



A novel enhanced region proposal network and modified loss function: threat object detection in secure screening using deep learning

Priscilla Steno¹ · Abeer Alsadoon¹ · P. W. C. Prasad¹ · Thair Al-Dala'in¹ · Omar Hisham Alsadoon²

Published online: 7 September 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Detection of threat objects concealed in passenger clothing and baggage poses a challenge to aviation security. At present, the detection technology is capable of detecting the presence of threats from the scanned images yet requires the involvement of human in determining what type of threat and where it is located. Deep learning-based object detection technique has not been successfully implemented to detect threats in the security screening processes. This research aims to improve the accuracy and the processing time of threat detection in security screening. Enhanced faster region-based convolutional neural network (faster R-CNN) with improved region proposals is used for better threat localization. The proposed system consists of an improved region proposal network that outputs object's region proposals with an object score to the detector module to accurately locate the threat in the human body. Furthermore, this system uses a modified loss function that strengthens the classification loss. Results obtained by the proposed model show a 15% improvement in object localization. Therefore, the enhanced faster R-CNN achieves an overall detection accuracy of 0.27 in terms of average precision and reduces the processing time by 0.19 s. The results obtained by the enhanced faster R-CNN for detection accuracy are superior to the state-of-the-art system. Also, this model focuses on localizing the threat and identifying its type, which makes the model suitable for threat detection security screening. Besides, the research also addresses the time consumption issue in detecting the threat object.

Keywords Threat detection · Region proposal network · Deep learning · Faster R-CNN · Cross-entropy loss · Airport security

✉ Abeer Alsadoon
alsadoon.abeer@gmail.com

¹ School of Computing and Mathematics, Charles Sturt University, Sydney, Australia

² Department of Islamic Sciences, Al Iraqia University, Baghdad, Iraq

1 Introduction

Transport Security Administration (TSA) operates an Advanced Imaging Technology (AIT) unit with a target recognition software for passenger screening [1]. This creates a generic outline of the human body on which a yellow box appears if the scanner detects an object. Whenever a potential threat is identified, a TSA officer is required to engage in a secondary, manual security screening process. The manual process creates significant bottlenecks at the checkpoints; further threat prediction algorithm is inaccurate in predicting the region where the threat is located [2]. According to Guimaraes and Tofighi [2], the Department of Homeland Security in the USA reports higher false alarm rates resulting from their passenger-screening algorithm used at the airports. Solving these issues through the use of deep learning-based object detection techniques will help TSA to improve the passenger experience while maintaining high levels of security.

Object detection has been dramatically improved through deep learning technology [3]. Rapid development in deep learning has introduced more powerful tools to learn semantic, high-level, and deeper features to address the problems existing in traditional object detection techniques. Deep learning-based object detection frameworks perform better with higher real-time detection speed than the traditional detection methods such as scale-invariant feature transform (SIFT) and speeded up robust features (SURF) [4, 5]. In the traditional methods, it is necessary to select which features are important in each given image before feature extraction. Moreover, all the parameters for each feature definition must be fine-tuned by the engineer. In contrast, deep learning-based object detection models work by locating and classifying existing objects in an image automatically. The domain of passenger screening requires algorithms to localize the threat object in the body accurately and generating the region proposal. According to Guimaraes and Tofighi [2], the current threat detection system fails to predict the accurate region of threat, thus misleads the security screening system. Locating the objects is done through region proposal networks that focus on regions of interest. Improving the region proposal networks for better threat localization will produce an efficient deep learning-based threat detection system for aviation security screening.

In order to improve detection accuracy and performance, deep learning-based object detectors are recommended to be used in security screening and cargo imagery. Current studies of deep learning-based object detectors show that region proposal-based object detection frameworks perform better than regression-based frameworks [3]. At present, deep learning-based passenger security screening system of Guimaraes and Tofighi [2] works by rotating the body image to predict the presence of a threat. Figure 1 shows a rotated body image with a threat [2]. However, this solution fails to recognize the kind of threat and exact location as to where it is found in the human body. Faster region-based convolutional neural network (R-CNN) model by Ren et al. [6] addresses the relationship between classification and detection of an object with the performance of 5–7 frame rate per second and an accuracy of 73.2% in terms of mean average precision (mAP).

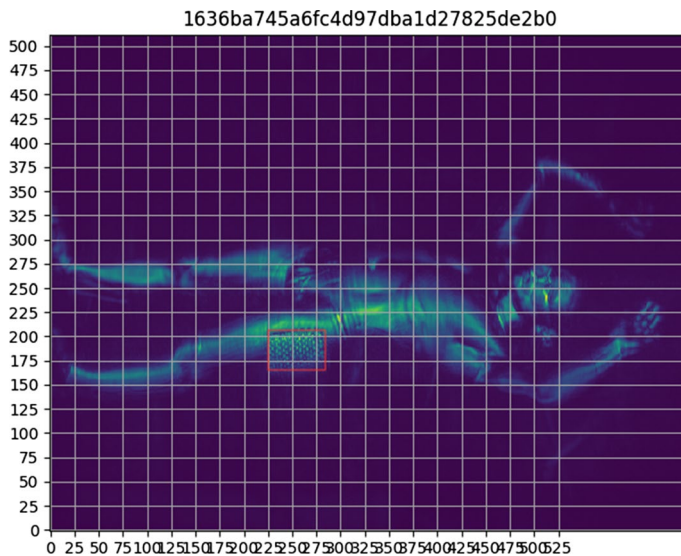


Fig. 1 A rotated body with a threat indicated with red [2] (color figure online)

This deep learning-based detection model by Ren et al. [6] classifies object into categories and indicates the exact location of the object. Therefore, it is possible to achieve an efficient security screening algorithm by utilizing an enhanced faster R-CNN model to detect the threat in its location.

The purpose of this paper is to introduce a threat detection system with higher threat detection accuracy and performance. The proposed system can be used in the domain of aviation security screening. Enhanced faster R-CNN with improved region proposals is used for better threat localization. The system consists of an improved region proposal network that output object's region proposals with an object score to the detector module to accurately locate the threat in the human body. Furthermore, this system uses an enhanced cross-entropy loss function to improve accuracy and Fast R-CCN detector to increase detection performance.

The rest sections of the paper are organized as follows: Sect. 2 presents a literature review that includes state of the art. Section 3 presents the proposed method. Section 4 presents the results and discussion of the proposed method. Section 5 presents the study conclusion.

2 Literature review

Deep convolutional neural network (CNN) has been taken into consideration for image classification and detection problem associated with security screening of baggage and a human body. The literature reviews are divided into four different sections based on pre-processing, classification, object detection, and detection

based on region proposals. Furthermore, the last section consists of the state-of-the-art solution.

Jaccard et al. [7] reviewed the past work on automating cargo X-ray image analysis. They identified that different image pre-processing methods such as image quality improvement, image manipulation, material discrimination, and segmentation, improve the accuracy of automated image understanding algorithms and corrects image errors that occur during image acquisition. Paper also suggests an automated threat detection method to improve accuracy. The study mentions a CNN model by Jaccard et al. [8] trained-from-scratch on an augmented data set, with real threat images projected. This model reported a false alarm rate of 0.8% given 90% detection. Image manipulation and image quality improvement techniques are suitable for the proposed solution. Akcay et al. [9] investigated the applicability of CNN in the domain of object classification of X-ray baggage images. They addressed binary classification and object classification of multiple classes. They used transfer learning to fine-tune the layers of CNN. A support vector machine (SVM) classifier trained on the fine-tuned network yielded the top performance and outperforming their previous work [10]. However, the model failed in detecting the foreground objects, which is increased false-negative occurrences. Though promising performance is achieved by CNN centred classification using transfer learning, more refined approaches are necessary to accomplish the joint localization of objects. Therefore, this solution is not suitable for the proposed solution. Svec et al. [11] investigated CNN models trained from scratch and focus on fine-tuned CNN and pre-trained CNN model. The first approach suffered from the problem of overfitting. To overcome this, they evaluated the two most popular pre-trained CNN models related to Image Net (_AlexNet and GoogleNet) [12]. The best performance was achieved by GoogleNet due to its nature of addressing the object detection problem, while AlexNet was a simple model built for classification. Since these pre-trained models mainly focus on classification, they are not suitable for detection of threats in the security screening system.

Krizhevsky et al. [13] trained a deep CNN that classifies the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. The architecture contains five convolutional layers and three fully connected layers with features of rectified linear unit nonlinearity to improve training time and local response normalization to reduce error rates significantly and drop out techniques to reduce overfitting. On ILSVRC-2010 competition, the model achieved results superior to the state-of-the-art solution of Sánchez and Perronnin [14]. Network's performance was dependent on the depth, i.e. the performance reduces if a single convolutional layer is detached. Although results are promising on large data set, the solution considers only the classification of images. The model requires object detection strategies to be incorporated to be used in the proposed solution. Guimaraes and Tofighi [2] work segmented human body into regions and labelled those regions. These were later fed into the classification model a CNN that configured with the solver-type stochastic gradient descent (SGD) using framework Tensorflow and the network AlexNet of Krizhevsky et al. [15]. This model could learn the body zones whether a threat is present or not. The model obtained an average accuracy for the validation data of 98.2863% and a loss of 0.091. Nevertheless, result produced

was highly accurate for predict zones but indicating overfitting of data. Also, the system fails to identify the kind of threat located. This study contributes to the proposed solution's classification module.

Intending to eliminate the need for a security officer to do a physical pat-down on passengers in order to detect threats, Bhattacharyya et al. [1] developed a deep learning-based project. The solution used 'AlexNet'. A CNN introduced by Krizhevsky et al. [15] with a convolutional layer, normalization layer, pooling layer, dropout layer, and fully connected layer. AlexNet was implemented on pre-processed images extracted from a full-body scan of a passenger. When tested, results suggested the algorithm as accurate by obtaining a log loss of 0.1221 and 0.0088 for two regions of the body. In addition, the solution is limited to just identifying the presence of a threat. Therefore, it seems that the solution needs to be further improved with object detection strategies to detect the type of threat found.

Riffo and Mery [16] and [17] have proposed an approach to detect objects in X-ray images in baggage screening automatically. The object detection task is automated through adapted implicit shape model (ASIM), which is an enhanced version of the implicit shape model in [18]. The solution uses SIFT descriptor to describe an object using several X-ray images from representative points of view. The visual vocabulary of the object parts is then used to characterize the object. Then, targets are detected by searching for similar visual words and spatial distributions. Object detector uses pose estimation and Q-learning computer vision technique. However, these techniques are not suitable for region-based threat detection. For the object detection task within X-ray security imagery, Akcay et al. [9] compared and evaluated four localization strategies sliding window—CNN classification, faster R-CNN, R-FCN, and YOLOv2. Results proved that strategies with a region proposal network, i.e. faster R-CNN and R-FCN performed better, while the worst performer was SW-CNN due to the lack of bounding box regression. It would, therefore, seem that the addition of a bounding box regression layer would boost performance and it is significant to the proposed solution.

The study by Caldwell et al. [19] investigated whether a CNN trained for ATD in one transport domain could be used in similar domains. Authors first present an ontology of X-ray images obtained during security screening to help in identifying types of changes by transfer learning. Then, they propose two distinct frameworks for ATD, i.e. training transfer and testing transfer. Results from the evaluation of these frameworks indicate that transformation of data to match with a pre-trained network is hard, whereas training a network using transformed data to is possible but training and testing with data are drawn from the same source performs better. In order to make use of data transfer in ATD application, transfer methods proposed the need to improved or different networks that learn generalizable representations should be introduced, and it is not significant to the proposed solution. Rogers et al. [20] investigated the use of CNN trained-from-scratch on complex dual-energy X-ray cargo images to detect small metallic threat (SMT). They have improved their prior work [21] by examining methods of cargo material discrimination, with three new variants and their performance when they are fed into the network through separate input channels. Results showed an 8.4% in the rate of detection and a 2.5% improvement in area under the curve (AUC) of precision/recall. The hypothesis

suppresses false-positives that CNN is capable of implicitly learning the material discrimination from the raw dual-energy images. This is unsuitable for the proposed solution as it works with dual-energy X-ray.

Zhao et al. [3] have done a systematic review of the frameworks for generic object detection architectures. As the result of the experimental analysis, they conclude as to region proposal-based frameworks, such as faster R-CNN and R-FCN that generates region proposals at first and then classifies each proposal into different object categories perform better than regression models. The comprehensive study of region proposal-based object detection techniques contributes to propose the region proposal network for the proposed threat detector. Region-based convolutional neural network establishes the relationship between image classification and object detection [22]. Taking this into consideration, Chen et al. [23] proposed an enhanced region proposal network that improved convolutional features output and used PSO-SVM as the classifier. The proposed model achieved a detection speed of 5.8 frames per second with the superior mAP than R-CNN of Girshick et al. [22]. However, the classification loss could be strengthened further to improve performance. As the solution provides a region-based convolutional neural network for object detection, it would be significant to the proposed threat detection system. Many studies used loss function with deep learning [24].

Islam et al. [25] presented an algorithm for passenger tracking at the airport with a deep learning approach. Optical flow detectors are combined with a deep-learning detector trained on overhead passenger images. A basic estimate of the passenger location provided by the flow-based method is refined by a fine-tuned version of faster R-CNN used in [26]. The proposed algorithm had the probability of detecting passenger (95.00%) and the probability of false alarm (7.5%). The false alarm rate is still unacceptable. Therefore, the proposed algorithm is unsuitable for the proposed threat detection system. Suhao et al. [27] investigated the application of CNN in vehicle target type detection and recognition. Faster R-CNN model is applied for detection. The model used pre-trained CNN (ZF/VGG16) with improved the objective function reducing multitasking loss of Fast R-CNN [28]. The model outperformed in terms of average target detection accuracy and detection rate when compared to Fast R-CNN. However, the model failed to improve the quality of the feature map that plays a significant role in detecting small objects. Therefore, it is not suitable for the proposed threat detection system.

2.1 State of the art

This section presents the good features of the current solution (highlighted inside broken blue lines in Fig. 2) and limitations (highlighted inside broken red lines in Fig. 2). Ren et al. [6] Proposed an object detection system called faster R-CNN by enhancing Fast R-CNN with regional proposal network (RPN) to generate high-quality region proposals. The model uses translation-Invariant and multi-scale anchors which are vital components for sharing features between the networks [6]. They merged region proposal network and Fast R-CNN into a single network by sharing their convolutional features [6]. The model provides better accuracy with

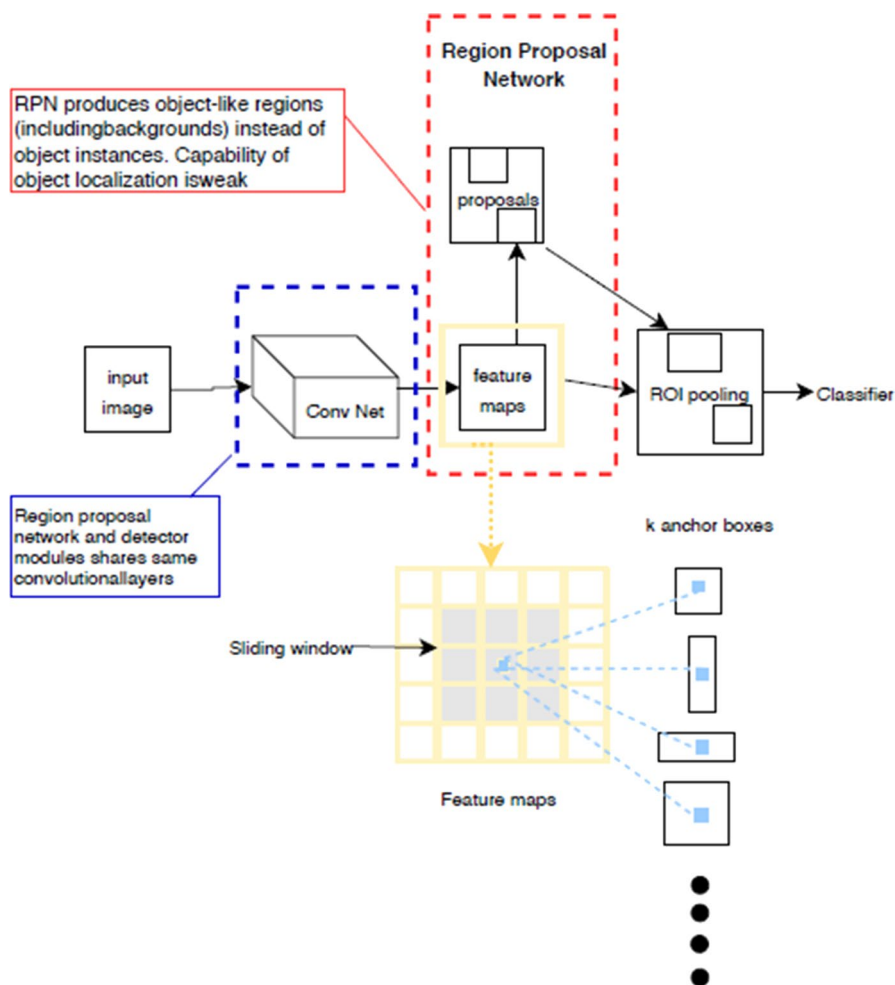


Fig. 2 Block Diagram of state of the art (faster R-CNN) [6]. [The blue borders show the good features of the state-of-the-art solution, and the red border refers to the limitation of it] (color figure online)

greater detection mAP than its state of the art with a processing time of 5–17 frames per second, respectively. This system consists of four stages, as shown in Fig. 2: pre-trained CNN module, region proposal network module, training RPN module, and training detector module.

2.1.1 Pre-trained CNN module

This work investigates two CNN: Zeiler and Fergus model (ZF) and Simonyan Zisserman model (VGG-16). ZF net has five shareable convolutional layers and VGG-16, which has 13 shareable convolutional layers that are shared among the RPN and Fast R-CNN modules [6].

2.1.2 Region proposal network module

RPN takes an image (of any size) as input and outputs a set of rectangular object proposals, with an objectness score [6]. In order to create region proposals, a minor network slides over the convolutional feature map output of the last convolutional layer shared by pre-trained CNN module. This small network takes as input an $n \times n$ spatial window of the input convolutional feature map. Later each window is mapped to a lower dimensional feature and sent to a box-regression layer and a box-classification layer [6]. Multiple region proposals are predicted simultaneously at each sliding-window location. Maximum possible proposals for each location is denoted as k . Reg layer's task is to encode the coordinates of k boxes, while the cls layer outputs scores that estimate the probability of object or not object for each proposal. Then, the k number of proposals are parameterized relative to k reference boxes, called anchors [6].

Anchors are centred at sliding window. The module also uses a translation-invariant feature to reduce the model size, i.e. a function translates an object in an image and the proposal [6]. The function also predicts the proposal in either location. The module uses multi-scale anchors as regression references to address multiple scales. These classify and regress bounding boxes with reference to anchor boxes of multiple scales and aspect ratios. Multi-scale design based on anchor allows the convolutional features computed on a single-scale image to be shared among the region proposal network and Fast R-CNN modules. RPN module minimized objective function following the multitask loss in Fast R-CNN. Uses a loss function to measure the error of classifier and box regressor [6].

But this module suffers from the low resolution of the top-level feature map that reduces the quality of region proposals. Also, the scale design and aspect ratios of anchor boxes are not ideal. Therefore, the ability of object localization is weak. Low quality of region proposals leads to inefficient object detection. Especially, small objects are difficult to be detected. With poor object localization capability, RPN produces object-like regions (including backgrounds) instead of object instances [23].

2.1.3 Training RPN module

This stage used Image-centric sampling strategy to train RPN. The loss function of a mini-batch was computed using a random sample of 256 anchors in an image. New layers were initialized by drawing weights from a zero-mean Gaussian distribution with a standard deviation of 0.01. The shared convolutional layers were initialized by pre-training a model for ImageNet classification [6].

2.1.4 Training detector module

Detector module is responsible for detecting the object using the proposed regions of RPN module. The module uses a four-step alternating training algorithm to learn shared features via alternating optimization [6]. RPN is trained as in the previous stage, and then the Fast R-CNN trains the detection network using

the generated region proposals. Later, RPN training is initialized by detector network; this fixes the shared convolutional layers while fine-tuning only the layers unique to RPN. Finally, the unique layers of Fast R-CNN are fine-tuned.

The state-of-the-art model improved detection accuracy by achieving higher mAP with a processing time of 5–17 frames per second [6]. The loss function is implemented in region proposal network module to measure the classification error and regression error, as shown in Eq. (1). However, accuracy can be increased by strengthening the classification loss, and processing time can be reduced.

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

i : index of an anchor
 p_i : predicted probability
 $p_i^* \begin{cases} 1 & \text{for +ve anchor} \\ 0 & \text{for -ve anchor} \end{cases}$
 N_{cls} : normalization term, number of anchors in mini batch
 L_{cls} : log loss over two classes(object versus not object)
 λ : constant value
 N_{reg} : normalization term, number of total anchors
 $p_i^* L_{reg}$: regression loss is activated only for positive anchors
 t_i : predicted box
 t_i^* : ground truth box
 $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$

(1)

The classification loss function $L_{cls}(p_i, p_i^*)$ does not focus on training multi-object categories and negative samples. The regression loss function is calculated as $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$, where R is the robust loss function (smooth L1), as shown in Eq. (2) [6].

$$\text{smooth } L_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases} \quad (2)$$

Bounding box regression adopts the parameterizations of the four coordinates as below:

$$\begin{aligned}
 t_x &= (x - x_a)/x_a, t_y = (y - y_a)/h_a, \\
 t_w &= \log(w/w_a), t_h = \log(h/h_a), \\
 t_x^* &= (x^* - x_a)/x_a, t_y^* = (y^* - y_a)/h_a, \\
 t_w^* &= \log(w^*/w_a), t_h^* = \log(h^*/h_a),
 \end{aligned}$$

where x , y , w , and h denote the box's centre coordinates and its width and height. Variables x , x_a , and x^* are for the predicted box, anchor box, and ground truth box, respectively.

Table 1 presents pseudocode for the RPN method flow to generate region proposals, and Fig. 3 presents the logical flow diagram of the region proposal network in faster R-CNN architecture.

In conclusion, the security screening algorithm has been developed by the combination of techniques studies in the literature. The pre-processing module of the proposed system makes use of image manipulation and quality improvement techniques. Inspired by faster R-CNN by Ren et al. [6]. The proposed solution employs an improved faster R-CNN with quality feature maps and fine-tunes it to work in this subdomain. Region proposal network of the proposed faster R-CNN uses features of an enhanced region proposal network by Chen et al. [23] while addressing its limitation in RPN loss function.

The proposed system solves these gaps in literature through the following contribution; a pre-processing module to prepare the image data, DE convolutional feature pyramid network to produce enhanced feature maps to increase the ability in

Table 1 Pseudocode for the RPN *method*

<p>Algorithm: RPN method to generate region proposals</p> <p>Input: An image (of any size) as input and pre-trained CNN network on image classification tasks</p> <p>Output: Set of rectangular object proposals, each with an objectness score</p>
<p>BEGIN</p> <p>Step 1: Initialize RPN with pre-train a CNN, obtaining conv feature map</p> <p>Step 2: Slide a small $n \times n$ spatial window over the conv feature map of the entire image.</p> <p>Step 3: Predict multiple region proposals simultaneously at the center of each sliding window.</p> <p style="padding-left: 40px;">(maximum possible proposals $\sim k$)</p> <p>Step 4: box-regression layer encodes the coordinates for k boxes</p> <p>Step 5: box-classification layer outputs $2k$ scores to estimate the probability of object or not for each proposal.</p> <p>Step 6: The k proposals are parameterized relative to k reference boxes(anchors)</p> <p style="padding-left: 40px;">For each location, k ($k=9$) anchor boxes are used (3 scales of 128, 256 and 512, and 3 aspect ratios of 1:1, 1:2, 2:1) for generating region proposals.</p> <p>Step 7: The classification loss over 2 classes and regression loss over bounding boxes are calculated</p> <p>Step 8: Pre-check which location contains an object.</p> <p>Step 9: Pass the corresponding locations and bounding boxes to detection network</p> <p>END</p>

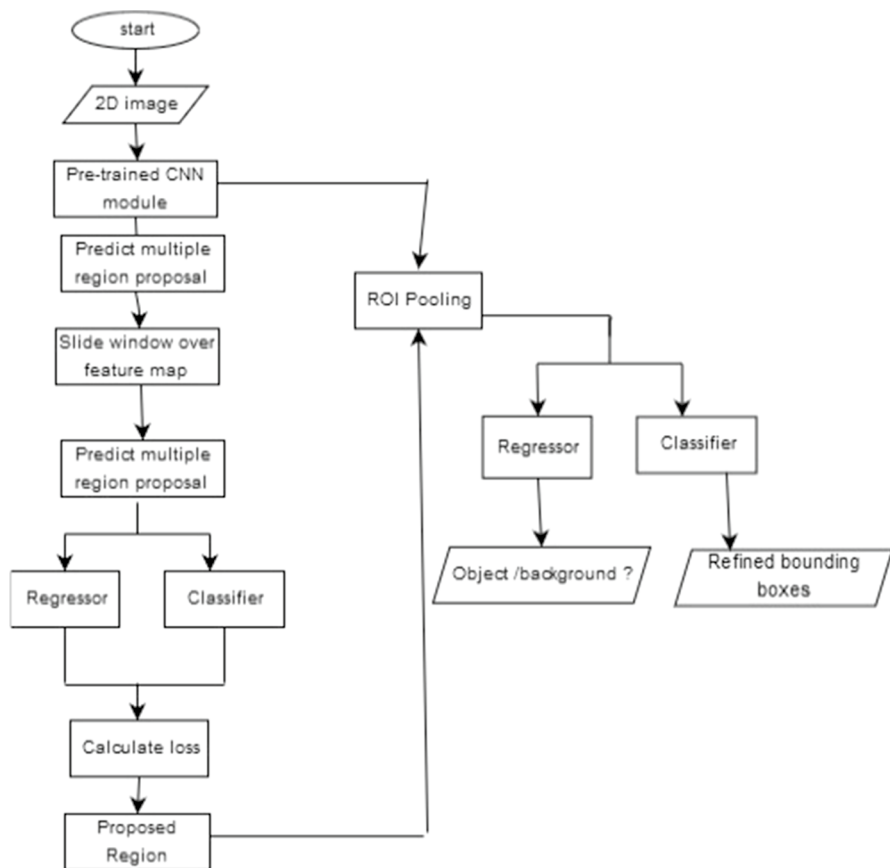


Fig. 3 The logical flow diagram of faster R-CNN algorithm (color figure online)

detecting small objects, and novel anchor boxes design to increase object localization capability of the system. Thus, the proposed system will have improved detection accuracy and performance.

3 Proposed system

After reviewing a range of object detection techniques for threat detection in security screening, benefits and drawback of each technique were analysed. The main issues to be considered were threat localization, accuracy, and processing time.

From the collected list of approaches in the review, the best one by Ren et al. [6] was selected as the basis for our proposed solution. The region proposal network module in the best solution is responsible for better object localization and faster iteration. Region proposal module generates region proposals as to where to look for objects. Module shares conv feature map with detector module and outputs a group

of rectangular boxes as object proposals, with each box having an objectness score to the detector module. Thus, a region proposal helps to localize the object. Detection networks mainly rely on additional methods that generate a candidate pool of isolated region proposal for localizing the object, which improves detection accuracy [3]. Secondly sharing of full-image convolutional features with the detector module reduces the region proposal computation bottleneck in improving efficiency [28]. Along with this, the proposed solution improves the region proposal network by introducing other features. Being inspired by the second-best solution, a work by Chen et al. [23], the proposed solution implements DE convolutional feature pyramid to improve the quality of region proposals and a novel anchor boxes design increases the capability of object localization. These features promote the performance of small object detection, accelerates object detection speed with only 200 top-ranked proposals and improves the detection accuracy by more than 4% mAP [23]. The proposed solution also incorporates a modified cross-entropy loss function that strengthens the classification loss of the CNN network.

The proposed system consists of five major modules (Fig. 4) called pre-processing module, trained CNN module, region proposal network module, training RPN module, and training detector module.

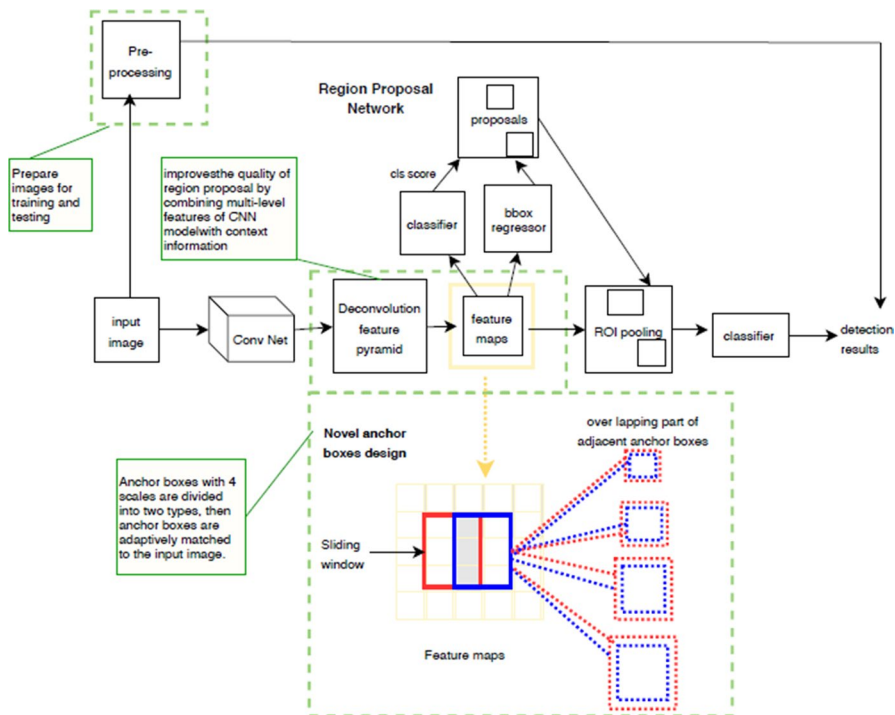


Fig. 4 Block diagram of the proposed threat detection system for security screening using enhanced RPN. [The green borders refer to the new parts in the proposed system] (color figure online)

3.1 Pre-processing module

This module resizes the image to be fed to the CNN module. Here, images are resized to fit the input layer of the CNN. The pre-processing module is fused with the detection module for locating the threat, see Fig. 4.

3.2 Trained CNN module

Trained CNN model is used to share convolutional layers among the RPN and Fast R-CNN modules [6]. However, the type of pre-trained CNN used by state of the art cannot be used in this domain as it is trained to detect common objects in RGB images only. So, the proposed system will make use of the network structure similar to the pre-trained module with modifications train the images specific to the domain. The CNN structure consists of convolution layer, relu layer, max-pooling layer, fully connected layer, SoftMax layer, and a classification layer. In addition, this basic CNN structure, DE convolutional feature pyramid network (DFPN) is introduced to enhance the top-level features with the context information. A DE convolutional layer is introduced to integrate lower feature maps with the higher feature maps and to extract more semantic features [23]. With the addition of a batch normalization layer after each convolutional layer, convolutional features are compressed into a uniform space. A synthesized pooling method, including max pooling and average pooling strategies, is introduced to the output concatenation features [23]. DFPN produces feature maps with rich information improving the quality of object proposals, thus increasing the ability to detect small threats (Fig. 5). See Table 2.

Table 2 Pseudocode for the enhanced RPN method to generate region proposals

Algorithm: Enhanced RPN method to generate region proposals
Input: An image as input and pre-trained CNN network on image classification tasks
Output: Set of rectangular object proposals, each with an objectness score
BEGIN
Step 1: Initialize RPN with pre-train a CNN, obtaining conv feature map
Step 2: Apply DFPN to integrate lower feature maps with the top-level feature maps.
Step 3: Generate novel anchor boxes based on the output features.
Step 4: box-regression layer encodes the coordinates for anchor boxes
Step 5: box-classification layer outputs scores to estimate the probability of an object or not for each proposal.
Step 4: The classification loss over 2 classes and regression loss over bounding boxes are calculated (Using the modified classification loss function in Eq. (5))
Step 8: Pre-check which location contains an object.
Step 9: Pass the corresponding locations and bounding boxes to detection network
END

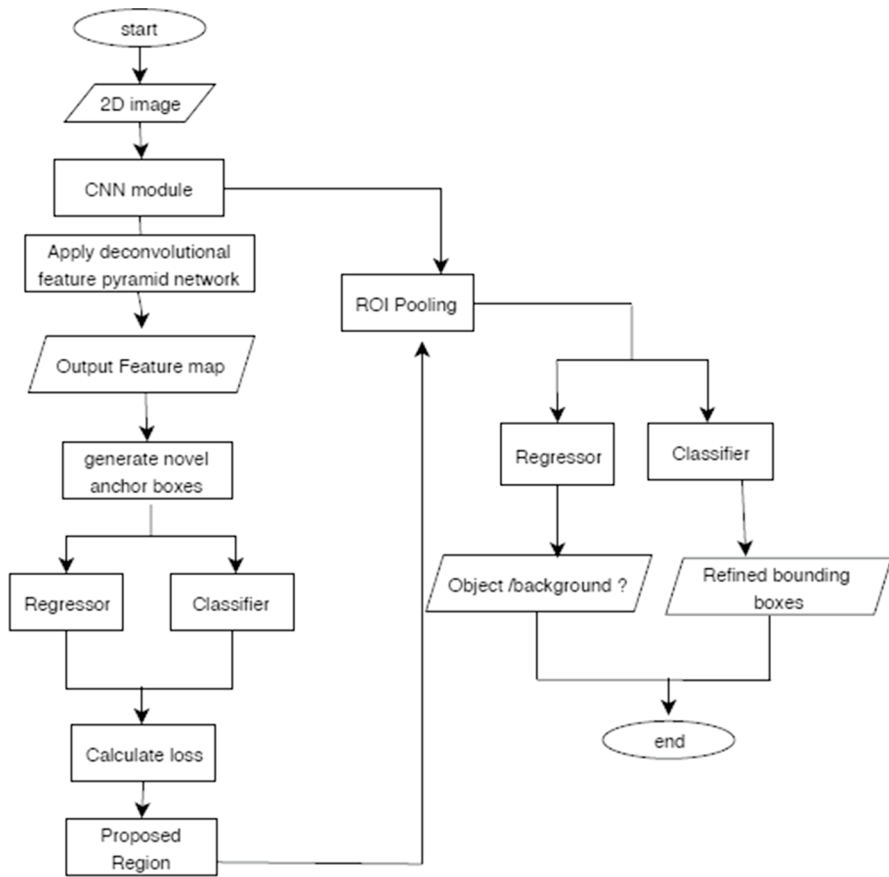


Fig. 5 The logical flow diagram of the proposed enhanced RPN method to generate region proposals

3.3 Region proposal network module

This module takes an image as input and outputs a group of rectangular boxes as object proposals with an object score for each proposal [6]. These proposals are generated through sliding windows over feature map output by the DE convolutional feature pyramid network introduced to trained CNN module. Each of these sliding windows is mapped to a lower dimensional feature and fed into a box-regression layer and a box-classification layer. At each sliding-window location, multiple region proposals are predicted simultaneously.

Anchors are centred at sliding window. State-of-the-art solution generated nine anchor boxes at each sliding position and nearly 20 k anchor boxes in total for a

convolutional feature map of a size $W \times H$ [4]. Translation-invariant feature was used to reduce model size. The proposed design of novel anchor boxes introduces anchor boxes with four scales which are divided into two types [23]. These two types of anchor boxes scales are interspersed for each 2×2 sliding window, as shown in Fig. 4. This design considers the relationship between the aspect ratio of the feature map and the aspect ratio of each anchor box. Anchor boxes are adaptively matched to the input image to improve the ability of object localization also increase the performance of object detection [23]. RPN module uses a loss function to measure the error of classifier and bounding box regressor [6]. Proposed system improves classification loss function by considering the influence of negative samples on the system. Balancing of the training classes will strengthen the classification loss.

3.4 Training RPN module

This stage uses an image-centric sampling strategy to train RPN. New layers will be initialized by drawing weights from a zero-mean Gaussian distribution with a standard deviation of 0.01 [6]. The trained model will initialize the shared convolutional layers from stage 2 for classification.

3.5 Training detector module

Detector module detects the threat object in an image using the proposed regions of enhanced RPN module. The module uses a four-step alternating training algorithm to learn shared features via alternating optimization [6].

An equation was introduced for the proposed system to enhance the performance of the neural network, and it is discussed in the following section.

3.6 Proposed equation

Minimized objective function following the multitask loss in Fast R-CNN and the loss function that is a combination of classification and regression loss, the classification loss uses a cross-entropy function that predicts the probability a threat presents and the regression loss function uses smooth L1 function [6].

Based on the modification that has been made on Eq. (1), this paper introduced an *Enhanced Loss Function (EL)* as it is illustrated in Eq. (3).

$$EL(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i ML_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

i : index of an anchor
 p_i : predicted probability
 $p_i^* \begin{cases} 1 & \text{for +ve anchor} \\ 0 & \text{for -ve anchor} \end{cases}$
 N_{cls} : normalization term, number of anchors in mini batch
 ML_{cls} : modified log loss over two classes(object versus not object)
 λ : constant value
 N_{reg} : normalization term, number of total anchors
 $p_i^* L_{reg}$: regression loss is activated only for positive anchors
 t_i : predicted box
 t_i^* : ground truth box
 $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$.

(3)

The objects or not objects are classified by a log loss L_{cls} , and it is defined by Eq. (4) [23]. However, this function cannot handle the imbalanced data. Therefore, classification function is modified to make the classifier aware of the imbalanced data by incorporating the weights of the classes into the objective function. The modified function (ML) is used to improve the balance of training samples, as shown in Eq. (5).

$$L_{cls}(p_i, p_i^*) = -p_i^* \log(p_i) - (1 - p_i^*) \log(1 - p_i)$$

$$L_{cls}(p_i, p_i^*) = \begin{cases} -\log(p_i) & \text{if } p_i^* = 1 \\ -\log(1 - p_i) & \text{if } p_i^* = 0 \end{cases}$$
(4)

$$ML_{cls}(p_i, p_i^*) = -w_0 p_i^* \log(p_i) - w_1 (1 - p_i^*) \log(1 - p_i)$$
(5)

where w_0 and w_1 weights of the two classes; positive class and negative class, respectively.

The regression loss L_{reg} is defined by Eq. (6) [6].

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$$
(6)

where R is the robust loss function (smooth $L1$), as shown in Eq. (2) [6].

3.7 Areas of improvement

As we discussed in the literature review section, none of the available solutions that applied faster R-CNN for threat detection have considered the use of a weighted cross-entropy function to measure the predicted probability divergence from the actual label in threat detection. An equation was proposed to enhance the performance of the neural network; the modified binomial cross-entropy function is

combined with classification loss and regression loss. The proposed Eq. (5) calculates the classification loss by incorporating the weights of the classes into the function. With the help of this Eq. (5), balancing of the training samples is achieved, which in turn improves the overall accuracy by accurate classification. With the pre-processing module, original images are resized to fit the input layers of the CNN. Introducing DE convolutional feature pyramid network enhances the top-level features with the context information and produces a rich feature map. Novel anchor boxes designed with interspersed scales and adaptive aspect ratios increase object localization capability of the system. The state-of-the-art system has no anchor design with adaptive aspect ratios and rich top-level features map for quality region proposals. Rich feature maps produce high quality of region proposals, increasing the ability to detect small objects. With the improvement on anchor box design, objects are located correctly within the image with a lower processing time as only top-ranked proposals are considered. These features improve the detection accuracy and reduce processing time, see Tables 3, 4, 5, 6, 7, and Fig. 9.

The region proposal network does two types of predictions; they are binary classification and the bounding box regression. RPN makes use of all the anchors from the mini-batch to calculate the classification loss using through binary cross-entropy loss function. The idea behind optimizing cross-entropy loss is to measure the dissimilarity between two vectors, in this case, the predicted outputs and the labelled ones. The existing classification function of RPN [6] does not consider the weight of negative samples. However, in a typical scenario, the number of negative samples is more than the number of positive samples. Therefore, the influence of negative samples is significant in training samples for classification. So, it is vital to address this problem and obtain a balance between training samples. The proposed classification loss function in Eq. (5) addresses the issue of balancing the training samples by adding in the weights of the binary classes into the function. This promotes classification accuracy by strengthening the classification loss function, which in turn improve the overall accuracy of the detector.

4 Results and discussion

A system configured with Intel(R) Core™ i7 3630QM CPU @ 2.40 GHz and 4.0 GB RAM was used. The proposed system was implemented in Matlab R2018b with deep learning Toolbox™ and parallel computing Toolbox™. The state-of-the-art model was built on Matlab with a pre-trained VGG-16 [6]. However, this pre-trained network accepts only RGB images. Therefore, the proposed system used the structure of VGG-16 with modifications. Convolutional neural network of VGG-16 was fine-tuned by replacing final layers with new layers to learn and create feature map specific to the training data set.

Parallel computing Toolbox and CUDA-capable NVIDIA™ GPU were required for training of the detector. The proposed system was trained using `trainFasterRCNNObjectDetector` function in Matlab. The images used in training and testing were X-ray images obtained from GDXray, a public database by Mery et al. [29]. The proposed system was trained using 3/4th of the data set

Table 3 Accuracy and processing time results for detection of clearly visible pistols


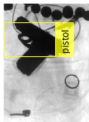
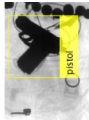

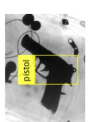
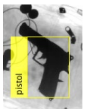

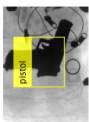
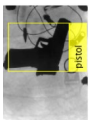
Sample no.	Original image	State of the art		Proposed solution			
		Processed sample	Localization Accuracy (in confidence score)	Processing time (second)	Processed sample	Accuracy (in confidence score)	Processing time (second)
1. Pistol clearly visible							
1.1			0.768	0.605		0.976	0.636
1.2			0.78	0.634		0.837	0.618
1.3			0.76	0.939		0.962	0.652

Table 4 Accuracy and processing time results for detection of a pistol with other objects








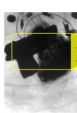

Sample no.	Original image	State of the art		Proposed solution			
		Processed sample	Localization accuracy (in confidence score)	Processing time (second)	Processed sample	Accuracy (in confidence score)	Processing time (second)
2. Pistol with other objects							
2.1			0.892	0.701		0.997	0.648
2.2			0.660 0.580	0.530		0.972	0.512
2.3			0.762	0.625		0.845	0.556

Table 5 Accuracy and processing time results for detection of pistol overlaid with an object










Sample no.	Original image	State of the art		Proposed solution			
		Processed sample	Localization accuracy (in confidence score)	Processing time (second)	Processed sample	Accuracy (in confidence score)	Processing time (second)
3. Pistol overlaid with object							
3.1			0.679	0.724		0.983	0.689
3.2			0.563	1.610		0.758	0.791
3.3			0.784	0.721		0.954	0.622

Table 6 Accuracy and processing time results for detection of images with 2 pistols



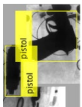
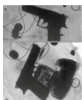
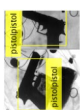
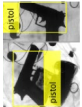


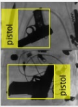

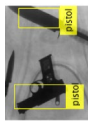






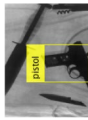
Sample no.	Original image	State of the art		Proposed solution			
		Processed sample	Localization accuracy (in confidence score)	Processing time (second)	Processed sample	Accuracy (in confidence score)	Processing time (second)
4. images with 2 pistols							
4.1			0.963	0.644		0.972 0.627	0.535
4.2			0.968 0.852	0.836		0.986 0.854	0.628
4.3			0.758	0.842		0.886 0.763	0.373

Table 7 Accuracy and processing time results for detection of pistol with a similar threat

Sample no.	Original image	State of the art		Proposed solution			
		Processed sample	Localization Accuracy (in confidence score)	Processing time (second)	Processed sample	Accuracy (in confidence score)	Processing time (second)
5. Pistol with similar threat							
5.1			0.734 0.503	0.736		0.945	0.456
5.2			0.750	0.663		0.940	0.486
5.3			0.810	0.770		0.935	0.380

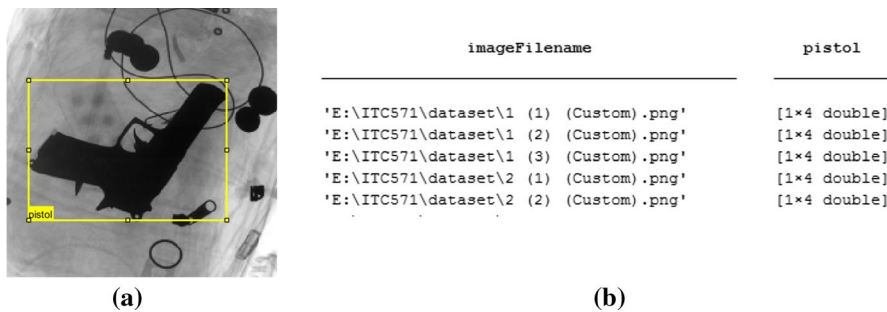


Fig. 6 Training samples from GDXray database [29]: (a) Image with a bounding box, (b) labels table for training

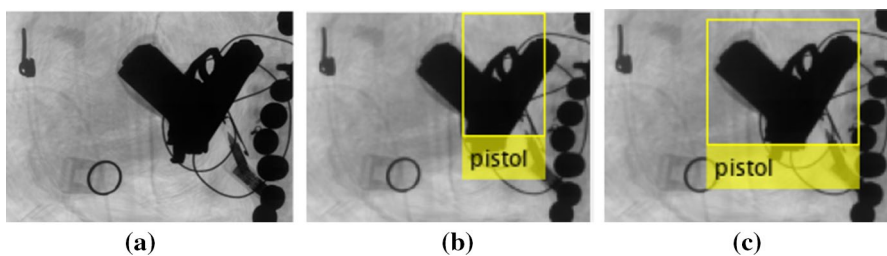


Fig. 7 Samples of detection system: (a) input image, (b) detection by the state of the art, (c) detection by the proposed system

while allocating 1/4th of the data set for testing purposes. The resolution of pistol images from the database was of 612 by 452 pixels and was resized to 153 by 113 pixels as more bits are required for a higher-quality image to be stored, transmitted, and processed. In our case, it is an acceptable level to suit the image input layer of the network structure. The proposed model was trained using images belonging to five different categories. Comprehensive samples of images from different angles and occlusion were used to test the model.

The CNN module is fine-tuned and trained with X-ray images. Sample training images with different occlusions used to train CNN. The CNN training module of the proposed system requires image labels with annotations. These images annotations for CNN training were created using the in-built image labeller app. Image labels and bounding boxes were given in the form of a table, as shown in Fig. 6a and b.

DE convolutional feature pyramid network extracts more semantic features to produce a feature map. Taking an input of 153×113 input image, DFPN outputs feature maps with a resolution of 50×50 . This feature map used to create anchor boxes. Region proposals are generated through the classification and regression operations. Region proposals are passed on to the region of interest pooling layer of the detection network to do final classification and regression. When comparing detection by the state of the art (Fig. 7b) and detection by the proposed

system (Fig. 7c), it is evident that proposed system's quality of region proposals has improved due to the DFPN and new design of anchor boxes.

Based on the implementation of the proposed system on the X-ray images data set, the sample pistols in images were detected and indicated. Sample results of state of the art and the proposed model are compared with the aid of graphs and the data reports. Test images for each category are obtained from baggage data sets from GDXray database of Mery et al. [29]. The results of the experiments on a few samples for threat detection are reviewed in Tables 3, 4, 5, 6, and 7. Comprehensive tests were carried out with three different test images for each category. The experiment was on the five pistol image categories: pistol clearly visible, pistol with other objects, pistol overlaid with an object, an image with two pistols, and a pistol with a similar threat.

Results tables tabulate the processing time and the localization accuracy of the detector models. The processing time is measured through performance measure function called tic-toc in MATLAB. The tic-toc function works by recording the internal time at execution of the tic command and displays the elapsed time with the toc command [30]. Thus, the time taken for the model to detect the threat is calculated. Further, the localization accuracy is measured by the confidence score in MATLAB, which represents how likely the model thinks this box really contains an object. The formula mentioned below is used to calculate the confidence score:

$$\begin{aligned} \text{Confidence score} &= p_r(\text{object}) * IoU \\ \text{where :} \\ p_r(\text{object}) &: \text{the probability that a box contains an object} \\ IoU &: \text{Intersection over union between the predicted box and the ground truth.} \end{aligned} \quad (7)$$

The standard deviation is a measure of how widely values are dispersed from the average value. The standard deviation of the processing time and confidence score is calculated as follows for the experimented samples:

$$\begin{aligned} s &= \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}} \\ \text{where :} \\ s &: \text{standard deviation of sample} \\ \bar{x} &: \text{sample mean} \\ x &: \text{values} \\ n &: \text{no of samples.} \end{aligned} \quad (8)$$

Estimated standard deviation based on test sample for processing time and localization accuracy of the proposed system is 0.116 and 0.068, respectively.

The overall accuracy of the faster R-CCN network is measured in terms of average precision returned by Matlab function evaluatesDetectionPrecision [31]. It works by comparing the detection results to ground truth data and returns the

precision statistics, i.e. precision, recall, and average precision. Precision and recall are calculated as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{Total Positive Results}} \quad (9)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}. \quad (10)$$

Figure 8 shows the precision vs recall curve plotted in Matlab for pistol detection in all image categories (pistol clearly visible, pistol with other objects, pistol overlaid with an object, an image with two pistols, and a pistol with a similar threat) in the test data set. The area under the precision–recall curve returns average precision (AP), i.e. the ability of the detector to make correct classifications. Figure 9 represents the average processing time of the detector for test data for each category. These results of detection system indicate that the proposed model has enhanced the localization accuracy through accurate region proposals, and besides reduced the processing time of the detector.

The reported results in Tables 3, 4, 5, 6, and 7 show the improvement in localization accuracy and processing time of the proposed solution compared to the state-of-the-art solution concerning threat detection. With the four-step training of RPN and detector, the proposed model enhances the processing time of the system on average by 0.19 s with the standard deviation of 0.116. The overall level of improvement in processing time is quantified by finding the difference between the average processing times consumed by each model to detect threats. Further, the localization

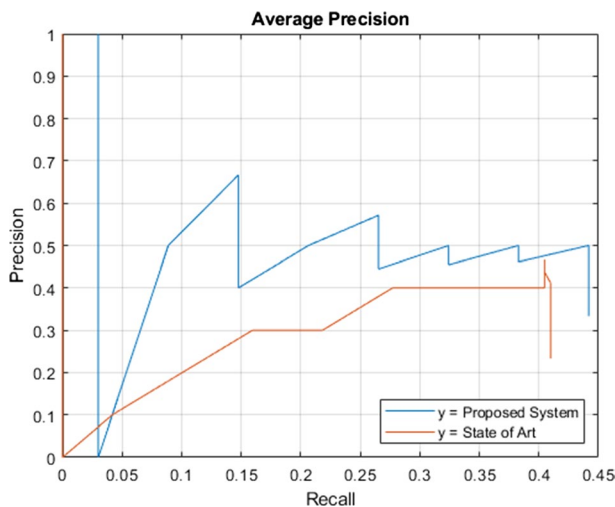


Fig. 8 Average precision results for proposed and state-of-the-art solution for threat detection. (a) Area under the blue line indicates the average precision by the proposed solution. (b) Area under the blue line indicates the average precision by the state of the art (color figure online)

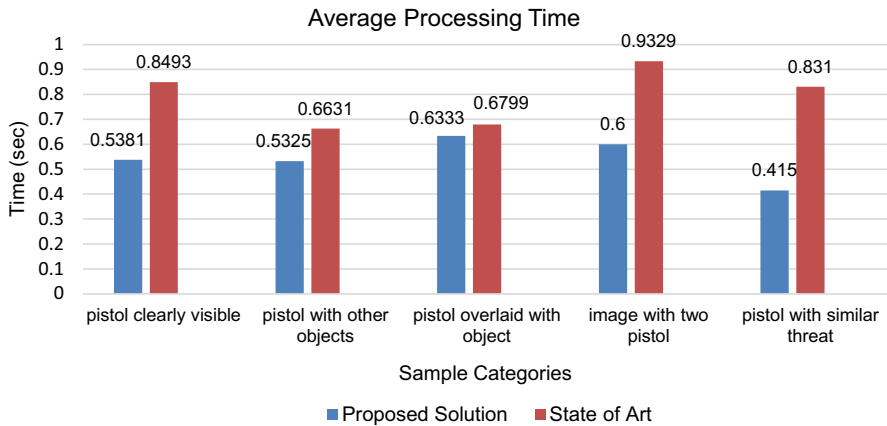


Fig. 9 Shows the processing time in seconds for the five different categories of pistol image detection. The blue colour indicates the processing time of the proposed solution, whereas the red colour indicates the processing time of the state-of-the-art solution. (a) First couple of bar graphs indicates the average processing time for pistol clearly visible images. (b) Second couple of bar graphs indicates the average processing time for a pistol with other objects. (c) Third couple of bar graphs indicates the average processing time for pistol overlaid with an object. (d) Fourth couple of bar graphs indicates the average processing time for an image with two pistols, and (e) Fifth couple of bar graphs indicates the average processing time for a pistol with a similar threat (color figure online)

accuracy of the proposed system is more than the state-of-the-art system. The state of the art obtains an average confidence score of 0.77 and is 77% confident about the threat in its region. Besides the proposed system obtains an average confidence score of 0.92, and it is 92% confident about the threat in its region with a confident score deviation of 0.068.

The proposed improvements to the feature map and anchor designs with adaptive aspect ratios and interspersed scales and have improved the accuracy of region proposals, thus leading to higher localization accuracy. These are evident from result displayed in Table 6 (4.1) and (4.3), state of the art fails to locate the second object as a pistol in the images with two pistols whereas proposed system detects both the pistols. Further, results from Table 4 (2.2) and (2.3) show that state of the art incorrectly locates the threat, by plotting a box over the other object as well, while proposed system correctly locates the threat identifying it as a pistol. The use of modified cross-entropy function has strengthened the classification loss by making classifier aware of the imbalanced data. This has enhanced the performance of the proposed system to classify it as a threat precisely. As shown in Table 7 (5.1), state of the art incorrectly detected another object as a pistol, whereas the proposed system accurately detects the pistol in its location. Thus, the system enhanced the detection accuracy and reduced the processing time of pistol detection in comparison with

the state-of-the-art model. In conclusion, the improved RPN of faster R-CNN has improved the threat detection by achieving better average processing time of 0.57 s and confidence score of 92% on average.

For the detection of threats in X-ray images, various techniques have been implemented. However, these techniques have been rapidly developed, based on the need for improved accuracy and processing time. This research has successfully solved the limitation of the current best solution with 572 ms against the 761 ms average processing time. The proposed solution also improves the accuracy of detection to 0.27 (average precision) against the current accuracy of 0.22 (average precision), this is supported by the implementation of the improvements to RPN network in the detector model. Furthermore, the proposed system displayed comparatively better accuracy and reduced processing time in all image categories, i.e. pistol clearly visible images, pistol with other objects, pistol overlaid with an object, an image with two pistols, and a pistol with a similar threat. Table 8 shows a comparison between the state-of-the-art and the proposed solutions.

5 Conclusion and future work

Accurate detection of threat is critical to security screening system as it provides information on where and what type of threat is concealed in the subject. Though faster R-CNN has been used in object detection, from the literature reviewed, it is evident that faster R-CNN has not been considered detection of threat in security screening. This research aims to improve the accuracy of threat localization and reduce the processing time of detection by using an enhanced faster R-CNN with improved region proposal network. The region proposal network is modified to increase object localization capability using a novel anchor box design. Improved RPN produces better quality feature map. Also, a modified cross-entropy function has been developed by introducing sample weights to the classification loss function, which strengthens the classification loss and improves the performance of multitasking loss function. The enhanced faster R-CNN tested in Matlab has shown to improve overall accuracy with an average precision of 0.27 and reduced the processing time by 0.19 s on average.

Future research may implement a threat detection system for passenger security screening by training this model with scan images of passengers taken from a larger data set with different angles.

Table 8 Comparative results of the state-of-the-art and proposed solution

Proposed solution		State-of-the-art solution
Name of the solution	Enhanced faster R-CNN-based threat detector	Faster R-CNN: towards real-time object detection with region proposal networks
Accuracy	Improved accuracy in terms of detection average precision (AP) greater than the state of the art in this domain	Improved accuracy in terms of detection mean average precision (mAP) than its state of the art. Providing mAP of 70.4
Processing Time	Decrease in average processing time by 2 s	Provides processing time of 5–17 frames per second
Proposed equation	Modified classification loss function: $ML_{cls}(p_i, p_i^*) = -w_0 p_i^* \log(p_i) - w_1 (1 - p_i^*) \log(1 - p_i)$	Loss function: $E(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$
Contribution 1	DE convolutional feature pyramid network enhanced the top-level features with the context information to produce feature maps, thus increasing the ability to detect small objects	Context information which contains highly semantic contents is not considered by the state-of-the-art system
Contribution 2	Novel anchor boxes designed increase object localization capability of the system; helps in determine the category of object and its position in an image	The state-of-the-art system does not provide anchor design with adaptive aspect ratios and rich top-level features map for quality region proposals

References

1. Bhattacharyya A, Lind CH, Shirpurkar R (2018) Threat Detection in TSA Scans using AlexNet
2. Guimaraes A, Tofighi G (2018) Detecting zones and threat on 3D body in security airports using deep learning machine. CoRR. <https://doi.org/10.5281/zenodo.1189345>
3. Zhao Z-Q, Zheng P, Xu S-T, Wu X (2018) Object detection with deep learning: a review. CoRR 14(8):3212–3232
4. Karami E, Shehata M, Smith A (2017) Image identification using SIFT algorithm: performance analysis against different image deformations. arXiv preprint arXiv:1710.02728
5. Bay H, Tuytelaars T, Van Gool L (2006) SURF: speeded up robust features. In: Leonardis A, Bischof H, Pinz A (eds) Computer vision—ECCV. Springer, Berlin, pp 404–417
6. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149
7. Jaccard N, Rogers TW, Morton EJ, Griffin LD (2017) Detection of concealed cars in complex cargo X-ray imagery using deep learning. J X-ray Sci Technol 25(3):323–339. <https://doi.org/10.3233/XST-161199>
8. Jaccard N, Rogers TW, Morton EJ, Griffin LD (2015) Using deep learning on X-ray images to detect threats In: Proceedings Cranfield Defence and Security Doctoral Symposium, 2015, pp 1–12
9. Akcay S, Kundegorski ME, Willcocks CG, Breckon TP (2018) Using deep convolutional neural network architectures for object classification and detection within X-Ray baggage security imagery. IEEE Trans Inf Forensics Secur 13(9):2203–2215. <https://doi.org/10.1109/TIFS.2018.2812196>
10. Akçay S, Kundegorski ME, Devereux M, Breckon TP (2016) Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery. In: IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA 2016, pp 1057–1061, <https://doi.org/10.1109/ICIP.2016.7532519>
11. Svec E, Mery D, Arias M, Rizzo V, Saavedra JM, Banerjee S (2017) Modern computer vision techniques for x-ray testing in baggage inspection. IEEE Trans Syst Man Cybern Syst 47(4):682–692
12. Szegedy C et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp 1–9
13. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. <https://doi.org/10.1145/3065386>
14. Sánchez J, Perronnin F (2011) High-dimensional signature compression for large-scale image classification. In: Computer Vision and Pattern Recognition (CVPR), 2011: IEEE pp 1665–1672
15. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, 2012, pp 1097–1105
16. Rizzo V, Mery D (2016) Automated detection of threat objects using adapted implicit shape model. IEEE Trans Syst Man Cybern Syst 46(4):472–482
17. Rizzo V, Flores S, Mery D (2017) Threat objects detection in x-ray images using an active vision approach. J Nondestruct Eval 36(3):44. <https://doi.org/10.1007/s10921-017-0419-3>
18. Leibe B, Leonardis A, Schiele B (2008) Robust object detection with interleaved categorization and segmentation. Int J Comput Vision 7(1–3):259–289
19. Caldwell M, Ransley M, Rogers TW, Griffin LD (2017) Transferring x-ray based automated threat detection between scanners with different energies and resolution. Counterterrorism Crime Fight Forensics Surveill Technol 10441:104410F
20. Rogers TW, Jaccard N, Griffin LD (2017) A deep learning framework for the automated inspection of complex dual-energy x-ray cargo imagery. In: Anomaly Detection and Imaging with X-Rays (ADIX) II, vol 10187, p 101870L, May 2017. <https://doi.org/10.3233/XST-161199>
21. Jaccard N, Rogers TW, Griffin DL (2014) Automated detection of cars in transmission X-ray images of freight containers. In: 2014 International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2014: IEEE, pp 387–392
22. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp 580–587
23. Chen YP, Li Y, Wang G (2018) An enhanced region proposal network for object detection using deep learning method. PLoS One 13(9):e0203897. <https://doi.org/10.1371/journal.pone.0203897>

24. Maharjan S, Alsadoon A, Prasad PWC, Al-Dalain T, Alsadoon OH (2020) A novel enhanced softmax loss function for brain tumour detection using deep learning. *J Neurosci Methods* 330:108520. <https://doi.org/10.1016/j.jneumeth.2019.108520>
25. Islam A, Zhang Y, Yin D, Camps O, Radke RJ (2018) Correlating Belongings with Passengers in a Simulated Airport Security Checkpoint. In: *Proceedings of the 12th International Conference on Distributed Smart Cameras 2018*: ACM, p 14
26. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *CoRR*. arXiv:1409.1556
27. Suhao L, Jinzhao L, Guoquan L, Tong B, Huiqian W, Yu P (2018) Vehicle type detection based on deep learning in traffic scene. *Procedia Comput Sci* 131:564–572
28. Girshick R (2015) Fast r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision 2015: The Computer Vision Foundation*, pp 1440–1448
29. Mery D et al (2015) GDXray: The database of X-ray images for nondestructive testing. *J Nondestr Eval* 34(4):1–12
30. Start stopwatch timer—MATLAB tic. Mathworks.com. <https://www.mathworks.com/help/matlab/ref/tic.html>. Accessed 30 January, 2019
31. "evaluateDetectionPrecision." <https://www.mathworks.com/help/vision/ref/evaluatedetectionprecision.html>. Accessed 30 January, 2019

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.