Automatic Threat Detection in Baggage Security Imagery using Deep Learning Models

Aditya Mithal
Department of Computer Science Engineering
Birla Institute of Technology and Science, Pilani, India
f20170157@pilani.bits-pilani.ac.in

Manit Baser
Department of Electrical and Electronics Engineering
Birla Institute of Technology and Science, Pilani, India
f20170370@pilani.bits-pilani.ac.in

Dhiraj
Intelligent Systems Group
CSIR-Central Electronics Engineering Research Institute, Pilani
dhiraj@ceeri.res.in

2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS) | 978-1-7281-8524-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICIIS51140.2020.9342691

***Abstract*—Automating object detection for surveillance purpose and threat detection is beneficial as it may compensate for the human error and will save time, which is of significant economic value. For the end-to-end classification process and feature extraction, the CNN approach requires large amounts of data. To overcome this limited availability of data, we have presented a transfer learning approach with various object detection models for single and multiple detections on two types of the dataset: Single-channelled(GDXray dataset) and Multichanneled(SIXray dataset). We have presented comparisons between the various models(Faster R-CNN with ResNet50, SSD with VGG16, YOLOv3 with ResNet50, and RetinaNet with ResNet50). The best results were achieved on Faster-RCNN(ResNet50) with 0.966 mAP for the four-class object detection problem(GDXray Dataset) and 0.845 mAP for the two-class object detection problem(SIXray Dataset).**

Index Terms- Surveillance, Faster R-CNN, RetinaNet, YOLOv3, SSD, X-Ray, Deep Learning, Transfer Learning, SVM, ConvNets, Object Detection.

I. Introduction

Airport baggage security is one of the crucial jobs while onboarding passengers on their respective planes. This job gets carried out by manual surveillance of the x-ray images. The task of manually detecting weapons and explosives is a very exhausting and demanding task, where the slightest of errors may lead to loss of life. It is a complex task due to the various types of baggage all around the world and tight packing.

The increased passenger throughput across the globe requires a faster and robust process for baggage screening. Fig. 1 shows the queue at the airport. Automating this process of surveillance can have a significant impact on the overall functioning of air-travel. It may compensate for the human error, which creeps in due to the close packing and the tremendous amount of baggage checked at the airports every day. Saving even a small portion of time can substantially benefit the passengers and the authorities. Hence, air travel can be made more convenient for everyone.

Many researchers are working on automating this detection process of dangerous objects[3]. The Rapiscan 620DV X-ray screening system presents a design suitable for aviation and high-security applications, which serves the purpose of data collection[8]. Previous work on image enhancement[1, 22] classification[21] and detection[5] tasks has provided a pipeline to explore further the real-time implementations of automated models for security screening. Given below are some of the explorations made in the classification and detection domain.

Fig. 1: Queue at Airport Security[15]

A. Classification:

Various applications use machine learning to detect and classify dangerous objects using hand-crafted features, and these features get fed to conventional classifiers like Support Vector Machine (SVM). More specifically, the use of an SVM and Bag-of-Visual-Words (BoVW) on X-ray images with feature representations. The use of scale-invariant feature transform (SIFT), in combination with the Difference of Gaussians (DoG), is given in [11].

Based on studies, the CNN approach with transfer Learning outperforms the BoVW method in the performance metric. [12] surveys ten different techniques of classification approaches using X-ray baggage datasets.

B. Detection:

Object classification is an essential factor in the identification of objects and differentiates between them being a threat or not. However, an essential job of detection remains, where regions of interest (ROI) are concerned. The use of bounding boxes often gets used for object localization within the image. [5] uses a linear SVM classifier and provides a sliding-window technique along with a histogram of oriented gradients (HOG). [4] explores the detection techniques in grayscale images via combinations of hand-curated features descriptor and detector, alongside the branch-and-bound method and an SVM classifier. Recent progress in the CNN literature related to object detection is presented in [2] and BoVW techniques in [13].

We have trained various deep learning models for threat detection in both X-ray and colored images. We present the use of Faster R-CNN, RetinaNet, YOLOv3, and SSD models and a metric to exhibit the results obtained after training the models on the two datasets. Section II defines the preprocessing on the GDXray[18] and SIXray[14] dataset for training and testing purposes. Section III presents classification techniques. Section IV introduces the various models we have trained and emphasizes more on their architecture. Section V presents the metric based on which we have evaluated the models. The section further exhibits the comparison between the various models and elaborates more on their performance. Section VI concludes the paper.

II. Dataset and Preprocessing

We have used two datasets, the GDXray dataset, and the SIXray dataset, for our training and testing purposes. The GDXray dataset provides grayscale images, and SIXray provides pseudo-colored images. More specifically, we have selectively annotated and augmented the images of guns, knives, shaving blades, and shurikens in the GDXray dataset and guns and scissors in the SIXray dataset. Fig. 2 and Fig. 3 shows the bounding boxes on threat objects in the GDXray and the SIXray dataset, respectively. The post augmentation counts are as presented in Table I and Table II. LabelImg is used for annotation of the data and converting it to Pascal Visual Object Classes (VOC) format. Albumentations is used further for augmentation purposes. The division of the augmented dataset into training and testing data is 80:20.

TABLE I: The GDXray dataset[18]

Class of Object	No. of Instances
Shaving blade	851
Knife	163
Shuriken	335
Gun	793

TABLE II: The SIXray sub-dataset[14]

Class of Object	No. of Instances
Scissors	214
Gun	656

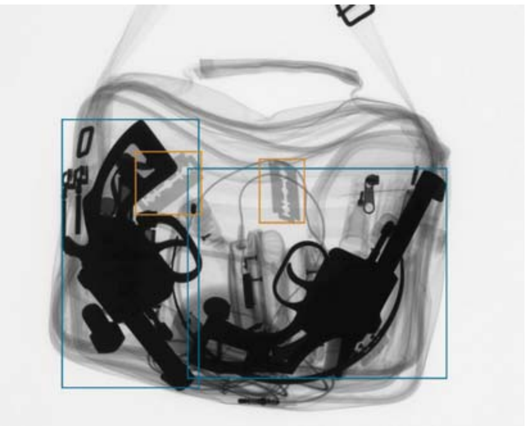

Fig. 2: Example of Labelled GDXray Dataset Image[18]

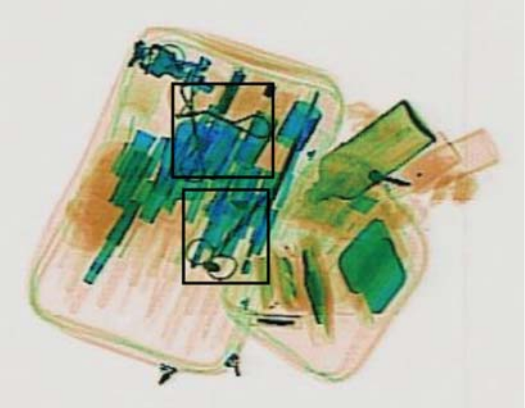

Fig. 3: Example of Labelled SIXray Dataset Image[14]

III. Classification

A. Convolutional Neural Networks

Convolutional Neural Network is a class of deep learning algorithms used for analyzing visual imageries. It expresses a single differentiable score function from the raw input image. Stacking up of Convolutional Layer, Pooling Layer, and Fully-Connected Layer builds up a ConvNet Architecture.

Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton developed AlexNet[7], a significant breakthrough in Convolutional Networks in Computer Vision. AlexNet proposed to stack Convolutional layers on top of each other, making it deeper instead of having a single Convolutional layer followed by a Pooling layer. The neural network has five convolutional layers with sixty million parameters. It gets supported by max-pooling layers, three FC layers, and a concluding 1000-way softmax.

GoogLeNet[20] introduced a futuristic algorithm for detection and classification. The newly designed Inception module with Average Pooling instead of stacking up Fully Connected layers reduced a tremendous amount of parameters in the model (4M, as opposed to AlexNet with 60M). GoogLeNet has 1×1 convolution and 128 filters for rectified direct activation and dimension reduction and a fully connected layer with 1024 units. Softmax loss with a linear layer acts as the classifier.

VGGNet[19], developed by Karen Simonyan and Andrew Zisserman, was the product of rigorous evaluation of increasing depths in the network. It showed a significant improvement in accuracy on increasing the depth to 16-19 weight layers. All Convolutional/FC layers have 3×3 filters to reduce the parameters in such dense networks. An enhancement of this model was to reduce the number of parameters. It showed that removing the FC layers did not downgrade the performance; it significantly reduced the parameters.

The ResNet[6] is based on VGG's full 3×3 Convolutional layer design: the 3×33×3 max-pooling layer supports the 7×77×7 convolutional layer with a stride of 2 and 64 output channels. It implements the ReLu activation function and Batch Normalization at the end of every Convolutional layer. By configuring different residual blocks and channels in the model, we can build various ResNet models, such as the deeper 152-layer ResNet-152. ResNets are considered to be by far state-of-the-art models.

B. *Transfer Learning*

The Transfer Learning approach concentrates on applying the previously acquired knowledge to a problem with related characteristics. CNN architectures such as GoogLeNet, VGG, and ResNet get trained on massive datasets containing thousands of distinct class labels with millions of data samples. Limiting dataset, parameter optimization, and time constraint give rise to the transfer learning approach.

The higher layers in the CNN Network act relatively more particularised to the actual objective, whereas the lower layers present a broad feature extraction tendencies. We fine-tune the weights of a pre-trained CNN, generalized for object detection towards our classification domain.

IV. Object Detection Models trained and their performance

A. *Various Models and their architecture*

1. Faster R-CNN:- It is the most widely used model of the Region-based CNNs (R-CNNs)[17]. Faster R-CNN utilizes Region Proposal Network (RPN), which replaces selective search (in Fast R-CNN). The input images pass into the convolutional network, which in turn returns the feature maps. The RPN apply on these feature maps and obtain the object proposals. ROI pooling layer brings down all the proposals to the same size. Then the proposals are transferred to a fully connected layer to classify and predict the bounding boxes. Fig. 4 demonstrates the overall process of detecting objects from an input image using Faster R-CNN.

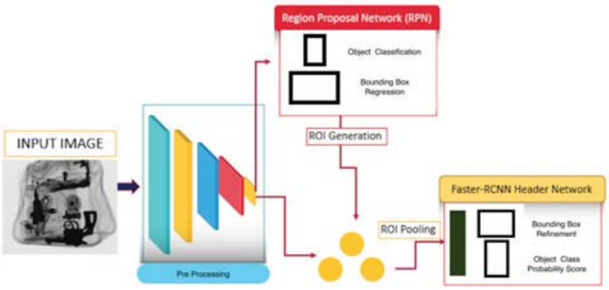

Fig. 4: The Faster R-CNN object detection network[17]

2. YOLO v3: YOLO v3[16] adopts a variant of Darknet, where the first 53 layer network is pre-trained on Imagenet. Also, 53 more layers are stacked onto it for the task of detection, hence having a 106 layer fully convolutional underlying architecture, which is why YOLO v3 is slower as compared to YOLO v2. YOLO v3 predicts by downsampling the input image dimensions by 32, 16 and 8, as shown in Fig. 5. Detections at various layers clear out the issue of detecting small objects. Along with the unsampled layer, the previous layers store the fine-grained features that help identify small objects.

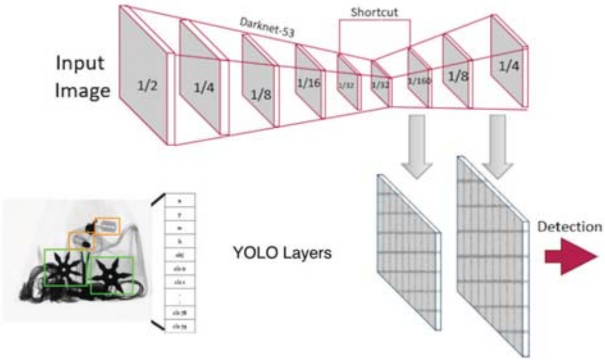

Fig. 5: YOLOv3 object detection network[16]

3. Single Shot MultiBox Detector: SSD's[10] architecture is built on that of VGG-16. A set of auxiliary convolutional layers is used in place of the primary VGG FC layers for feature extraction at various scales and progressively reduce the input size for each subsequent layer as shown in Fig. 6. Szegedy's work on MultiBox inspires the bounding box regression technique of SSD.

4. RetinaNet: RetinaNet[9] is an outstanding one-stage object detection model that operates extremely well on the detection of dense and small-scale objects. Aerial and satellite imagery often use this model for detection purposes. It uses ResNet and FPN as a backbone for feature extraction, as demonstrated in the first two stages of Fig. 7. For every anchor box and object class at each spatial location, the classification network predicts the object's probability (Fig. 7). For each ground truth object, the regression network regresses the offset for the bounding boxes from the anchor boxes.

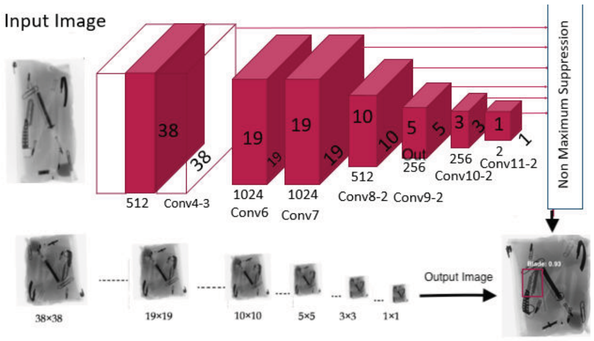

Fig. 6: SSD object detection network[10]

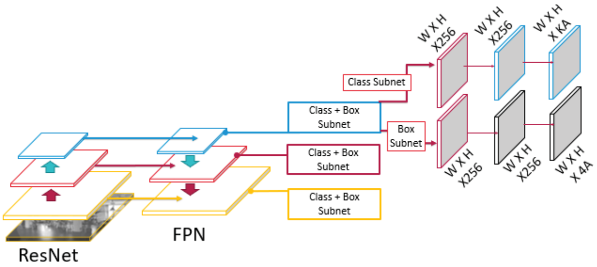

Fig. 7: RetinaNet object detection network[9]

V. Evaluation

The various models' performance on both the datasets is evaluated based on their mean average precision (mAP) values accepted in the Pascal VOC object detection challenge. For the calculation of mAP values, we first need to calculate the following parameters given ahead. The confidence score is the probability of an anchor box containing an object. Intersection over Union (IoU) is formalized as:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}$$

where B_{gt} is the ground truth box and B_p is the predicted bounding box.

Detection is a true positive (TP) when it satisfies all the following cases: 1) confidence score $>$ threshold, 2) the predicted class should be the same as ground truth class, and 3) IoU of the predicted bounding box $>$ threshold (0.5) with the ground truth. If either of the second or third conditions is not satisfied, then the case is considered as a false positive (FP). If the confidence score of detection, which is presumed to detect a ground truth $<$ threshold, it is taken to be a false negative (FN). If the confidence score of detection, which is not presumed to detect anything $<$ threshold, is considered a true negative (TN). Precision and recall are given by:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The interpolated precision for a recall level r is formalized as:

$$p_{interp}(r) = \max p(r')$$

for any level $r' \geq r$.

Average Precision (AP) is the calculated area under the interpolated precision-recall graph, given as:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1})$$

and r_1,r_2,... are the recall levels at which the precision was interpolated. Finally, the mean average precision (mAP) is given by:

$$mAP = \frac{\sum_{i=1}^{M} AP_i}{M}$$

where M defines the number of classes. Table IV and V shows the results for the GDXray and SIXray dataset, respectively.

We have employed a transfer learning approach for object detection using the pre-trained networks on the Pascal-VOC dataset. We observe that Faster R-CNN on ResNet50 outperforms on both GDXray Dataset (mAP: 0.966) and SIXray Dataset (mAP: 0.845). Figures 8, 9, 10, and 11 show the bounding boxes over detected objects along with the classification labels and the confidence scores.

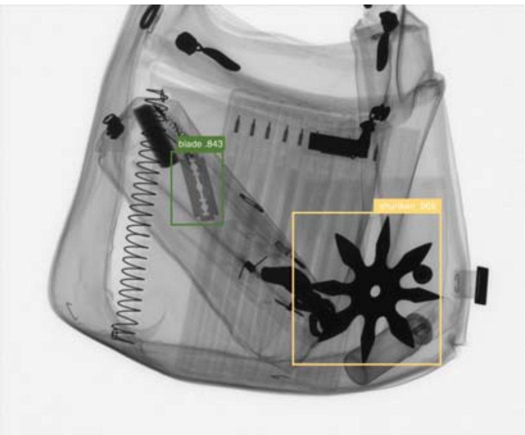

Fig. 8: Resulted Label and Score on GDXray Dataset Image[18]

VI. Conclusion

To improve dangerous object detection in tight-packed items of baggage, we explored and experimented with various approaches.

First, we presented an overview of different classification and detection methods and the same fields' progress. Transfer learning has made a massive impact on classification methodology and has made it significantly robust and efficient.

Secondly, we annotated and augmented the grayscale images of the GDXray dataset using multiple tools in Pascal VOC format. We also used the colored SIXray sub-dataset

TABLE III: Detection results on the GDXray dataset[18]

Model	Network	mAP	gun	shuriken	knife	shaving blades
SSD[10]	VGG16	0.959	0.986	**1.000**	0.951	0.899
YOLO v3[16]	Darknet53	0.902	0.907	**1.000**	0.839	0.865
Faster R-CNN[17]	ResNet50	**0.966**	0.980	**1.000**	**0.992**	0.892
RetinaNet[9]	ResNet50	0.933	**0.990**	0.999	0.789	**0.951**

TABLE IV: Detection results on the SIXray dataset[14]

Model	Network	mAP	scissors	gun
SSD[10]	VGG16	0.839	0.788	0.889
YOLO v3[16]	Darknet53	0.754	0.781	0.725
Faster R-CNN[17]	ResNet50	**0.845**	**0.798**	**0.891**

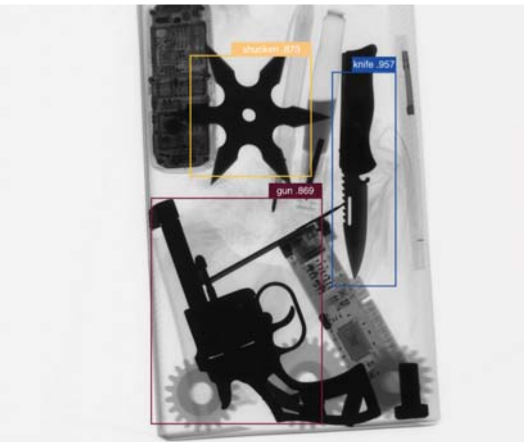

Fig. 9: Resulted Label and Score on GDXray Dataset Image[18]

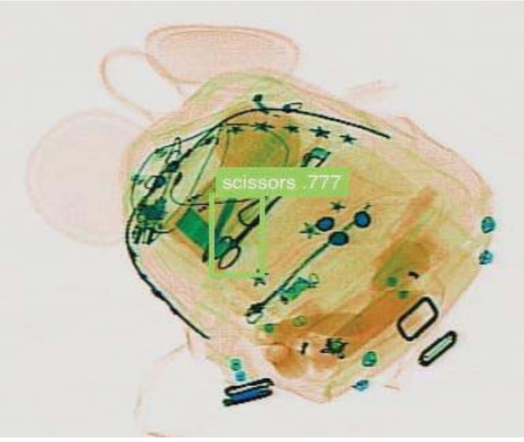

Fig. 11: Resulted Label and Score on SIXray Dataset Image[14]

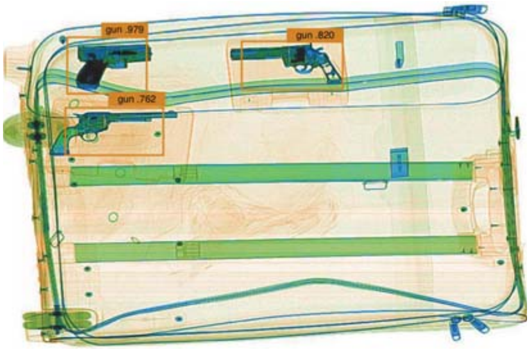

Fig. 10: Resulted Label and Score on SIXray Dataset Image[14]

to train and test the models. We have presented various modernistic object detection models along with discussing their architecture. These models are Faster R-CNN, SSD, YOLO v3, and RetinaNet.

We experimented on the discussed models with transfer learning and evaluated them based on their mAP values for Pascal VOC format. In the GDXray dataset, which contains grayscale images, the mAP values of each of the four models cross 0.9 and exhibit significant results for each of the objects' classes. The SIXray sub-dataset, which contains colored images, has mAP values of 0.845 for Faster R-CNN ResNet50, 0.839 for SSD VGG16, and 0.754 YOLOv3 Darknet53. Tables III and IV show that there is a significant difference in mAP values when the models get trained on either dataset. It may indicate that detection is more robust on **single-channel** images. Further testing on bigger datasets may lead to improved weights for the models.

The trade-off between speed versus accuracy among the various models is presented and evaluated. The detection is Faster R-CNN takes place in two stages, and the time taken is decided on the RPN proposed number of regions. In comparison, SSD computations take place in a single stage. In YOLOv3, object classification takes place at the same time as that of bounding box regression.

As observed from the mAP evaluation, Faster R-CNN exhibits maximum accuracy in object detection compared to YOLOv3 and SSD. SSD shows similar results with slightly off accuracy but is relatively faster than Faster R-CNN. One reason for this is that in SSD, the RPN, and the classification and localization computations entirely happen in a single stage. Also, Faster R-CNN detects smaller objects like knives in the GDXray dataset more efficiently. The difference reduces the detection of larger objects like guns.

References

[1] B. R. Abidi et al. “Improving Weapon Detection in Single Energy X-Ray Images Through Pseudocoloring”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 36.6 (2006), pp. 784–796.

[2] S. Akçay et al. “Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 1057–1061.

[3] Samet Akcay et al. “Using Deep Convolutional Neural Network Architectures for Object Classification and Detection Within X-Ray Baggage Security Imagery”. In: *IEEE Transactions on Information Forensics and Security* PP (Mar. 2018), pp. 1–1.

[4] Muhammet Bastan, Wonmin Byeon, and Thomas Breuel. “Object Recognition in Multi-View Dual Energy X-ray Images”. In: Jan. 2013, pp. 130.1–130.11. ISBN: 1-901725-49-9. DOI: 10.5244/C.27.130.

[5] Thorsten Franzel, Uwe Schmidt, and Stefan Roth. “Object Detection in Multi-view X-Ray Images”. In: *Pattern Recognition*. Ed. by Axel Pinz et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 144–154. ISBN: 978-3-642-32717-9.

[6] K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems* 25 (Jan. 2012). DOI: 10.1145/3065386.

[8] Kevin Liang et al. *Toward Automatic Threat Recognition for Airport X-ray Baggage Screening with Deep Convolutional Object Detection*. Dec. 2019.

[9] Tsung-Yi Lin et al. “Focal Loss for Dense Object Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (July 2018), pp. 1–1. DOI: 10.1109/TPAMI.2018.2858826.

[10] Wei Liu et al. “SSD: Single Shot MultiBox Detector”. In: *ArXiv* abs/1512.02325 (2016).

[11] David G. Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision* 2 (1999).

[12] D. Mery et al. “Modern Computer Vision Techniques for X-Ray Testing in Baggage Inspection”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.4 (2017).

[13] Domingo Mery, Erick Svec, and Marco Arias. “Object Recognition in Baggage Inspection Using Adaptive Sparse Representations of X-ray Images”. In: Feb. 2016. ISBN: 978-3-319-29450-6. DOI: 10.1007/978-3-319-29451-3_56.

[14] Caijing Miao et al. “SIXray: A Large-Scale Security Inspection X-Ray Benchmark for Prohibited Item Discovery in Overlapping Images”. In: June 2019, pp. 2114–2123. DOI: 10.1109/CVPR.2019.00222.

[15] Wayne Mullins. https://securitytoday.com/blogs/reaction/2017/05/Airport-Security-and-Screening-How-the-Rich-and-Famous-Bypass-the-Queues.aspx. 2017.

[16] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *ArXiv* abs/1804.02767 (2018).

[17] S. Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149.

[18] Vladimir Riffo, Hans Lobel, and Domingo Mery. “GDXray: The Database of X-ray Images for Nondestructive Testing”. In: *Journal of Nondestructive Evaluation* (May 2015).

[19] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv 1409.1556* (Sept. 2014).

[20] C. Szegedy et al. “Going deeper with convolutions”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9.

[21] D. Turcsany, A. Mouton, and T. P. Breckon. “Improving feature-based object recognition for X-ray baggage security screening using primed visualwords”. In: *2013 IEEE International Conference on Industrial Technology (ICIT)*. 2013, pp. 1140–1145.

[22] Zhiyu Chen et al. “A Combinational Approach to the Fusion, De-noising and Enhancement of Dual-Energy X-Ray Luggage Images”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Workshops*. 2005, pp. 2–2.