Journal of Real-Time Image Processing (2021) 18:923–935
https://doi.org/10.1007/s11554-020-01051-1

ORIGINAL RESEARCH PAPER

An automated detection model of threat objects for X-ray baggage inspection based on depthwise separable convolution

Yiru Wei[1] · Zhiliang Zhu[1] · Hai Yu[1] · Wei Zhang[1]

Received: 30 March 2020 / Accepted: 19 November 2020 / Published online: 2 January 2021

Abstract
X-ray baggage inspection is an essential task to detect threat objects at important controlled access places, which can guard personal safety and prevent crime. Generally, it is carried out by screeners to visually determine whether or not a bag contains threat objects. Whereas, manual detection exhibits distinct shortcomings, from high detection errors to different detection results produced by screeners. These limitations can be addressed by introducing automated detection model of threat objects for X-ray baggage inspection. However, existing automated detection methods cannot realize end-to-end detection and the detection results include only classification without location. In this paper, we propose an automated detection model of threat objects based on depthwise separable convolution. Our model is able to not only categorize the threat object but also locate it simultaneously. The network model has the advantage of high detection accuracy, fast computational speed, and a few parameters. Meanwhile, the precision of threat object regions is enhanced with the help of multi-scale prediction. A deformation layer is added in our model, which can provide invariance to affine warping. The experiments on the GDXray database (Mery et al. in J Nondestr Eval 34(4):42, 2015) demonstrate that the overall performance of our proposed model is superior to YOLOv3 (Redmon J and Farhadi A in YOLOv3: an incremental improvement, 2018) model, SSD (Liu et al. in SSD: single shot multibox detector. In: European Conference on Computer Vision (ECCV), pp. 21–37, 2016) model, and Tiny_YOLO (Redmon et al. in You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2015) model.

Keywords X-ray baggage inspection · Detection model of threat objects · Depthwise separable convolution · Deep learning

1 Introduction

Baggage inspection through X-ray security inspection machine is an important task in public spaces and security checkpoints, which can guard personal safety and reduce the risk of crime [22, 32]. The main purpose of X-ray baggage inspection is to prevent forbidden items from passing the security checkpoints. Considering that threat objects are challenging to identify when being placed in cluttered package, detection of threat objects is an extremely complicated task [3].

When the bags pass through the X-ray security inspection machine, detection of threat objects is usually performed by screeners, who visually inspect the X-ray images and decide whether or not a bag has threat objects. The work of identifying a large number of different shapes, sizes, and substances (metals, and organic and inorganic substances) needs a great deal of concentration. Meanwhile, screeners have only a few seconds to recognize threat objects in a bag, the likelihood of missed and incorrect detection is comparably high over a long period of time. Although manual detection has been widely applied in the field of X-ray security inspection, this method mainly relies on the experience of screeners, which results in the fact that the accuracy cannot be assured and the detection results always vary among different screeners. Some literatures show that the detection performance is only 80–90% [20].

In addition to the manual detection method, automated detection methods of threat objects [1, 7, 8, 16, 18, 26, 30, 31] have already been developed for years. Among them,

✉ Zhiliang Zhu
zzl_neu@yeah.net

Yiru Wei
weiyiru0228@yeah.net

[1] Software Collage, Northeastern University, Shengyang 110819, China

early methods always use low-level features or various hand-crafted features to detect threat objects. Due to the problems of noise, occlusion, and clutter among others objects, appearance of threat objects become greatly difficult to comprehend. Therefore, automated detection of threat objects is far from satisfactory. Recently, deep learning-based methods [4, 6, 9, 13, 21, 28, 33] have shown promising results in image processing tasks. A detection method of threat objects based on deep learning is also proposed [19] in the past years. However, this method fails to train an ad-hoc detection model of threat objects with the GDXray database [17], but uses the existing model as generic feature extractor. In the testing stage, a simple nearest-neighbor classifier is used, which means that the label of a test image is that of its nearest neighbor in the training set.

Generally speaking, the existing detection methods of threat objects have the following defeats. (1) Manual detection of threat objects mainly relies on the experience of screeners, which consumes large amount of time and labor. Besides, the detection results of different screeners are not same. Therefore, it is difficult to ensure the robustness of this method. (2) Early automatic detection methods need hand-crafted features as the input, thus failing to meet the demand for automation. (3) Although detection model of threat objects based on deep learning can extract the features automatically, this method does not train an ad-hoc detection model of threat objects using X-ray images. Moreover, the computational speed cannot meet the real-time requirements of X-ray baggage inspection. (4) Detection of threat objects is considered as a classification problem. The detection results only include classification information without position information.

We, in a different way, explore the computational advantage of depthwise separable convolution, which can significantly reduce the amount of calculation and parameters without performance decrease. We compared three lightweight networks based on depthwise separable convolution on the GDXray database [17]. Therefore, we employ a lightweight backbone network: improved MobileNetV2 based on depthwise separable convolution as backbone to extract multi-scale features. The features in low layers contain more spatial information that can be used to recognize the objects' boundaries. The features in high layers retain more semantic information and thus can help locate the threat objects. Thus, the finer detection results can be obtained through multi-scale feature aggregation. Besides, we employ a deformation layer followed the backbone network to deal with the deformation of threat object in X-ray images.

Our main contributions are summarized as follows:

- We propose an automated detection model of threat objects for X-ray baggage inspection. Similar to YOLOv3 [25], we represent object as a boundary box; yet employ a lightweight backbone network: improved MobileNetV2 [29] to achieve real-time response.
- We utilize a deformation layer to enhance the robustness of our model to nonrigid deformation. Meanwhile, we employ multi-scale prediction to aggregate the multi-level features to generate the finer detection results.
- To evaluate the performance of our proposed model, a series of experiments are conducted on the GDXray database [17]. The experiment results show that our model can achieve better performance than YOLOv3 [25], SSD [15], and Tiny_YOLO [23].

2 Related work

The detection of threat objects in X-ray images is a challenging task, because the X-ray images are shadow images that are actually perspective projections of objects. For the recognition of threat objects from mono-energy X-ray images, methods like the adapted implicit shape model (ISM) based on visual codebooks [26], and adapted sparse representations (XASR+) [18] have been used. In the case of dual-energy X-ray images, Gabor texture features [31], bag of words (Bows) [1, 30], and support vector machines (SVM) classifiers [8] have been used. Algorithms based on 3-D features for 3-D objects recognition have been developed. For example, rotation invariant feature transform and scale-invariant feature transform (SIFT) descriptors [7], 3-D visual cortex modeling 3-D Zernike descriptors, and histogram of shape index [16]. In all of these traditional methods, detection of threat objects has two steps: (1) threat object characterization and (2) detection. Each class of threat objects is represented and detected separately. In conclusion, previous methods cannot realize end-to-end detection, which is difficult to meet the automation requirements of X-ray baggage inspection.

Due to the tremendous success of deep learning, especially the convolutional neural networks (CNNs), a detection model of threat objects based on CNNs has been proposed [19]. This model considers the detection of threat objects as a classification problem, which only exploits high-level features to detect the threat objects. The detection results only include classification information without location information. Moreover, this model includes several fully connected layers, which is computationally expensive and makes the speed much slower than real time.

Recently, CNN-based object detectors generally adopt anchor-based method to detect objects, which have exhibited superior performance for object detection. It is classified into two-stage detectors and one-stage detectors. The anchor-based two-stage detectors are proposed for better accuracy and efficiency, but it is computationally expensive. On the other hand, anchor-based one-stage detectors can accomplish detection and classification tasks on output feature maps at the same time. For this reason, one-stage detectors are widely regarded as the key to real-time detection task. Therefore, high accuracy and fast computational speed can be achieved by the combination of one-stage detectors and detection of threat object for X-ray baggage inspection.

3 Detection model of threat objects

It is widely known that one-stage detectors such as YOLO methods [23–25] and SSD [15] are widely applied to real-time detection tasks, which can directly predict object category and bounding box location. Among them, YOLOv3 [25] and SSD [15] have outstanding performances in objection detection, but the number of parameters and computational cost are large for our detection task of threat objects. Tiny_YOLO [23], as an improved version of YOLO [23], is a faster network model. However, its detection accuracy and generalization ability are unsatisfied. Accordingly, the existing one-stage detectors cannot achieve a good balance between detection precision and computational speed for detection task of threat objects. Therefore, this paper focuses on developing an automated detection model of threat objects for X-ray baggage inspection that has both high detection accuracy and fast computational speed.

3.1 Fine-tuning depthwise separable convolution model

Depthwise separable convolution is composed of depthwise (DW) convolution and pointwise (PW) convolution. The operation processes of DW convolution and PW convolution are shown in Fig. 1. To verify the computational advantage of depthwise separable convolution, we will compare the calculation amounts between the standard convolution and depthwise separable convolution. To be brief, the convolution kernel is $D_k \times D_k$, and the input and output channels are M and N, respectively. The resolution of input image is $W \times H$. The calculation amounts F_s in the standard convolution are defined as:

$$F_s = D_k \times D_k \times M \times N \times W \times H. \tag{1}$$

The calculation amounts F_{dw} of DW convolution can be represented as:

$$F_{dw} = D_k \times D_k \times M \times W \times H. \tag{2}$$

The calculation amounts F_{pw} of PW convolution can be represented as:

$$F_{pw} = M \times N \times W \times H. \tag{3}$$

Therefore, the calculation amounts F_{ds} of depthwise separable convolution can be represented as:

$$F_{ds} = F_{dw} + F_{pw} = D_k \times D_k \times M \times W \times H + M \times N \times W \times H. \tag{4}$$

The ratio of calculation amounts R between depthwise separable convolution and standard convolution is:

$$\begin{aligned} R = \frac{F_{ds}}{F_s} &= \frac{D_k \times D_k \times M \times W \times H + M \times N \times W \times H}{D_k \times D_k \times M \times N \times W \times H} \\ &= \frac{1}{N} + \frac{1}{D_k \times D_k}. \end{aligned} \tag{5}$$

According to Eq. (5), the calculation amounts in the depthwise separable convolution have been greatly reduced. Based on the computational advantage of depthwise separable convolution, it is employed in backbone sub-network for extracting features.

In off-the-shelf CNNs models, Xception [5], MobileNetV1 [10], and MobileNetV2 [29] all use depthwise separable convolution. In Sect. 4.4, three pre-trained networks on ImageNet [27] are used as backbone network for feature extraction and followed by three-scale prediction network. The three models are denoted as Xception-T, MobileNetV1-T, and MobileNetV2-T. We compare the performances of the three detection models on the GDXray database [17] in Table 1, from which when the MobileNetV2 [29] is used as backbone, the performance is best.

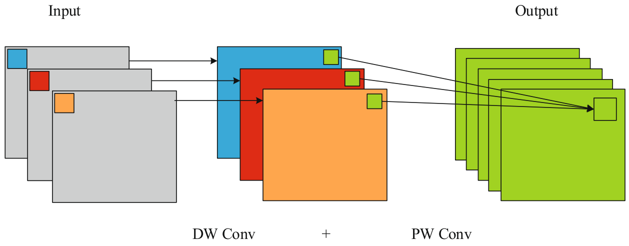

Fig. 1 Convolution process of depthwise separable convolution. It is composed of depthwise (DW) convolution and pointwise (PW) convolution. DW convolution is operated by channel-wise fashion; PW convolution is standard convolution with 1 × 1 kernels

3.2 The improved depthwise separable convolution model

The inverted residual structure is employed in MobileNetV2 [29], the principle of which is to expand the input channels and then get more feature maps in the depthwise convolution operation. Denoting the depthwise convolution with 3 × 3 filters as DWConv_3, standard convolution with 1 × 1 filters as Conv_1, and the activation function Relu6 with $Relu6$. Supposing x is input feature maps in inverted residual block, which is first processed by a 1 × 1 convolution and the activation

Table 1 The detection performance comparison among three detection models of threat objects that are different in backbone networks

Model	P (%)	R (%)	F_1 (%)	AP			mAP (%)
				Handgun (%)	Shuriken (%)	Razor blade (%)	
Xception-T	89.56	97.43	93.32	80.24	94.52	95.06	89.94
MobileNetV1-T	97.85	89.11	93.28	88.37	81.82	95.92	88.7
MobileNetV2-T	97.37	96.19	96.78	97.67	90.91	97.4	95.33

Three pre-trained networks on ImageNet [27] (namely Xception [5], MobileNetV1 [10]. and MobileNetV2 [29]) are used as backbone network for feature extraction and followed by three-scale prediction network. These three models are denoted as Xception-T, MobileNetV1-T, and MobileNetV2-T

function Relu6 to expand the number of channels. which can be written as:

$$x_1 = \mathrm{Relu6}(\mathrm{Conv}_1(x)). \tag{6}$$

Then, x_1 is processed by the depthwise convolution with 3×3 filters and the activation function Relu6, which can be represented as:

$$x_2 = \mathrm{Relu6}(\mathrm{DWConv}_3(x_1)). \tag{7}$$

When stride is 1, the output feature map can be represented as:

$$x_3 = x + \mathrm{Conv}_1(x_2). \tag{8}$$

When stride is 2, the output feature map can be represented as:

$$x_3 = \mathrm{Conv}_1(x_2). \tag{9}$$

As the depthwise convolution is operated by channel-wise fashion, the feature information can only be transferred in one channel. The activation function Relu6 is performed after the depthwise convolution, which would destroy the original information. We improve the inverted residual structure, as shown in Fig. 2. The activation function Relu6 after depthwise convolution is replaced by linear activation function, which can be described as:

$$x_2 = \mathrm{DWConv}_3(x_1). \tag{10}$$

Equation (7) is replaced by Eq. (10) in our improved inverted residual block. In Sect. 4.4, the improved MobileNetV2 and original MobileNetV2 are used as backbone and followed by three-scale prediction network, which are denoted as Im_MobileNetV2-T and MobileNetV2-T. We compare their performance on the GDXray database [17] in Table 2. The experimental results show that Im_MobileNetV2-T model can enhance the mAP by about 0.17% with respect to the MobileNetV2-T model.

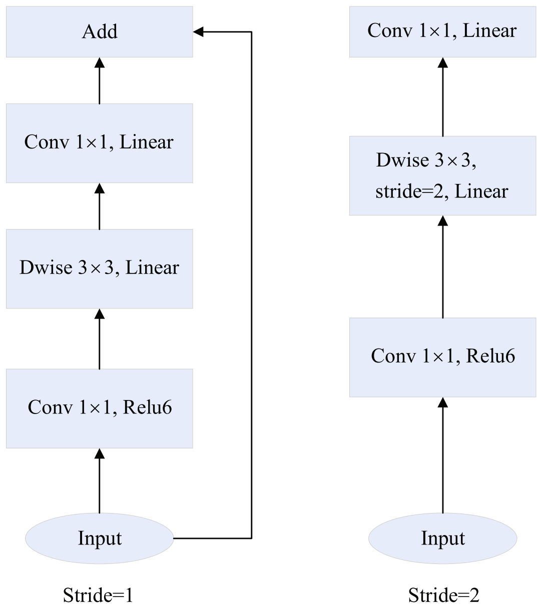

Fig. 2 The improved inverted residual structure: the Relu6 after depthwise convolution is replaced by linear

3.3 ThreNet: an ad-hoc detection model of threat objects based on depthwise separable convolution

To provide invariance to affine warping, a deformation layer [11] is inserted in backbone network, which learns an adaptive geometric transformation to act on input images or feature maps. The deformation layer learns to compute the parameters of an affine transformation which rotates, translates, scales, and crops an input image or a feature map to deal with object nonrigid deformations. A deformation layer consists of the localisation network, grid generator, and sampler, as shown in Fig. 3. The input feature map is first processed by the localisation network (a small CNN classifier network) that outputs a six-dimensional transformation parameter vector $\theta = \{\theta_1, \ldots, \theta_6\}$. The grid generator defines mapping relationships between input and output coordinate points by use of transformation matrix A_θ. $G_i = (x_i^t, y_i^t)$ is the coordinate point of output feature

Table 2 The improved MobileNetV2 is used as backbone and followed by three-scale prediction network, which is denoted as Im_ MobileNetV2-T

Model	P (%)	R (%)	F_1 (%)	AP			mAP (%)
				Handgun (%)	Shuriken (%)	Razor blade (%)	
MobileNetV2-T	97.37	96.19	96.78	97.67	90.91	97.4	95.33
Im_MobileNetV2-T	98.2	96.19	97.18	97.67	91.23	97.62	95.5

The performance comparison between the Im_MobileNetV2-T and the MobileNetV2-T

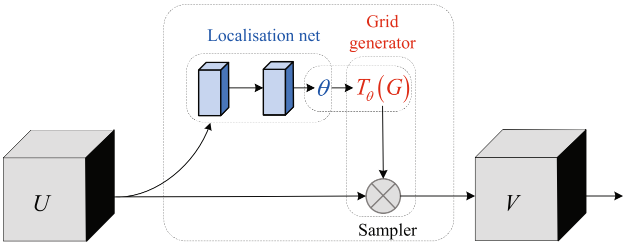

Fig. 3 The architecture of a deformation layer. It consists of the localisation network, grid generator, and sampler

map V. (x_i^s, y_i^s) is the coordinate point of input feature map U. The affine transformation T_θ is defined by:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = T_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}. \tag{11}$$

Finally, a sampler module uses bidirectional interpolation to obtain pixel values of the output feature map. This deformation layer can be either directly applied to the input images as a pre-processing layer or be inserted after one of the convolutional layers—to process the deformations of higher-level features.

Ultimately, our model—called ThreNet—consists of the improved MobileNetV2 network, a deformation layer, and three-scale prediction network. We tested the performance of different network architectures by varying the position of the deformation layer. According to the experiment results shown in Sect. 4.5, the best network architecture is shown in Fig. 4.

3.4 Multi-scale prediction

In the high layers of the detection network, the feature maps have rich semantic information and their receptive fields are large, which are suitable for detecting large objects. Whereas in the low layers, the feature maps have small receptive fields and are suitable for detecting small objects. Based on these properties, the feature pyramid network [14], as the output of the network, is employed to locate the threat objects regions on the last three layers to realize multi-scale prediction, as shown in Fig. 5.

After the multi-scale prediction, we must choose the best one from multiple boundary boxes. Soft-NMS algorithm (NMS) [2] is used to update the score for each boundary box, which can be formulated as:

$$s_i = \begin{cases} s_i, & IOU(H, b_i) < N_t \\ s_i(1 - IOU(H, b_i)), & IOU(H, b_i) \geq N_t, \end{cases} \tag{12}$$

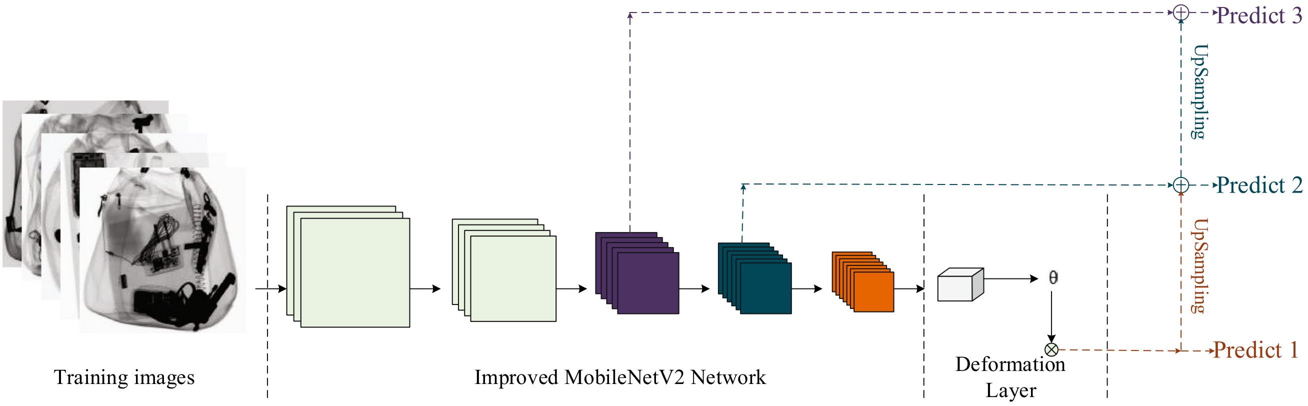

Fig. 4 Overview of the ThreNet architecture: combination of the improved MobileNetV2 network, a deformation layer, and three-scale prediction network. The improved MobileNetV2 is backbone network for feature extraction. A deformation layer is employed to deal with the nonrigid deformation of threat objects. The three-scale prediction network outputs three-scale feature maps: 13 × 13, 26 × 26 and 52 × 52

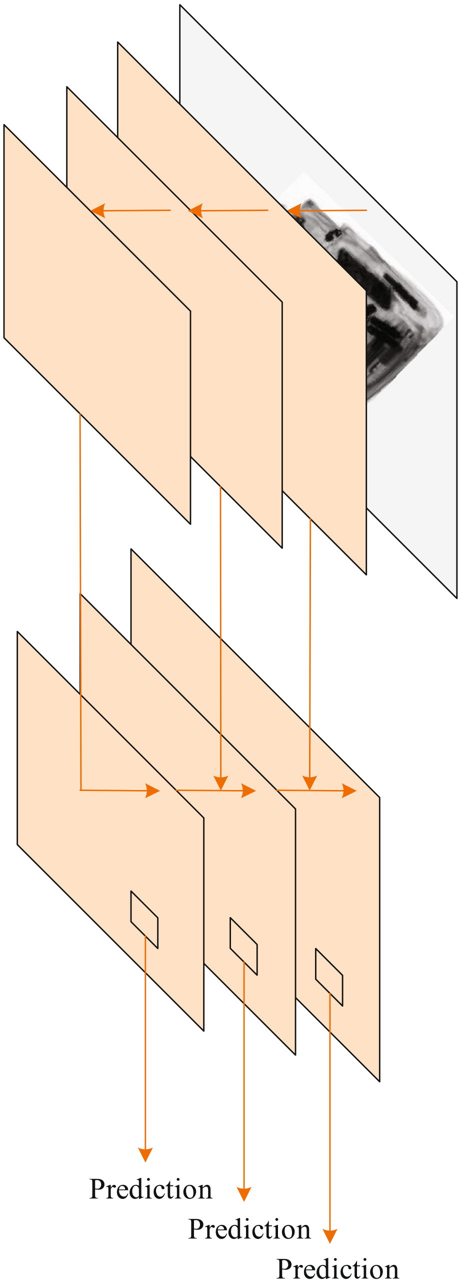

Fig. 5 Feature pyramid network for multi-scale prediction

where H is the boundary box that has the highest score, b_i is the initial detection boundary box, the threshold value of intersection over union (IOU) is set to N_t, and IOU can be expressed as:

$$IOU = \frac{\text{area}(A) \cap \text{area}(B)}{\text{area}(A) \cup \text{area}(B)}, \tag{13}$$

where A is the predicted boundary box and B is the ground truth boundary box.

When IOU is smaller than the threshold value N_t, the score of the boundary box remains unchanged. However, when IOU is larger than the threshold value N_t, the score of the boundary box is equal to the product between the score and $(1 - IOU)$.

3.5 Loss function

In this paper, our proposed detection model is also anchor-based one-stage detector, which uses features extracted from the entire input image to predict categories and positions of threat objects simultaneously. It also integrates different components into one single detection model. This means that our proposed model can realize end-to-end detection for real-time detection task of threat objects.

In our work, we use the K-means clustering algorithm to obtain the size of anchor boxes. A total of nine anchor boxes are obtained. The nine anchor boxes on the GDXray database [17] are: (29 × 127), (39 × 103), (52 × 111), (72 × 194), (105 × 197), (123 × 192), (172 × 188), (176 × 248), and (204 × 196). In terms of allocation, large anchor boxes (172 × 188), (176 × 248), and (204 × 196) are assigned to the 13 × 13 output feature maps (with the largest receptive field), which are suitable for detecting large objects. Anchor boxes of medium size (72 × 194), (105 × 197), and (123 × 192) are assigned to the 26 × 26 output feature maps (with the medium size of receptive field), which are suitable for detecting objects of medium size. Small anchor boxes (29 × 127), (39 × 103), and (52 × 111) are assigned to the 52 × 52 output feature maps (with the smallest receptive field), which are suitable for detecting small objects.

In Fig. 6, each-scale output divides the input image into $K \times K$ grids. Each grid cell predicts three bounding boxes that has the same shape as the assigned anchor boxes. We compute the IOU between each predicted box of each grid cell and the ground truth box. The bounding box with the largest IOU value is responsible for detecting the corresponding threat object. Prediction information of each bounding box consists of position offset (x, y), width w, height h, confidence score, and C class probabilities. Note that (x, y) is the center coordinate of the bounding box relative to the bounds of the grid cell. The width and height are relative to the whole image. The confidence score indicates whether a threat object is included in the bounding box. The confidence score can be expressed as $Pr(\text{Object}) \times IOU_{pred}^{truth}$. If there is no threat object in that cell, the confidence score is zero. Otherwise, the confidence score is equal to IOU between the predicted box and the ground truth box. The class probabilities can be represented as $P(\text{Class}_i \mid \text{Object})$, which is conditioned on the grid cell including a threat object. In our experiments with the GDXray database [17], the size of input image is set to 416 × 416 and the feature map sizes of three-scale output are set to 13 × 13,

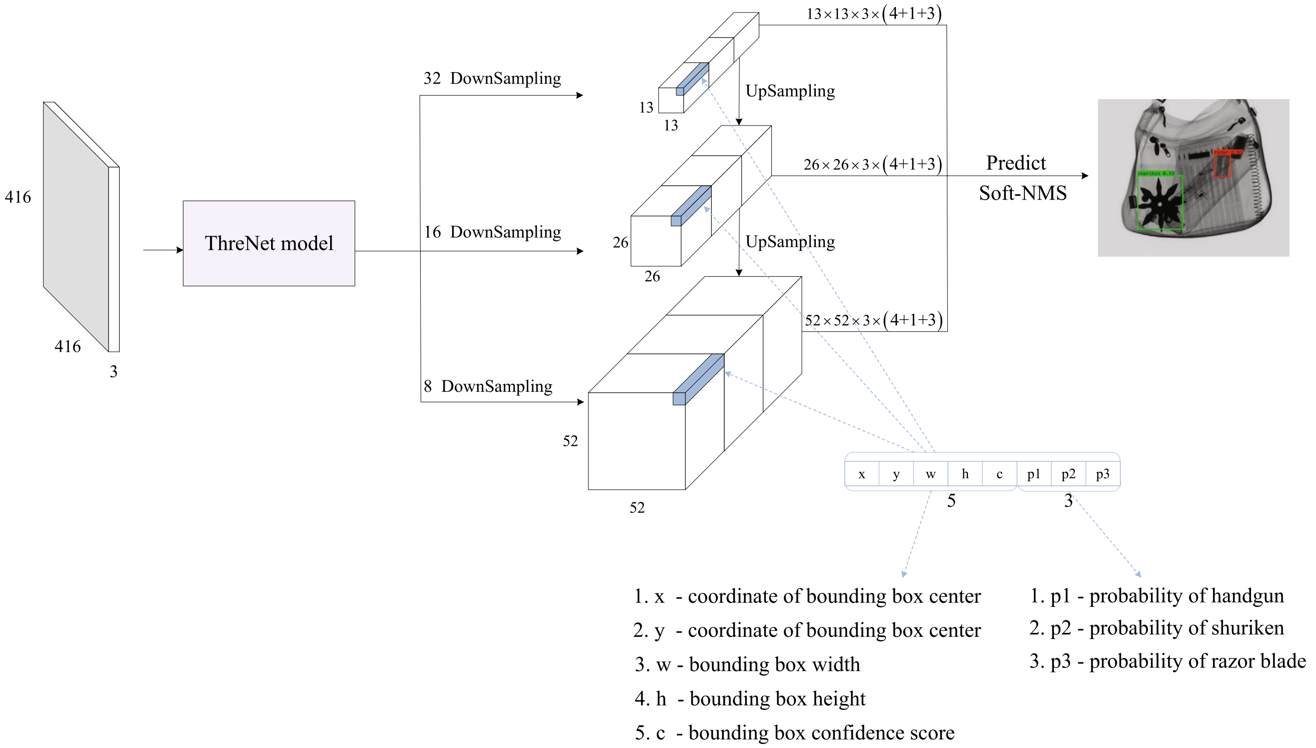
13×13×3×(4+1+3)
32 DownSampling
UpSampling
13
13
416
26×26×3×(4+1+3)
Predict
Soft-NMS
ThreNet model
16 DownSampling
26
26
UpSampling
416
3
52×52×3×(4+1+3)
8 DownSampling
52
52
x
y
w
h
c
p1
p2
p3
5
3
1. x - coordinate of bounding box center
2. y - coordinate of bounding box center
3. w - bounding box width
4. h - bounding box height
5. c - bounding box confidence score
1. p1 - probability of handgun
2. p2 - probability of shuriken
3. p3 - probability of razor blade

Fig. 6 Prediction process of bounding boxes of threat objects. Our proposed detection model (ThreNet) performs multiple convolutional layer with stride of 2 to down-sampling the input image with factor 8, 16, 32 and outputs three-scale candidate bounding boxes: 13 × 13 × 24, 26 × 26 × 24 and 52 × 52 × 24. Finally, Soft-NMS [2] is performed on the candidate bounding boxes to produce the detection results

26×26, and 52×52, respectively. Thus, we set K to 13, 26, 52 on three-scale output layers. There are three kinds of threat objects, so C is set to 3.

The loss function consists of the coordinate error, the confidence value error, and class probability error. The coordinate error can be expressed as:

$$
\begin{aligned}
coordErr = {} & \lambda_{cood} \sum_{i=1}^{K^2} \sum_{j=1}^{M} R_{ij}^{obj} \left[\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \right] \\
& + \lambda_{\text{cood}} \sum_{i=1}^{K^2} \sum_{j=1}^{M} R_{ij}^{obj} \\
& \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right].
\end{aligned} \tag{14}
$$

The confidence value error can be represented as:

$$
\begin{aligned}
confidErr = {} & \sum_{i=1}^{K^2} \sum_{j=1}^{M} R_{ij}^{obj} \left[\hat{C}_i \log\left(C_i\right) \right. \\
& \left. + (1 - \hat{C}_i) \log\left(1 - C_i\right) \right] \\
& + \lambda_{noobj} \sum_{i=1}^{K^2} \sum_{j=1}^{M} (1 - R_{ij}^{obj}) \left[\hat{C}_i \log\left(C_i\right) \right. \\
& \left. + (1 - \hat{C}_i) \log\left(1 - C_i\right) \right].
\end{aligned} \tag{15}
$$

The class probability error is shown as:

$$clsErr = \sum_{i=1}^{K^2} R_i^{obj} \sum_{c \in classes} \Big[\hat{p}_i(c) \log \left(p_i(c) \right) + (1 - \hat{p}_i(c)) \log \left(1 - p_i(c) \right) \Big]. \tag{16}$$

Here, K is the number of grid (set as 13, 26, 52, respectively), M is the number of bounding boxes detected by each grid (set as 3), λ_{cood} is weight coefficient of the coordinate error (set as 4), λ_{noobj} is weight coefficient of the confidence error for the cell with no object (set as 0.6), x_i, y_i is the ground true center coordinate of bounding box that falls in cell i, $\hat{x}_i, \hat{y}_i$ is the predicted center coordinate of bounding box that falls in cell i, w_i, h_i is the ground true width and height of bounding box that falls in cell i, $\hat{w}_i, \hat{h}_i$ is the predicted width and height of bounding box that falls in cell i, C_i is the ground true confidence score, $\hat{C}_i$ is the predicted confidence score, R_{ij}^{obj} is the value of 1 if the jth bounding box predicted by cell i is responsible for that prediction (otherwise 0), R_i^{obj} is the value of 1 if object falls in cell i (otherwise 0), $p_i(c)$ is the ground true class probability of class c, and $\hat{p}_i(c)$ is the predicted class probability of class c.

The training process is to optimize the loss function of each part to achieve the overall optimal. The final loss function can be expressed as:

$$loss = coordErr + confidErr + clsErr. \tag{17}$$

4 Experimental results

4.1 Data description

In this paper, all X-ray images belong to the GDXray database [17], in which there are three kinds of threat objects: handgun, shuriken. and razor blade, as shown in Fig. 7. To increase the amount of data, the X-ray images containing threat objects are processed by means of data augmentation such as translation, rotation, cutting, reversal, and so on. Then, the threat objects in the X-ray images are labeled by the labelImg tool.

There are 2250 and 250 X-ray images in the training and validation set, in which the proportion of handgun, shuriken, and razor blade is 1:1:1. Meanwhile, 1200 images are used as the testing set to verify the generalization detection ability of our proposed model.

4.2 Training detail

The experiments are run on a computer with a 3.7 GHz CPU, 12 GB RAM and an Nvidia Titan X GPU. Keras and TensorFlow are employed to implement the network model. All models are trained using the Adam optimization algorithm [12], with the momentum factor set as 0.9. The epoch and batch sizes are set to 500 and 32, respectively. Meanwhile, the initial learning rate is 0.001 and it will be decreased 10 times if the loss function is not declined after 3 epochs. The training will stop early if the loss function is not declined after 10 epochs.

4.3 Evaluation criteria

In this paper, the metrics of precision (P), recall (R), F_1 score, average precision (AP), and mean average precision (mAP) are used to evaluate the performances of the model. Precision (P) can be defined as:

$$P = \frac{T_{\mathrm{P}}}{T_{\mathrm{P}} + F_{\mathrm{P}}}. \tag{18}$$

Recall (R) can be defined as:

$$R = \frac{T_{\mathrm{P}}}{T_{\mathrm{P}} + F_{\mathrm{N}}}. \tag{19}$$

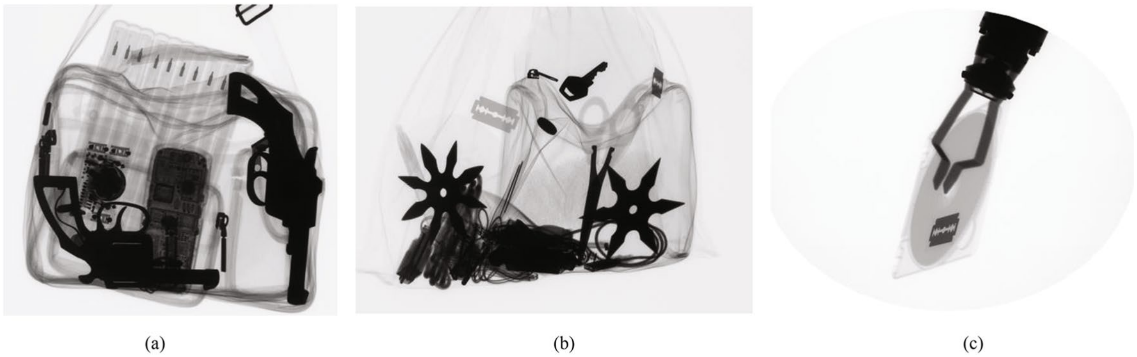

Fig. 7 X-ray images containing threat objects. **a** X-ray image containing two handguns. **b** X-ray image containing two shurikens and two razor blades. **c** X-ray image containing a razor blade

F_1 score is represented as:

$$F_1 = \frac{2 \times P \times R}{P + R}, \tag{20}$$

where T_P is the number of true positive samples, F_P is the number of false-positive samples, and F_N is the number of false-negative samples. A set of recall thresholds [0, 0.1, 0.2, ⋯ , 0.9, 1] is set, and average precision (AP) can be calculated as:

$$AP = \frac{1}{11} \sum_{R \in \{0,0.1,\ldots,0.9,1\}} P_{\mathrm{m}}(R), P_{\mathrm{m}}(R) = \max_{\tilde{R}, \tilde{R} \geq R} P(\tilde{R}), \tag{21}$$

where $P_{\mathrm{m}}(R)$ is the maximum precision when the recall meets $\tilde{R} \geq R$, and $\tilde{R}$ denotes the recall corresponding to the maximum precision. Mean average precision (mAP) can be defined as:

$$mAP = \frac{\sum_{i=1}^{T} \mathrm{AP}_i}{T}, \tag{22}$$

where T is the number of threat objects categories. In addition, we will compare the model size and computational speed (FPS). When the computational speed reaches to 30 FPS, it could be considered as real time.

4.4 Performance of different detection models of threat objects

In this section, we will show the results achieved by multiple detection models of threat objects that are different in backbone networks. Three pre-trained networks on ImageNet [27] (namely Xception [5], MobileNetV1 [10], and MobileNetV2 [29]) are used as backbone network for feature extraction and followed by three-scale prediction network. The three models are denoted as Xception-T, MobileNetV1-T, and MobileNetV2-T. First, we compare performance of the three detection models on the GDXray database [17]. The achieved results are given in Table 1, from which we can find that MobileNetV2-T achieves the best performance (mAP of 95.33%). Then, the improved MobileNetV2 is used as backbone and followed by three-scale prediction network, which is denoted as Im_MobileNetV2-T. Second, we compare the performance between the Im_MobileNetV2-T and the MobileNetV2-T on the GDXray database [17] in Table 2. According to Table 2, Im_MobileNetV2-T enhances the mAP of about 0.17% with respect to the MobileNetV2-T detection model.

4.5 Performance of our detection model of threat objects: ThreNet

To design the best architecture of our detection model of threat objects for X-ray baggage inspection, we conduct several experiments using different configurations varied by the position of the deformation layer. In particular, for the best performance, we design two schemes: (a) before all layers, acting directly on the input X-ray images (configuration represented with a "Yes" in Table 3) to face nonrigid deformation, and (b) after the down-sampling layer (represented with "Yes: X" in Table 3, that is, after the Xth down-sampling layer) to process nonrigid deformation of small regions. For all experiments, the deformation layer scales down the input feature maps by a factor of eight and processes them through the localization network. The localization network consists of three convolutional layers with 16, 32, 64 feature maps, respectively (with 7×7, 5×5 and 3×3 kernels) to compute the deformation parameters.

Table 3 shows the experiment results for different architecture configurations. The best architecture consists of the improved MobileNetV2 network, a deformation layer after the fifth down-sampling layer, and the three-scale prediction network. This version is our detection model of threat objects—ThreNet.

When the deformation layer is added after the fourth or fifth down-sampling layer, there is a performance increase. However, the detection ability worsens when the deformation layer is inserted at the beginning or after the first, second, and third down-sampling layer. The reason is twofold:

Table 3 Performance comparison of different configurations for ThreNet

Deform	P (%)	R (%)	F_1 (%)	AP			mAP (%)
				Handgun (%)	Shuriken (%)	Razor blade (%)	
Yes	96.52	89.17	92.7	90.7	88.46	94.73	91.3
Yes: 1	96.87	90.25	93.48	92.14	88.75	96.5	92.46
Yes: 2	97.37	95.41	96.38	95.34	90.81	97.4	94.52
Yes: 3	97.84	95.68	96.75	96.5	90.86	97.36	94.91
Yes: 4	98.21	97.08	97.64	97.45	91.58	97.76	95.6
Yes: 5	99.07	97.2	98.13	97.67	92.84	98.1	96.15

The "Deform" column represents the position of the deformation layer: "Yes" illustrates that the deformation layer is the first layer and acts directly on the input X-ray images; "Yes: X" represents that the deformation layer has been added after a specific down-sampling layer

(a) acting on the input images or the large feature maps has the effect of eliminating some differences that may be discriminative for the subsequent convolution operation; (b) detection of threat objects strictly depends on some specific region rather than on the entire image; thus, it is more crucial to process the nonrigid deformation of some specific regions than those of the whole images'

Finally, we compare the performance of the ThreNet, Xception-T, and MobileNetV2-T detection model (we exclude MobileNetV1-T, since, among the tested models, its performance is the worst) as well as the Im_MobileNetV2-T detection model. The results are shown in Table 4, from which we can see that our proposed detection model: ThreNet outperforms all the other detection models of threat object, reaching the *mAP* of 96.15%.

4.6 Comparison with the state of the art

In this section, to further evaluate the performance of ThreNet and to compare its performance with existing anchor-based one-stage object detectors, we test the performance of ThreNet, YOLOv3 [25], SSD [15], and Tiny_YOLO [23] on the testing set. Comparison results of the four models are presented in Table 5. It can be seen that ThreNet has a better precision, recall, and mean average precision than YOLOv3 [25], SSD [15], and the size of ThreNet is smaller than the YOLOv3 [25] and SSD [15] models. Moreover, the computational speed of ThreNet is significantly faster than YOLOv3 and SSD [15] models, achieving more than 60 FPS. It obviously meets the real-time requirement of X-ray screening system. Although the computational speed of Tiny_YOLO [23] model is faster than YOLOv3 [25], SSD [15], and ThreNet models, the detection ability of Tiny_YOLO [23] is the worst. Therefore, it means that our proposed ThreNet model performs better than the other three models in the detection task of threat objects for X-ray baggage inspection.

Figure 8 shows the detection results of our ThreNet, YOLOv3 [25], SSD [15], and Tiny_YOLO [23] with different X-ray images. It can be seen that the handguns can be successfully located by four models, but the bounding boxes of our ThreNet output are the closest to the real handgun regions. For the razor blade and shuriken, Tiny_YOLO [23] can either not detect the regions or the bounding boxes are much larger than the real regions, and YOLOv3 [25] and SSD [15] can correctly detect the regions, while the bounding boxes are a bit larger or smaller than the real regions. Compared with Tiny_YOLO [23], YOLOv3 [25], and SSD [15], our proposed ThreNet model can detect the threat object regions well and bounding boxes are very close to the real regions. According to the above analysis, it can be concluded that our proposed ThreNet model has faster computational speed and better detection accuracy than the other three detection models, and the bounding boxes can cover the threat object regions exactly. Therefore, our ThreNet model is very suitable for real-time detection of threat objects for X-ray baggage inspection.

5 Conclusion

In this paper, we employ the deep learning methods for detection of threat objects in X-ray inspection images. We have tested several existing deep learning models on the GDXray database [17] and proved that deep learning methods are able to solve the detection problem of threat objects in X-ray

Table 4 The detection performance comparisons of our ThreNet, Xception-T, MobileNetV2-T, and Im_MobileNetV2-T

Model	*P* (%)	*R* (%)	F_1 (%)	*AP*			*mAP* (%)
				Handgun (%)	Shuriken (%)	Razor blade (%)	
Xception-T	89.56	97.43	93.32	80.24	94.52	95.06	89.94
MobileNetV2-T	97.37	96.19	96.78	97.67	90.91	97.4	95.33
Im_MobileNetV2-T	98.2	96.19	97.18	97.67	91.23	97.62	95.5
ThreNet	99.07	97.2	98.13	97.67	92.84	98.1	96.15

Table 5 The detection performance comparisons between the ThreNet and existing object detectors: YOLOv3 [25], SSD [15], and Tiny_YOLO [23]

Model	*P* (%)	*R* (%)	F_1 (%)	*AP*			*mAP* (%)	Size (MB)	Speed (FPS)
				Handgun (%)	Shuriken (%)	Razor blade (%)			
YOLOv3	97.37	96.19	96.78	90.7	90.91	96.65	92.75	235.47	28.5
SSD	96.6	95.43	96.01	93.23	92.91	90.7	92.28	447.6	15.8
Tiny_YOLO	93.1	70.33	80.7	60.47	84.85	58.38	67.9	33.21	215.2
ThreNet	99.07	97.2	98.13	97.67	92.84	98.1	96.15	93.11	63.4

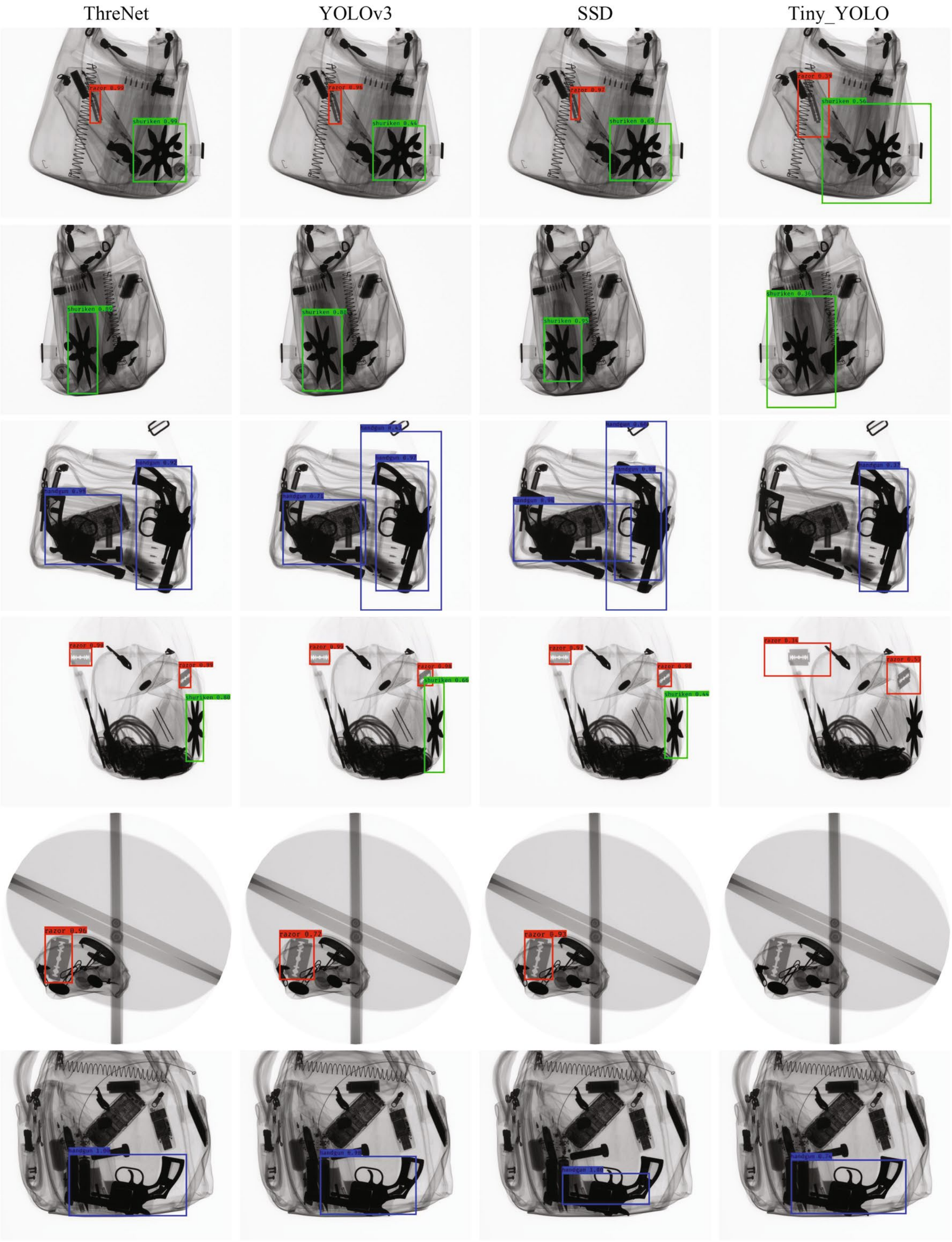

Fig. 8 The detection results of the ThreNet, YOLOv3 [25], SSD [15], and Tiny_YOLO [23] models

inspection images. We designed a novel automated model termed ThreNet based on the depthwise separable convolution to realize the detection of threat objects precisely and quickly for X-ray baggage inspection. To demonstrate the expression ability of the proposed model, we have conducted a series of experiments to compare its performance to other anchor-based one-stage object detection models. The experiment results demonstrate that our ThreNet model performs better than YOLOv3 [25], SSD [15], and Tiny_YOLO [23] in terms of detection performance and computational speed.

Our proposed model can be an auxiliary tool to assist the safety screeners to detect threat objects.

Further work could explore the integration of different kinds of features and the quantitative analyses of deep features. Moreover, it is possible to employ CNNs-based anchor-free object detection method to solve detection problem of threat objects to obtain better detection performance.

Acknowledgements This research is supported by the National Natural Science Foundation of China (Grant Nos. 61374178, 61402092, 61603182), the Fundamental Research Funds for the Central Universities (N171704004), the online education research fund of MOE research center for online education, China (Qtone education, Grant No.2016ZD306), and the Ph.D. Start-Up Foundation of Liaoning Province, China (Grant No. 201501141).

References

1. Bastan, M., Yousefi, M.R., Breuel, T.M.: Visual Words on Baggage X-ray Images. Springer, New York (2011)
2. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-NMS-improving object detection with one line of code. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5561–5569 (2017)
3. Bolfing, A., Halbherr, T., Schwaninger, A.: How Image Based Factors and Human Factors Contribute to Threat Detection Performance in X-ray Aviation Security Screening. Springer, New York (2008)
4. Chen, Y., Kang, X., Shi, Y.Q., Wang, Z.J.: A multi-purpose image forensic method using densely connected convolutional neural networks. J. Real Time Image Process. 16(3), 725–740 (2019)
5. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800–1807 (2016)
6. Christ, P.F., Ettlinger, F., Kaissis, G., Schlecht, S., Ahmaddy, F., Grun, F., Valentinitsch, A., Ahmadi, S.A., Braren, R., Menze, B.: SurvivalNet: Predicting patient survival from diffusion weighted magnetic resonance images using cascaded fully convolutional and 3D convolutional neural networks. In: IEEE International Symposium on Biomedical Imaging, pp. 839–843 (2017)
7. Flitton, G., Breckon, T.P., Megherbi, N.: A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery. Pattern Recogn. **46**(9), 2420–2436 (2013)
8. Franzel, T., Schmidt, U., Roth, S.: Object Detection in Multi-view X-ray Images. Springer, New York (2012)
9. Gao, X., Lin, S., Wong, T.Y.: Automatic feature learning to grade nuclear cataracts based on deep learning. IEEE Trans. Biomed. Eng. **62**(11), 2693–2701 (2015)
10. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: efficient convolutional neural networks for mobile vision applications (2017)
11. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. pp. 2017–2025 (2015)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014)
13. Krishnaraj, N., Elhoseny, M., Thenmozhi, M., Selim, M.M., Shankar, K.: Deep learning model for real-time image compression in internet of underwater things (IoUT). J. Real Time Image Process. **17**, 2097–2111 (2020)
14. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 936–944 (2017)
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: European Conference on Computer Vision (ECCV), pp. 21–37 (2016)
16. Megherbi, N., Han, J., Breckon, T.P., Flitton, G.T.: A comparison of classification approaches for threat detection in CT based baggage screening. In: IEEE International Conference on Image Processing, pp. 3109–3112 (2013)
17. Mery, D., Riffo, V., Zscherpel, U., Mondragón, G., Lillo, I., Zuccar, I., Lobel, H., Carrasco, M.: GDXray: The database of X-ray images for nondestructive testing. J. Nondestr. Eval. **34**(4), 42 (2015)
18. Mery, D., Svec, E., Arias, M.: Object recognition in baggage inspection using adaptive sparse representations of X-ray images. In: Pacific-rim Symposium on Image and Video Technology, pp. 709–720 (2015)
19. Mery, D., Svec, E., Arias, M., Riffo, V., Saavedra, J.M., Banerjee, S.: Modern computer vision techniques for X-ray testing in baggage inspection. IEEE Trans. Syst. Man Cyberne. Syst. **47**(4), 682–692 (2017)
20. Michel, S., Ruiter, J.C.D., Hogervorst, M., Koller, S.M., Schwaninger, A.: Computer-based training increases efficiency in X-ray image interpretation by aviation security screeners. In: IEEE International Carnahan Conference on Security Technology, pp. 201–206 (2007)
21. Payan, A., Montana, G.: Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. In: 2015 International Conference on Pattern Recognition Applications and Methods (ICPRAM), pp. 355–362 (2015)
22. Pozzo, F.R.D.: Aviation Security. Springer, New York (2015)
23. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2015)
24. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525 (2016)
25. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement (2018)
26. Riffo, V., Mery, D.: Automated detection of threat objects using adapted implicit shape model. IEEE Trans. Syst. Man Cybern. Syst. **46**(4), 472–482 (2017)
27. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2014)
28. Sajjad, M., Khan, S., Hussain, T., Muhammad, K., Sangaiah, A.K., Castiglione, A., Esposito, C., Baik, S.W.: CNN-based anti-spoofing two-tier multi-factor authentication system. Pattern Recogn. Lett. **126**(1), 123–131 (2019)
29. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: inverted residuals and linear bottlenecks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
30. Turcsany, D., Mouton, A., Breckon, T.P.: Improving feature-based object recognition for X-ray baggage security screening using primed visual words. In: IEEE International Conference on Industrial Technology, pp. 1140–1145 (2013)
31. Uroukov, I., Speller, R.: A preliminary approach to intelligent X-ray imaging for baggage inspection at airports. Signal Process. Res. **4**(5), 1–11 (2015)

32. Zentai, G.: X-ray imaging for homeland security. In: IEEE International Workshop on Imaging Systems & Techniques, pp. 1–6 (2008)
33. Zheng, H.T., Chen, J.Y., Yao, X., Sangaiah, A.K., Zhao, C.Z.: Clickbait convolutional neural network. Symmetry **10**(5), 138 (2018)

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

Yiru Wei received the B.E degree in Software Engineering from Wuhan Institute of Technology, China, and the M.S. degree in Computer Architecture from North China Electric Power University, China, in 2010 and 2013, respectively, now is pursuing the Ph.D. degree in the Software College, Northeastern University, China. Her main research interests include image processing and machine learning.

Zhiliang Zhu received an M.S. degree in Computer Applications and a PhD degree in Computer Science from NortheasternUniversity, China. He is a Fellow of the China Institute of Communications. His main research interests include information integration, complexity software systems, network coding and communication security, chaos-based digital communications, applications of complex network theories, and cryptography. He has authored and co-authored over 130 international journal papers and 100 conference papers. In addition, he has published five books, including Introduction to Communication and Program Designing of Visual Basic .NET. He is also the recipient of nine academic awards at national, ministerial, and provincial levels. Prof. Zhu has served in different capacities at many international journals and conferences. Currently, he serves as Co-Chair of the 1st–13th International Workshop on Chaos-Fractals Theories and Applications. He is a senior member of the Chinese Institute of Electronics and the Teaching Guiding Committee for Software Engineering under the Ministry of Education.

Hai Yu received a B.E. degree in Electronic Engineering from Jilin University, China, in 1993 and a PhD degree in Computer Software and Theory from Northeastern University, China, in 2006. He is currently an Associate Professor of Software Engineering at the Northeastern University, China. His research interests include complex networks, chaotic encryption, software testing, software refactoring, and software architecture. At present, he serves as an Associate Editor for the International Journal of Bifurcation and Chaos, Guest Editor for Entropy, and Guest Editor for the Journal of Applied Analysis and Computation. In addition, he was a Lead Guest Editor for Mathematical Problems in Engineering during 2013. Moreover, he has served different roles at several international conferences, such as Associate Chair for the 7th IWCFTA in 2014, Program committee Chair for the 4th IWCFTA in 2010, Chair of the Best Paper Award Committee at the 9th International Conference for Young Computer Scientists in 2008, and Program committee member for the 3th–13th IWCFTA and the 5th Asia Pacific Workshop on Chaos Control and Synchronization.

Wei Zhang received a PhD degree in Computer Science and Technology from Northeastern University, China, in 2013. He currently works as an Associate Professor in the Software College of Northeastern University. His research interests include signal processing and multimedia security.