Journal of Information Hiding and Multimedia Signal Processing ©2022 ISSN 2073-4212
Ubiquitous International Volume **13**, Number **3**, September **2022**

Contraband detection based on deep learning

Hao Chen and Zhe-Ming Lu

School of Aeronautics and Astronautics
Zhejiang University
No. 38, Zheda Road, Hangzhou 310027, P. R. China
*Corresponding author: zheminglu@zju.edu.cn

Received March 2022; revised July 2022
(Communicated by Zhe-Ming Lu)

Abstract. *This paper focuses on the detection of contraband scanned by X-ray. Due to the complexity of passenger's luggage, contraband in X-ray images have the characteristics of small target, fuzzy features and complex background. This paper proposes a scheme based on deep learning to detect the contraband in X-ray images. The paper mainly adopts YOLO (You Only Look Once) algorithm, which regards object detection as a regression task and obtains all information through one neural network, including the probability and bounding boxes. After investigation, the paper uses YOLOv3 algorithm, and the accuracy for the dev set achieves 45.2813%. The paper also explores two tasks in object detection, one is the number of bounding boxes for every grid in the YOLO algorithm, the other is over-fitting in training neural network.*

Keywords: Object detection, deep learning, YOLO (You Only Look Once) algorithm, x-ray picture, Convolution Neural Networks (CNN)

1. **Introduction.** In order to protect public property and people's safety, it has become a consensus of the whole society to block some dangerous goods that may cause accidents in public places. Especially in transportation hubs such as airports and stations, it is very important for authorities to protect the safety of passengers and the normal operation of public facilities. However, the complexity of passenger luggage and the small size of the contraband, such as thief and lighter, make the security more difficult. Hence, many new technologies have been going into airport security, the most common way is that using X-rays to non-intrusively analyze passengers' luggage, speeding up checks while avoiding direct contact with the luggage. X-ray security inspection machine greatly improves the efficiency of security inspection [1][2][10]. With the popularity of X-Ray security screening machines, new problems have emerged. X-Ray imaging technology has erased many of the goods' characteristics [12], making it more difficult to detect contraband by eyes and requiring more time for security officers to determine whether the luggage contains contraband. The huge number of passengers means that security checks need to be carried out as quickly as possible, which makes it harder to identify contraband manually.

In view of the disadvantages of identifying X-Ray contraband by eyes, we put forward an idea, hoping to realize the detection of contraband in X-Ray images by deep learning. The luggage scanned by X-Ray security inspection machine is used as the input of neural network, and the results are displayed on the output of it, telling us whether the luggage contains contraband [11]. The application of this method can greatly reduce the labor cost of airport security, the security inspection process is no longer completely dependent on manual identification, and further improve the efficiency of airport security check.

Along with contraband detection, this paper describes solutions to two tasks in object detection, one is the number of bounding boxes in the YOLO algorithm, the other is overfitting in the NN training process. This paper is based on YOLO algorithm. In Section 2 , we show the data processing and two methods used in this paper. InSection 3 , we show the experimental results. In Section 4 , we summarize the paper, and put forward our own ideas for the existing model.

2. Contraband Detection with Deep Learning.

2.1. Dataset. The data set used in this paper is from 'Alibaba Tianchi Competition - Jinnan Digital Manufacturing Algorithm Challenge'. The dataset contains two sub-datasets, one called Restricted containing the object and the other called Normal containing no object. The dataset contains 2024 images from Restricted and 3000 images from Normal. In the course of machine learning, Andrew Y. Ng told us that the combination of dataset Restricted and dataset Normal is more conducive to the training of models. After mixing, we divided all the X-Ray images into three parts in a 18:1:1 ratio, which were train set, dev set and test set. The train set is used to train the weights of the neural network, the dev set is used to compare with the training set, and the test set is used to calculate the final error rate of the system. We can also keep two data sets, the train set and the test set. However, according to the research of Y. LeCun et al., it is better to divide all data sets into three [7].

This data set contains five categories of contraband, including Iron lighter, black nail lighter, knife, the battery capacity and scissors. This data set already provides the category and bounding box of each contraband on the image, and presents these information in a JSON file.

The X-Ray images used in this paper are all from the luggage scanned by the airport security inspection machine. Usually, the interior of a passenger's suitcase is very complicated. After X-Ray scanning, the items in suitcase is interlaced with each other, forming a very complex background in x-ray images. In general, the contraband we carry, including small objects such as scissors, lighters, and batteries, is hardly detect even by eyes. The complexity of the suitcase, the image distortion caused by X-ray scanning, and the small size of the object all greatly increase the difficulty of contraband detection.

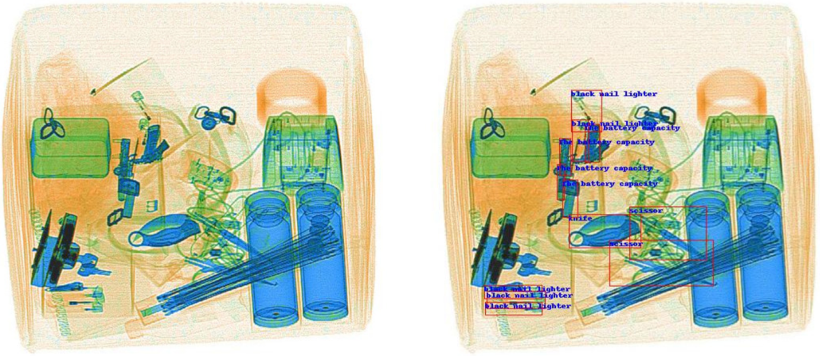

Figure 1. X-Ray contraband image (left) and labeled X-Ray contraband image (right)

After proposing several different schemes, we compare the error rate of train set with the error rate of dev set in every algorithm. There is no doubt that the error rate of train set is very low. If the error rate of dev set is also at a low level, we think the neural network performs well. After putting test set into the neural network, we will obtain the lowest error rate. The error rate of the train set and the dev set can also be used to determine the state of the neural network, fitting or over-fitting. If the error rate of a neural network on the train set is very low, and high on the dev set, the neural network is considered to be over-fitting. We should appropriately reduce the number of training iterations and the number of layers of the NN which helps to ease the fitting degree on the train set.

For a single picture, this paper uses the traditional digital image processing method to enhance the profile of the object. In view of the blurry feature of X-Ray images, we process the original images in three dimensions: contrast, brightness and color. In order to avoid the interference of the complex background, we adopt the spatial filtering to enhance the edge features of the image and blur the background information of the image. The image processing method used in this paper is a simple method for X-Ray pictures.

For images in the same neural network, they must have the same size. In this paper, we use bicubic interpolation to enlarge or shrink the original image, so that the size of all images remains the same. The formula of bicubic interpolation is:

$$v(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j \tag{1}$$

2.2. **Label.** YOLO does not regard deep learning as a classification task, but as a regression task [3]. In the YOLO algorithm, we not only predict the classification probability, but also the bounding box of the object in one neural network [4].

In the YOLO algorithm, in order to reduce the complexity of calculation, we do not extract the regional proposals in advance, but segment the original image and classify the bounding box. In this paper, the image is divided into 8*8 grids, and the objects on the whole image are allocated to each grid on the picture. If the object's center point is within the grid, it is considered that the object belongs to the grid [4], even if the object's bounding box is beyond the boundary of the grid (see Figure 2).

After analyzing all the images in the data set, we find that the grid divided on an image contains at most 10 objects. We will show later that the optimal number of bounding boxes is not the maximum number of objects in grids. In this paper, the number of bounding boxes is 5.

In this paper, we have five different categories of objects to be classified. For one object, we need the coordinate of the object's center point and the length and width of the object's bounding box to mark its position on the image. The label Pc is used to determine whether the object is included in the picture. BX, BY, BW and BH respectively represent the four parameters of the bounding box, the horizontal and vertical coordinates of the center point and the length and width of the bounding box. C1, C2, C3, C4 and C5 respectively represent the probability that the object is of five different categories. As a result, we need 10 parameters to detect one object, and thus the label for one object is y = [Pc, BX, BY, BW, BH, C1, C2, C3, C4, C5]. It is worth noting that BX and BY are not the actual coordinates of the center point, but the relative coordinates of the center point for the upper left corner of the grid, and then divided by the length and width of the grid. BW and BH are the relative length and width of the object border, which divided by the length and width of the grid [3][4].

Here, let's summarize what has been said above. In this paper, for a picture, we divide it into 8*8 grids. For a grid, we assume that there are at most 5 objects in the grid. For a object, we need 10 labels to indicate the category of the target and the position of the object border. Therefore, the label size for an image is 8*8*5*10. For convenience,

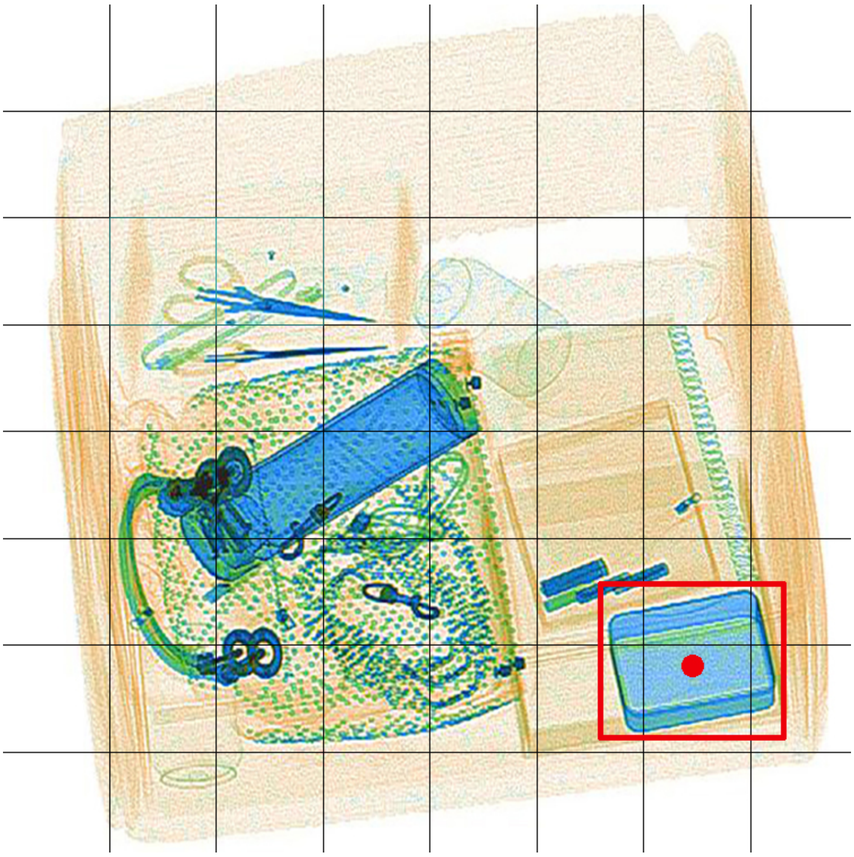

Figure 2. The image segmentation by 8*8. The red rectangle marks object border , which is detected by the grid where the center point of border is.

we convert it into 8*8*50. Therefore, in the neural network, the number of labels for an image is 8*8*50.

2.3. **Two Different Projects.** During the research, this paper adopts two deep learning-based schemes for object detection. One is the classical convolutional neural network based on sliding window method [7], and the other is the convolutional neural network based on YOLO [9]. The results of VGG-16 in RGB images classification is not ideal. The curve of the loss function trained by VGG network with the number of iterations fails to converge, while the curve of YOLO algorithm in object detection can converge. Therefore, I finally choose YOLO algorithm as the backbone of this paper.

2.3.1. *Classification Algorithm Based on VGG-16.* When I first came into this project, I adopt the traditional VGG-16 network to classify the objects. Firstly, we segment the part containing the target in the image, and train the neural network with segmentation. In the test, we segment the image with sliding window method, and the different parts of the image are judged separately. In this model, the input part of the neural network is a clipped picture, and the output of the neural network is a one-dimensional matrix.

VGG-16 convolutional neural network consists of 13 convolutional layers, 5 pooling layers, and 3 fully connected layers. The output feature vector of the last pooling layer is a three-dimensional matrix, while the input of the full connection layer is a one-dimensional matrix. Therefore, PCA dimension reduction algorithm is adopted here to reduce the dimension of the feature vector.

In training the neural network, after several iterations, the loss function of the train set is always unable to converge. Figure 4 shows the result of 200 iterations of training

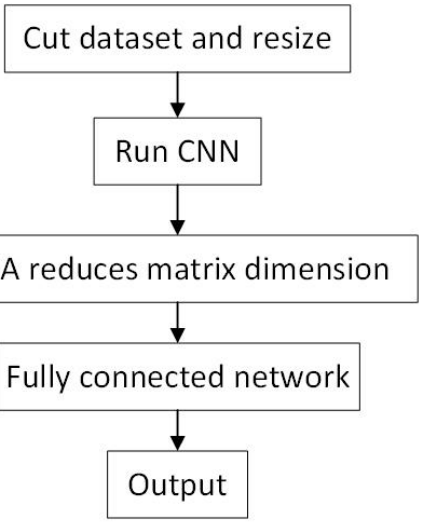

Figure 3. Classification algorithm flow chart based on VGG-16

vgg-16 with 40 images. The yellow line is the curve of the loss changing with iterations, while the blue line is the curve of the accuracy changing with iterations. It shows that both the loss and the accuracy have a violent oscillation, which indicates that the curve of the loss function is not a smooth curve with one minimum value point, and the cost function always fails to converge. After judgment, VGG-16 classical convolutional neural network is abandoned in this paper.

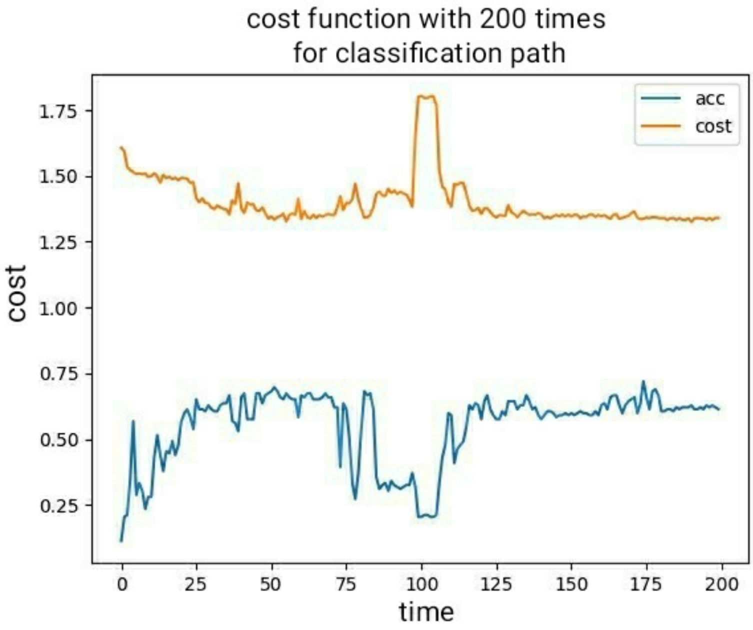

Figure 4. Curve of accuracy (blue line) and loss value (orange line) with iterations, the curve does not converge

2.3.2. *Object Detection Based on YOLO.* YOLO doesn't segment the picture, but segment the grids and uses each grid to detect objects. Different from the sliding window method, YOLO algorithm directly divides the original image, similar to the sliding window method with the same window [4]. This scheme also brings some drawbacks. The excessive grid segmentation leads to the increase of parameters for neural network, and the detection probability of the same object for adjacent grids increases, which improves the difficulty of multi-object detection [4]. Too little grids segmentation result in a lot of objects to be detected in a same grid, and the interference between them also increase [7]. Similarly, an excessive number of bounding boxes result in the large wasted parameters in each grid.

After comprehensive analysis, the size of the image in the Darknet-53 network is (512,512,3). The output of the neural network was a matrix of 8*8*5*10, in which the number of bounding boxes was 5.

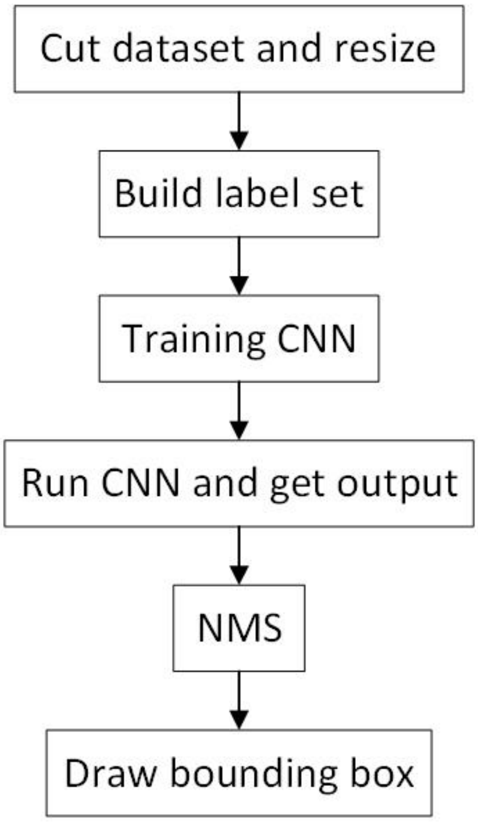

Figure 5. Flowchart of object detection algorithm based on YOLO

In this paper, we use a fixed size of learning rate, and set it to 0.001. And we train the network with 4523 X-Ray images as train set. After 100 iterations, we obtain the curve of the loss function with the number of iterations. The curve was smooth and could converge to meet the requirement in training the neural network.

3. **Performance Evaluation.** The data set used in this paper contains 5025 pictures, the train set contains 4523 pictures, and the dev set and the test set contain 251 pictures respectively. This paper uses the train set to train the neural network, uses the dev set to judge whether the state of the neural network is underfit or overfit, and uses the test set to judge the effect of the neural network. In order to show the experimental effect, while keeping the YOLO framework unchanged, this paper adopts three different backbones, Darknet53, Resnet34 and Darknet19, to compare the influence of network structure and layer number on the training and detection accuracy.

3.1. **Three Different Neural Networks.** Table 1 shows the comparison of three kinds of neural network structures. Darknet-19 is used in the classic YOLO 9000 algorithm [6]. On the basis of the original YOLO network, this network uses more 3*3 convolution

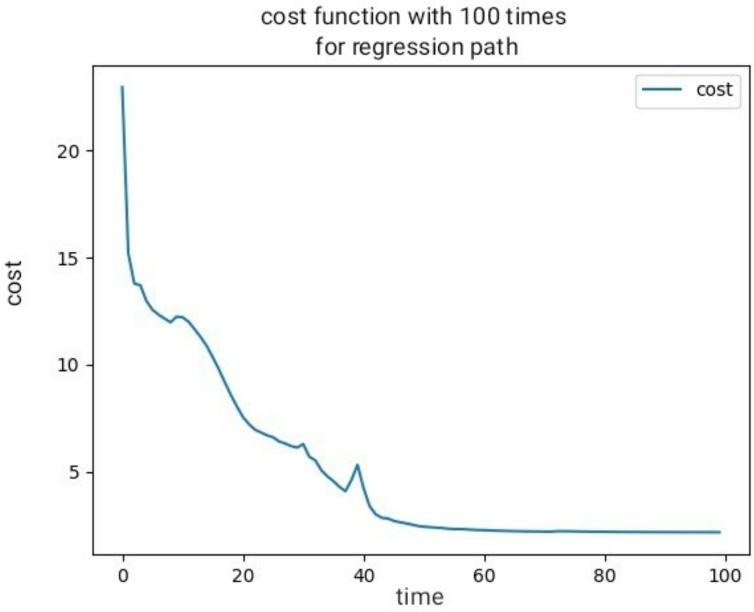

Figure 6. The curve of loss value (blue line) of object detection algorithm based on YOLO with iterations, the curve converges.

kernels, and replaces one layer of high-dimensional convolution kernels with multi-layer low-dimensional convolution kernels, so that the neural network can fit more complex non-linear functions. Resnet-34 is mentioned in the paper by H. Kaiming et al. [8]. The network is compared with VGG network and 34-layer plain network. Addition of residual block greatly improves the accuracy of the network and the training speed. However, due to the limitations of cognition at that time, the first layer of the network still adopts the high-dimensional convolution kernel of 7*7. Darknet-53 network is the main backbone used in this paper.

The number of convolution layers of the three neural networks increases successively. By training three neural networks, we can compare the effect of the layer number of neural networks. At the same time, in view of the possible overfitting of Darknet53 training small data sets, the use of shallow neural network is also conducive to exploring the neural network overfitting.

3.2. **Numbers of Bounding Boxes.** At the beginning of the paper, considering the data set and the characteristics of YOLO algorithm, in order to detect all objects in each grid, the paper adopted the maximum number of the detected object in the grid as the number of bounding boxes, namely BBs (Bounding Box Numbers) =10. However, in the actual experiment, we found that the more bounding boxes in each grid is not the better. There was only one grid that could reach 10 objects. The huge number of bounding boxes caused a waste of parameters. For most images, nearly half of the tags were used to describe meaningless backgrounds.

Figure 7 shows the distribution of the number of objects in each grid after the picture is segmented into grids (grids without objects belong to background and no statistics are made). Observing the distribution diagram, we found that lots of grids contain less than 5 objects, and the proportion of grids with more than 5 objects is very small. Considering that there are 5 categories to be detected in this data set, we set the number of bounding boxes as 5 in the subsequent experiments, that is, BBs=5.

Table 1. Comparison of three network structures [5][6][8]

<table>
<tr><th>Darknet19</th><th>Resnet34</th><th>Darknet53</th></tr>
<tr><th>19 conv layers</th><th>34 conv layers</th><th>53 conv layers</th></tr>
<tr><td colspan="3">Input(512*512*3 RGB image)</td></tr>
<tr><td>3*3 conv,32</td><td>7*7 conv,64,/2</td><td>3*3 conv,64,/2</td></tr>
<tr><td>2*2 Max pooling,/2</td><td>2*2 Max pooling,/2</td><td>3*3 conv,64,/2</td></tr>
<tr><td>3*3 conv,64
×1</td><td>3*3 conv,64
3*3 conv,64
Residual
×3</td><td>1*1 conv,32
3*3 conv,64
Residual
×1</td></tr>
<tr><td>2*2 Max pooling,/2</td><td>3*3 conv,128,/2
3*3 conv,128</td><td>3*3 conv,128,/2</td></tr>
<tr><td>3*3 conv,128
1*1 conv,64
3*3 conv,128</td><td>3*3 conv,128
3*3 conv,128
Residual
×3</td><td>1*1 conv,64
3*3 conv,128
Residual
×2</td></tr>
<tr><td>2*2 Max pooling,/2</td><td>3*3 conv,256,/2
3*3 conv,256</td><td>3*3 conv,256,/2</td></tr>
<tr><td>3*3 conv,256
1*1 conv,128
3*3 conv,256</td><td>3*3 conv,256
3*3 conv,256
Residual
×5</td><td>1*1 conv,128
3*3 conv,256
Residual
×8</td></tr>
<tr><td>2*2 Max pooling,/2</td><td>3*3 conv,512,/2
3*3 conv,512</td><td>3*3 conv,512,/2</td></tr>
<tr><td>3*3 conv,512
1*1 conv,256
3*3 conv,512
1*1 conv,256
1*1 conv,512</td><td>3*3 conv,512
3*3 conv,512
Residual
×5</td><td>1*1 conv,256
3*3 conv,512
Residual
×8</td></tr>
<tr><td>2*2 Max pooling,/2</td><td></td><td>3*3 conv,1024,/2</td></tr>
<tr><td>3*3 conv,512
1*1 conv,256
3*3 conv,512
1*1 conv,256
1*1 conv,512</td><td></td><td>1*1 conv,512
3*3 conv,1024
Residual
×8</td></tr>
<tr><td colspan="3">2*2 avg pooling,/2
Fc 1000 (1*1 conv,1000)
Fc 50 (1*1 conv,50)
Output (8*8*50)</td></tr>
</table>

Therefore, this experiment is based on a hypothesis that there are at most 5 objects in a grid, and each category can only appear at most once in a grid.

3.3. **Training.** In order to avoid over-fitting, this deign observes loss function curve of training set and dev set during the training process.

In this paper, the same training data, learning rate and iteration times are used to train different neural networks. As is shown in Figure 8, the training process of Darknet-19, Resnet-34 and Darknet-53 is respectively shown. The blue curve represents the train set's loss, while the orange curve represents the dev set's loss. Figure 8 shows that the

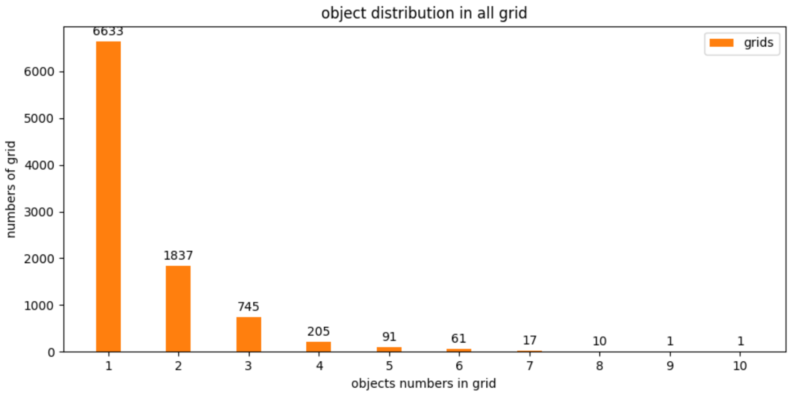

Figure 7. Object number distribution in the grids

training process of the three networks is roughly the same, and they can converge with train set in the end. Resnet-34 network converges at first and the number of convolution layers is not the least. However, addition of residual block improves the training speed of the neural network. Darknet-19 has the least number of convolution layers and the convergence speed is slower. Darknet-53 has the largest number of convolution layers, but the convergence speed is greatly accelerated after addition of residual block. In three different backbones, the loss of the dev set is not ideal, but with iterations increase, the loss value of dev set tends to decline.

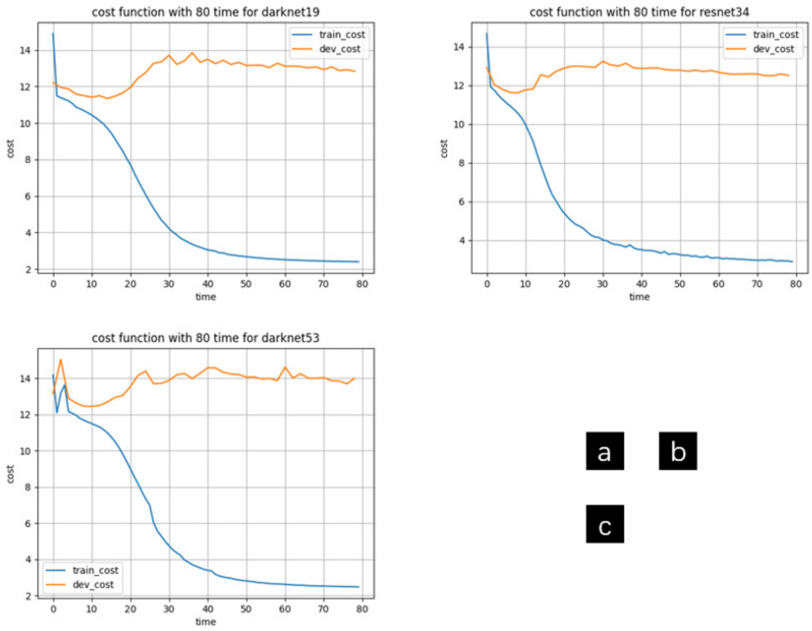

Figure 8. During the training of three neural networks Darknet-19 (Figure a), Resnet-34 (Figure b) and Darknet-53 (Figure c), the loss value of train set (blue line) and dev set (orange line) varies with iterations

3.4. **Imitative Effect in the Training Set.** Figure 9 respectively shows the effect of three different neural networks on the train set. Two pictures after training are selected to display the training results. From these two images, we can see the complexity of this data set, and the contraband in the images is difficult to see even with eyes. In Fig 9 (Fig. d), the number of objects is relatively small, and the object border is relatively clear, used to compare the accuracy of the algorithm for object border. As the number of objects in Fig 9 (Fig. h) is large and the objects overlap each other, it is difficult to identify the border of objects, used to compare the recognition and classification of multi-objects.

For Resnet-34, small objects were difficult to detect in pictures of multi-objects (see yellow circles in pictures b and f in figure 9), while object border fits inaccurate in Darknet-53 (see green circles in pictures c and g in figure 9). The detection effect is better in Darknet-19. However, three networks all have different degrees of missed detection with multi-object aliasing.

In general, after enough iterations, all three networks can fit the train set, although the degree of fitting varies. From the perspective of practical application, the training cannot iterate infinitely, and we need to consider the fitting of the dev set to the neural network. The infinite cycles will make the network overfit the train set, and the effect of the dev set and the test set will be worse.

The data set is characterized by a lot of objects in one image, and the lack of X-Ray contraband images made the task very difficult. We consider not only the fitting of the neural network to the train set, but also the matching of layer number and data set. It is not that the more layers and complex structure, the better the training effect will be.

3.5. **Imitative Effect in the Dev Set.** From the curve of the loss function with iterations, we see that three neural networks begin to converge after 44 iterations. In this paper, each neural network is iterated 78 times, and the weights of the neural network are recorded in each of the two iterations. From the 44th iteration to the 78th iteration, we recorded the imitative effect of the neural network for the dev set, and recorded information once every two iterations, including 16 neural networks.

Table 2 shows the imitative effect of three neural networks to the dev set. The data in the table comes from the average value of the accuracy of each image in the dev set. The accuracy of Darknet19 network for dev set gradually increased, reached the peak after 70 iterations and gradually decreased. Based on the imitative effect of the dev set and the training process, we judge that after 70 iterations, Darknet-19 starts over-fitting. Therefore, we think 0.273896 is the best imitative effect that Darknet19 network achieves for this data set. The imitative effect of Resnet34 network on the dev set has always been at a low level. Considering the imitative effect of Resnet-34 network on the train set, it can be concluded that this network is not suitable for the object detection task in this data set. The imitative effect of Darknet-53 on the dev set reached the highest level in this paper. After 70 and 78 iterations, the accuracy of the dev set reached 0.452813 and 0.452548, which is a huge improvement compared with the previous two networks. It can be seen that Darknet53 has the best performance for data sets.

3.6. **Optimization of the Model.** This paper uses three different backbones, carries on the same iteration number to the same data set. The whole task has achieved some results, but the training process needs optimizing.

The algorithm adopted in this paper is YOLO. The biggest characteristic of YOLO is to obtain all the results through one neural network, including the probability and the border. Because of this characteristic, the algorithm runs very fast. However, YOLO algorithm is only a rough detection of the object, because of its simplicity. In order to solve this problem, a widely adopted method is to increase the number of convolution layers. YOLO 9000 and YOLOv3 all adopt this scheme. In view of the experimental results, this paper proposes the optimization scheme of YOLO algorithm.

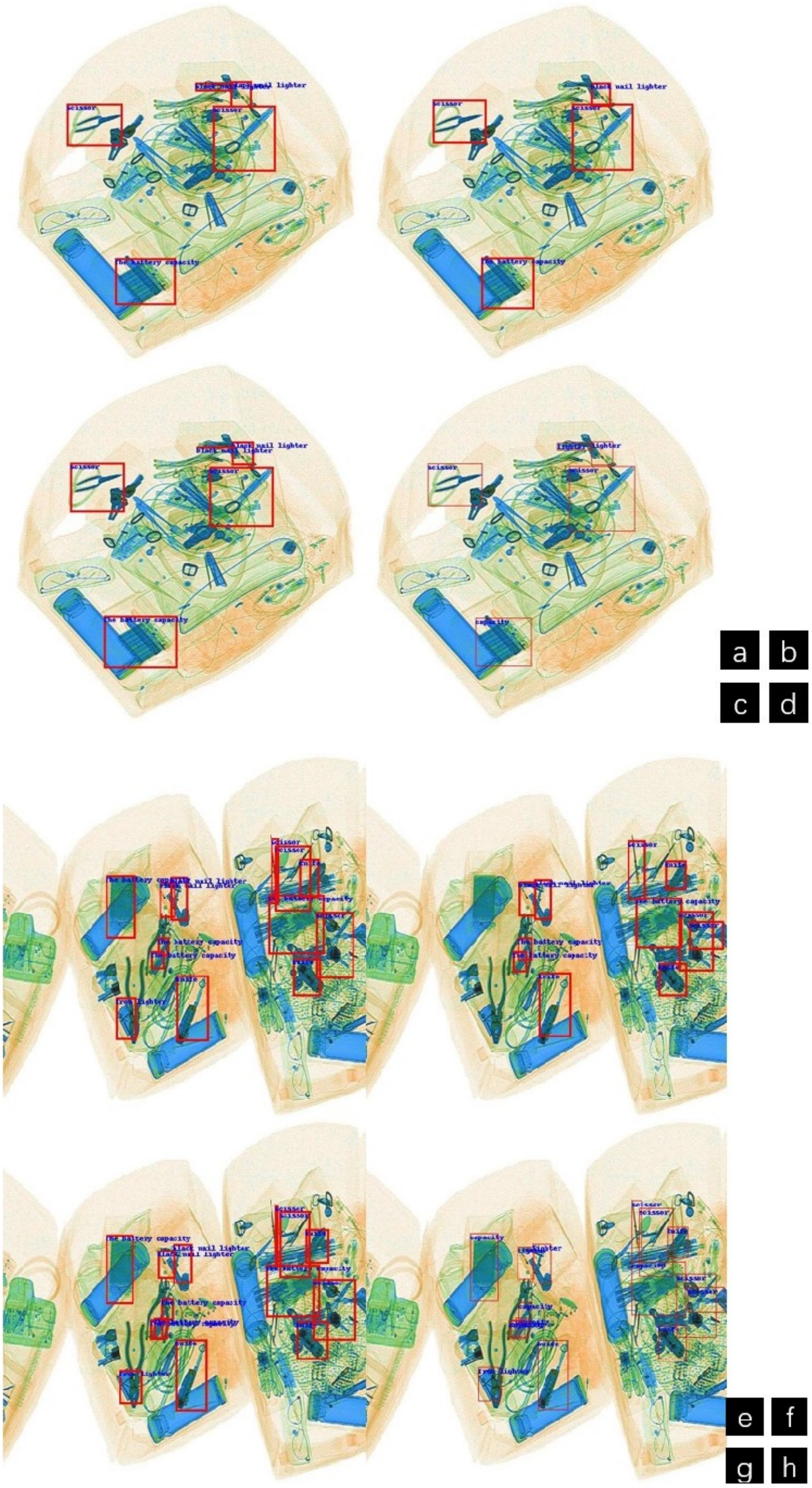

Figure 9. Training results of three neural network, an X-ray image with fewer objects (Figure d) is recognized by three neural networks Darknet-19 (Figure a), Resnet-34 (Figure b) and Darknet-53 (Figure c). An X-Ray image with many objects (Figure h) is recognized by three networks Darknet-19 (Figure e), Resnet-34 (Figure f) and Darknet-53 (Figure g)

Table 2. The variation of accuracy with iterations of three networks (for dev set)

iterations	**darknet19**	**resnet34**	**darknet53**
44	0.186615	0.098363	0.167470097
46	0.166995	0.101842	0.136491
48	0.233382	0.127432	0.165079
50	0.124935	0.121869	0.143393
52	0.20079	0.097774	0.115481
54	0.219028	**0.177135**	0.201854
56	0.211089	0.137745	0.255054
58	0.201459	0.120875	0.227709
60	0.254528	0.09517	0.291989
62	0.170558	0.098363	0.268141
64	0.173331	0.09164	0.321155
66	0.217717	0.080102	0.373723
68	0.156366	0.077097	0.417333
70	**0.273896**	0.06822	0.452813
72	0.257998	0.083731	0.313066
74	0.261479	0.07277	0.403762
76	0.178561	0.090504	0.320061
78	0.215504	0.101052	**0.452548968**

Since YOLO algorithm is only crude detection of objects, the recognition rate of small targets is low. The data set contains many small objects, such as batteries, lighters and knives. In this paper, the image is divided by the way of 8*8. The size of each grid is 64*64. The size of the detection target is smaller than the size of the grid. The grid size is similar to the object size, which is more conducive to object detection.

In this paper, after analyzing the characteristics of the data set, we assume that a grid contains five objects. Five bounding boxes were used to describe the object border in each grid, which save the cost of calculation and improved fitting degree of the data set.

4. **Conclusions.** At the end of this paper, we review the whole paper. This paper carries out for specific task and tries to detect the contraband in X-Ray images by deep learning. First, we collect the data, handle and annotate the dataset. For contraband detection, the initial idea is to take it as a classification task, segmenting the images and classifying them. After trying, the loss function could not converge in the training, so the idea was abandoned and we turn to the object detection algorithm. In the two paths of object detection, the YOLO algorithm of "all results obtained by one detection" was selected. The loss function converges quickly with YOLO algorithm. YOLOv3 fits the train set quickly, but the effect is not great for the dev set. Therefore, considering the possible overfitting of the network, we try to reduce the number of layers and iterations in training. Three backbones with different convolutional layers, Darknet-19, Resnet-34 and Darknet-53, were used in YOLO framework, and the network was trained with different iterations in order to get the best result. The result shows that Darknet-53 is still the best one.

Acknowledgment. This research work is also partially supported by Ningbo Science and Technology innovation 2025 major project under Grants No. 2021Z010 and No. 2021Z063.

REFERENCES

[1] Y. Wei and X. Liu, Dangerous goods detection based on transfer learning in X-ray images, *Neural Computing & Applications*,vol. 32, no. 12, pp. 8711-8724, Jun 2020.

[2] D. Saavedra, S. Banerjee, and D. Mery, Detection of threat objects in baggage inspection with X-ray images using deep learning, *Neural Computing & Applications*,2020.

[3] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. D. Wu, Object Detection With Deep Learning: A Review, *Ieee Transactions on Neural Networks and Learning Systems*,Review vol. 30, no. 11, pp. 3212-3232, Nov 2019.

[4] Redmon, Joseph, et al., You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*,2016

[5] Redmon, Joseph, and Ali Farhadi, Yolov3: An incremental improvement, *arXiv preprint*, arXiv:1804.02767 (2018).

[6] Redmon, Joseph, and Ali Farhadi, YOLO9000: better, faster, stronger, *Proceedings of the IEEE conference on computer vision and pattern recognition.*, 2017.

[7] LeCun, Yann, et al, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*,86.11 (1998): 2278-2324.

[8] He, Kaiming, et al, Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[9] Li-Sang Liu, Dong-Wei He, Ying Ma, Ying Ma, Jing Huang, Jun-Nan Li and Jin-xin Yao, A Novel License Plate Location Method Based on Deep Learning, *Journal of Network Intelligence*, Vol. 5, No. 3, pp. 93-101, August 2020.

[10] Chien-Chih Lin, Jin-Xiang Liao, Ming-Sung Chiu, Ming-Tsung Yeh, Yi-Nung Chung and Chao-Hsing Hsu, Applying Deep Learning Algorithm to Cell Identification, *Journal of Network Intelligence*, Vol. 6, No. 3, pp. 401-410, August 2021.

[11] Jia-Lin Cui, Bin Lian, Zhe-Ming Lu and Hao-Lai Li, Research on Wheel X-ray Defect Recognition Algorithm Based on Deep Learning, *Journal of Network Intelligence*, Vol. 6, No. 4, pp. 753-762, November 2021.

[12] Jia Zhai, Hao-guang Zhao, Qiang Ji and Xiao-dan Xie, Computational Resource Constrained Deep Learning Based Target Recognition from Visible Optical Images, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 3, pp. 558-566, May 2018.