Automated Threat Objects Detection with Synthetic Data for Real-Time X-ray Baggage Inspection

Kunal Chaturvedi[1], Ali Braytee[2], Dinesh Kumar Vishwakarma[1], Muhammad Saqib[2], Domingo Mery[3], Mukesh Prasad[2]

[1] Department of Information Technology, Delhi Technological University, India
[2] School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
[3] Department of Computer Science, Pontificia Universidad Catolica de Chile, Chile

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9533928

***Abstract*— With the recent surge in threats to public safety, the security focus of several organizations has been moved towards enhanced intelligent screening systems. Conventional X-ray screening, which relies on the human operator is the best use of this technology, allowing for the more accurate identification of potential threats. This paper explores X-ray security imagery by introducing a novel approach that generates realistic synthesized data, which opens up the possibility of using different settings to simulate occlusion, radiopacity, varying textures, and distractors to generate cluttered scenes. The generated synthetic data is effective in the training of deep networks. It allows better generalization on training data to deal with domain adaptation in the real world. The extensive set of experiments in this paper provides evidence for the efficacy of synthetic datasets over human-annotated datasets for automated X-ray security screening. The proposed approach outperforms the state-of-the-art approach for a diverse threat object dataset on mean Average Precision (mAP) of region-based detectors and classification/regression-based detectors.**

Keywords*— *X-ray, Threat, Baggage, Synthetic data, Screening, Object detection.

I. INTRODUCTION

Security at public places has been synonymous with the safety of an individual. Massive systems are deployed to safeguard the aviation industry, goods and passenger transportation, and high-risk zones such as defense areas, etc. X-ray security screening is one such system that enables inspection through the scanning of baggage, cargo, vehicle, and full-body scanning [1]. This paper focuses on the X-ray screening for baggage inspection. For accurate identification of suspicious items, security screening requires continuous vigilance from security personnel. However, the limited attention span of humans and the presence of occluded substances in cluttered X-ray scans adversely affect the detection rate of suspicious items. Due to the multitude of suspicious items, there is often confusion in the decision making to identify such items in each baggage, thus making it a time-consuming and tedious operation. Every year millions of baggage go through security screening at checkpoints. To prevent delays during rush hour and consider the number of items to be screened, the baggage screening must be highly effective and accurate. This could be achieved using a robust automated baggage inspection mechanism to assist human inspectors.

Object detection plays a significant role in identifying particular objects, such as distinguishing between threat and non-threat objects. It is a challenging process as it incorporates the two most important tasks: image classification and localization of objects with bounding box coordinates. The advent of deep learning techniques using Convolutional Neural Networks (CNN) replaced conventional computer vision techniques for X-ray baggage screening [2]. However, because of the unavailability of data, the CNN models may suffer from overfitting problems. Unfortunately, the dataset is not readily accessible in the X-ray security imagery domain due to privacy concerns. The available datasets are imbalanced, containing fewer instances of threat objects. This issue can be resolved using synthetic data generation, which is better than augmentation techniques. Synthesized data allows introducing new classes of prohibited items, variation in poses, scales, occlusion, and training on such dataset contributes to highly efficient detectors. This paper performs threat detection using the techniques that can be categorized into the one-step framework and two-steps framework. The one-step framework is based on the traditional object detection pipeline, which at first generates region proposals, and then classifies them into separate objects, and the two-steps framework adopts a unified framework to address object detection as a regression or classification problem called single-shot detectors. Taking into consideration that each technique has its drawback, there must be a trade-off between the two approaches to make screening fast and accurate.

The main contributions of the work are as follows:

- It generates a Synthesized annotated dataset through a mechanism that requires minimal effort and provides good visual coverage appearance of the prohibited items.
- It presents a comparative study of region proposal-based and regression/classification-based object detection pipelines for X-ray security.
- It conducts extensive experiments on the benchmark dataset, namely GDXray [3].

The organization of the remaining paper is as follows. Section II discusses the related work. The methodology is described in detail in Section III, followed by experiment results in Section IV. Finally, Section V concludes the paper with future work.

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

II. Related Works

This section briefly addresses the previous methodologies for X-ray security imagery and approaches for generating synthetic data. Security screening has become a major concern for organizations across the world due to a massive increase in the influx and efflux of passengers, goods, and cargo in the past decades. Due to this recent surge of interest in the field of X-ray security imagery [4], conventional machine learning approaches for classification [5] [6] [7], segmentation [8], [9], and detection [10] have been extensively studied to automate the process. A majority of previous classification work was carried out using bag of visual words (BoVW) [5] approach with SVM classifier, and sparse representations [6]. A transfer-learning based approach [11] was proposed for threat object classification. Hu et al. [12] proposed Security X-ray Multi-label Classification Network (SXMNet) to deal with overlapping in X-ray images classification. However, the object detection is more beneficial over the classification of objects for the detailed analysis of the contents in the X-ray imagery of baggage. Automated localization with bounding box dimensions and identification of threat object assists security officers to avert bizarre human-made disasters. Franzel et al. [13] conducted experiments on multiple-view X-ray imagery and compares it to single-view detection that demonstrated superior detection efficiency over single-view detection for handguns. With the advent of deep learning in object detection, CNN based frameworks replaced the approaches based on hand-crafted features. Samet et al. [14] introduced the region-based object detection methodologies, Faster RCNN [15] for this purpose, but the analysis does not take into account the speed of the detector, which is important for rapid threat detection. Dhiraj et al. [16] conducted experiments on the X-ray scans of the GRIMA X-ray Database (GDXray) that proposed by Mery et al. [3]. The dataset consists of instances of guns, shuriken, razor blades, and knives using different object detection frameworks.

However, the training of deep learning models becomes challenging due to the restricted availability of private security screening datasets and the skewed ratio of non-threatening objects to threatening objects. Different strategies have been adopted to compensate for the data for X-ray security imagery by generating synthetic dataset using General Adversarial Networks (GAN) [17] [18] [19], Threat Image Projection [10], and Logarithmic X-ray Imaging Model [20]. Yang et al. [17] and Zhu et al. [18] used GANS that are limited to generating synthetic instances of foreground objects including handgun, knife, etc. ignoring complex baggage information. These approaches do not take into consideration of cluttered background situations, including noise, and the use of non-threatening objects. Most of the recent literature on GANs [21][22] revolves around learning a mapping from a single image source, i.e. image translation that restricts the ability of full control over the rotation, scaling, radiopacity, occlusion, etc. This paper generates simulated X-ray images using real background and object instances with operations that allow full user control over the rotation, scaling, radiopacity, occlusion, etc. on the foreground object to preserve both realism and domain adaptation.

III. Methodology

Due to the lack of available data, the proposed approach (AXSD) aims to generate accurate synthesized images from an existing dataset GDXray. The generated images are evaluated using different object detection methods.

A. AXSD: Automated X-ray Synthetic Data generation

This section presents an automated X-ray synthetic data generation (AXSD) to minimize the empirical limits with the availability of enormous training data for deep learning based object detection. AXSD consists of two components: data collection and object superposition.

i. Data Collection

In the first component, the sampled data are randomly gathered of X-ray images of empty baggage and instances of threat/non-threat objects to perform the experiments in this paper. These instances of objects are extracted using the methodology described as shown in Fig. 1.

Edge Detection: As the first step, for foreground/background segmentation, we use a cellular nonlinear network-based [23] edge detector to predict fine edge-map from the foreground X-ray image. The network takes up three arguments: input image, number of iterations, and a combination of 19 parameters, defined as cloning template. Here, these parameters under optimization are iteratively adjusted so that its output converges to the output of the ideal edge detector. After the template learning, the optimised detector slides through the pixels, computing operations to evaluate edges. The predicted edge-map is the union of edge sets detected along the x and y directions respectively.

Noise Removal: Pre-processing technique such as median blur is applied on the predicted edge-map to eliminate noise. This step is necessary before contour detection, which generates a hierarchy of contours.

Foreground Extraction: Further, the graph-based GrabCut algorithm [24] is applied to the contour with the largest area to extract the foreground mask, which in our case, belongs to the instances of the threat item / non-threat item.

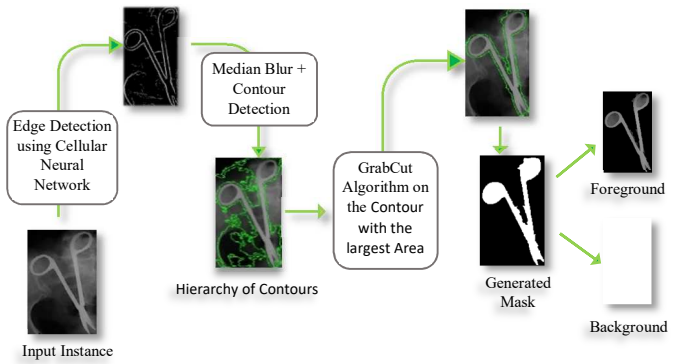

Fig. 1 Illustration of the proposed foreground/background methodology for the extraction of the instances of prohibited items.

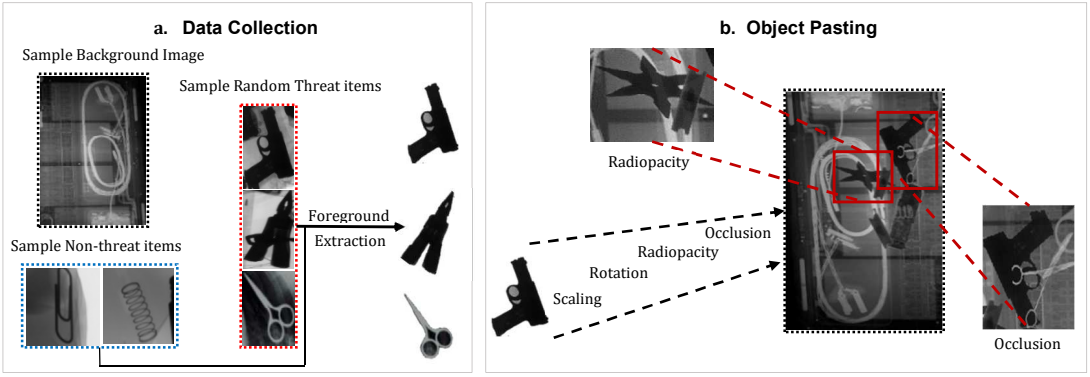

Fig. 2. A detailed explanation of our approach for a) Data collection and b) Object superposition

ii. Object Superposition

The extracted foreground items are superimposed randomly within a small section of the luggage in the background X-ray images. Contour detection is used to determine the boundary regions of the luggage. Due to the limited data, data augmentation techniques are applied to the foreground items to generate synthetic images. The main steps of the proposed method are shown in Fig. 2b and described as follows:

Scaling: A random scale with values ranging from 0.5 to 1 is chosen based on the size of the foreground object and the background luggage image.

Rotation2D: Different 2D viewpoints from the same object with an angle of rotation from -180 to 180 degrees are possible inside the baggage as the position of the object does not matter.

Non-Threat Items: Multiple distractors are superimposed onto the baggage X-ray image to simulate real-world scenarios to prevent bias in the training algorithm.

Radiopacity: The relative inability of electromagnetic radiation to pass through a material varies by thickness and the form of the material. These conditions are simulated by adjusting the alpha channel as shown in Fig. 3.

Occlusion: Partial overlapping with a maximum Intersection over Union (IOU) of 0.75 between different threat items and non-threat items is used when superimposing it on the background image to generate a genuine simulated image. The use of occlusion with the radiopacity principle on both threat objects and distractors increases the X-ray image's scene realism while maintaining global consistency with the existing dataset. A summary of the proposed framework is shown in Fig. 4.

Fig. 3. The radiopacity parameter is adjusted according to the intermediate values between 0 and 1. The higher the value, the less radiopaque the object will be.

B. Evaluate the synthetic data using object detection methods

We evaluated different object detection strategies. These detectors are based on the categorization of the conventional object detection system, which evaluates the regional proposals and then classifies each proposal as a threat or non-threat element. Another one uses a regression or classification problem-based unified framework to evaluate bounding-box coordinates and classification probabilities.

i. Region Proposal based Architectures

Faster RCNN proposed by Ren et al. [15] substitute the selective search algorithm with the introduction of the Region Proposal Network (RPN). It assigns a score to each anchor box that is of varying size and scale generated by sliding a fixed-sized window

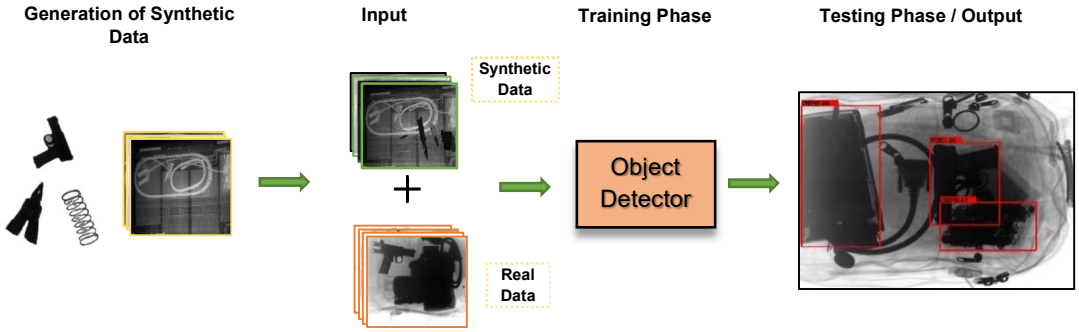

Fig. 4. Overview of our proposed approach. We first use the object superimposition to generate realistic data samples. Then, we use a combination of real and generated synthetic data for the object detection pipeline.

over the activation maps generated by the base network's last convolutional layer. After applying an ROI pooling layer, the proposals generated are passed onto the RCNN. It uses two different fully connected layers that output the bounding box and classifies whether a suspicious entity is present or not. Any redundant entity (overlapping bounding box) is further eliminated using non-maximum suppression (NMS).

Cascade RCNN [25] developed by Cai et al., is a multi-staged framework based on the popular RCNN. At each stage, cascaded bounding box regression combined with a sequence of detectors operates by changing the bounding boxes to be more selective against the close false positives used in the next stage of training. This reduces the risk of overfitting during training and overcomes a quality mismatch at inference, ensuring high-quality detection performance over Faster RCNN when the items are placed next to one another in the luggage.

ii. Classification/Regression based Architectures

Single Shot MultiBox Detector (SSD) [26] by Liu et al. uses a unified framework for both localizing objects and classification, eliminating the need for a proposal stage. It also resolves the issue of low-resolution object detection within the YOLO [27] framework by incorporating the idea of making faster predictions for bounding boxes and their labels at each feature map than Faster RCNN [15]. Feature maps of varying scales allow the SSD to detect objects of different scales and sizes.

RetinaNet [28] addresses the issue of the class imbalance (foreground/background) problem faced by unified object detection frameworks with a reconfiguration of cross-entropy loss. Focal loss decreases the share of high-probability detections that can dominate the model's performance. It incorporates the feature pyramid network, a regression subnetwork that evaluates bounding box coordinates using a regression head and classification subnetwork to allocate classes to these boxes.

IV. Experiment Results and Discussions

In this section, a comprehensive study is carried out on different combinations of synthesized data and human-annotated data using region-based and classification/regression-based CNN frameworks. Average precision (AP) is used as the evaluation criteria for the different frameworks and precision, and recall is determined by thresholding the Intersection-over-Union (IoU) for detection. The dataset is split into a ratio of 70:30 for training and testing, respectively. The number of epochs is fixed to 50 and the batch size to 4. The PyTorch framework is used for all the experiments. We first describe the characteristics of the dataset. We then report the results of our method using different proportions of synthetic images mixed

Table 1. Comparison of $GDXray$ dataset and $GDXray + Samples\ of\ Synthetic_{\alpha}$ dataset based on the average precisions (AP) metrics using architectures based on the two-stage framework and the one-stage framework.

Model	Backbone Network	Dataset	Average Precision							mAP
			Mobile	Chip	Shuriken	Revolver	Gun	Knife	Blade	
Faster RCNN	ResNet50	GDXray	0.907	0.907	0.917	0.996	0.909	0.905	0.909	0.921
		10% syn	0.910	0.924	0.925	0.979	0.899	0.847	0.913	0.913
		20% syn	0.915	0.925	0.931	0.979	0.902	0.873	0.920	0.920
		30% syn	0.915	0.919	0.932	0.981	0.909	0.898	0.921	0.925
		40% syn	0.903	0.908	0.931	0.999	0.952	0.948	0.921	**0.937**
		50% syn	0.917	0.915	0.929	0.968	0.909	0.901	0.922	0.923
Cascade RCNN	ResNet50	GDXray	0.908	0.904	0.909	0.995	0.906	0.907	0.913	0.920
		10% syn	0.891	0.891	0.902	0.984	0.960	0.899	0.901	0.918
		20% syn	0.899	0.902	0.903	0.989	0.969	0.903	0.901	0.922
		30% syn	0.907	0.901	0.914	0.991	0.959	0.937	0.911	0.931
		40% syn	0.904	0.907	0.929	0.999	0.961	0.954	0.923	**0.939**
		50% syn	0.901	0.906	0.911	0.999	0.961	0.939	0.913	0.932
SSD512	VGG16	GDXray	0.899	0.888	0.897	0.941	0.928	0.834	0.907	0.889
		10% syn	0.894	0.857	0.897	0.901	0.914	0.817	0.878	0.879
		20% syn	0.903	0.833	0.901	0.898	0.925	0.851	0.879	0.884
		30% syn	0.912	0.878	0.915	0.906	0.918	0.863	0.901	0.899
		40% syn	0.903	0.903	0.907	0.964	0.907	0.859	0.909	**0.907**
		50% syn	0.904	0.903	0.897	0.901	0.906	0.858	0.909	0.896
RetinaNet	ResNet50	GDXray	0.905	0.893	0.909	0.953	0.933	0.842	0.909	0.906
		10% syn	0.908	0.875	0.906	0.907	0.904	0.864	0.874	0.891
		20% syn	0.912	0.884	0.913	0.903	0.931	0.844	0.918	0.900
		30% syn	0.925	0.915	0.912	0.922	0.926	0.856	0.924	0.911
		40% syn	0.911	0.907	0.929	0.970	0.907	0.877	0.919	**0.917**
		50% syn	0.915	0.916	0.915	0.927	0.926	0.871	0.927	0.913

Table 2. Comparison results of the architectures trained on $Synthetic_{\alpha}$ dataset with 10% and 20% of GDXray dataset used as test samples.

Model	Backbone Network	Test Real Dataset	Average Precision							
			Mobile	Chip	Shuriken	Revolver	Gun	Knife	Blade	
Faster RCNN	ResNet50	10% syn	0.908	0.903	0.967	0.936	0.914	0.848	0.856	0.904
		20% syn	0.899	0.839	0.929	0.909	0.903	0.801	0.848	0.875
Cascade RCNN	ResNet50	10% syn	0.877	0.890	0.969	0.949	0.935	0.885	0.842	0.906
		20% syn	0.885	0.861	0.913	0.916	0.882	0.795	0.831	0.869
SSD512	VGG16	10% syn	0.853	0.850	0.947	0.931	0.901	0.806	0.813	0.871
		20% syn	0.888	0.856	0.887	0.901	0.901	0.834	0.762	0.861
RetinaNet	ResNet50	10% syn	0.893	0.873	0.946	0.933	0.901	0.808	0.832	0.883
		20% syn	0.874	0.849	0.907	0.891	0.915	0.820	0.799	0.865

Table 3. Comparison of the average precisions (AP) on $GDXray + Samples\ of\ Synthetic_{\beta}$ dataset, consisting of additional classes of prohibited items.

Model	Backbone Network	Dataset	Average Precision										mAP
			Mobile	Chip	Shuriken	Revolver	Gun	Knife	Blade	Pliers	Scissor	Hammer	
Faster RCNN	ResNet50	30% syn	0.901	0.892	0.896	0.938	0.875	0.858	0.887	0.894	0.847	0.968	0.895
		40% syn	0.909	0.909	0.908	0.997	0.905	0.907	0.909	0.909	0.896	1.000	**0.924**
		50% syn	0.905	0.879	0.936	0.999	0.901	0.877	0.901	0.910	0.833	0.971	0.911
Cascade RCNN	ResNet50	30% syn	0.878	0.865	0.899	0.931	0.802	0.886	0.901	0.928	0.889	0.979	0.895
		40% syn	0.908	0.907	0.911	0.997	0.899	0.908	0.909	0.989	0.905	0.998	**0.933**
		50% syn	0.917	0.898	0.908	0.989	0.901	0.902	0.899	0.967	0.911	0.989	0.928
SSD512	VGG16	30% syn	0.758	0.846	0.846	0.891	0.799	0.812	0.863	0.872	0.829	0.939	0.845
		40% syn	0.779	0.867	0.898	0.906	0.860	0.837	0.899	0.921	0.896	0.994	0.885
		50% syn	0.791	0.878	0.902	0.899	0.889	0.857	0.891	0.934	0.903	0.999	**0.894**
RetinaNet	ResNet50	30% syn	0.821	0.861	0.881	0.959	0.896	0.857	0.865	0.889	0.891	0.939	0.885
		40% syn	0.814	0.902	0.906	0.992	0.910	0.878	0.902	0.906	0.884	0.996	0.909
		50% syn	0.832	0.879	0.912	0.981	0.917	0.891	0.889	0.910	0.901	0.999	**0.911**

with real images. Finally, we compare our proposed method with the state-of-the-art.

A. Datasets

The Grima X-ray dataset (GDXray) was collected by Mery et al. [3]. The dataset consists of five groups of X-ray scans with more than 21,100 images: castings, welds, baggage, nature, and settings. However, for this research work, we used the baggage group consisting of 1371 X-ray images of prohibited objects. The prohibited items include five classes i.e., shuriken, knives, blades, firearms (guns and revolvers), and electronic items (chips and mobiles). Chips and mobiles are taken into consideration as they are restricted in high-risk environments such as defense areas.

In this work, an exhaustive set of experiments are conducted to choose the best performing synthetic dataset. For this purpose, we evaluate the effectiveness of combining the fixed ratio of the synthetic dataset with the real dataset. We synthesized and created two types of datasets of the same size as the GDXray dataset, $Synthetic_{\alpha}$ and $Synthetic_{\beta}$ using the same dataset generation procedure discussed earlier and randomly took samples in the ratio of 50%, 40%, 30%, 20%, and 10% from $Synthetic_{\alpha}$ and 30%, 40%, and 50% from $Synthetic_{\beta}$. $Synthetic_{\alpha}$ is composed of synthetically generated instances of the classes only from the GDXray dataset and $Synthetic_{\beta}$ consists of three additional classes, namely pliers, scissors, and hammers. The $GDXray + Synthetic_{\beta}$ dataset contains the largest number of classes of prohibited items available.

B. Results

As shown in Table 1 and Table 3, multiclass detection is performed, and the performance is compared in terms of average precision (AP) for each class of prohibited item using different architectures i.e., Faster RCNN, Cascade RCNN, RetinaNet with ResNet50 as configuration and SSD512 with VGG16 [29] as configuration. Table 1 investigates and compares the $GDXray$ dataset and the $GDXray + Synthetic_{\alpha}$ dataset and Table 3 evaluates the effectiveness of the $GDXray + Synthetic_{\beta}$ a dataset comprising ten classes of prohibited items including three additional classes. Table 1 shows that the networks trained on $GDXray +$ samples of $Synthetic_{\alpha}$ dataset achieve a performance gain up to ~2% over the $GDXray$ dataset. As shown in Table 1, combination of $GDXray$ and 40% of $Synthetic_{\alpha}$ resulted in a better and more robust performance than a network trained on $GDXray$ alone for all the architectures.

Moreover, the performance of the models is significant as shown in Table 2 when using a synthetic dataset as the training set and the real dataset as the test set. We used the $Synthetic_{\alpha}$ dataset and randomly chose 10% and 20% of GDXray as the test set. One of the crucial factors observed was the domain

adaptation when using the synthetic dataset. The results show that the generalization learned through the model with the use of only synthetic data overcomes the issue of domain adaptation on the test set.

Table 3 shows that the two-stage frameworks, namely Faster RCNN and Cascade RCNN reported higher mAP with 40% of $Synthetic_{\beta}$ Dataset + GDXray while the one-stage frameworks achieved better mAP on 50% of $Synthetic_{\beta}$ Dataset + GDXray. This result is attributed to the increased amount of data for the classes pliers, scissors, and hammers which resulted in inefficiently trained networks. Also, the source background image and target foreground image for the additional classes of $Synthetic_{\beta}$ are obtained using a different X-ray film scanner. However, the networks resulted in superior performance over the $Synthetic_{\beta}$ dataset + GDXray. The Cascade RCNN architecture with ResNet50 achieved the best performance over the $GDXray + 40\%\ of\ Synthetic_{\beta}$ dataset with a mAP of 0.933, while SSD512 performed the worst with a mAP of 0.885.

Table 4. Comparison of average precision (AP) by omitting different settings on $GDXray + 40\%\ Synthetic_{\alpha}$ dataset

Model	No Radiopacity	No Augmentation	No Non-Threat Object	All Operations
Faster RCNN	0.931	0.899	0.926	0.937
Cascade RCNN	0.928	0.902	0.924	0.939
SSD512	0.899	0.879	0.896	0.907
Retinanet50	0.911	0.891	0.910	0.917

A maximal AP of 0.910 for the class gun was observed on the RetinaNet. The visual results are shown in Fig. 6, using different object strategies on the GDXray and Synthesized dataset.

Table 4 shows the results obtained by omitting different settings of our synthetic data generation process. A minor decrease in the mAP for all the architectures was observed with the exclusion of the radiopacity principle in the images. However, the decrease in mAP was significant when augmentation techniques such as rotation and scaling, etc. are not applied. The results also emphasize the importance of the use of distractors which include non-threat objects to force the network to learn to ignore the nearby patterns.

Further, we evaluate and compare different object detectors on the $GDXray + 40\%\ of\ Synthetic_{\beta}$ dataset using the precision-recall curve. For a comparison of the region-based detector and the classification/regression-based detectors, both Faster RCNN and Cascade RCNN outperformed SSD and RetinaNet using the metrics of mAP. We also evaluate and report the inference speed of different models using Nvidia RTX 2080Ti on the GDXray dataset in Table 5. SSD512 based on VGG16 outperformed Faster RCNN, Cascade RCNN, and RetinaNet in speed, however Faster RCNN with 21.8 fps surpassed RetinaNet in both accuracy and speed, which is crucial for security screening. Although Cascade RCNN is comparable to Faster RCNN in terms of mAP, it is not feasible to use it due to its low inference speed. Also, it was observed that the training on the synthesized dataset is challenging due to the diverse nature of the prohibited objects with a diverse cluttered background in the dataset which enabled the network trained on this dataset to be better generalized.

Table 5. Comparison of different models in terms of inference speed on GDXRAY dataset using Nvidia RTX 2080 Ti with batch size 4.

Model	Backbone	Inf(fps)
Faster RCNN	ResNet50	21.8
Cascade RCNN	ResNet50	13.7
SSD512	VGG16	26.1
RetinaNet	ResNet50	16.5

C. Comparison with generative models

The proposed method AXSD is compared with the state-of-the-art generative adversarial models [21][22]. The officially released code is used to produce the results of the GAN models.

BigGAN [22] is a class-conditional GAN model for high fidelity natural image synthesis. The model emphasizes the benefits of scaling up both the batch size and the model size for high-quality image generation. However, there is a lack of

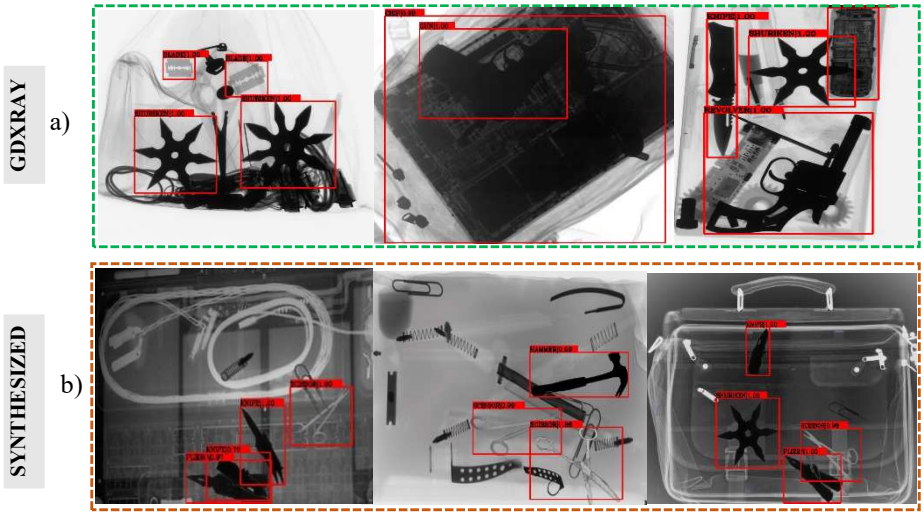

Fig. 6. Performance of the different threat detection pipelines on exemplar images from a) GDXRAY dataset and b) Synthesized dataset. It was observed that the frameworks achieved superior results for cluttered scenes, occluded objects, and illumination changes.

diversity in the generated images which is attributed to the fact that the model truncates large values of the noise vector to improve the image quality [30].

SinGAN [21] is an unconditional image generation technique that uses single image for serial training. The model consists of a pyramid of fully convolutional GANs, that is trained using a multi-resolution and multi-stage approach. It uses internal patch distribution within the single image to generate diversified samples. However, this method does not consider the relationship between the two images, thus ignoring distribution variations between the images.

Here, we use the Fréchet Inception Distance (FID) [31] as a measure of image quality. It compares the statistics of generated samples to the ground truth samples. A lower FID score indicates better model performance that generates images similar to the real images. As shown in Table 6, the proposed method outperforms the state-of-the-art GAN based methods with the lowest FID score. The GAN-based techniques are resource-intensive and less automated for the controlled synthesis process to deal with complex data.

Table 6. Comparison with the state-of-the-art GAN-based methods. Lower FID indicates better image quality.

Method	FID
BigGAN [22]	334.3
SinGAN [21]	80.6
Ours	**57.9**

V. Conclusions

This paper proposes a novel approach for generating synthetic datasets using different settings to address the scarcity of data in the X-ray imaging domain. The proposed approach generates highly realistic diversified synthetic X-ray scans with less cost and time that assists human-annotated datasets for training robust neural networks. The performance of the combinations of the GDXray and synthesized dataset is validated through an extensive set of experiments on two different detection pipelines. The results obtained are superior compared to the models trained on the real dataset. Also, this paper demonstrates that the performance of models trained purely on the synthetic dataset is comparable to the performance of models trained on the mixture of the real and synthesized dataset. In future, the proposed approach can be extended to other existing classification and detection problems for further improvements in the efficiency and robustness of the trained networks. Also, It can aim to generate realistic multi-view radiographs in the medical domain and implement few-shot learning techniques to reduce dependency on huge amounts of data.

References

[1] S. Akcay and T. Breckon, "Towards Automatic Threat Detection: A Survey of Advances of Deep Learning within X-ray Security Imaging," Jan. 2020, doi: 2001.01293.

[2] S. Akcay, M. E. Kundegorski, M. Devereux, and T. P. Breckon, "Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 1057–1061, doi: 10.1109/ICIP.2016.7532519.

[3] D. Mery, V. Riffo, U. Zscherpel, G. Mondragón, I. Lillo, I. Zuccar, et al., "GDXray: The Database of X-ray Images for Nondestructive Testing," *J. Nondestruct. Eval.*, vol. 34, no. 4, pp. 1–12, Dec. 2015, doi: 10.1007/s10921-015-0315-7.

[4] D. Mery, D. Saavedra, and M. Prasad, "X-Ray Baggage Inspection With Computer Vision: A Survey," *IEEE Access*, vol. 8, pp. 145620–145633, 2020, doi: 10.1109/ACCESS.2020.3015014.

[5] M. Baştan, "Multi-view object detection in dual-energy X-ray images," *Mach. Vis. Appl.*, vol. 26, no. 7–8, pp. 1045–1060, Nov. 2015, doi: 10.1007/s00138-015-0706-x.

[6] D. Mery, E. Svec, and M. Arias, "Object Recognition in Baggage Inspection Using Adaptive Sparse Representations of X-ray Images," 2016, pp. 709–720.

[7] D. Mery, V. Riffo, I. Zuccar, and C. Pieringer, "Object recognition in X-ray testing using an efficient search algorithm in multiple views," *Insight - Non-Destructive Test. Cond. Monit.*, vol. 59, no. 2, pp. 85–92, Feb. 2017, doi: 10.1784/insi.2017.59.2.85.

[8] M. Singh and S. Singh, "Image segmentation optimisation for x-ray images of airline luggage," in *Proceedings of the 2004 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, 2004. CIHSPS 2004.*, pp. 10–17, doi: 10.1109/CIHSPS.2004.1360198.

[9] G. Heitz and G. Chechik, "Object separation in x-ray image sets," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 2093–2100, doi: 10.1109/CVPR.2010.5539887.

[10] N. Bhowmik, Q. Wang, Y. F. A. Gaus, M. Szarek, and T. P. Breckon, "The Good, the Bad and the Ugly: Evaluating Convolutional Neural Networks for Prohibited Item Detection Using Real and Synthetically Composited X-ray Imagery," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.11508.

[11] 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," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 4, pp. 682–692, Apr. 2017, doi: 10.1109/TSMC.2016.2628381.

[12] B. Hu, C. Zhang, L. Wang, and Q. Zhang, "Multi-label X-ray Imagery Classification via Bottom-up Attention and Meta Fusion."

[13] T. Franzel, U. Schmidt, and S. Roth, "Object Detection in Multi-view X-Ray Images," 2012, pp. 144–154.

[14] S. Akcay and T. P. Breckon, "An evaluation of region based object detection strategies within X-ray baggage security imagery," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 1337–1341, doi: 10.1109/ICIP.2017.8296499.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.01497.

[16] Dhiraj and D. K. Jain, "An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery," *Pattern Recognit. Lett.*, vol. 120,

pp. 112–119, Apr. 2019, doi: 10.1016/j.patrec.2019.01.014.

[17] J. Yang, Z. Zhao, H. Zhang, and Y. Shi, "Data Augmentation for X-Ray Prohibited Item Images Using Generative Adversarial Networks," *IEEE Access*, vol. 7, pp. 28894–28902, 2019, doi: 10.1109/ACCESS.2019.2902121.

[18] Y. Zhu, H. Zhang, J. An, and J. Yang, "GAN-based data augmentation of prohibited item X-ray images in security inspection," *Optoelectron. Lett.*, vol. 16, no. 3, pp. 225–229, May 2020, doi: 10.1007/s11801-020-9116-z.

[19] D. Saavedra, S. Banerjee, and D. Mery, "Detection of threat objects in baggage inspection with X-ray images using deep learning," *Neural Comput. Appl.*, 2020, doi: 10.1007/s00521-020-05521-2.

[20] D. Mery and A. K. Katsaggelos, "A Logarithmic X-Ray Imaging Model for Baggage Inspection: Simulation and Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jul. 2017, pp. 251–259, doi: 10.1109/CVPRW.2017.37.

[21] T. R. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a Generative Model From a Single Natural Image," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 4569–4579, doi: 10.1109/ICCV.2019.00467.

[22] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1809.11096.

[23] L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Trans. Circuits Syst.*, vol. 35, no. 10, pp. 1257–1272, Oct. 1988, doi: 10.1109/31.7600.

[24] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut,'" *ACM Trans. Graph.*, vol. 23, no. 3, p. 309, Aug. 2004, doi: 10.1145/1015706.1015720.

[25] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High Quality Object Detection and Instance Segmentation," Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.09756.

[26] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," Dec. 2015, doi: 10.1007/978-3-319-46448-0_2.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Jun. 2015, doi: 1506.02640.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," Aug. 2017, [Online]. Available: http://arxiv.org/abs/1708.02002.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556.

[30] A. Razavi, A. van den Oord, and O. Vinyals, "Generating Diverse High-Fidelity Images with VQ-VAE-2," Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.00446.

[31] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.08500.