Threat Object Detection in X-ray Images Using SSD, R-FCN and Faster R-CNN

Jola Koçi
Department of Computer Engineering
Epoka University
Tirana, Albania
jkoci@epoka.edu.al

Ali Osman Topal
Department of Computer Engineering
Epoka University
Tirana, Albania
aotopal@epoka.edu.al

Maaruf Ali
Department of Computer Engineering
Epoka University
Tirana, Albania
mali@epoka.edu.al

***Abstract*—Inspection of baggage for threat objects (such as explosives) using X-ray images is a priority task for preventing terrorist attacks. Currently, the checking of baggage is based on a semi-automated system that consists of both a human operator and also assisted by image detection. In order to apply effective object detection, accurate object detection models like the: Single Shot Detector, Region-based Fully Convolutional Network (R-FCN) and Faster R-CNN (Region-based Convolutional Neural Network) must be considered. These models are used by applying different techniques of feature extraction, such as: Inception-v2, MobileNet-v2 and ResNet101. In conclusion, the best detection was achieved by the combination of the Faster R-CNN detection models and the ResNet101 feature extractor, achieving an accuracy of 87.58% (±0.75% error margin).**

Keywords—object detection, threat objects, X-ray images, data stimulation, data augmentation

I. Introduction

In every country, one of the highest priorities is security. So as to accomplish better outcomes in automated methods of being secure, object detection is a needed task. Now almost every airport has a semi-automated scanner system, showing that automatisation is becoming widespread and proving its improving reliability. Automation is mainly based on scanning luggage into X-Ray scanners [1] and then the next part is left to the security guard to determine whether there is any potential threat revealed. One main drawback with this strategy is that it leaves a window open for human error, which sometimes can be critically catastrophic. This is why, automatic object-detection is a promising approach, since it does not suffer from these human limitations. Even though detection of threat objects based on material properties, most used being that of metal object detection, is already used in practice. However, image-based detection of objects in X-ray images is not yet common [2].

II. Literature Review

During recent years, automated inspection models have been created and tested in detection of threat objects.

A. Object Detection

There is a lot of research that uses many different methods to get better views from X-Ray images. Reference [3], suggested using image segmentation before going into object detection. Another approach [4] introduced pseudo-colour algorithms for weapon detection. Also [5] used automated detection of threat objects with single or multiple views on a single or dual-energy X-ray system.

After researching the visualisation of X-Ray images, the next step is to investigate some effective threat object classification approaches. Classification of X-ray images could be a difficult task when working with images. The explanation is that images from X-rays can normally be occluded by different objects and cannot be recognised simply once scanned [6]. To resolve this drawback, researchers planned to change the detection of threat objects. Reference [7] initially suggested working with an SVM (Support Vector Machine) in detection of threat objects. The planned technique was evaluated in gun recognition and mobile detection. This methodology was proven to be very effective, since over time it was the main ideology used in many other research. Another research [8] experimented upon four different threat classes such as blade, gun, knife and *shuriken*. A "*shuriken*" (Japanese: 手裏剣) literally means a "hidden hand blade", is a Japanese concealed weapon. After using the ideology of Bags of Words (BoWs) and codebooks, introduced in [9], they applied the new concept of transfer learning into their data. [9] defined transfer learning or information transfer as a way of re-using a model trained from an outsized dataset to unravel another drawback or task. Their experimental results showed that by using transfer learning with information augmentation and fine-tuning, threat objects may be classified with 99.5% accuracy. This can be considered the method with the highest accuracy. However, this accuracy was limited by being tested on only four different types of objects. To make it more reliable, [9] considered future work for their method, testing with more and varied types of threat objects. This approach is being taken into consideration in this paper.

B. Dataset Decription

Based on the literature review, it is concluded that most of the recent approaches are mainly being tested using the GDB X-ray dataset of images [10]. On that dataset as shown [9], the highest possible accuracy was achieved with 99.5%. Based on the future work [9], it is suggested to try their recommended approach using another dataset. Taking into consideration their suggestion and due to the lack of other data, the dataset for this small-scale study included some simulated data using other classes of threat objects. The GDB X-Ray dataset, as mentioned, has overall distribution as shown in Table 1.

TABLE I. GDXray luggage Dataset Object Distribution [10].

Threat Object	*Count of Objects in all Images*
Handgun	250
Knife	250
Razorblade	250
Shuriken	250

978-1-7281-8488-3/20/$31.00 ©2020 IEEE

III. METHODOLOGY

The methodology used was an experimental one, since there will be different approaches applied to achieve a higher accuracy. The threat objects that was taken into consideration are: handguns, knives, blades, *shurikens*, wires, batteries and mortars. The methodology was divided into two main steps: processing of images and object detection.

A. X-ray Image Processing

The main purpose of this processing phase is to be able to deliver into training an image which would assure the detection algorithm the best accuracy possible.

The main part of image processing in this paper is data simulation. Data simulation as a concept consists of generating new random images based on other old images. In order for the data simulation on the images not to have interference with the object detection model accuracy, the simulation was based on [23]. The object simulation inside the image was achieved through manual insertion as shown in Fig.1.

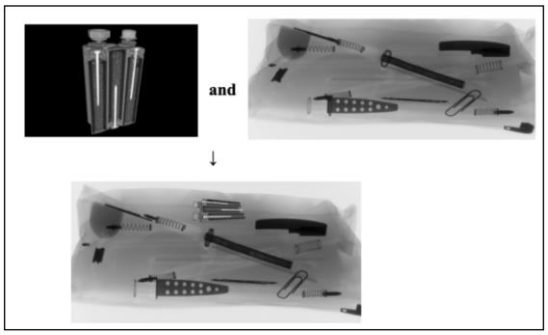

Fig. 1. The threat objects (top left), background (top right) and bag with the objects (bottom).

After applying the data simulation, the new dataset consisted of a total of 1450 threat objects in the images. The separation between the different threat classes are shown in Table 2.

TABLE II. THE DISTRIBUTION OF THE THREAT OBJECTS IN THE SIMULATED DATASET.

Threat Object	*Count of Objects in all Images*
Handgun	250
Knife	250
Razorblade	250
Shuriken	250
Battery	150
Wires	150
Mortar	150

The next step of the image processing phase is based on the labelling of the images. The labelling is considered as the returning point from the image dataset into a features dataset where from every image is extracted the following information (such as the image label and bounding box pixels) in a .XML format.

B. Transfer Learning.

Transfer learning is a machine learning technique where a trained model on one task is reused on a second training model. Transfer learning only works in deep learning if the model features learned from the first task are general, since it is trying to detect from another predefined dataset. In this way there is no need to train the data from the beginning, since that would require a lot of time. By using these predefined training models that are also trained, the training step can be quite fast and more accurate since even the amount of data is larger. In other words, instead of using random initialisation of each of the features, transfer learning can be used as an already initialised model that has approximated the parameters, since the data is already trained.

The most important decision when using transfer learning is mostly based on the selection of object detection models and feature extraction strategies. An object detection model is trained to detect the presence and placement of multiple categories of objects. The models that will be taken into consideration in this thesis are: Faster-RCNN, Single Shot Detector (SSD) and Region-based Fully Convolutional Networks (R-FCN).

1) Faster R-CNN

This CNN network is defined as the pre-trained model which is used for transfer learning and as a feature extractor. The reason why Faster R-CNN uses transfer learning is because features that are learned by a particular layer are generally transferable to other tasks outside the predefined network. This makes it more reliable. After the features are extracted, they are passed to the next phase of the RPN (Region Proposal Network). In RPN, a set of anchors are accepted and after that it outputs the suggested proposals for where the image would be. RPN does not specify where the object really is, but just proposes main areas where it could be. The next step consists of RoIP (Region of Interest Pooling), that extracts feature maps from each proposed region that has come as an input. Finally, a R-CNN is used to make the final label predictions and further define the locations for better accuracy.

2) Single Shot Detector

The SSD model adds a few component layers to the furthest limit of a base system. The primary methodology of SSD is creating classification features and box counterbalances for a fixed arrangement of default jumping boxes utilising little convolutional channels applied to the highlight maps. To accomplish high detection accuracy, SSD produces expectations of various scales from the included maps of various scales and expressly isolates forecasts by viewpoint proportion.

3) Region-based Fully Convolutional Network

Region based Fully Convolutional Networks or *R-FCN* is a R-CNN based network approach. Like the other R-CNN based detectors, even R-FCN processes the object detection in two different stages. The first stage consists

of generating region of interest (RoI) proposals areas and the second stage makes classification and localisation predictions from the ROIs. The typical use case for convolutional layers is for image data where, as required, the features are local.

C. Detection Architectures

Faster R-CNN, Single Shooting Detectors and the R-FCN can be considered the three architectures of CNN-based detectors. By changing their parameters, their performance can significantly be changed. In order to determine what parameters each object detection algorithm should take, there can be used some already defined models. Some of these models are MobileNet, Resnet and Inception. The following feature extractor models were taken into consideration:

i. Inception-v2 [11]: Inception v2 and Inception v3 were described in [11]. The reason why version two of the Inception-based architecture is chosen is mainly due to the fact of batch normalisation and reduction of overfitting. When talking about batch normalisation it can be referred to the fact of normalised inputs to every layer before sending the data to the activation function. On the other hand, it can also affect the lowering of overfitting levels. The other versions of the Inception architecture become more and more reductive in each layer by making the convolution even 1×n then n×1. This could mean there is a lot of overfitting of the data, something that should be avoided. This is why the 3×3 architecture seems the best choice between the other versions.

ii. ResNet-101: ResNet is an architectural design that is mainly used due to its deeper training layers and good performance. This architecture performs very well in image recognition because it achieves deeper training. Most of the other architectures cannot go into the deeper layers due to the loss of accuracy and increased computational cost. RestNet can even go to the deepest layers without having any decrease in its performance. This is mainly due to the fact that it works based on adding skip connections and then forming a residual block.

iii. MobileNet-v2 [12]: MobileNets are little, low-latency, low-power models parameterised to satisfy the resource constraints of a range of use cases. It is a very effective feature extractor for object detection and segmentation. For detection, once paired with a Single Shot Detector, MobileNetV2 is 35% quicker with an equivalent accuracy than using MobileNetV1.

Table 3 shown a combination for each object detection model with its own object detection architecture.

TABLE III. THE SELECTED OBJECT DETECTION MODELS AND RESPECTIVE ARCHITECTURES.

Object Detection Model	*Object Detection Architecture*
SSD	Inception-V2
SSD	MobileNet-V2
R-FCN	ResNet-101
Faster R-CNN	ResNet-101
Faster R-CNN	Inception-V2

IV. RESULTS AND DISCUSSION

In order to fully complete the transfer learning approach, the Tensorflow API was utilised.

A. Simulation Results

In order to make sure that the approach chosen, to make the new simulated images not affect the accuracy of the detection models, the following procedure was taken explained next. Since the missing objects from the dataset were: battery, wire and mortar, the testing cannot be done by using those objects. Instead, in the dataset will be stimulated the handgun as an object where it is missing and then will be trained. So, the same algorithm will be chosen to initially detect the old dataset and the new one together with the simulations and will come into a conclusion of how this simulation affects the accuracy of the algorithm. The stimulation is done as shown in Fig. 2.

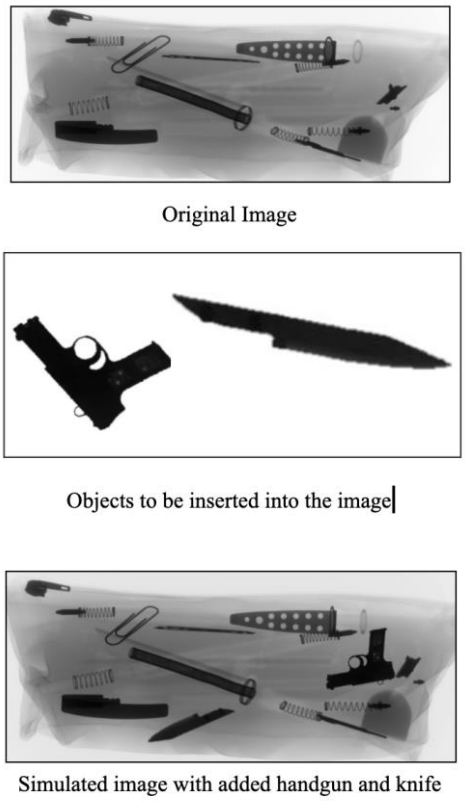

Fig. 2. Stimulated Results.

As mentioned previously, the dataset contains 250 images for each of the classes including: handgun, knife,

razor-blade and *shuriken*. After making the insertions into the images of the threat objects of handgun and knife their count in total went up to: 320 handguns and 300 knives in total. The results after training in Faster R-CNN previously and after are shown in Fig3.

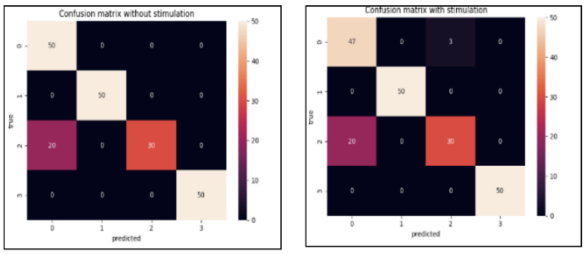

Fig. 3. Confusion Matrix of Faster R-CNN with Inception-v2 applied to the old dataset (on the left) and to the simulated dataset (on the right), both containing 50 objects in testing.

Based on the results from the results of the confusion matrix it can be seen that the Faster R-CNN algorithm applied to the data without stimulation consists of only 30 FN (false negative) values where the knives were predicted as handguns. The confusion matrix consists of the classes distributed as follows: 0: handgun, 1: razor-blades, 2: knife, 3: *shuriken*. Translating this result in terms of accuracy, it would mean that the results were as following:

✓ Accuracy of data without stimulation: 90%
✓ Accuracy of data with simulation: 88.5%
✓ This leads to an approximate absolute error of 1.6%.

B. *Object Detection Results*

The results of the experiment were calculated based on different metrics, but the main one considered was the Accuracy. The experiments were done using (1).

$$mAP = \frac{1}{C}\sum_{i=1}^{C} A\,P_i \qquad (1)$$

In the graphical plots shown in Figs. 4-6, the monitoring of the output was taken for every step for each of the chosen algorithms, noting that for each algorithm is passed in 2K or 2,000 steps in training.

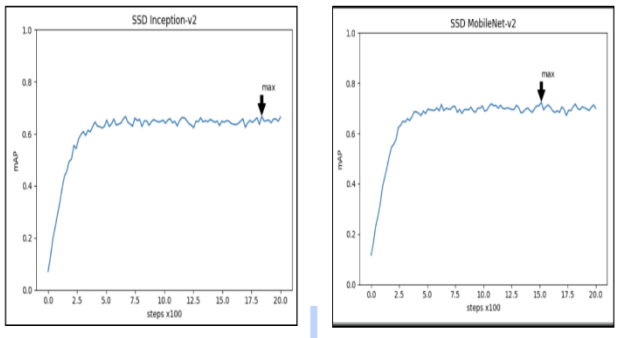

Fig. 4. mAP monitoring in SSD: a) on the left Inception-v2 architecture; b) MobileNet-v2.

Firstly, consider the SSD model with the respective architectures. As shown in Fig. 4, the left shows the result for the Inception-v2 architecture. In these results, the highest possible mean average precision of 66.74% was achieved as shown in the figure by the arrow.

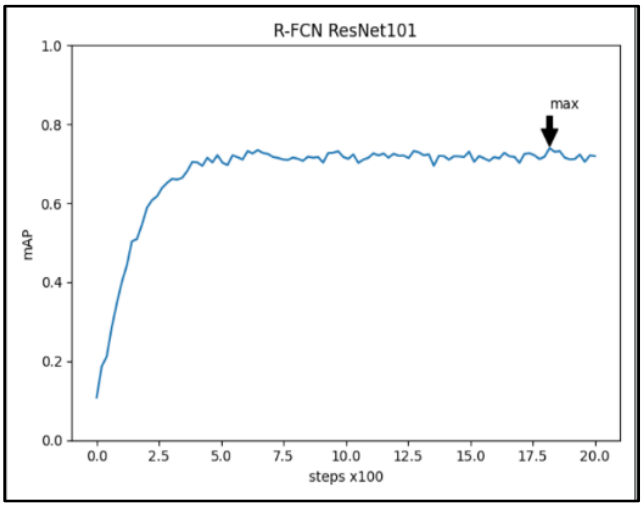

Fig. 5. mAP monitoring in R-FCN ResNet101.

Secondly, the R-FCN and ResNet101 architecture result is shown in Fig. 5. As it can be seen, this model achieves a higher mAP by 73.96%. Even though, it is quite close to outperforming the SSD-MobileNet101 outcome, it can be seen that R-FCN takes more number of steps to converge.

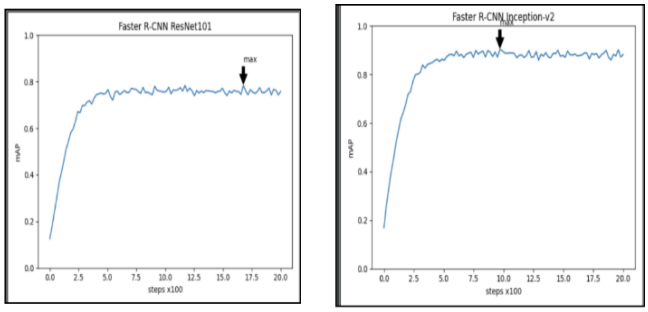

Fig. 6. mAP monitoring in R-FCN ResNet101

The results of Faster R-CNN and the selected architectures, is shown in Fig. 6. It can be seen that for the ResNet101 architecture, the algorithm does not take a long time to converge. Even exceeding the highest mAP achieved so far, by achieving a 78.43% maximum value of mean average precision. On the other hand, the Inception-v2 architecture achieved the maximum value of mAP at 1000 steps and with a precision of 87.58%.

After taking into account the mean average precision of each of the selected detection models and architectures, Fig. 7, shows the output images from all the five different models.

Considering the results achieved with Faster R-CNN, there should be recognition of the standard deviation effect achieved by utilising the simulated data. So, when considering the maximum average, an absolute error of 1.6% should be applied.

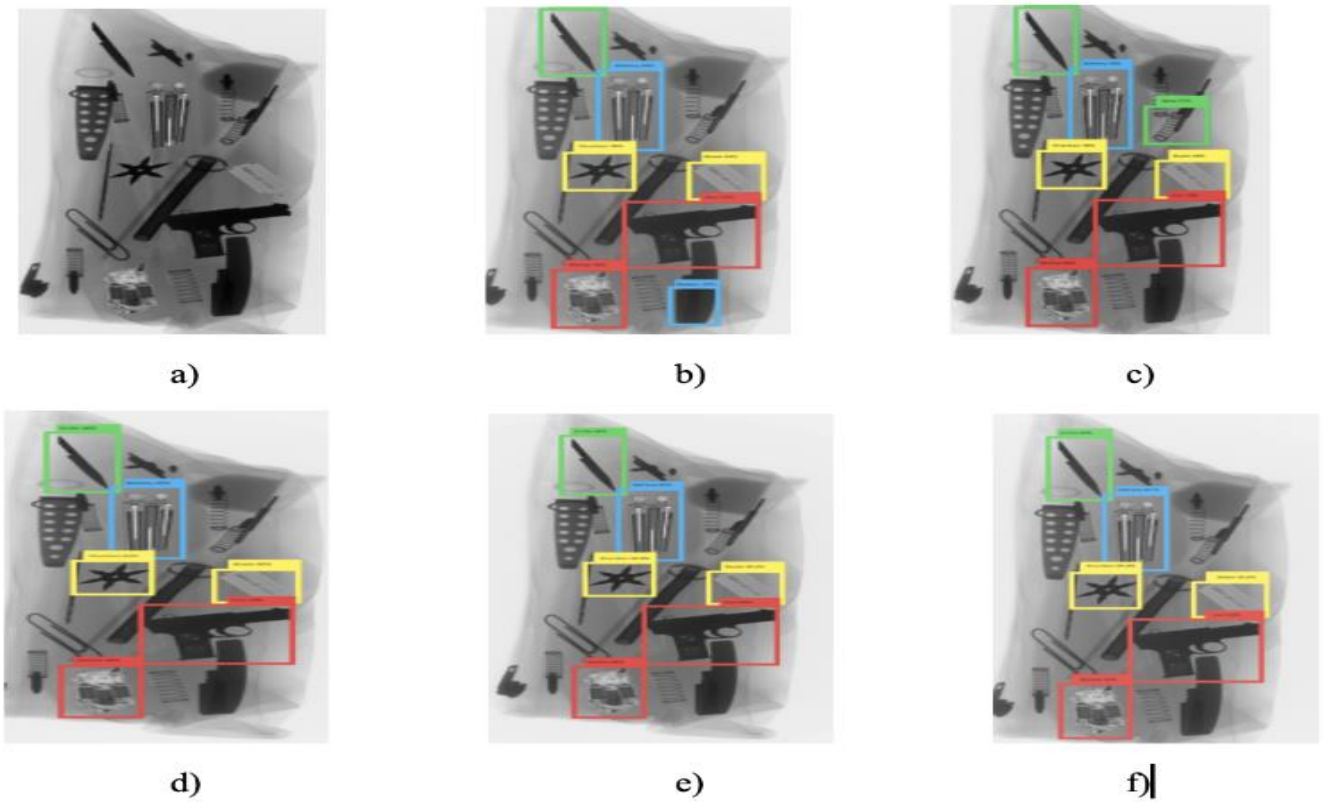

Fig. 7. Best case of detection; a) tested image containing only six threat objects, b) output of SSD MobileNet-v2, c) output of SSD Inception-v2, d) output of R-FCN ResNet-101, e) output of Faster R-CNN ResNet-101, f) output of Faster R-CNN Inception-v2.

C. Discussion

Based on the results, the highest performance was achieved using the Faster R-CNN and Inception-v2 models with 87.58% of average and a maximum of 89.58% accuracy. Faster R-CNN outperformed SSD significantly mainly due to the fact that SSD is used mostly in real time object detection and consists of scanning only once so it does not go into the deeper layers.

Faster R-CNN outperformed R-FCN due to the fact that R-FCN is a position-based algorithm and in our dataset, the object could not be detected based on the position because same objects are positioned differently in different images.

Considering the results achieved by utilising the Faster R-CNN algorithm, there should be recognition for the standard deviation smoothing effect achieved by the use of the simulated data. So, when considering the maximum average, an absolute error margin of ***1.6***% should be applied to the results.

V. Conclusions

This research has presented a new simulated dataset, which is made out of X-ray images of threat items most commonly found in scanning of baggage. The dataset included seven main classes of detection: handgun, razor-blade, knife, *shuriken*, battery, wires and mortar.

The best in class CNN-based article location models and object detection architectures were assessed and looked at its exhibition utilising the stimulated dataset. It was necessary to spotlight pre-trained models originally trained with completely different datasets from those employed in this research. The results show the success of using the simulated data. The results also indicated that the Faster R-CNN Inception-v2 accomplishes the most accurate precision attaining an accuracy of 87.58% with the test data set. Utilisation of further learning with information is expected to increase the accuracy rate.

References

[1] G. Zentai, “X-ray imaging for homeland security”, IEEE International Workshop on Imaging Systems and Techniques (IST 2008), Sept.2012, pp. 1-6.

[2] T Franzel, U. Schmidt, S. Roth, “Object Detection in Multi-View X-Ray Images”. In: Pinz A., Pock T., Bischof H., Leberl F. (eds) Pattern Recognition. DAGM/OAGM 2012. Lecture Notes in Computer Science, vol 7476., 2012, pp. 144-154. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-32717-9_15

[3] N. Megherbi, T. P. Breckon, G. T. Flitton, “Investigating existing medical CT segmentation techniques within automated baggage and package inspection”, Proc. SPIE 8901, Optics and Photonics for Counterterrorism, Crime Fighting and Defence IX; and Optical Materials and Biomaterials in Security and Defence Systems Technology X, 89010L, 16 October 2013. https://doi.org/10.1117/12.2028509.

[4] B. Abidi, Y. Zheng, A. Gribok, M. Abidi, “Improving weapon detection in single energy X-ray images through pseudocoloring”, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 36, no. 6, pp. 784-796, Nov. 2006, doi: 10.1109/TSMCC.2005.855523.

[5] M. Bastan, W. Byeon, T. Breuel, “Object recognition in multi-view X-ray images”. In Proceedings British Machine Vision Conference 2013. Pp. 130.1-130.11. http://dx.doi.org/10.5244/C.27.130

[6] A. Bolfing, T. Halbherr A. Schwaninger, “How Image Based Factors and Human Factors Contribute to Threat Detection Performance in X-Ray Aviation Security Screening”, In: Holzinger A. (eds) HCI and Usability for Education and Work. USAB 2008. Lecture Notes in Computer Science, vol 5298, pp. 419-438, 2008 Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-89350-9_30

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

[8] R. L. Galvez, E. P. Dadios, A. A. Bandala and R. R. P. Vicerra, “Threat Object Classification in X-ray Images Using Transfer Learning”, 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 2018, pp. 1-5, doi: 10.1109/HNICEM.2018.8666344.

[9] D. Mery, E. Svec, M. Arias, V. Riffo, J. M. Saavedra and S. Banerjee, “Modern Computer Vision Techniques for X-Ray Testing in Baggage Inspection”, in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 4, pp. 682-692, April 2017, doi: 10.1109/TSMC.2016.2628381.

[10] D. Mery, V. Riffo, U. Zscherpel, G. Mondragón, I. Lillo, I. Zuccar, H. Lobel and M. Carrasco, “GDXray: The database of X-ray Images for Nondestructive Testing”, Journal of Nondestructive Evaluation, 34, article no. 42, 2015. https://doi.org/10.1007/s10921-015-0315-7

[11] H. E. Martz, C. M. Logan, D. J. Schneberk and P. J. Shull, X-ray Imaging: fundamentals, industrial techniques, and applications. CRC Press, October 2016. ISBN-13: 978-0849397721.