blocks. These blocks allow the encoder to generate accurate features for preserving the contextual and semantic information about the prohibited items during the scan decomposition. Moreover, there are 13 shape preservation blocks (SPBs) and five residual blocks (RBs) in the encoder. SPB is composed of four convolutions, four batch normalization (BN), and two ReLUs. RB contains three convolutions, three BNs, two ReLUs, and one max pooling. The encoder features f_e, obtained after optimizing encoder weights, effectively discriminate the contraband data from the rest of the baggage content. However, it also results in some false positives towards distinguishing occluded threat object regions [29]. To overcome this issue, we enforce the separation of inter-class distributions by merging f_e with the transformer projections p_t via convolution, which yield the fused feature representations $f_d = f_e * p_t$ in which the similar features between f_e and p_t are amplified, and the heterogeneous representations are suppressed, ensuring a significant reduction in the number of false positives. These combined features are forwarded to the decoder end to segment the contraband data.

3.3.2. Transformer block

The transformer block, within the proposed model, consists of t encoders, where t is empirically determined to be 3, i.e., $t = 3$, yielding, T-1, T-2, and T-3 encoders. These encoders are joined together in a sequential fashion to produce p_t. The prime objective of using transformers within the proposed system is to boost the detection performance using attention mechanism. For this purpose, we decompose the baggage X-ray image x into non-overlapping squared patches $x^p \in R^{P\times P\times C_h}$, where P represents the resolution of x^p, such that $P = \sqrt{\frac{RC_h}{n_p}}$, and n_p corresponds to the number of patches. Afterward, we generate the positional embeddings x_i^e corresponding to patch x_i^p, i.e., $x^e \in R^{P\times P\times C_h}$, from which the flatten projections are computed, i.e., $f_p(x_i^e)$. Similarly, we obtain the linear projection for x_i^p, i.e., $l_t(x_i^p)$, and then we remap $f_p(x_i^e)$ and $l_t(x_i^p)$ to l dimensions and generate the sequenced embeddings (for x_i^p) by adding $l_t(x_i^p)$ with $f_p(x_i^e)$, i.e., $q_i = l_t(x_i^p) + f_p(x_i^e)$. Repeating the same process for all n_p patches produces combined projections q^o, as written below:

$$q^o = [l_t(x_0^p); l_t(x_1^p); \ldots; l_t(x_{n_p-1}^p)] + [f_p(x_0^e); f_p(x_1^e); \ldots; f_p(x_{n_p-1}^e)], \tag{2}$$

or

$$q^o = [q_0; q_1; \ldots; q_{n_p-1}]. \tag{3}$$

The combined projections q^o is passed to T-1 where, at head j, q_j^o is normalized to produced $q_j^{'o}$. Afterward, $q_j^{'o}$ is linearly decomposed into query (Q_j), key (K_j), and value (V_j) pairs via learnable weights, such that, $Q_j = q_j^{'o} w_q$, $K = q_j^{'o} w_k$, and $V = q_j^{'o} w_v$. To compute the contextual self-attention at head j, i.e., A_j, Q_j and K_j are combined together via scaled dot product, and their resultant scores are fused with V_j, as formulated below:

$$A_j(q_j^{'o}; Q_j, K_j, V_j) = \sigma\left(\frac{Q_j K_j^T}{\sqrt{l}}\right) V_j, \tag{4}$$

where $\sigma(.)$ denotes the softmax function. Moreover, the proposed contextual self-attention maps from multiple heads are concatenated together to produce contextual multi-head self-attention distribution $\varphi_{CMSA}(q^{'o})$, as expressed below.

$$\varphi_{CMSA}(q^{'o}) = [A_0(q_j^{'0}; Q_0, K_0, V_0); A_1(q_j^{'1}; Q_1, K_1, V_1); \ldots; A_{h-1} \times (q_j^{'h-1}; Q_{h-1}, K_{h-1}, V_{h-1})]. \tag{5}$$

Apart from this, $\varphi_{CMSA}(q'^{o})$ is added with q^{o}, where the resultant embeddings are normalized and are passed to the normalized feedforward block to produce T-1 latent projections (p_{T1}):

$$p_{T1} = \phi_f((\varphi_{CMSA}(q'^{o}) + q^{o})') + (\varphi_{CMSA}(q'^{o}) + q^{o})', \tag{6}$$

where $\phi_f(.)$ denotes the learnable feed-forward function. After computing p_{T1}, it is passed to T-2 which produces p_{T2} in a similar manner, and p_{T2} is passed to T-3 encoder which produces p_{T3} projections, where $p_t = p_{T3}$. p_t is fused with f_e to generate f_d, and f_d is passed to the decoder block to extract the threat object instances.

3.3.3. Decoder block

After convolving f_e with p_t, a new set of fused features f_d is obtained, which is passed to the decoder. The decoder block consists of five levels where each level contains one max unpooling and two to three rescaling blocks. There are also 13 rescaling blocks in the decoder where each block contains the convolution, batch normalization, and the ReLU activation layer. Moreover, the skip-connections are established between encoder and decoder block to overcome the degradation problem of the model during threat objects segmentation. Also, at the head of the decoder lies the softmax layer that classifies each pixel to one of the threat object instance classes.

3.4. Proposed incremental loss functions

To train the proposed model E_j for $j \geq 2$ to incrementally recognize multiple instances of the contraband items, we designed a loss function that is based on knowledge distillation [32]. The loss function is expressed below:

$$l_{il} = l_{old} + l_{new}, \tag{7}$$

where

$$l_{old} = -\frac{\beta}{b_s} \sum_{u=0}^{b_s-1} \sum_{v=0}^{c_o-1} t^{o}_{u,v} \log(p(l^{o}_{u,v})), \tag{8}$$

and

$$l_{new} = \frac{1}{b_s} \sum_{u=0}^{b_s-1} \sum_{v=0}^{c_n-1} q(t^{n}_{u,v}) \log\left(\frac{q(t^{n}_{u,v})}{p(l^{n}_{u,v})}\right), \tag{9}$$

where c_o and c_n denote the number of previously learned and newly added suspicious items' instance classes, b_s is the pixel count, respectively. The rationale for devising l_{il} is to overcome catastrophic forgetting during incremental training where l_{old} ensures that the network E_j (in jth iteration) remember its existing knowledge (about the previously learned instances) through distilled representations, and l_{new} ensures that the network E_j learns the newly added j overlapping suspicious item instances from their corresponding training examples (in the jth incremental iteration). Compared to similar methods adopting knowledge distillation schemes, like [33], we introduce β as a hyper-parameter to control the network remembrance ability during incremental training. The higher value of β produces high resistance (of the network E_j) to catastrophic forgetting. However, it also makes it biased towards retaining old knowledge while paying less attention to the newly fed classes (in jth iteration). Similarly, a lower value of β removes this bias but at the expense of a compromise towards retaining its prior knowledge. Through extensive ablative analysis, we determined the optimal value of β that gives the best incremental instance segmentation performance (at the inference stage) without compromising the network's previously learned knowledge. More on this is presented in Section 5.1.1. Moreover, when $l_{old} = 0$ (i.e., if l_{old} is eliminated in Eq. (7)), we get a conventional transfer-learning based instance segmentation model.

Apart from this, $p(l^{o}_{u,v})$ denotes the predicted softmax probability of the logit ($l^{o}_{u,v}$) generated from the training examples of the old instances (learned in iterations 1 to $j-1$), $q(t^{n}_{u,v})$ denotes the probability of the one-hot ground truth label of the uth training sample belonging to the vth class added in the current jth iteration, $p(l^{n}_{u,v})$ represents the predicted softmax probability of the logit $l_{u,v}$ generated from the uth training example for the vth category (added in current iteration j).

For the regressor $\mathcal{R}$, we used the mean absolute error l_{mae} as expressed in Eq. (10):

$$l_{mae} = \frac{1}{b_{sz}} \sum_{u=0}^{b_{sz}-1} \left(|y_u - \hat{y}_u|\right), \tag{10}$$

where $\hat{y}_u$ and y_u denote the predicted and true values for the number of samples (instances of the contraband items within each scan patch), and b_{sz} represents the batch size.

Table 1
Number of scans used in the incremental training iterations versus the scans allocated by the dataset standard for conventional fine-tuning.

Dataset	Iteration	Training scans used	Dataset standard
GDXray [35]	1	210	400
	2	0	
SIXray [23]	1	42,030	113,959
	2	1250	
	3	1300	
	4	100	
	5	115	
	6	120	
SIXray10 [23]	1	32,156	78,575
	2	1500	
	3	1400	
	4	210	
	5	220	
	6	230	
SIXray100 [23]	1	102,030	721,463
	2	2000	
	3	1300	
	4	400	
	5	410	
	6	420	
SIXray1000 [23]	1	321,458	841,042
	2	3200	
	3	2800	
	4	650	
	5	700	
	6	750	
OPIXray [27]	1	3556	7109
	2	230	

4. Experimental setup

This section presents the datasets, the implementation details, and the evaluation metrics. It should be noted that to ensure fair comparison of the proposed framework with state-of-the-art methods across each dataset, we have used the same hyperparameters, training configurations, and the evaluation metrics in all the experimentation.

4.1. Datasets

The proposed system has been thoroughly evaluated on three public datasets, namely the GDXray [35], SIXray [23], and the OPIXray [27]. Due to space constraints, a detailed description of each dataset is presented within the supplementary material document submitted along with this manuscript.

4.2. Training and implementation details

As mentioned above, the training of the proposed model is performed incrementally where in the first increment, the model E_1 is trained to perform standard semantic segmentation to extract different

contraband items (e.g., a *gun* overlapped on the *knife*). In the consecutive training increments, the derived instance of E_1 is tuned to differentiate between different instances of the same contraband items (e.g., a *knife* overlapped on another *knife*) via l_{il} loss function.

Moreover, to perform the proposed incremental instance segmentation across each dataset, the convolutional transformer model requires a reduced portion of training examples as compared to the actual dataset standard. For example, the model E_j in jth increment ($j > 1$) requires approximately 46.95% lesser data (for each dataset) as per their standard defined for the conventional transfer learning or fine-tuning models (see Table 1). This is appreciable because providing large-scale and well-annotated data in order to train or fine-tune the conventional deep learning frameworks for aviation screening (in real-world) is an impracticable option, which limits their capacity for deployment [1]. Apart from this, the total training increments across each dataset vary as the maximum overlapping item instances, in each dataset, differs from one another.

We also highlight that in order to train the proposed framework on GDXray, OPIXray, and SIXray datasets, we produced instance-level pixel-wise labels for each scan within these datasets. However, these labels were marked in accordance with the box-level annotations which were originally marked in the dataset, i.e., only those items instances were marked pixel-wise, which already had box-level annotations marked by the dataset authors. Therefore, the distribution of the total number of instances contained within the scans of each dataset is same as the one which is mentioned in their original papers. Moreover, these annotations were marked using the MATLAB Image Labeler application and will be released publicly in the source code repository upon paper acceptance.

On the GDXray and OPIXray datasets, we have at most two instances of suspicious items. For example, in GDXray, we have two instances of *guns*, two instances of *knives*, two instances of *shuriken*, etc. Similarly, in OPIXray, we have two instances of different *knife* categories, such as *straight knife*, *folding knife*, *multi-tool knife*, *utility knife*, and *scissors*. In the SIXray dataset, we have 1,050,302 negative scans (which only had background '0' labels), and 8929 positive scans (which have up to six instances of the contraband items, e.g., six instances of *guns*, etc.). Moreover, the name of the ground truth files was the same as the name of the original scans (with the same size and a PNG extension). Also, for the SIXray subsets, i.e., SIXray10, SIXray100, and SIXray1000, the corresponding labels were automatically chosen from the original dataset based on their names.

The proposed framework has been implemented using TensorFlow 2.2.0, Python 3.7.4 on a machine with Intel Core i7-9750H@2.6 GHz, 32 GB RAM, with NVIDIA RTX 2080 Max-Q GPU having cuDNN v7.5 and a CUDA Toolkit 10.1.243. To ensure reproducibility, the source code and demo videos will be released on GitHub[1]. We also released the summaries of model E, and the regressor $\mathcal{R}$ within the source code repository.

4.3. Evaluation metrics

To evaluate the proposed system, we have used the mean intersection-over-union (μIoU), mean dice coefficient (μDC), and a mean average precision (mAP) at the IoU threshold ≥ 0.5, and 0.75 denoted respectively as mAP_{50} and mAP_{75}. We also used the mAP score computed at $0.5 \leq IoU \leq 0.95$ in the step of 0.05, denoted as mAP_a, and the area under the curve (AUC) scores. To evaluate the $\mathcal{R}$'s performance, we have used the root mean square error (RMSE) and the Pearson correlation coefficient.

Table 2
Performance evaluation of the proposed framework across GDXray [35], SIXray [23] and OPIXray [27] dataset (at the inference stage) in terms of mAP_a by varying β. Bold indicates the optimal performance.

β	GDXray [35]	SIXray [23]	OPIXray [27]
5	0.7432	0.5206	0.6917
10	**0.7896**	**0.5974**	**0.7569**
15	0.7673	0.5413	0.7236
20	0.7338	0.5295	0.7092

5. Results

In this section, we report the experimental results of the proposed framework across each dataset. We also present an extensive list of ablation experiments to determine different hyper-parameters and backbone networks of the proposed framework. We also want to highlight that the instances for each item within all the datasets are fixed, and these instances are marked manually within each dataset. However, the way in which the model learns these instances incrementally allows the proposed scheme to learn new instances of the contraband items as well in the future.

5.1. Ablation studies

The ablative aspect of this work includes (1) determining the optimal β parameter to control the network remembrance ability; (2) comparing the performance of the incremental instance-aware learning approach with the traditional fine-tuning baseline; (3) analyzing the effect of encoder and transformer blocks within the proposed model; (4) analyzing the effect of including $\mathcal{R}$ within the proposed framework for producing optimal segmentation results; (5) evaluating the performance and statistical significance of $\mathcal{R}$ through the Pearson correlation coefficient; and (6) comparing different convolutional and transformer models for baggage threat instance segmentation.

5.1.1. Determining the β parameter

The β factor in the l_{old} loss function in Eq. (8) controls the ability of the proposed framework to remember its prior knowledge while learning about the new instance classes. Decreasing β emphasizes learning the newly added classes, whereas increasing its values makes the proposed framework more inclined to retain its prior learned knowledge. We conducted a series of experiments where we observed the overall performance of the proposed framework when varying β. As shown in Table 2, we notice that for GDXray [35], SIXray [23] and OPIXray [27] datasets (at inference stage), the optimal performance (in terms of mAP_a) is achieved for $\beta = 10$.

5.1.2. Comparison with a fine-tuning baseline

After determining the β factor, we compared the detection performance of the proposed incremental instance segmentation framework with its fine-tuned variant. For the fine-tuned variant, we first added all the suspicious items' instance classes within the last layer of the proposed network and then constrained it via categorical cross-entropy loss function to learn these instances. The performance comparison of both incrementally-trained and conventionally-trained variants of the proposed network is shown in Table 3. We can observe here that although the fine-tuned variant is leading the incremental variant by 2.76% on GDXray [35], 3.77% on SIXray [23], and 1.71% on OPIXray [27] in terms of mAP_{50}, the performance of the incremental variant is still competitive, especially considering the fact that it utilizes 35.0%, 42.8%, and 46.7% less training data from GDXray [35], SIXray [23] and OPIXray [27] dataset, respectively to produce these results. Here, we do not expect to beat the performance of the fine-tuned variant using the incrementally trained variant. Rather, our aim is to achieve the comparable performance with ±5% difference. It is

Table 3
Performance comparison of the proposed framework with its fine-tuned variant in terms of mAP_{50} at the inference (test) stage, and training time. Here, the training time reflects the total time taken by each variant.

Dataset	Incremental variant			Fine-tuned variant		
	mAP_{50} on test dataset	Training scans used	Training time (min)	mAP_{50} on test dataset	Training scans used	Training time (min)
GDXray	0.9481	260	14	0.9751	400	46
SIXray	0.8162	44,915	192	0.8539	78,575	427
OPIXray	0.8643	3786	78	0.8892	7109	293

Table 4
Performance comparison between the proposed scheme (incremental variant) with state-of-the-art methods in terms of total training time, as well as training time in each increment. Training time is reported in minutes. Moreover, bold indicates the best scores and the second best scores are underlined across each dataset.

Dataset	Method	Training time (Total)	Training time (Each increment)
GDXray	Proposed	**14**	**7.0**
	TST-l_{il} [24]	<u>18</u>	<u>9.0</u>
	RefineNet-l_{il} [3]	23	11.5
SIXray	Proposed	**192**	**32.0**
	TST-l_{il} [24]	<u>238</u>	<u>39.6</u>
	RefineNet-l_{il} [3]	304	50.6
OPIXray	Proposed	**78**	**39.0**
	TST-l_{il} [24]	<u>102</u>	<u>51.0</u>
	RefineNet-l_{il} [3]	146	73.0

natural that fine-tuning variant would perform better as it is trained on a large size the dataset with exhaustive training rounds as compared to the proposed incrementally trained variant. But, with significantly lesser amount of training data and iterations, the incrementally trained variant of the proposed scheme achieved 69.56%, 55.03%, and 73.37% more computationally efficiency than the fine-tuned variant across GDXray, SIXray, and OPIXray datasets, respectively, during the training stage (see Table 3 for more details). In addition to this, Table 4 reports the comparison of the proposed scheme with state-of-the-art models in terms of computational speed during the training phase, where it outperforms them significantly under the fair comparison (see Table 4 for more details).

5.1.3. Effects of encoder and transformer blocks

From Fig. 2, we can observe that the proposed model consists of encoder, transformer, and decoder blocks. The latent features produced by the encoder block are fused with the transformer projections in order to improve the discrimination between cluttered object instances by paying attention to their contextual and semantic attributes. Moreover, the fused features are then passed to the decoder end to extract the cluttered contraband objects. To analyze the extent of improvements that the proposed model brings by conjuncting encoder and transformer features, we conducted another series of experiments, in which we trained the proposed model with and without the inclusion of the encoder and transformer blocks, respectively, and measured their performance across GDXray, SIXray and OPIXray datasets. The quantitative results of these experiments are reported in Table 5. The utilization of aggregated features from transformer and encoder models led to an increased gain in performance compared to transformer-based features (which ranked second-best) in terms of mAP_a. Specifically, in the context of GDXray, SIXray, and OPIXray datasets, the aggregated features (transformer + encoder) achieved additional mAP_a scores of 3.65%, 4.72%, and 2.96%, respectively. It is worth noting that the performance gain in the case of the OPIXray dataset was slightly lower. This can be attributed to the presence of more diverse data samples in comparison to the other two datasets.

Table 5
Analyzing the effect of including encoder and transformer blocks within the proposed model in terms of mAP_{50}, mAP_{75}, and mAP_a scores across GDXray [35], SIXray [23], and OPIXray [27] datasets. Bold indicates the best score, while the second-best performance is underlined.

Dataset	Backbone	mAP_a	mAP_{50}	mAP_{75}
GDXray	Transformer projections	<u>0.7531</u>	<u>0.9073</u>	**0.8241**
	Encoder features	0.7442	0.8916	0.7506
	Transformer + Encoder	**0.7896**	**0.9481**	<u>0.8039</u>
SIXray	Transformer projections	<u>0.5502</u>	<u>0.7586</u>	<u>0.5910</u>
	Encoder features	0.5315	0.7493	0.5738
	Transformer + Encoder	**0.5974**	**0.8162**	**0.6214**
OPIXray	Transformer projections	<u>0.7273</u>	<u>0.8318</u>	<u>0.7419</u>
	Encoder features	0.7054	0.8071	0.6956
	Transformer + Encoder	**0.7569**	**0.8643**	**0.7682**

Table 6
Analyzing the effect of using regressor model that selects E_k model instead of E_m in order to perform k-instance segmentation tasks (where $k < m$). The comparison is performed in terms of mAP_{50}, mAP_{75}, and mAP_a scores across GDXray [35], SIXray [23], and OPIXray [27] datasets. Bold indicates the best score.

Dataset	Variant	Model instance	mAP_a	mAP_{50}	mAP_{75}
GDXray	Without $\mathcal{R}$	E_m	0.7453	0.8706	0.7697
	With $\mathcal{R}$	E_k	**0.7896**	**0.9481**	**0.8039**
SIXray	Without $\mathcal{R}$	E_m	0.4598	0.7054	0.5186
	With $\mathcal{R}$	E_k	**0.5974**	**0.8162**	**0.6214**
OPIXray	Without $\mathcal{R}$	E_m	0.7141	0.8286	0.7309
	With $\mathcal{R}$	E_m	**0.7569**	**0.8643**	**0.7682**

5.1.4. $\mathcal{R}$ for optimal segmentation

The proposed framework incorporates regression model $\mathcal{R}$ to determine the suitable model instance E_k for k-instance segmentation by identifying the instances k with the highest degree of overlap within the candidate test scan. However, utilizing model instance E_m (i.e., without $\mathcal{R}$) for k-instance segmentation, where $k < m$, can result in excessive segmentation (false positives). To avoid this, the regressor ($\mathcal{R}$) is employed to select the appropriate model instance E_k. The effect of using the regressor model, which selects E_k instead of E_m, for performing k-instance segmentation tasks (where $m < k$), is reported in Table 6 and Fig. 4. From Table 6, we can observe that the inclusion of $\mathcal{R}$ resulted in notable improvements, with an increase in mAP_a scores of 4.43%, 13.76%, and 4.28% for the GDXray, SIXray, and OPIXray datasets, respectively. Similarly, from Fig. 4(A)–(C), we can also notice a significant reduction in the pixel-wise false positives and false negatives that is obtained by selecting the E_k model instead of E_m to perform the k-instance segmentation, where the selection of E_k model is ensured through the inclusion of $\mathcal{R}$. Therefore, in the rest of the experimentation, we preferred the utilization of $\mathcal{R}$ within the proposed framework to perform optimal instance segmentation.

5.1.5. The regressor performance and its statistical significance

In order to ensure that $\mathcal{R}$ is able to correctly select the model instance E_k to perform k-instance segmentation tasks, we thoroughly tested its capacity to recognize the isolated and overlapping item instances from baggage X-ray scans during the validation and inference stage. Afterward, we computed the Pearson correlation coefficient (r_c)

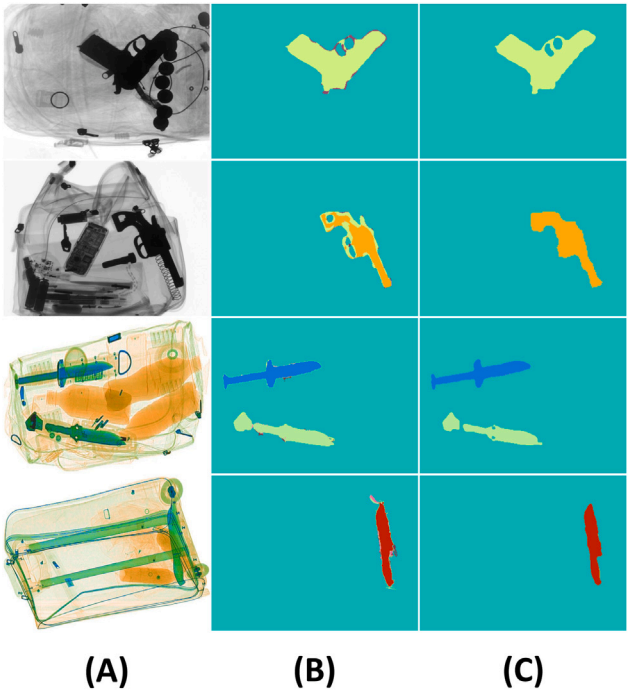

Fig. 4. Visual examples showing the comparison of k-instance segmentation using E_m and E_k models, where $k < m$. (A) denotes the original scans, (B) shows the over-segmentation results obtained by E_m model, and (C) represents the optimal segmentation results obtained through E_k model.

Table 7
Evaluation of the regressor $\mathcal{R}$ (at the validation and inference stage) through Pearson correlation coefficient (r_c), its statistical significance (p-value) and RMSE values.

Metric	GDXray [35]		SIXray [23]		OPIXray [27]	
	Validation	Inference	Validation	Inference	Validation	Inference
r_c	0.9294	0.9063	0.9016	0.8754	0.9145	0.8914
p-value	0.0162	0.0278	0.0467	0.0481	0.0302	0.0251
RMSE	0.1618	0.2203	0.2072	0.2635	0.1902	0.2148

between the predicted and true overlapping instances within each cropped patch of the X-ray scans (across each dataset) and measured the statistical significance of r_c through p-value (see Table 7). We can observe that on GDXray dataset, $\mathcal{R}$ achieved the r_c score of 0.9294 and 0.9063 with a p-value of 0.0162 and 0.0278 at the validation and inference stage, respectively. Similarly, on the SIXray, it scores an r_c of 0.9016 and 0.8754 with the p-value of 0.0467 and 0.0481 at the validation and inference stages, respectively. Also, on OPIXray, $\mathcal{R}$ obtained the r_c score of 0.9145 and 0.8914 with a p-value of 0.0302 and 0.0251 at the validation and inference stages, respectively. These r_c scores and the p-values below 0.05 confirms the robustness and statistical significance of $\mathcal{R}$. Apart from this, the performance of the regressor $\mathcal{R}$ is also evaluated through the RMSE values for each dataset reported in Table 7 showing that it achieved an overall RMSE score of 0.2203, 0.2635, and 0.2148 on the GDXray, SIXray, and OPIXray respectively at the inference stage which is quite appreciable.

5.1.6. Choice of segmentation network

Lastly, we evaluated the proposed incremental instance segmentation scheme by coupling within it the proposed convolutional transformer and many popular transformer, scene parsing, encoder–decoder, and fully convolutional-based models as a backbone, such as MedT [36], PSPNet [37], SegNet [38], and FCN-8 [39]. From Table 8, we can see that the proposed convolutional transformer is able to produce 4.87%, 3.91%, and 3.60% better extraction of the contraband items in terms of mean IoU as compared to the second-best SegNet [38] model on GDXray [35], OPIXray [27] and SIXray [23] dataset, respectively. Moreover, it leads all of its competitors by achieving the mean DC score of 0.9146 on GDXray [35], 0.8866 on OPIXray, and a mean DC score of 0.8467 on SIXray [23] dataset.

Table 8
Evaluation of the proposed framework using different encoder–decoder, transformer, fully convolutional, and scene parsing networks. Bold indicates the best scores and the second-best scores are underlined.

Metric	Dataset	SegNet	PSPNet	FCN-8	MedT	Proposed
μIoU	SIXray	0.6983	0.6741	0.6428	0.6582	**0.7341**
	OPIXray	0.7651	0.7489	0.7063	0.7345	**0.7963**
	GDXray	0.8123	0.7794	0.7214	0.7593	**0.8427**
μDC	SIXray	0.8224	0.8053	0.7826	0.7939	**0.8467**
	OPIXray	0.8669	0.8564	0.8279	0.8469	**0.8866**
	GDXray	0.8964	0.8760	0.8382	0.8632	**0.9146**

Table 9
Comparison of proposed framework with state-of-the-art methods on GDXray [35] dataset. Bold indicates the best performance while the second-best performance is underlined. Moreover, the abbreviations are: CST-l_{il}: CST [28] trained incrementally using the proposed l_{il} loss function, TST-l_{il}: TST [24] trained with l_{il}, RefineNet-l_{il}: RefineNet [3] trained with l_{il}, CIE-l_{il}: CIE [25] trained with l_{il}, DTSD: Dual-Tensor-Shot Detector [40], Mask R-CNN-l_{il}: Mask R-CNN [16] trained with l_{il}, TP-l_{il}: TP [29] trained with l_{il}. Also, '–' indicate that the metric is not computed for the given method.

Dataset	Method	mAP_a	mAP_{50}	mAP_{75}	AUC
GDXray	Proposed	**0.7896**	**0.9481**	**0.8039**	**0.9639**
	CST-l_{il} [28]	0.7041	0.8863	0.7365	0.9073
	TST-l_{il} [24]	0.7219	0.9056	0.7632	0.9248
	Mask R-CNN-l_{il} [16]	0.3882	0.6673	0.4219	0.7841
	Mask R-CNN[a] [16]	0.4723	0.7205	0.4901	0.8256
	RefineNet-l_{il} [3]	0.6256	0.8157	0.6608	0.8414
	CIE-l_{il} [25]	0.4813	0.7691	0.4958	0.8543
	YOLACT[a] [41]	0.3644	0.6548	0.3807	0.8092
	DTSD[a] [40]	–	0.9162	–	–
	TP-l_{il} [29]	0.7486	0.9237	0.7805	0.9418

[a] Indicates that the respective model is trained as per its original training configurations.

5.2. Evaluations on GDXray dataset

The scans in the GDXray dataset contain up to two isolated or overlapped instances of the same item. Table 9 reports the quantitative evaluation of the proposed framework in terms of mAP scores. We can observe that the proposed framework achieved the mAP_a score of 0.7896 and a mAP_{75} score of 0.8039 where it lead the state-of-the-art by 5.19% in terms of mAP_a and 2.91% in terms of mAP_{75}. Furthermore, it leads the tensor pooling (TP) framework [29] by 2.57%, the dual-tensor-shot detector (DTSD) [40] by 3.36%, and the cascaded structure tensors (CST) by 6.51% in terms of mAP_{50} under fair comparison. Moreover, considering the fact that both CST [28] and DTSD [40] are only object detection frameworks (lacking the capacity to generate object masks), we believe that the improvements produced by the proposed framework over CST [28] and DTSD [40] on the GDXray dataset are substantial. Apart from this, in terms of AUC scores, the proposed framework outperforms its competitors by achieving an AUC score of 0.9639, leading the TP [29] by 2.29% and trainable structure tensors (TST) framework [24] by 4.05%.

Apart from this, we report some qualitative evaluations of the proposed framework in Fig. 5. Here, scans (J and L) show only one occluded item (i.e., the *gun*), whereas the rest of the scans show more than one isolated or occluded item. Scan (B) shows two isolated instances of the *gun*, scan (N) shows isolated *gun* and *knives* instances, scan (D, F, and P) shows occluded instances of the *shuriken* and *razors*, whereas scan (H) showcase overlapping *knives*. Through these examples, we can appreciate the proposed framework's capacity to recognize concealed contraband items even from the challenging grayscale scans.

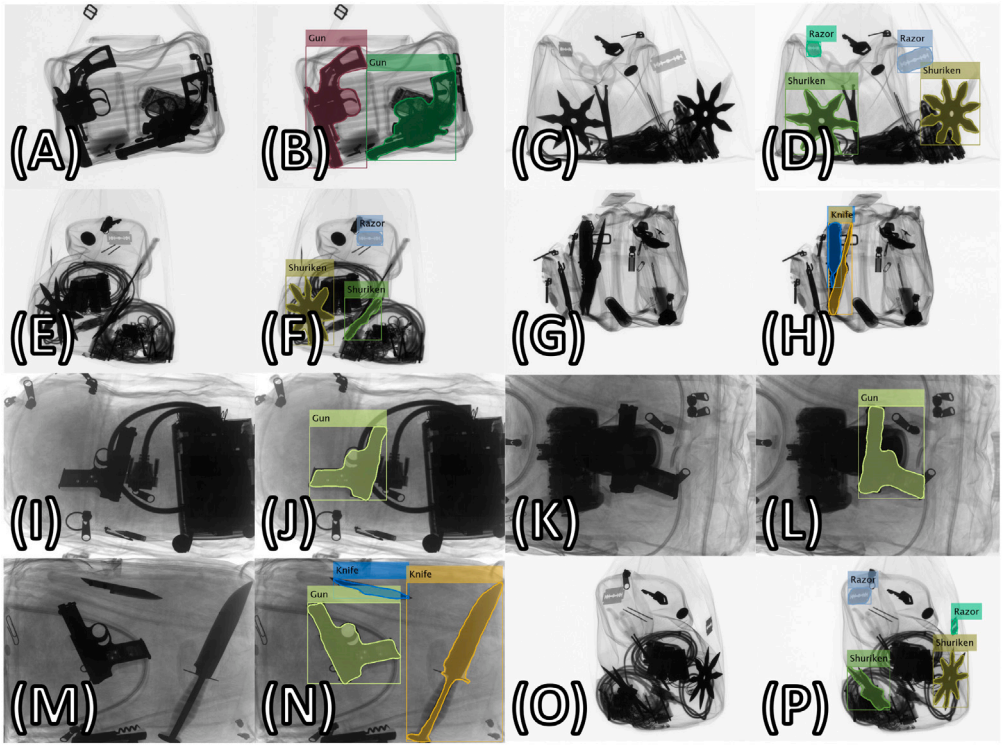

Fig. 5. GDXray: Examples of occluded and overlapping objects detection. The first and third column shows the original scans. Please zoom-in to best see the results.

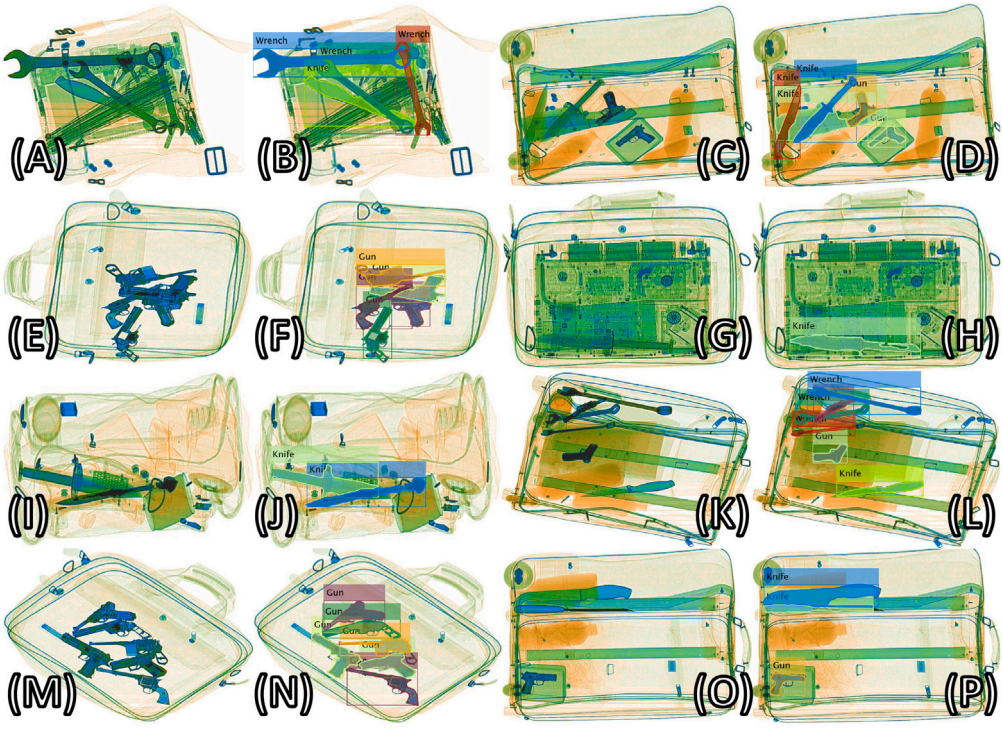

Fig. 6. SIXray: Examples of occluded and overlapping objects detection. The first and third column shows the original scans. Please zoom-in to best see the results.

5.3. Evaluations on SIXray dataset

Table 10 reports the thorough evaluation of the proposed framework and its comparison with state-of-the-art in terms of mAP_a, mAP_{50}, mAP_{75} and AUC scores. From Table 10, we can notice that, on the SIXray dataset, the proposed framework achieved the overall mAP_a score of 0.5974, mAP_{75} score of 0.6214, and mAP_{50} score of 0.8162, leading its competitors by 1.33%, 1.22%, and 1.32%, respectively in terms of mAP_a, mAP_{75}, and mAP_{50}. Similarly, on the SIXray10 subset, the proposed framework leads the state-of-the-art by 4.45% in terms of mAP_{75}. On SIXray100, the proposed framework is leading other methods by 3.71%. Moreover, on SIXray1000 (or SIXray1k) subset, the proposed framework is outperforming the second-best TP framework [29] (trained incrementally using l_{il} loss function) by 3.19%.

Table 10
Comparison of proposed framework with state-of-the-art methods on SIXray [23] dataset and its subsets. Bold indicates the best performance while the second-best performance is underlined. Moreover, the abbreviations are: CST-l_{il}: CST [28] trained incrementally using the proposed l_{il} loss function, TST-l_{il}: TST [24] trained with l_{il}, CHR: Class-Balanced Hierarchical Framework [23], CIE-l_{il}: CIE [25] trained with l_{il}, DTSD: Dual-Tensor-Shot Detector [40], Mask R-CNN-l_{il}: Mask R-CNN [16] trained with l_{il}, TP-l_{il}: TP [29] trained with l_{il}. Also, '–' indicate that the metric is not computed for the given method.

Dataset	Method	mAP_a	mAP_{50}	mAP_{75}	AUC
SIXray	Proposed	**0.5974**	**0.8162**	**0.6214**	**0.9893**
	CST-l_{il} [28]	0.5467	0.7765	0.5891	0.9752
	TST-l_{il} [24]	0.5702	0.7941	0.5915	0.9714
	Mask R-CNN-l_{il} [16]	0.2657	0.5213	0.2926	0.8594
	Mask R-CNN[a] [16]	0.3452	0.5986	0.3609	0.9243
	CIE-l_{il} [25]	0.4369	0.6641	0.4578	0.9371
	YOLACT[a] [41]	0.2964	0.5673	0.3193	0.8862
	CHR[a] [23]	–	0.7635	–	–
	DTSD[a] [40]	–	0.6457	–	–
	TP-l_{il} [29]	0.5894	0.8054	0.6138	0.9854
SIXray10	Proposed	**0.6466**	**0.8286**	**0.6983**	**0.9925**
	CST-l_{il} [28]	0.5982	0.7843	0.6173	0.9804
	TST-l_{il} [24]	0.6001	0.8021	0.6216	0.9761
	Mask R-CNN-l_{il} [16]	0.2793	0.5381	0.3072	0.8976
	Mask R-CNN[a] [16]	0.3498	0.6019	0.3703	0.9316
	CIE-l_{il} [25]	0.4549	0.6796	0.4711	0.9580
	YOLACT[a] [41]	0.3586	0.5891	0.3732	0.9092
	CHR[a] [23]	–	0.7956	–	–
	DTSD[a] [40]	–	0.8053	–	–
	TP-l_{il} [29]	0.6239	0.8174	0.6672	0.9857
SIXray100	Proposed	**0.5335**	**0.6882**	**0.5627**	**0.9252**
	CST-l_{il} [28]	0.4964	0.6462	0.5273	0.9016
	TST-l_{il} [24]	0.5048	0.6536	0.5429	0.9084
	Mask R-CNN-l_{il} [16]	0.1186	0.4059	0.1473	0.7162
	Mask R-CNN[a] [16]	0.2217	0.4804	0.2385	0.7424
	CIE-l_{il} [25]	0.3243	0.5586	0.3412	0.7825
	YOLACT[a] [41]	0.1735	0.4421	0.2054	0.7253
	CHR [23]	–	0.5992	–	–
	DTSD[a] [40]	–	0.6791	–	–
	TP-l_{il} [29]	0.5197	0.6626	0.5508	0.9113
SIXray1000	Proposed	**0.4476**	**0.5724**	**0.4537**	**0.8961**
	CST-l_{il} [28]	0.3842	0.5163	0.4045	0.8639
	TST-l_{il} [24]	0.4073	0.5349	0.4161	0.8765
	Mask R-CNN-l_{il} [16]	0.0981	0.2895	0.1109	0.6045
	Mask R-CNN[a] [16]	0.1796	0.3614	0.2071	0.6651
	CIE-l_{il} [25]	0.2753	0.4398	0.2942	0.6809
	YOLACT[a] [41]	0.1688	0.3246	0.1963	0.6253
	CHR [23]	–	0.4836	–	–
	DTSD [40]	–	0.4527	–	–
	TP-l_{il} [29]	0.4316	0.5571	0.4392	0.8926

[a] Indicates that the respective model is trained as per its original training configurations.

Apart from this, in terms of mAP_{50}, the proposed framework achieves 12.93% and 15.51% improvements over the CHR method [23] on the SIXray100 and SIXray1000 subsets. In addition to this, in terms of AUC, the proposed framework achieves a score of 0.9925, 0.9252, and 0.8961 on the SIXray10, SIXray100, and SIXray1000, respectively, which is also appreciable. These improvements stem from the fact that the proposed framework is trained incrementally to recognize individual instances of the contraband items where the catastrophic forgetting phenomena is controlled through the β hyperparameter. Furthermore, the unique integration of SPBs, RBs, along with the transformer projections allowed the proposed model to accurately generate the distinct feature representations of the cluttered items' instances, enabling the decoder end to effectively extract them from the baggage X-ray scans at the inference stage.

Although, we can notice that the difference between the AUC and the mAP scores is substantial on SIXray [23] dataset (especially on SIXray100, and SIXray1000). This difference is explained by the fact that AUC scores are computed pixel-wise, where the background class (true negatives) is also included. Since the negative scans in the SIXray dataset (especially its subsets) contain only the background pixels, the correct classification of these background pixels results in a drastic increase in AUC scores.

Table 11
Comparison of proposed framework with state-of-the-art methods on OPIXray [27] dataset. Bold indicates the best performance while the second-best performance is underlined. Moreover, the abbreviations are: CST-l_{il}: CST [28] trained incrementally using the proposed l_{il} loss function, RefineNet-l_{il}: RefineNet [3] trained with l_{il}, TST-l_{il}: TST [24] trained with l_{il}, CIE-l_{il}: CIE [25] trained with l_{il}, TP-l_{il}: TP [29] trained with l_{il}. Also, '–' indicate that the metric is not computed for the given method.

Dataset	Method	mAP_a	mAP_{50}	mAP_{75}	AUC
OPIXray	Proposed	**0.7569**	**0.8643**	**0.7682**	**0.9702**
	CST-l_{il} [28]	0.6852	0.8046	0.7235	0.9342
	RefineNet-l_{il} [3]	0.6217	0.7359	0.7136	0.8472
	TST-l_{il} [24]	0.7053	0.8234	0.7257	0.9402
	CIE-l_{il} [16]	0.6834	0.8029	0.7383	0.9416
	DOAM[a] [27]	–	0.8241	–	–
	TP-l_{il} [29]	0.7354	0.8378	0.7391	0.9538
	YOLOv8 [42]	0.7219	0.8183	0.7316	0.9381
	FCOS [43]	0.6528	0.7519	0.6629	0.9104
	DETR [44]	0.7108	0.8043	0.7175	0.9238

[a] Indicates that the respective model is trained as per its original training configurations.

We also want to highlight here that many researchers have used Faster R-CNN [45] and Mask R-CNN [16] to recognize baggage threats from the security X-ray scans [46]. Some of them have also tested them on SIXray10 [23] subset of the SIXray [23] dataset for detecting *guns* and *knives* [46] where they achieved the mAP_{50} score of 0.86 using Faster R-CNN [45], and a mAP_{50} score of 0.84 using Mask R-CNN [16]. However, the comparison of the proposed framework with this method would be unfair as these studies have only used Mask R-CNN [16] to extract *guns* and *knives* whereas we have extracted all the suspicious items marked within the original dataset [23].

Moreover, in Fig. 6, we report some qualitative examples depicting the performance of the proposed framework in accurately extracting the concealed, cluttered, and overlapping contraband items. In the first row of Fig. 6, we can appreciate the detection of the concealed *knife*, *guns*, *wrenches* in the scan (B) and (D). In the second, third, and fourth rows, depicting the scans having up to six isolated and overlapping instances of the same item, we can observe the capacity of the proposed framework in picking the overlapping *wrenches*, *knives*, and *guns* (F, H, J, L, N, and P).

5.4. Evaluations on OPIXray dataset

The third dataset on which we evaluated the proposed framework is the OPIXray [27] dataset. From Table 11, we can observe that the proposed framework, in terms of mAP_{50}, achieved 3.06% improvement as compared to the second-best TP-l_{il} framework [29], and lead the DOAM framework [27] (backboned through FCOS [43]) by 4.65% in terms of mAP_{50}. Furthermore, in terms of mAP_a, mAP_{75}, and AUC scores, the proposed framework lead the state-of-the-art by 2.84%, 3.78%, and 1.69%, respectively. These performance improvements, under fair comparison, emanates from the capacity of the proposed convolutional transformer block to generate unique feature representations of the threat objects from the cluttered baggage content which allows their accurate extraction irrespective of the scanner specifications. In addition to this, we have compared the performance of the proposed framework across different occlusion levels defined by the OPIXray dataset. From Table 12, we can observe that at OL1, OL2, and OL3, the proposed framework lead the second-best TP [29] by 1.98%, 3.36%, and 0.829%.

In Fig. 7, we report the qualitative results of the proposed framework on the OPIXray [27] dataset. We can observe here that the proposed framework has accurately extracted the contraband items (see scans B, D, and N). Especially under extremely cluttered scenarios, as observed in scans (F, H, J, L, and P), the extraction of the *straight knife* and *scissor* is remarkable.

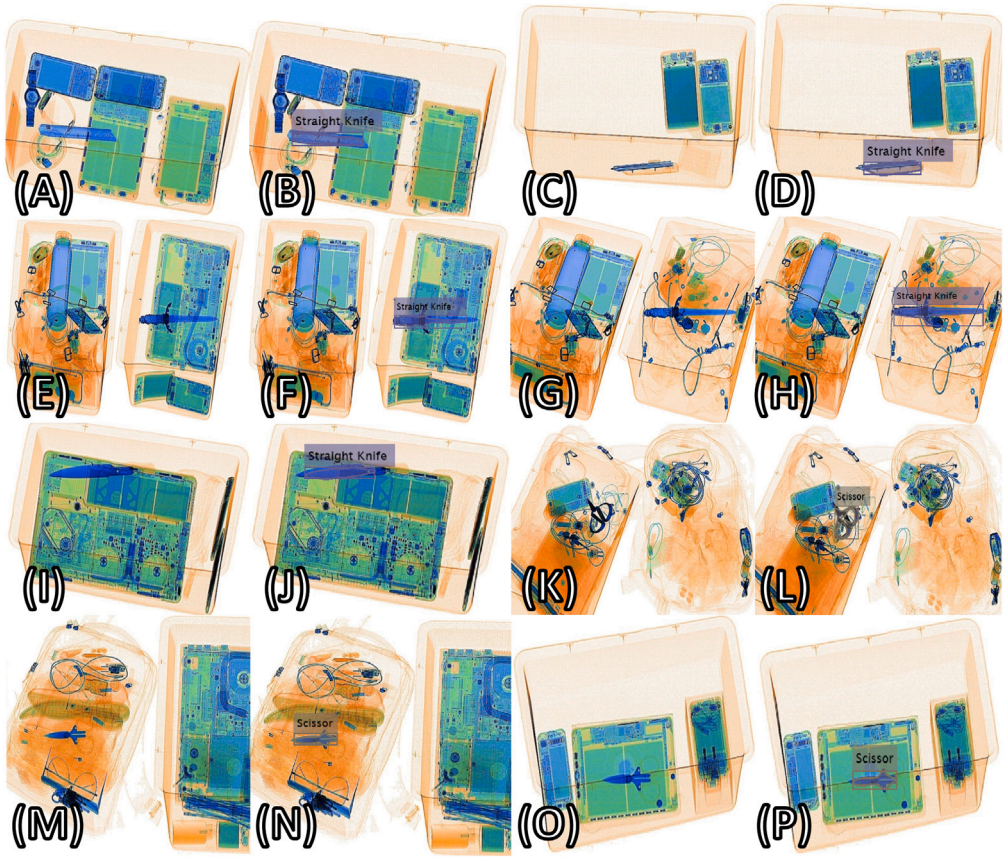

Fig. 7. Qualitative evaluation of the proposed framework on OPIXray [27] dataset. Please zoom-in for better visualization.

Table 12
Performance comparison of proposed framework with TP [29], DOAM [27] (backboned through SSD [47]) on different occlusion levels of OPIXray [27] dataset.

Method	OL1	OL2	OL3
Proposed framework	**0.8107**	**0.7639**	**0.7352**
Tensor pooling [29]	0.7946	0.7382	0.7291
DOAM + SSD [27]	0.7787	0.7245	0.7078
SSD [47]	0.7545	0.6954	0.6630

Table 13
Run-time performance in frames per second (FPS) on SIXray [23], GDXray [35], and OPIXray [27]. Bold indicates the best performance while the performance of the proposed model is underlined. '–' shows that the respective model is not tested for the given dataset.

Method	GDXray	SIXray	OPIXray	Mean
Mask R-CNN [16]	4.9	5.8	–	5.3
YOLACT [41]	**23.7**	**24.1**	**24.6**	**25.4**
Tensor pooling [29]	4.3	5.3	5.8	5.1
CIE [25]	5.6	6.2	6.7	5.8
Proposed	6.5	7.3	7.9	7.1

5.5. Qualitative comparison

Apart from quantitatively evaluating the proposed framework against state-of-the-art, we have also qualitatively compared it through visual examples as shown in Fig. 8. Here, we can observe that, although, many recent state-of-the-art frameworks were able to localize the contraband data through masks. Still, the quality of the masks produced by each method (when trained in a fair manner) varies significantly. For example, comparing (U) with (AD, AM, AV), and (X) with (AG, AP, AY), we can observe how accurately the proposed framework generated the mask for the *gun*. Similarly, comparing the extraction of *straight knife* in (Z) with (AI, AR, AAA) and in (AA) with (AJ, AS, AAB), we can appreciate the capacity of the proposed convolutional transformer, which generate unique shape preserving embeddings and self-attention projections to segment the cluttered contraband data, as compared to the state-of-the-art.

5.6. Run-time performance

We have also compared the proposed framework's run-time performance with the state-of-the-art models in terms of frames per second (FPS). The results depicted in Table 13 report that, on average, the proposed framework achieved the run-time performance of 7.1 FPS which is although lagging behind YOLACT by a significant gap but outperforms other competitive frameworks by 18.30%, in terms of FPS, which is appreciable. Furthermore, comparing the detection versus run-time performance of the proposed framework and YOLACT [41], where the proposed framework achieved 30.93%, 30.49%, 28.90%, 35.75%, and 43.29% improvements in terms of mAP_{50} over YOLACT [41] across GDXray, SIXray, SIXray10, SIXray100, and SIXray1000 datasets (see Table 11), we believe that the applicability of the proposed framework is quite significant as compared to YOLACT [41] for threat object detection.

5.7. Limitations

This section highlights and discusses the potential limitations of the proposed framework and their remedies. The limitations of the proposed framework are highlighted in Fig. 9. The first one is related to the incapacity of the proposed framework to generate small masks for the extremely cluttered or less frequently observed items like the *gun* in (B) and the *scissor* in (D and L). Apart from this, the second limitation of

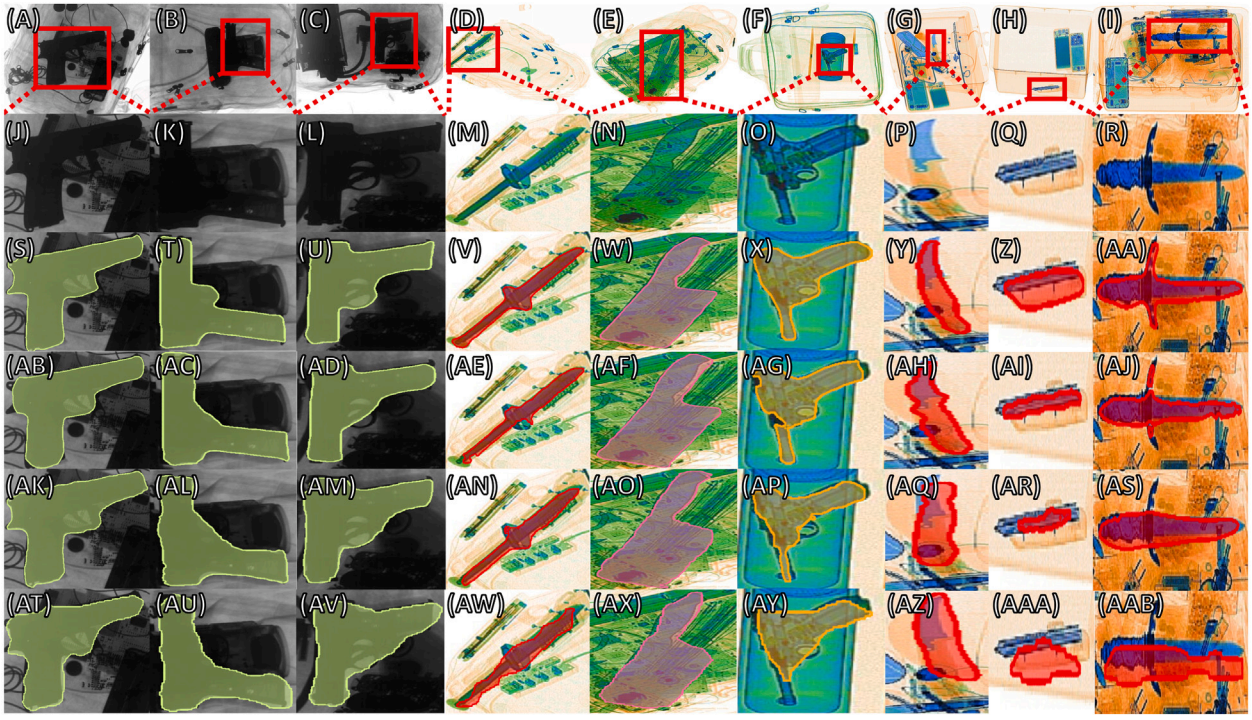

Fig. 8. Qualitative comparison of the proposed framework with state-of-the-art methods towards extracting the masks of the contraband objects depicted within the baggage X-ray scans of the GDXray, SIXray, and OPIXray datasets. (A-I) shows original scans, (J-R) shows the cropped region within the scans containing suspicious objects, (S-AA) shows the masks of the suspicious items generated by the proposed framework, (AB-AJ) shows the masks extracted by TP [29], (AK-AS) shows the masks extracted by CIE [25], and (AT-AAB) shows the masks extracted by TST [24]. Please zoom-in for better visualization.

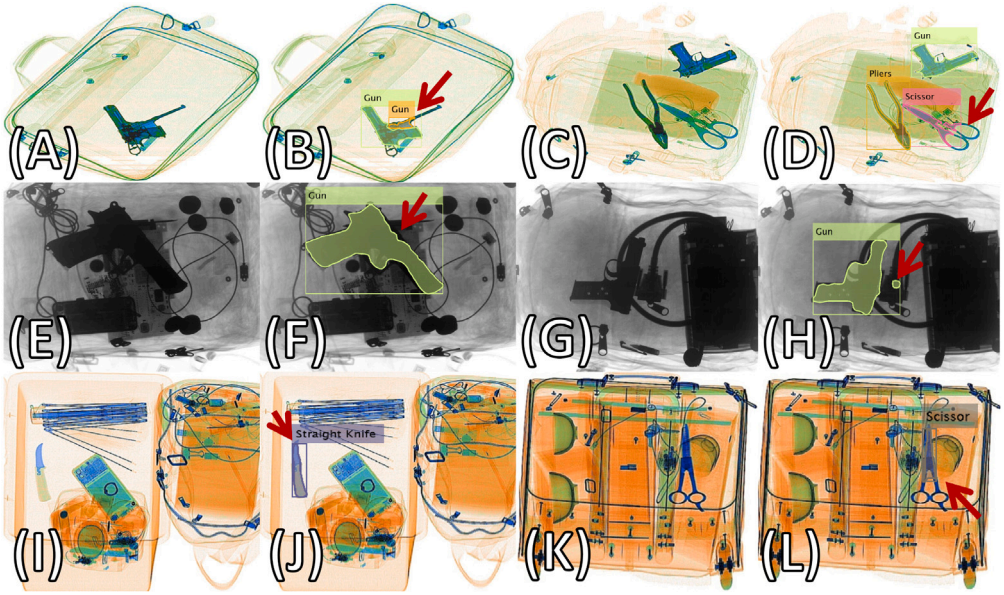

Fig. 9. Visual examples showcasing the limitations of the proposed framework. A red arrow in second and fourth columns indicates each limitation. Please zoom-in for better visualization.

the proposed framework is its inability to generate accurate masks for cluttered items. For example, see the inconsistencies produced within the mask of *gun* in (F), and *straight knife* in (J). Moreover, the third limitation of the proposed framework is in its inability to generate pixel-level false positive, as it can be seen for the *gun* in (H). The first limitation can be addressed easily by employing morphological opening operations as a postprocessing step. The second limitation, although, is not a major problem because the masks shown in (F) and (J) are still of decent quality, and due to the fact that the proposed framework can extract better masks than state-of-the-art as evident from Fig. 8. Still, it can be addressed by employing sophisticated segmentation loss functions (e.g., dice or IoU loss) as an objective function within l_{il} to constrain the model in preserving the exact shape of the segmented objects. The third limitation of the proposed framework can be addressed by employing morphological blob opening operations as a postprocessing step.

6. Discussion

This paper presents a novel convolutional transformer and an incremental instance segmentation scheme to recognize concealed and cluttered instances of contraband items via baggage X-ray imagery. The proposed system requires a significantly less computational supervision, and training resources than the ones which are required in the conventional training (or fine-tuning) process to teach the segmentation model to recognize all the classes at once. We further wanted to ensure that the proposed framework does not employ multiple encoders to process one scan, and, in each training increment, we only train one instance of the model and save it for later usage. At the inference stage, we select the appropriate instance k of the proposed model E_k (using the regressor $\mathcal{R}$) and use it to perform the required k-instance segmentation. Indeed, it is true that we save all the trained instances of the model on the hard disk and load the best one during the inference stage, but the proposed approach neither takes more computational time nor computational memory than the conventional instance segmentation methods. Because at inference stage, only a single regressor and at most two networks (one for semantic segmentation and one for instance segmentation) are loaded into the memory. Even at training, only one model is kept in the memory in the current j increment whereas all the previously trained models (in increment 1 to $j-1$) are saved on the hard disk. We note also that using E_m to perform k-instance segmentation task might lead to over-segmentation that can degrade the detection performance of the proposed system.

We also want to highlight that the proposed framework uses only a single convolutional transformer network to perform instance segmentation, where we incrementally add new classes in the network by modifying the final layer of the decoder (during training phase) and update the weights of the proposed network through l_{il} loss function. Also, we do not use multiple encoders and decoders in each training increment. Rather, we only update the weights of the model instance that is trained in the previous training increment.

Although, the proposed scheme is generalizable to any number of overlapping instances. For example, to recognize 'm' instances, the proposed scheme requires 'm' incremental iterations to perform m-instance segmentation tasks. Similarly, if needed in the future, with one more incremental iteration, the proposed scheme can be tuned to recognize '$m+1$' overlapping instances without requiring exhaustive re-training or conventional fine-tuning rounds. Apart from this, we concur that the complexity of the segmentation problem increases linearly with the maximum number of overlapping instances, thus, putting more constraints on the underlying segmentation network. But, for the baggage threat detection problem, practically we do not expect the number of these overlapping instances to exceed more than six. Even on the highly complex SIXray dataset (which contains unrealistically challenging examples [23]), we found only a few scans that contained at most six overlapping instances of the suspicious items, which we were able to extract as shown in Fig. 6.

In addition to this, the proposed incremental learning scheme requires lesser number of training samples in each increment as compared to the fine-tuning variant. Similarly, the total time it takes to complete the incremental training is significantly lesser than the conventional fine-tuning as reported in Table 3 of the revised manuscript. In addition to this, the proposed scheme can easily learn new types of threat items (or their instances) in a very few iterations using lesser number of samples. However, the conventional fine-tuning variant requires an exhaustive routine in which the model will be re-learning the already learned classes along with the new instance classes so that it does not forget its prior knowledge [31–33].

Also, in the proposed incremental learning approach, storing and loading multiple models can be computationally expensive. Therefore, it is important to strike a balance between efficiency and effectiveness in incremental learning according to the domain application of a deep learning model.

7. Conclusion

In this paper, we present a novel approach for detecting contraband items using security X-ray scans. The proposed scheme is driven by a convolutional transformer that is distinguished by its incremental learning aspect, allowing it to continually adapt itself and learn new instances of prohibited items periodically. The in-built design of the proposed framework drastically increases its applicability to perform baggage threat detection tasks in the real world without having to re-train itself repeatedly from scratch in order to learn new emerging instances of threatening objects. Additionally, deploying the proposed framework in the real world can reduce the significant load of security staff towards detecting the concealed and cluttered instances of contraband objects from the X-ray scans. Manually screening such items is a subjective process, especially during rush hours and tiring work schedules. But the proposed framework, when deployed in the real world, can overcome this problem as it can reliably and efficiently screen the baggage X-ray scans against potentially dangerous and threatening items. Furthermore, it does not require hectic training workflows to learn new instances of threat object categories.

Apart from this, the proposed framework has been thoroughly evaluated on three public X-ray datasets, namely GDXray [35], SIXray [23], and OPIXray [27]. The series of experiments conducted on these datasets evidenced the expansive capacity of the proposed system to screen extensively cluttered suspicious objects and demonstrated its superiority over existing state-of-the-art solutions across multiple evaluation metrics. For example, across GDXray [35] dataset, the proposed system achieved a mAP_a score of 0.7896, outperforming the state-of-the-art by 5.19%. Similarly, across SIXray [23], and OPIXray [27] datasets, the proposed framework leads the state-of-the-art by 1.33%, and 2.84% by achieving a mAP_a score of 0.5974, and 0.7569, respectively. The performance improvements across these datasets emanate from the capacity of the proposed model to generate unique feature representations of the threat objects from the cluttered baggage content, which allows their accurate extraction irrespective of the scanner specifications.

Our next research direction will focus on detecting 3D-printed items in X-ray scans. Due to their low reflectance, such items represent an exciting challenge for imaging modalities. Investigating the extent to which the current learned framework can accommodate this category of objects will be our first investigation stage. Similarly, we have also interpreted and confirmed the reliability of the proposed system's performance through an extensive set of ablation experiments. But conducting a proper uncertainty analysis [48,49] would be worthwhile to ensure scalability and robustness of the proposed scheme during the deployment phase. Moreover, the threat items can have origins in different associated and individual parts. Although, to date, there are no public X-ray datasets that contain such types of dismantled or 3D-printed threat objects. Nevertheless, recognizing such contraband items still poses a considerable challenge, even to the security staff, and exploring the proposed framework to extract and detect such types of items across these datasets would be a worthwhile investigation in the future. Apart from this, exploring the advanced segment anything models (SAM) within the baggage threat detection territory is also important. However, they require a decent set of text prompts generated by the expert security staff [23], and to the best of our knowledge, there is no public baggage threat detection dataset till date which contains such kind of information for applying SAMs [23,27,35]. Therefore, we also envisage it as one of the possible future directions related to this work.

CRediT authorship contribution statement

Taimur Hassan: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Bilal Hassan:** Data

curation, Formal analysis, Investigation, Methodology, Visualization, Writing – original draft. **Muhammad Owais:** Software, Validation, Writing – original draft. **Divya Velayudhan:** Visualization, Writing – original draft. **Jorge Dias:** Resources, Validation, Writing – review & editing. **Mohammed Ghazal:** Methodology, Validation, Visualization, Writing – review & editing. **Naoufel Werghi:** Data curation, Funding acquisition, Methodology, Supervision, Validation, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All the data that is used in this research is publicly available.

Acknowledgments

This research is supported by funds from the Higher Education Commission (HEC), Pakistan, Ref: 20-15089/NRPU/R&D/HEC/2021, and Khalifa University, UAE: Ref: CIRA-2021-052 and RC1-2018-KUCARS-8474000136.

References

[1] S. Akçay, T. Breckon, Towards automatic threat detection: A survey of advances of deep learning within X-ray security imaging, tPatRec (February 2022).
[2] G. Flitton, A. Mouton, T.P. Breckon, Object classification in 3D baggage security computed tomography imagery using visual codebooks, Pattern Recognit. (2015).
[3] M. Shafay, T. Hassan, D. Velayudhan, E. Damiani, N. Werghi, Deep fusion driven semantic segmentation for the automatic recognition of concealed contraband items, in: SoCPaR, 2020, pp. 550–559.
[4] K. Shaukat, S. Luo, V. Varadharajan, A novel deep learning-based approach for malware detection, Eng. Appl. Artif. Intell. (2023).
[5] K. Shaukat, et al., Performance comparison and current challenges of using machine learning techniques in cybersecurity, Energies (2020).
[6] K. Shaukat, et al., A survey on machine learning techniques for cyber security in the last decade, IEEE Access (2020).
[7] F. Lu, Z. Zhang, T. Liu, C. Tang, H. Bai, G. Zhai, J.C. c, X. Wu, A weakly supervised inpainting-based learning method for lung CT image segmentation, Pattern Recognit. (2023).
[8] K. Shaukat, et al., A review of time-series anomaly detection techniques: A step to future perspectives, in: FICC, 2021.
[9] K. Shaukat, et al., The impact of artificial intelligence and robotics on the future employment opportunities, Trends Comput. Sci. Inf. Technol. (2020).
[10] K. Shaukat, et al., A review on security challenges in internet of things (IoT), in: 26th International Conference on Automation and Computing, ICAC, 2021.
[11] K. Shaukat, et al., A socio-technological analysis of cyber crime and cyber security in Pakistan, in: Sociology, Computer Science, 2017.
[12] M. Shafay, T. Hassan, E. Damiani, N. Werghi, Temporal fusion based multi-scale semantic segmentation for detecting concealed baggage threats, in: IEEE International Conference on Systems, Man, and Cybernetics, SMC, 2021, August.
[13] U. Tariq, I. Ahmed, A.K. Bashir, K. Shaukat, A critical cybersecurity analysis and future research directions for the internet of things: A comprehensive review, Sensors (2023).
[14] K. Shaukat, S. Luo, S. Chen, D. Liu, Cyber threat detection using machine learning techniques: A performance evaluation perspective, in: International Conference on Cyber Warfare and Security, ICCWS, 2020.
[15] K. Shaukat, S. Luo, V. Varadharajan, A novel method for improving the robustness of deep learning-based malware detectors against adversarial attacks, Eng. Appl. Artif. Intell. (2022).
[16] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: ICCV, 2017, pp. 2961–2969.
[17] A. Mouton, T.P. Breckon, Materials-based 3D segmentation of unknown objects from dual-energy computed tomography imagery in baggage security screening, Pattern Recognit. (2015).
[18] D. Velayudhan, T. Hassan, E. Damiani, N. Werghi, Recent advances in baggage threat detection: A comprehensive and systematic survey, tACMCS (2022).
[19] M. Bastan, Multi-view object detection in dual-energy X-ray images, Mach. Vis. Appl. (2015) 1045–1060.
[20] D. Mery, E. Svec, M. Arias, Object recognition in baggage inspection using adaptive sparse representations of X-ray images, in: PSIVT, 2016, pp. 709–720.
[21] N. Jaccard, T.W. Rogers, L.D. Griffin, Automated detection of cars in transmission X-ray images of freight containers, in: AVSS, 2014, pp. 387–392.
[22] S. Akcay, M.E. Kundegorski, C.G. Willcocks, T.P. Breckon, Using deep convolutional neural network architectures for object classification and detection within X-ray baggage security imagery, tIEEETIFS 13 (9) (2018) 2203–2215.
[23] C. Miao, L. Xie, F. Wan, C. Su, H. Liu, J. Jiao, Q. Ye, Sixray: A large-scale security inspection X-ray benchmark for prohibited item discovery in overlapping images, in: CVPR, 2019, pp. 2119–2128.
[24] T. Hassan, S. Akçay, M. Bennamoun, S. Khan, N. Werghi, Trainable structure tensors for autonomous baggage threat detection under extreme occlusion, in: Asian Conference on Computer Vision, ACCV, 2020.
[25] T. Hassan, S. Akcay, M. Bennamoun, S. Khan, N. Werghi, A novel incremental learning driven instance segmentation framework to recognize highly cluttered instances of the contraband items, IEEE Trans. Syst. Man Cybern. Syst. (2021).
[26] T. Hassan, S. Akçay, M. Bennamoun, S. Khan, N. Werghi, Unsupervised anomaly instance segmentation for baggage threat recognition, J. Ambient Intell. Hum. Comput. (2021) July.
[27] Y. Wei, R. Tao, Z. Wu, Y. Ma, L. Zhang, X. Liu, Occluded prohibited items detection: An X-ray security inspection benchmark and de-occlusion attention module, ACM Multimedia (2020).
[28] T. Hassan, S. Akcay, B. Hassan, M. Bennamoun, S. Khan, J. Dias, N. Werghi, Cascaded structure tensor for robust baggage threat detection, Neural Comput. Appl. (2023).
[29] T. Hassan, S. Akcay, M. Bennamoun, S. Khan, N. Werghi, Tensor pooling driven instance segmentation framework for baggage threat recognition, tNCApp (2022).
[30] Z. Li, D. Hoiem, Learning without forgetting, in: ECCV, 2016.
[31] A. Dong, J. Liu, G. Zhang, Z. Wei, Y. Zhai, G. Lv, Momentum contrast transformer for COVID-19 diagnosis with knowledge distillation, Pattern Recognit. (2023).
[32] R. Xu, L. Xing, B. Liu, D. Tao, W. Cao, W. Liu, Cross-domain few-shot classification via class-shared and class-specific dictionaries, Pattern Recognit. (2023).
[33] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, Icarl: Incremental classifier and representation learning, in: CVPR, 2017.
[34] H. Samet, M. Tamminen, Efficient component labeling of images of arbitrary dimension represented by linear bintrees, IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI) (1988).
[35] D. Mery, V. Riffo, U. Zscherpel, G. Mondragón, I. Lillo, I. Zuccar, H. Lobel, M. Carrasco, Gdxray: The database of X-ray images for nondestructive testing, J. Nondestruct. Eval. 34 (4) (2015).
[36] J.M.J. Valanarasu, et al., Medical transformer: Gated axial-attention for medical image segmentation, in: MICCAI, 2021.
[37] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: CVPR, 2017, pp. 2881–2890.
[38] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.
[39] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: CVPR, 2015.
[40] T. Hassan, M. Shafay, S. Akçay, S. Khan, M. Bennamoun, E. Damiani, N. Werghi, Meta-transfer learning driven tensor-shot detector for the autonomous localization and recognition of concealed baggage threats, Sensors (2020).
[41] D. Bolya, C. Zhou, F. Xiao, Y.J. Lee, YOLACT: Real-time instance segmentation, in: ICCV, 2019, pp. 9157–9166.
[42] D. Reis, J. Kupec, J. Hong, A. Daoudi, Real-time flying object detection with YOLOv8, 2023, arXiv:2305.09972.
[43] Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in: CVPR, 2019.
[44] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, ECCV, 2020.
[45] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. (2015).
[46] Y.F.A. Gaus, et al., Evaluating the transferability and adversarial discrimination of convolutional neural networks for threat object detection and classification within X-Ray security imagery, in: ICMLA, 2019.
[47] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot MultiBox detector, in: ECCV, 2016.
[48] D. Zhang, Y. Fu, Z. Zheng, UAST: Uncertainty-aware siamese tracking, in: ICML, 2022.
[49] H. Hojjati, T.K.K. Ho, N. Armanfard, Self-supervised anomaly detection: A survey and outlook, in: ESWA, 2023.

Taimur Hassan received the Ph.D. degree in computer engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2019. He is currently working as an Assistant Professor in the Department of Electrical, Computer and Biomedical Engineering at Abu Dhabi University, United Arab Emirates. Dr. Hassan has led many local and foreign research projects. His research interests lie in the fields of computer vision, medical imaging, deep learning, the Internet of Things, and robotics.

Dr. Hassan's Ph.D. research won the Gold Award in the Research and Development Category at the Pakistan Software Houses Association for IT and ITeS (P@SHA) ICT Awards in 2016, the Gold Award in the Research and Development Category at the Asia Pacific ICT Alliance (APICTA) Awards in 2016, and the Gold Award in the Artificial Intelligence category at P@SHA ICT Awards in 2018. He was a recipient of many national and international awards.

Bilal Hassan received the Ph.D. degree in pattern recognition and intelligent systems from Beihang University, Beijing, China, in 2022. He is currently a Post-Doctoral Fellow with the Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. His research interests lie in the domain of computer vision, medical imaging, wireless communication, and control systems.

Muhammad Owais received the B.S. and M.S. degrees in computer engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2014 and 2016, respectively. He is completed his Ph.D. in electronics and electrical engineering from Dongguk University, Seoul, South Korea. His research interests include medical image analysis and deep learning.

Divya Velayudhan completed her M. Tech. in Signal Processing from Sree Chitra Thirunal College of Engineering, India. She is currently pursuing her Ph.D. in Electrical Engineering and Computer Science (EECS) at Khalifa University, United Arab Emirates. Her interests are in Signal Processing, Computer Vision and Machine Learning.

Jorge Dias received the Ph.D. degree in electrical engineering from the University of Coimbra, Coimbra, Portugal, in 1994. He coordinated the Artificial Perception Group, Institute of Systems and Robotics, University of Coimbra. He is currently a Full Professor with the Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates, where he is also the Deputy Director of the Center of Autonomous Robotic Systems. His expertise is in the area of artificial perception (computer vision and robotic vision) and has contributions on the field since 1984. He has been a principal investigator and a consortia coordinator from several research international projects and coordinates the research group on computer vision and artificial perception from the Khalifa University Center for Autonomous Robotic Systems (KUCARS). He has published several articles in the area of computer vision and robotics that include more than 300 publications in international journals and conference proceedings and recently published book on probabilistic robot perception that addresses the use of statistical modeling and artificial intelligence for perception, planning, and decision in robots. He was the Project Coordinator of two European Consortium for the Projects "Social Robot" and "GrowMeUP" that were developed to support the inclusivity and wellbeing for of the elderly generation.

Mohammed Ghazal received the B.Sc. degree in computer engineering from the American University of Sharjah and the M.S. and Ph.D. degrees in electrical and computer engineering (ECE) from Concordia University, Canada, in 2006 and 2010, respectively. He is currently a Professor of ECE and the Chair of a department at Abu Dhabi University (ADU). He led the B.Sc. programs in ECE to achieve ABET and WASC accreditation. He also leads ADU E-Learning initiative. He has authored over 50 publications in technical journals and international conferences, including IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Transactions on Consumer Electronics. His current research interests include pattern recognition, computer vision, machine learning and medical imaging. He is a member of the ACM and the BMES. He was awarded the Alexander Graham Bell Fellowship by NSERC and the QFRNT by the Government of Quebec. He was also awarded the President's Cup and the Ministry of Education Shield for Creative Thinking at the American University of Sharjah. During his time at ADU, he received multiple awards, including the University Teaching Award, in 2013, and the Distinguished Faculty Award, in 2014 and 2016.

Naoufel Werghi received the Ph.D. degree in computer vision from the University of Strasbourg, Strasbourg, France, in 1996. He was a Research Fellow with the Division of Informatics, The University of Edinburgh, Edinburgh, U.K., and a Lecturer with the Department of Computer Sciences, University of Glasgow, Glasgow, U.K. He was a Visiting Professor with the Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY, USA. He is currently a Professor of Computer Vision in the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates. His research interests include image analysis and interpretation, where he has been leading several funded projects in the areas of biometrics, medical imaging, and intelligent systems.