IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 34, No. 4, August, 2020, pp. 495-500
Journal homepage: http://iieta.org/journals/ria

Evaluating One Stage Detector Architecture of Convolutional Neural Network for Threat Object Detection Using X-Ray Baggage Security Imaging

Malarvizhi Subramani[1*], Kayalvizhi Rajaduari[1], Siddhartha Dhar Choudhury[2], Anita Topkar[3], Vijayakumar Ponnusamy[1]

[1] Department of Electronics and Communication Engineering, SRMIST, Chennai 600023, India
[2] Department of Computer Science Engineering, SRMIST, Chennai 600023, India
[3] Electronics Division, Bhabha Atomic Research Centre, Trombay, Mumbai 400085, India

Corresponding Author Email: malarvig@srmist.edu.in

https://doi.org/10.18280/ria.340415

Received: 18 June 2020
Accepted: 26 July 2020

Keywords:
deep learning, x-ray baggage screening, object detection, RetinaNet, SSD

ABSTRACT

Neural networks can map complex functions between input and a target, and they have produced state-of-the-art results in the field of computer vision. These neural network based models have superseded the conventional computer vision algorithms for X-ray imaging. In this paper, we propose a deep neural network based solution for a subset of the X-ray imaging problem of detecting sharp items in a baggage X-ray. Existing reports were region based CNN architecture for an object detection in X-ray imaging systems. We propose Deep learning method as a Single Shot Detector (SSD) and RetinaNet, which are a one-shot technique for object detection and are able to do inference in real time 15-30 frame per seconds (fps) videos. These techniques are Fully Convolutional Network (FCN) and have the capability to do both classification and regression with the same shared weights. These networks return a bounding box around the object of interest along with the class of that particular object. This technique has been used in training single stage detectors for four objects of interest - knife, scissors, wrench and pliers. We have achieved good detection accuracy with mean average precision of a 60.5% for SSD and of 60.9% for RetinaNet using the SIX-ray10 database, which contains harmful items and non-harmful items. The ratio of number of harmful to non-harmful items is very low, making the problem a daunting one. Through various experimentations we have come up with the best possible results using various pre- trained networks as the feature extractor in tandem with these object detection algorithms. With further improvements on the achieved results, it would be possible to deploy this technique in airports to minimize human error and improve security in such environments.

1. INTRODUCTION

Population mobility has become essential due to the development of society and economic trade activities, resulting in the rapid increase of number of people using the airports for travel. Aircraft is one of the primary targets for the terrorists necessitating the scanning of the passenger baggage with better accuracy and high speed of checking /inspection process. Baggage inspection using X-ray scanning is also the technique used to protect public spaces like railway stations, shopping malls, hotels, offices, etc., from dangerous events happening. Object identification/detection for security applications is a highly challenging area as when the objects are placed in a closely packed bag, it is overlapped by other objects, thus offering an unrecognizable view of the object in cluttered X-ray images. Manual detection of sharp objects is a tedious task in a limited time as in the suitcases, the items are randomly stacked and overlap each other. Though the human eyes provide us with the best readability test, it is difficult for the operator to identify the threat objects with the naked human eyes due to the clutter of the objects. Hence, it is necessary to automate the X-ray baggage scanning system to aid the human operator. This technique of automating the identification process at the airports has not been fully implemented yet due to the low accuracy levels and the improper ratio of data in the training set of harmful and non-harmful items. The objective of this paper is to develop a technique to reduce the time of inspection at the checking station time. This would reduce the required human force and efforts for baggage inspection and in turn save the passengers time for check in as well as improve the security measure required at the airport. This paper is organized with related published work followed by Deep learning technique adopted for object detection, and results and conclusions.

2. RELATED WORK

In recent years, the evolvement in the area of security imaging resulted in the use of computer vision tools to analyze X-ray images. In this line, Mery et al. [1] used adaptive sparse method to identify four class of dangerous objects like razor blades, shuriken, handguns and clips, the recognition rate was more than 95% in every class in the X-ray images. Khotanzad and Hong [2] used Zernike moment features to identify objects in two stages object charactersation and object matching. Shape based fuzzy KNN classifier to detect pistol in X-ray baggage image of size is 310*1035 and evaluated only 15

image samples [3]. Bastan et al. [4] work of Bag-of-Visual-Words (BoVW) and Support Vector Machine (SVM) classification with SIFT feature descriptors within X-ray baggage imagery they obtained performance of recall, precision and average precision 0.7, 0.29, 0.57 respectively. A similar approach, extending the work of Turcsany [5], using BoVW with SURF feature descriptors and SVM classification together achieved true positive 99.07%, and false positive 4.31% on firearms detection over 2000 examples. Mery et al. [6] provides the work which has an impact in the field of object recognition in X-ray testing by evaluating different computer vision techniques such as Bag of Words (BoW), KNN-Based in Sparse Reconstruction Object Recognition, Adaptive Implict Model (AISM), and Adaptive Sparse Representations. The use of various features such as pseudocolor, texture, edge, shape features, LBP, SIFT, SURF as well as various classification models like KNN, SVM, ISM, BOW, among them 95% accuracy achieved based on visual vocabularies and deep features in the detection of sharp objects. Akçay et al. [7] have introduced Convolutional Neural Network CNN in the field of X-ray baggage security imagery. Akcay et al. [8] compares the CNN and Handcrafted feature detector and descriptor for classification of multiple objects. The presented experimentation demonstrates that CNN features achieve superior performance to Handcrafted BoVW features. Hand crafted feature detector/descriptor showed accuracy of 94% and CNN demonstrated the accuracy of 98.6% for same images. Recently CNN-based deep learning architectures [9-11] have been considerably implemented in X-ray baggage applications. Liu et al. [10] proposed Single Shot Detector (SSD) and of the study [12] proposed yolo model for natural light image classification and these methods were also applied to object detection. Russakovsky et al. [13] used ImageNet evaluation algorithms for object detection and image classification at large scale, and the fine-tuning training method was used to transfer the deep neural network learned in the ImageNet. Lin et al. [14] employed RetinaNet and achieved comparable detection performance, when trained with 30,000 images synthetically generated via Threat Image Projection (TIP) with 5000 X-ray cargo containers and 544 firearms. Miao et al. [15] proposed class-balanced hierarchical refinement (CHR) method to detect six classes of objects like wrench, pliers, scissor, knife, gun and hammer to achieve overall mean Average precision (mAP) of 0.439 with SIX-ray database. In the work of Akcay and Breckon [16], a detailed survey related to harmful object detection in X-ray baggage security application has been discussed.

3. PROPOSED WORK

In this work, we employ the SSD and RetinaNet object detection algorithm for the detection of threat objects. We used SIX-ray10, the subset of SIX-ray database to detect four different harmful item classes - knife, scissors, wrench, and pliers. The major difficulty using this dataset is the ratio of harmful to non-harmful objects which is around 1:10 in the SIX-ray database. This ratio introduces immense overfitting in the results, which we have tried to overcome through various experimentations, by a trade off of some accuracy. In the training of model, we used the darknet architecture for training SSD and RetinaNet, which allowed faster iterative testing during our experiments, easing the process of both SSD and RetinaNet training. Inception V3 and ResNet-50 architectures have been used to extract important features from the images. SSD stands for Single Shot Detector. It is a light weight object detection algorithm based on deep neural networks (convolutional neural networks to be specific) that provides real-time object detection capability. The SSD algorithm consists of two different steps – extracting feature maps, and applying convolution filters to detect objects. The original paper used pre-trained VGG 16 to extract feature maps. In our case we used Inception V3 as it is smaller than VGG 16 and performs equally well. Retina Net is an object detection algorithm based on convolutional neural networks. This performs exceptionally well even when the object to image pixel ratio is very small thus making it a good option for baggage x-ray detection problem. Retina Net consists of three different parts – Backbone network that extracts features from the images, classification subnet that predicts the classes of the objects detected in an image, and a regression subnet that predicts the location and size of the bounding box created during object detection phase. This thus provides a bounding box along with the class of the object that is identified in the image. SSD is trained with an InceptionV3 backend. ResNet-50 is a pre-trained model which is trained on the ImageNet database. The model consists of 50 layers of convolution and fully connected layer and is capable of predicting one out of one thousand (1,000) different classes that are present in the ImageNet database. This network introduced the concept of Residual Blocks which allows features from previous layers to be fed to later layers so that the model does not forget the lower level features when predicting the higher level features. Inception V3 or Google LeNet is a pre-trained convolutional neural network trained on the ImageNet database. This model consists of convolutional and fully connected layers amounting to a total of three hundred and eleven (311) layers and approximately 24 million parameters. RetinaNet is trained with ResNet-50 backend. These convolutional neural network architectures are trained on the ImageNet database that contains 1.4 million images belonging to a thousand classes. These networks thus have the capability of extracting complex features from the images which can save the time for training the low level feature maps in the convolution layers, and also reasonably focus on the high level features and regression task for computing the bounding box. These models involve enormous matrix computations and contain millions of parameters which need fast computational speeds. We implemented all our experiments on Intel7 processor desktop with 64GB RAM and RTX 2080 Ti Graphical Processing Unit (GPU) for parallel computation of these large matrices.

3.1 Object detection algorithm using single shot detector

SSD is an object detection algorithm that draws bounding boxes around the object of interest or multiple objects. Figure 1 shows the architecture of SSD. The original SSD paper [10] uses VGG-16 in its backend which is trained on the ImageNet database. The last fully connected layers of VGG-16 model are sliced off, as they are specific for the classification tasks. The VGG-16 backend helps in extracting features from the image, since it is trained on the ImageNet database; it can learn kernels to perform this step. In our experimentation, we used InceptionV3 instead of VGG-16 to train the SSD model. As was done in the original paper, we chopped off the final fully connected layers, which were responsible for the classification of an image into a particular class (one out of thousand classes). For the convolution layers, we froze the values of parameters in the convolution layers to obtain good feature maps out of the image given as the input.

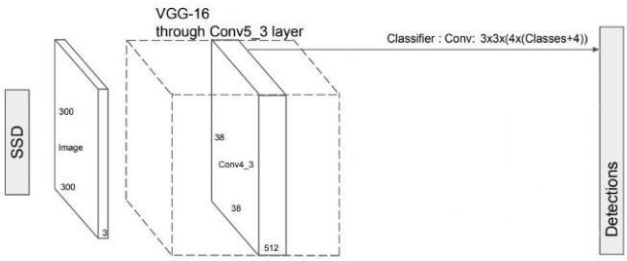

Figure 1. Architecture of SSD from Ref. [17]

3.2 Inception V3

ImageNet database consists of 42 layers, and is used to train on InceptionV3 network. The last layer of ImageNet database is a softmax layer, which contains thousand different nodes and the node represents number of classes in the ImageNet database. The last few layers are not of importance to our classification problem, hence we detached their weights and the remaining were used with weights till the final inception module C of the network. These weights were sufficient for extracting important features, both simple and complex from the images, and encoding them in feature map activations of very less dimensionality than the original image size (224*224*3), which was the original. These features were then passed to the SSD network for computing the bounding box around the four classes of interest such as knife, scissors, wrench and plier So we have proposed 'Model1'of SSD with InceptionV3 as backend. Figure 2 shows the Schematic diagram of inceptionV3.

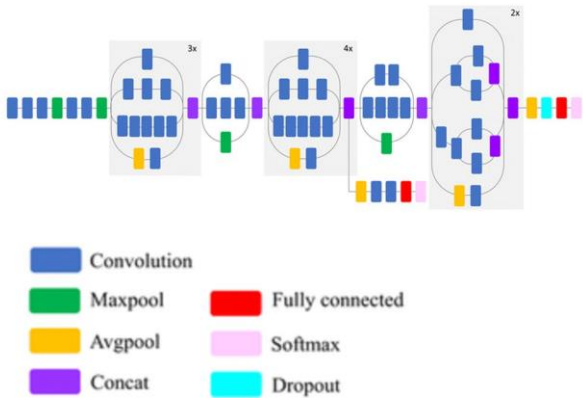

Figure 2. Schematic diagram of inceptionV3 from Ref. [18]

3.3 Model1: SSD with InceptionV3 backend

In this model, we have used Inception V3 to extract the features from an image. Inception V3 architecture is already pretrained and available for various deep learning frameworks. In our model, Inception V3 architecture has been trained by the ImageNet database. This database contains thousands of different classes of images. So this architecture is used to extract features from an input image, and then it can be passed into the SSD architecture. Each image is first passed through the convolution blocks of Inception V3 to get a low dimensional representation of the features of the particular image and then it is passed on to SSD for calculating the bounding box and detecting the class of the object inside a particular bounding box. Figure 3 shows the architecture of SSD with inceptionV3 at the backend.

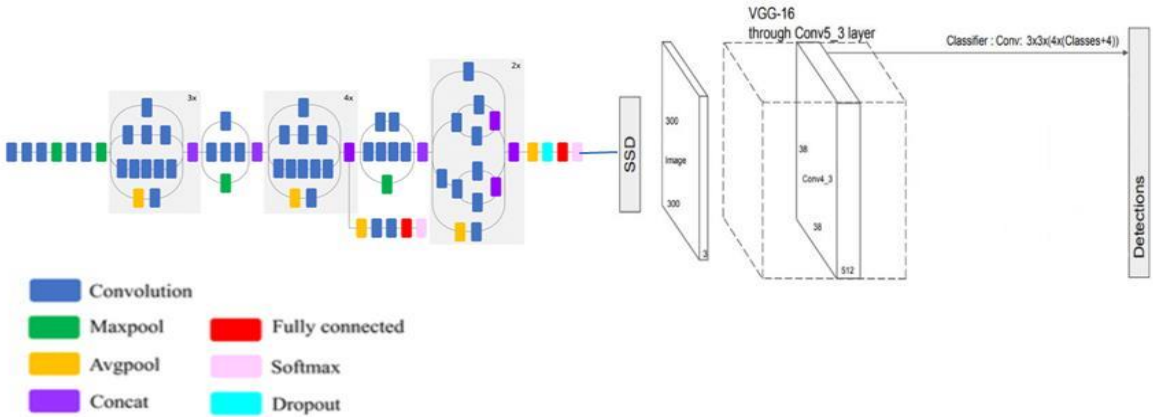

Figure 3. Model1: SSD with Inception V3

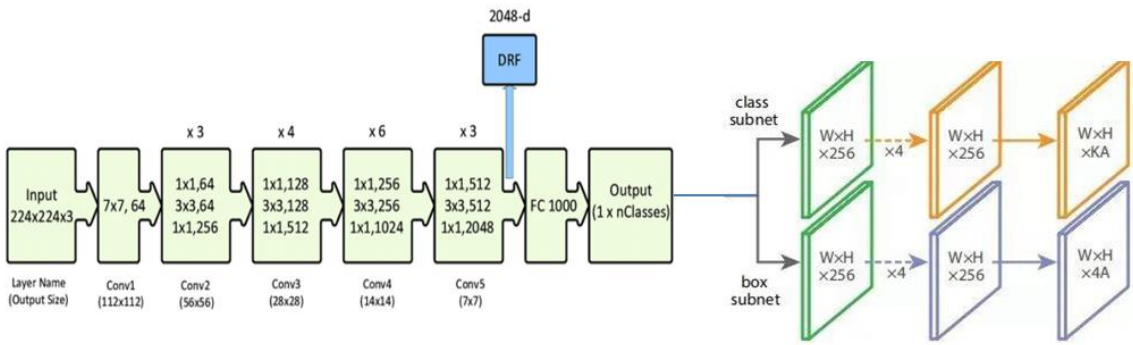

Figure 4. Model2: RetinaNet with ResNet-50

3.4 RetinaNet

RetinaNet is a two stage detector which uses ResNet and Feature Pyramid Network (FPN). The ResNet variant used in the original paper [19] is ResNet-50. In our experimentation also we used ResNet-50 for training the RetinaNet model. The core idea of the RetinaNet model lies in its special loss function - Focal Loss. RetinaNet is capable of real time inference on edge devices and performs better than other object detection algorithms like SSD and YOLO in most of the use cases.

3.5 ResNet-50

ResNet-50 is a CNN, as the name implies, it contains 50 layers deep in its combination of convolution and fully connected layers. A unique feature of this network is the presence of the residual blocks, this allows the model to continually track low level features, while extracting relatively complex features. It improves the overall performance of the model accuracy. With the above framework, we have proposed 'Model2 RetinaNet' with ResNet-50 backend.

3.6 Model2: Retina Net with ResNet-50 backend

The Model2 uses ResNet-50 in its backend that helps in extracting the important features from the image. Similar to RetinaNet, ResNet-50 are also trained by the ImageNet database. In this model the final dense layers were removed from ResNet-50 and the pre-trained weights of the convolutional layers were freezed during training of the RetinaNet model. It saves computational time for extracting features from the image. The input images were first passed through the ResNet pre-trained architecture, which extracts the important low dimensional features from the image, and then it is passed to the convolutional layers of Retina Net for computing the class of object and drawing a bounding box around the particular object of interest, ResNet-50 architecture [20] and RetnaNet architecture [21]. Figure 4 depicts the Model2 architecture.

4. EVALUATION METHODS

The evaluation metric that we presented in this paper is called mean average precision. Mean average precision is the mean of average precision. This measures the quality of the object detector. The value ranges between 0 and 1 and the closer the mAP is to 1 the better is the resulting object detector. mAP score is computed using the AP scores which is in turn computed using three different values – precision, recall and IoU (Intersection over Union). Since we presented mAP score in this paper thus we left out these rather basic metrics which are finally incorporated into the mAP score.

The term 'Precision' refers to accuracy of predictions andterm. 'Recall' refers to the measure of the accuracy with which one can find all predictions. These metrics are represented as Eq. (1) and (2).

$$Precision = \frac{TP}{(TP + FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

where, TP-True positive, FP-False positive, FN-False Negative.

Average precision represents area under precision-recall curve, precision p(r) is a function of recall. This measure is able verify the proposed framework to correctly identifying object proposals of each class. AP (for each class) is computed by sorting the images based on confidence scores and marking the bounding box predictions positive or negative. Afterwards, the precision and Recall computed using Eq. (1) and (2), are used to generate AP as follows

$$AP = \sum_{r=0}^{r=1} P(r)\nabla T(r) \tag{3}$$

where, $\nabla T(r)$ indicates the change in consecutive recall values. Mean Average Precision is computed from Average precision as follows

$$mAP = \frac{1}{T_c} \sum_{s=0}^{T_c-1} AP(s) \tag{4}$$

where, T_c denotes number of classes in the database. This measures the quality of the object detector. The value ranges between 0 and 1 and the closer the mAP is to 1 the better is the resulting object detector.

5. RESULTS AND DISCUSSION

In our experiments, the objects of interest are knife, wrench, pliers and scissors which are to be classified in the SIX-ray database. We obtained the overall mAP of Model1 to be 60.5%. The object wise mAPis listed in Table 1.

Table 1. mAP for four objects for model1

Four Classes (mAP)			
Knife	Wrench	Pliers	Scissors
61.3%	62.5%	58.6%	59.5%

For Model2, the overall mAP is 60.8%. The following Table 2 lists object wise mAP.

Table 2. mAP for four objects for model2

Four Classes (mAP)			
Knife	Wrench	Pliers	Scissors
60.32%	62.15%	63.68%	57.21%

Table 3. Comparison of time performance

Average time performance in seconds			
Average time	Model1	Model2	Ref. [22]
Training	621.80	571.80	677.09
Testing	0.026 s per image	0.019s per image	0.019 s per image

Sigma is used in the loss function used in SSD algorithm. This is called the regularized area. When the value of this hyperparameter is high then the loss function becomes similar to L1 loss which results in a sparse and faster running model, if on the other hand it is small then the loss function is

smoother allowing the model to move down the gradient descent curve to reach the global optimum with high probability.

Gamma is used in the cross entropy loss (used for classification of the type of object inside a bounding box). With increasing gamma the loss reduces when the probability of predicting the correct category is high.

Alpha is called the learning rate of the optimization algorithm, which sets the speed with which the model will learn (descend on the gradient descent curve towards the global optimum value).

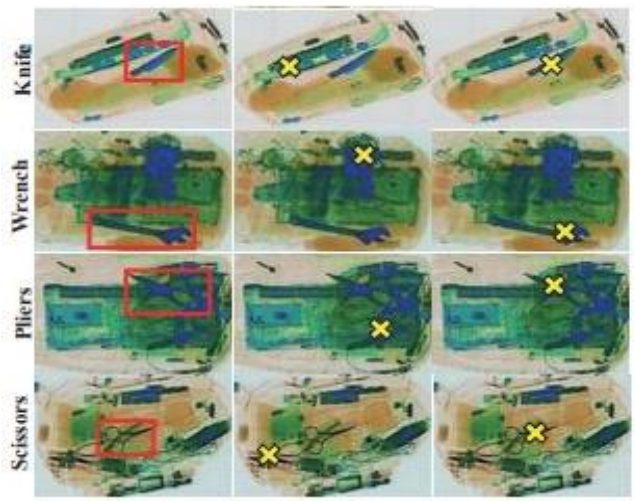

Figure 5. Sample images for the detection of four classes

Real time implementation, we considered time as performance metric. Hassan et al. [22] has implemented Faster RCNN with ResNet-50 for detecting objects in SIX-ray database. They obtained training time 19,600seconds and inference time 0.55seconds per image and as well author proposed method with ResNet-50 backend comparison of time performance with we proposed Model A and Model B mentioned in the Table 3.

From the above table model B obtained best time performance compared to model A with existing methods mentioned in references.

The results of our experimentation showing sample images containing the objects of interest bounded by boxes are shown in Figure 5.

6. CONCLUSIONS

We performed experiments with SIX-ray10 database to develop a CNN model, which can perform classification for four objects namely knife, wrench, pliers and scissors using the SSD and the RetinaNet. In this work, we focused only on detecting and drawing bounding boxes around objects that are harmful. The SIX-ray10 database contains five different types of harmful objects from which we considered four objects. The experiments made use of InceptionV3 and ResNet-50 backends which eased the process of feature extraction which would otherwise have been a computationally expensive process. In earlier reported work [15], the detection of six classes of objects involving wrench, pliers, scissor, knife, gun and hammer using SIX-ray database mAP of 0.439 was achieved. Our experiments using RetinaNet with ResNet-50 backend works slightly better with approximately 0.38% increase in mAP over SSD with InceptionV3 backend. Hence, it can be concluded that ResNet-50 is a better option as it is able to capture better features from an X-ray image than InceptionV3 and can be considered in future work for training object detection models on X-ray image databases to improve the accuracy.

REFERENCES

[1] Mery, D., Svec, E., Arias, M. (2015). Object recognition in baggage inspection using adaptive sparse representations of X-ray images. Image and Video Technology. PSIVT 2015. Lecture Notes in Computer Science, 9431: 709-720. https://doi.org/10.1007/978-3-319-29451-3_56

[2] Khotanzad, A., Hong, Y.H. (1990). Invariant image recognition by Zernike moments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(5): 489-497. https://doi.org/10.1109/34.55109

[3] Chen, W.K. (1993). Linear Networks and Systems. Wadsworth, Belmont, 123-135.

[4] Bastan, M., Yousefi, M.R., Breuel, T.M. (2011). Visual words on baggage x-ray images. Computer Analysis of Images and Patterns, 6854: 360-368. https://doi.org/10.1007/978-3-642-23672-3_44

[5] Turcsany, D., Mouton, A., Breckon, T.P. (2013). Improving feature-based object recognition for X-ray baggage security screening using primed visual words. IEEE International Conference on Industrial Technology (ICIT), Cape Town, pp. 1140-1145. https://doi.org/10.1109/ICIT.2013.6505833

[6] Mery, D., Svec, E., Arias, M., Riffo, V., Saavedra, J.M., Banerjee, S. (2016). Modern computer vision techniques for x-ray testing in baggage inspection. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(4): 682-692. https://doi.org/10.1109/TSMC.2016.2628381

[7] Akçay, S., Kundegorski, M.E., Devereux, M, Breckon, T.P. (2016). Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery. IEEE International Conference on Image Processing (ICIP), Phoenix, Arizona, USA, pp. 1057-1061. http://dx.doi.org/10.1109/ICIP.2016.7532519

[8] Akcay, S., Kundegorski, M.E., Willcocks, C.G., Breckon, T.P. (2018). Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. IEEE Transactions on Information Forensics and Security, 13(9): 2203-2215. https://doi.org/10.1109/TIFS.2018.2812196

[9] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In European Conference on Computer Vision, pp. 740-755. https://doi.org/10.1007/978-3-319-10602-1_48

[10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). SSD: Single shot MultiBox detector. In European Conference on Computer Vision, pp. 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[11] Mery, D. (2013). X-ray testing by computer vision. In Proceedings of the IEEE Conference on Computer

Vision and Pattern Recognition Workshops, pp. 360-367. https://doi.org/10.1109/CVPRW.2013.61

[12] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, pp. 779-788. https://doi.org/10.1109/CVPR.2016.91

[13] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C. (2015). ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 115(3): 211-252. https://doi.org/10.1007/s11263-015-0816-y

[14] Lin, T.Y., Goyal, P., Girshick, R., He, K.., Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, pp. 2980-2988. https://doi.org/10.1109/ICCV.2017.324

[15] Miao, C., Xie, L., Wan, F., Su, C., Liu, H., Jiao, J., Ye, Q. (2019). Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, pp. 2119-2128. https://doi.org/10.1109/CVPR.2019.00222

[16] Akcay, S., Breckon, T. (2020). Towards automatic threat detection: A survey of advances of deep learning within X-ray security imaging. arXiv preprint arXiv:2001.01293.

[17] Wang, Y., Wang, C., Zhang, H., Zhang, C., Fu, Q. (2017). Combing Single Shot Multibox Detector with transfer learning for ship detection using Chinese Gaofen-3 images. In IEEE Progress in Electromagnetics Research Symposium-Fall (PIERS-FALL), Singapore, pp. 712-716. https://doi.org/10.1109/PIERS-FALL.2017.8293227

[18] Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., Zhang, Y. (2018). Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. Remote Sensing, 10(7): 1119. https://doi.org/10.3390/rs10071119

[19] Zhang, H., Chang, H., Ma, B., Shan, S., Chen, X. (2019). Cascade RetinaNet: Maintaining consistency for single-stage object detection. arXiv preprint arXiv:1907.06881.

[20] Mahmood, A., Ospina, A.G., Bennamoun, M., An, S., Sohel, F., Boussaid, F., Kendrick, G.A. (2020). Automatic hierarchical classification of kelps using deep residual features. Sensors, 20(2): 447. https://doi.org/10.3390/s20020447

[21] Milton, M.A.A. (2019). Towards Pedestrian Detection Using RetinaNet in ECCV 2018 Wider Pedestrian Detection Challenge. arXiv preprint arXiv:1902.01031

[22] Hassan, T., Akcay, S., Bennamoun, M., Khan, S., Werghi, N. (2020). Cascaded structure tensor framework for robust identification of heavily occluded baggage items from X-ray scans. arXiv preprint arXiv:2004.06780.