micromachines

*Article*

# Towards More Efficient Security Inspection via Deep Learning: A Task-Driven X-ray Image Cropping Scheme

**Hong Duc Nguyen, Rizhao Cai, Heng Zhao, Alex C. Kot and Bihan Wen ***

School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore; e200215@e.ntu.edu.sg (H.D.N.); rzcai@ntu.edu.sg (R.C.); zhaoheng@ntu.edu.sg (H.Z.); eackot@ntu.edu.sg (A.C.K.)

* Correspondence: bihan.wen@ntu.edu.sg

**Abstract:** X-ray imaging machines are widely used in border control checkpoints or public transportation, for luggage scanning and inspection. Recent advances in deep learning enabled automatic object detection of X-ray imaging results to largely reduce labor costs. Compared to tasks on natural images, object detection for X-ray inspection are typically more challenging, due to the varied sizes and aspect ratios of X-ray images, random locations of the small target objects within the redundant background region, etc. In practice, we show that directly applying off-the-shelf deep learning-based detection algorithms for X-ray imagery can be highly time-consuming and ineffective. To this end, we propose a Task-Driven Cropping scheme, dubbed TDC, for improving the deep image detection algorithms towards efficient and effective luggage inspection via X-ray images. Instead of processing the whole X-ray images for object detection, we propose a two-stage strategy, which first adaptively crops X-ray images and only preserves the task-related regions, i.e., the luggage regions for security inspection. A task-specific deep feature extractor is used to rapidly identify the importance of each X-ray image pixel. Only the regions that are useful and related to the detection tasks are kept and passed to the follow-up deep detector. The varied-scale X-ray images are thus reduced to the same size and aspect ratio, which enables a more efficient deep detection pipeline. Besides, to benchmark the effectiveness of X-ray image detection algorithms, we propose a novel dataset for X-ray image detection, dubbed SIXray-D, based on the popular SIXray dataset. In SIXray-D, we provide the complete and more accurate annotations of both object classes and bounding boxes, which enables model training for supervised X-ray detection methods. Our results show that our proposed TDC algorithm can effectively boost popular detection algorithms, by achieving better detection mAPs or reducing the run time.

**Keywords:** X-ray imaging; objective detection; image cropping; deep learning; features extraction

check for updates

**Citation:** Nguyen, H.D.; Cai, R.; Zhao, H.; Kot, A.C.; Wen, B. Towards More Efficient Security Inspection via Deep Learning: A Task-Driven X-ray Image Cropping Scheme. *Micromachines* **2022**, *13*, 565. https://doi.org/10.3390/mi13040565

Academic Editor: Marc Desmulliez

Received: 31 December 2021
Accepted: 28 March 2022
Published: 31 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

X-ray screening is a commonly-used security measure at airports, border checkpoints and public transportation due to the merits such as real-time imaging and non-invasiveness. The investigation of X-ray images was typically done by human screeners, while such a manual process is expensive, inefficient, tiring, and can be affected by factors such as mental exhaustion and workplace conditions [1,2]. Additionally, the different view angles of clutter and overlapped objects in security X-ray images will further increase the risk of missing prohibited items [3,4]. Owing to these reasons, it is desired to automate the X-ray screening process using advanced object detection algorithms.

While classic methods [5–8] exploited image processing and model-based optimization for X-ray object detection, recently proposed deep learning algorithms [9–13] have proved to be a better method with higher detection accuracy. Along with an increase in the number of available detection algorithms, more security X-ray image datasets [13–16] are published and made publicly available from 2016 onward, which enable more deep learning methods to solve the X-ray image detection or classification problems. Although providing valuable

auto-detection benchmarks, these datasets all share the same common trait: the images are manually cleaned and processed, i.e., the objects of interest are manually amplified and centered to fit the viewport. However, in the realistic situations, the scanned items do not always fit the viewport and may appear very small compared to the redundant background. Figure 1 show some examples of the manually process X-ray images from the GDXray [14] and OPIXray [13] datasets, compared to the realistic images from Sixray [17]. Besides, to run the trained detection models, X-ray images are typically required to be resized to square, but such process may distort the original resolution and aspect ratios of the relevant objects. Research has also pointed out that the smaller sizes of the input image can decrease the detector's accuracy [18–20]. The redundant spaces also contribute nothing to the detection's ability but more processing time, as large size input images tend to increase the inference time of the model, so a method of removing such spaces is much needed. Conventional cropping techniques such as center cropping or edge cropping might not perform well on security X-ray images due to the variety in size and objects' location along with X-ray artifacts [21,22]. Alternatively, deep learning-based cropping or retargeting techniques [23–26] using feature extraction mainly focus on the aesthetic aspect of images. They focus on improving the visual quality and often neglect the inference time of the model and the effect of cropping on detection time and accuracy.

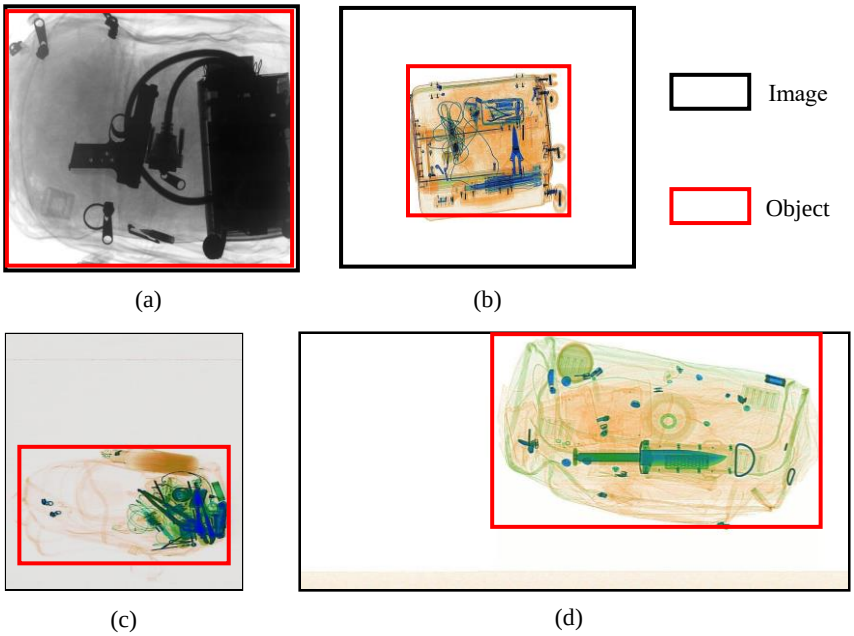


**Figure 1.** Examples of security X-ray images from popular public datasets: (**a**) GDXray [14], (**b**) OPIXray [13], (**c**,**d**) SIXray [17].

In this work, we propose Task-Driven image Cropping (TDC)—a novel and efficient scheme of cropping security X-ray images using activation output of convolutional layers of the detection network to simultaneously achieve two objectives, i.e., reduction of run time and improvement of detection accuracy. By utilizing the change in the energy of the feature maps extracted from the network backbone, we can efficiently crop the unwanted background and preserve regions of interest. To test the cropping performance of TDC for real-life classification and detection problems, we select SIXray [17] because it provides extra challenges with multiple viewpoints, complex backgrounds, and overlapping objects. However, only the test SIXray dataset has bounding box-level annotation and some samples are wrongly annotated as negative (without prohibited items) in the dataset. To provide better annotations, we propose a novel X-ray image detection benchmark, named SIXray-D, which is a fully annotated detection dataset with more positive images and objects.

Our contribution can be summarized as follows:

1. We propose SIXray-D, an improved dataset based on the popular SIXray [17] as a fully annotated dataset for contraband items detection. SIXray-D provides a comprehensive detection benchmark, which can be used to evaluate and improve the effectiveness of deep X-ray detection networks.
2. We propose TDC, a task-driven X-ray image cropping pipeline to efficiently remove redundant background and preserve the task-related objects by utilizing the features extracted from the network's backbone.
3. We conduct experiments to evaluate several state-of-the-art single-stage detectors on the proposed SIXray-D. We show that TDC can effectively improve the detection methods such as RFB-Net, by achieving better mAPs or reducing the inference time.

The remainder of this paper is organized as follows. Section 2 briefly introduces the background and related works. Section 3 presents our proposed SIXray-D dataset for X-ray image detection tasks. Section 4 describes the details of the proposed TDC scheme and how it improves the X-ray image detection tasks. The experimental results are presented in Section 5, and Section 6 provides several concluding remarks.

## 2. Related Works

**X-ray security inspection task.** X-ray screening is a universal security measure at border checkpoints, airports, and public transportation due to the thoroughly real-time imaging and non-invasiveness. The procedure can be conducted on both passengers and their luggage to identify any prohibited and potentially dangerous items carried through the border checkpoints or stations. The inspection of X-ray images is mostly done by security personnel, and human factors such as physical well-being, mental health, and work satisfaction could affect the process and lead to errors in detecting contraband items. According to a survey on airport security professionals in 18 Brazilian airports [2], 61% of the professionals admit that they have committed an error during security inspection and try to correct it. The main factors contributing the most to the errors are the tiring, repetitive and monotonous jobs, the neglect of following work procedures, and the complacency on the jobs. Furthermore, training the human operators for threat detection is expensive and requires a great amount of effort, hence automated X-ray screening process is desired using object detection algorithms.

**X-ray security dataset.** There are several public datasets in the X-ray security field such as the GDXray dataset [14] and OPIXray dataset [13]. Each dataset has a different way of labeling images, and the number of classes is also varied. These datasets contain non-complex X-ray images with no redundant background, and there is usually only one object of interest per image. Moreover, the proportion between positive and negative samples does not reflect realistic scenarios, thus making such datasets non-ideal for security inspecting applications. HiXray [27] is another security X-ray inspection dataset with high-quality images, multiple objects of interest per image, and object occlusion. However, it focuses on airport cabin baggage as the classes consist of phone charger, water bottle, mobile phone, tablet, and laptop, which are usually scanned separately with airplane cabin baggage. On the other hand, these items are allowed in checked-in baggage and general security X-ray inspection such as subway station or land border checkpoint. Furthermore, the images are pre-processed, and the luggage is situated at the center of the viewport, thus making the dataset less realistic. Meanwhile, SIXray [17] provides a complex dataset with extra challenges such as overlapped objects, various image sizes, and difficult background content. It proves to be the largest security X-ray dataset up to date with over 1 million images, including both positive and negative samples. However, the manual annotations for SIXray are done coarsely with only the test set annotated with bounding boxes, hence more detailed annotations on the whole dataset can lead to improvement in the accuracy of the detection model.

**Single-stage detectors.** Top-performing image detectors are usually two-stage detectors using deep convolutional neural networks (CNN) backbones such as Inception [28], Mask-RCNN [29], or ResNet [30] with the trade-off of high computational complexity and

slow runtime. Alternatively, several one-stage detectors provide real-time speed and performance comparable to these two-stage detectors, namely RetinaNet [31], SSD [32]. RetinaNet utilizes feature pyramid networks [33], and couples with the ResNet [30] backbone network for feature extraction. It also introduces FocalLoss to handle the class imbalance problem of a single-stage detector to provide a good detection performance. SSD introduces the application of dividing the image into a grid with the pre-defined anchor boxes in each grid cell. The anchor boxes have different sizes and aspect ratios, and they are responsible for matching the objects of interest during the training and detecting process. Liu et al. [34] proposed a lightweight detector consisting of Receptive Field Block (RFB) modules inspired from Receptive Fields (RFs) in human eyes to make features more distinguishable on top of the SSD network. Besides, with the recent work of YOLOv5 [35] and image transformer for one-shot detector [36], the performance of single-stage detectors could surpass their double-stage detector counterparts in some experiments. X-ray image detection at a security checkpoint requires fast inference time, thus by using single-stage detectors on the task we can achieve good accuracy and relatively real-time detection speed.

**Image retargeting/cropping methods.** Image cropping and retargeting are popular methods to remove the redundant area of an image. The main goal of image retargeting [25,37–39] is improving the aesthetic of the image by warping but with a risk of warping objects of interest and generating artifacts. Alternatively, image cropping is much simpler, as it only selects an area that contains saliency regions. The SIXray dataset provides a challenge to auto image cropping methods such as center cropping due to the different locations of objects and variety of image sizes. Additionally, X-ray artifacts [21] such as vertical or horizontal bars prevent cropping using conventional methods such as edge detection [40] to crop the interest region. Furthermore, image retargeting usually requires a target aspect ratio [25,38,39], which is hard to determine due to the dataset's characteristics.

**Object detection in security X-ray inspection.** The methods for automated object recognition in X-ray can be categorized into conventional image analysis, machine learning-based approach, and deep learning-based approach. Classic methods range from fusion, de-noising, and enhancement of dual-energy X-ray images [6] to Threat Image Projection (TIP) [16] for enhancing X-ray detection performance. Such methods exploit image processing and threshold-based optimization to improve the operators' performance and alertness. Before the rise in the number of deep learning-based algorithms, the bag of visual words (BoVW) was the popular machine learning method for both object classification [41,42] and object detection [43,44]. Besides BoVW, some other common approaches [45,46] based on feature descriptors and k-NN classifier [47] were used for multi-view X-ray images. Recently, with the introduction of deep learning algorithms, object detection methods based on both single-stage detectors [9,11,48] and double-stage detectors [48–50] prove to be the better choice for automated X-ray image inspection. Furthermore, pixel-level analysis [51] can be conducted to enhancing the performance of the deep learning detector for large-scale X-ray security images. They achieve high accuracy and reasonable inference time on different X-ray image datasets such as DBF3 [52], GDXray [14], and SIXray [17] datasets.

## 3. SIXray-D Dataset

GDXray [14] and OPIXray [13] are popular public security X-ray datasets serving the detection task, and the key attributes of these datasets can be summarized in Table 1. In GDXray and OPIXray datasets, images are manually cleaned, and redundant background and noise are removed as shown in Figure 1. The GDXray only contains simple settings and non-clutter baggage with one contraband item per image, hence the dataset does not align with the realistic situation where objects of interest appear together with other items and are hard to be detected. On another hand, the image in OPIXray has object occlusion, but it also has only one prohibited object per baggage. Besides, the dataset only consists of knives and scissors classes, thus excluding some contraband classes such as guns or wrenches. Furthermore, both GDXray and OPIXray have the trouble of retrieving the datasets from the sources.

**Table 1.** Key attributes of popular detection security X-ray datasets, namely GDXray [14] and OPIXray [13], as well as the proposed SIXray-D dataset.

| Dataset | Class Types | Positive Images | Negative Images | Multiple Objects per Image | Object Occlusion | Real X-Ray Artifacts | Realistic Orientation of Luggage |
|---|---|---|---|---|---|---|---|
| GDXray | Shuriken, gun, knife | 8850 | 10,550 | ✗ | ✗ | ✗ | ✗ |
| OPIXray | Scissors and variants of knife | 8885 | 0 | ✗ [1] | ✓ | ✗ | ✗ |
| SIXray-D | Scissors, pliers, gun, wrench, knife | 11,401 | 1,050,302 | ✓ | ✓ | ✓ | ✓ |

[1] For OPIXray, there are only 35 out of 8885 images with multiple objects per image.

Due to the aforementioned limitations, we choose to prepare our new benchmark using SIXray [17] as our data baseline, which provides more complex X-ray images, multiple contraband items per image, and easier accessibility. The original SIXray dataset has 8,929 positive images with contraband items and 1,050,302 negative images. The ratio between positive and negative samples of SIXray is around 1:1000, and it reflects the realistic frequency of appearance of prohibited items. Though SIXray intends to provide 6 object classes, images containing hammers were not available for download. Thus, in the proposed SIXray-D dataset, we proceeded with only 5 item classes, namely Pliers, Scissors, Gun, Knife, and Wrench. More importantly, the SIXray is designed for the classification task, hence originally only 1200 positive images were annotated at the bounding box level in the testing set. In practice, training a detection deep model with only a few annotated images usually leads to overfitting, e.g., the detection mAP for the original SIXray is only 65.62 [17] when using DenseNet [53].

To this end, we propose a new X-ray object detection benchmark by utilizing the image data from SIXray [17] with more comprehensive annotation for detection, named SIXray-D. There are 8823 positive images in total that are publicly available from SIXray and around 1 million negative images. We train the RFB network for object detection using the positive images. Then, we apply the network to conduct detection on the negative images. The network detects some contraband items in these negative images (inspected set). We manually inspect the inspected set and we find that there are two categories. The first one is images that contain contraband items, but are the false-negative image of the SIXray dataset (wrongly annotated as negative). Such false-negative samples usually have a large area of redundant background with multiple overlapped objects, and the contraband items are small compared to the other components. Figure 2 illustrates some of the Scissors and Knife images that are false-negative. The second category does not contain contraband images but is marked as positive by our network. The second catagory is the false-positive images to the detector, and Figure 3 illustrates such images. In the proposed SIXray-D dataset, we complete the annotation for the first category images and add them to the positive set. For the second category image, we separate them from the negative images to prevent confusion and misdetection. Table 2 summarizes the detailed information about our contribution to SIXray-D.

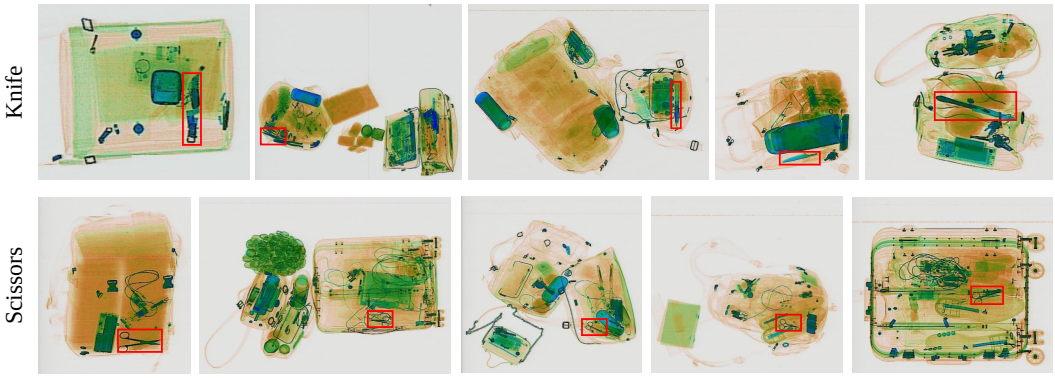


**Figure 2.** False-negative images from the SIXray dataset [17]. The red bounding boxes that are newly annotated by SIXray-D indicate the contraband items from the classes Knife and Scissors.

**Figure 3.** Negative images that are marked as positive by the detection network.

**Table 2.** Comparison between SIXray [17] and SIXray-D datasets.

| Dataset | SIXray | SIXray-D |
|---|---|---|
| Supervised task | Classification | Detection |
| Bounding box annotations | Test Set | Train + test set |
| Positive images | 8823 | 11,401 |
| Positive objects | 20,729 | 23,470 |

## 4. Task-Driven Image Cropping by Deep Feature Extraction

With the new SIXray-D dataset available, our goal is to develop a more efficient and effective X-ray image detection pipeline that can overcome practical challenges, such as varied image sizes, aspect ratios as well as arbitrary locations of small target objects. We propose the Task-Driven image Cropping (TDC) module, which utilizes activation feature maps from convolutional layers for determining the redundant data to be removed. Figure 4 shows the proposed TDC-based X-ray image detection pipeline, which consists of the TDC module based on the feature map generated by the task network backbone, and the detection process using the output of the cropping module. As shown in Figure 4, the cropping process can utilize the same network backbone with the detection network to save memory and computation. Figure 5 illustrates the detailed structure of the TDC module. Next, we will first introduce preliminaries about CNN feature maps, followed by the detailed TDC module and how it works for adaptive image cropping.

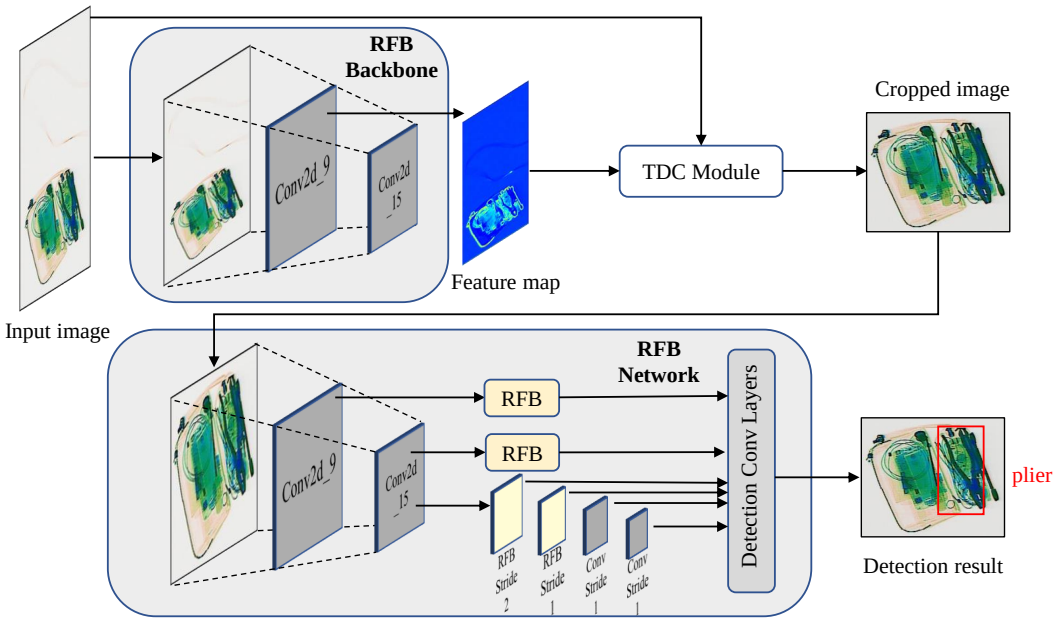


**Figure 4.** The proposed X-ray image detection pipeline with the TDC module for task-driven cropping. The network backbone for the feature extraction is the same in the detection network. In this work, we use RFB-Net [34] for the detection task, while TDC could also be used for other single-stage detection networks.

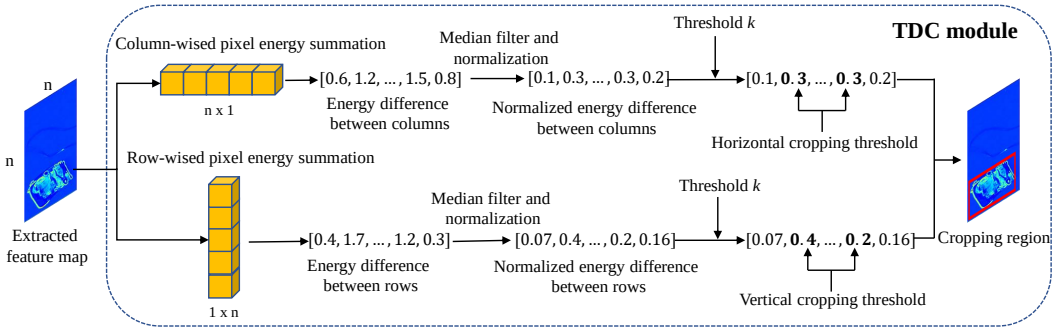


**Figure 5.** Construction of TDC module by combining the columns/rows wised summation of pixel energy, the median filter to filter out the X-ray artifact spikes, the min-max normalization, and the threshold-based cropping. The cropping region is then combined with the input image to produce the output cropped image.

### *4.1. Feature Map Generation*

In a CNN model, the feature map $\mathcal{F} \in \mathbb{R}^{W \times H \times C}$ is the result of the input image after the convolution process, where W and H are the width and height of the feature map, and C is the number of channels. Usually, the first few layers provide details about low-level features such as edges and colors. The deeper layers will provide information about high-level features like positions and shapes of salient objects [54,55].

Let $\mathcal{F}$ be the output from a convolutional layer, in which $\mathcal{F} = \{F_1, F_2, ..., F_C\}$ where $C$ is the number of channels and $F_i$ is the output per channel. To generate an overall feature map of a layer, the sum of magnitudes of each channel output is calculated and then averaged as follows:

$$\mathcal{F}_{avg} = \frac{1}{C} \sum_{i=1}^{C} |F_i|, \tag{1}$$

where $|F_i|$ is the element-wise absolute value of $F_i$ and $\mathcal{F}_{avg}$ is also a 2D array.

For scanned security X-ray images, objects of interest overlap with the container or luggage. Hence, to provide a suitable feature map that distinguishes the items from the

background, we decide to extract middle convolutional layers. Activation from such layer contains moderate information about objects, but does not omit the suitcases containing them, which helps distinguish against the plain background of the X-ray image. Figure 6 presents the original image and three different feature maps from three layers: the 2nd convolutional layer, the 15th convolutional layer, and our choice, the 9th layer of the backbone of RFB Net. Note that we can use the output from any convolutional layer to perform feature extraction. The ablation study in Section 5.4 proves that the 9th layer gives the best detection accuracy improvement after the image cropping process.

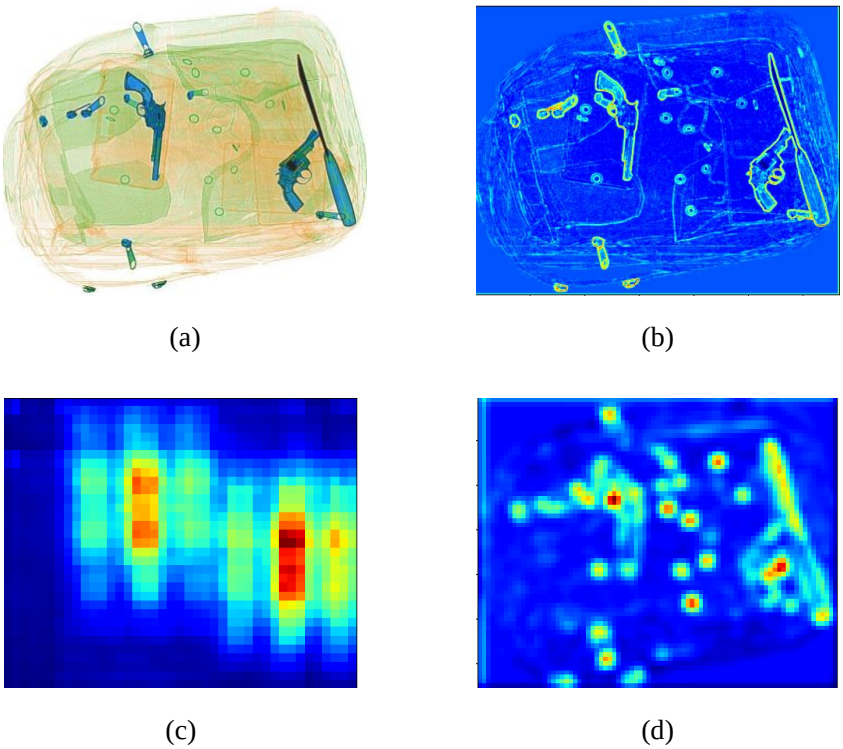

**Figure 6.** Original X-ray image and feature outputs from different convolutional layers from the backbone of RFB Net. (**a**) is the original image, (**b**) is the 2nd layer output, (**c**) is the 15th layer output and (**d**) is the 9th layer output.

### *4.2. TDC Module and Image Cropping*

**Dimension reduction of feature matrix.** Figure 5 summarizes our TDC module for efficient task-driven feature-based cropping. We propose that the pixel magnitude of the feature map in the regions of interest is higher in other areas. To prove this idea and utilize it for the cropping process, we construct a column-wised feature matrix $\mathcal{F}_c$ through summation of all pixels $\mathcal{F}_{avg}(x, y)$ in the feature map $\mathcal{F}_{avg}$ vertically:

$$\mathcal{F}_c = \sum_{j=1}^{H} \mathcal{F}_{avg}(x, j), \tag{2}$$

where $H$ is the vertical resolution of the feature map $\mathcal{F}_{avg}$. $\mathcal{F}_c = \{F_{c1}, F_{c2}, .., F_{cW}\}$ where $F_{ci}$ is the magnitude of column $i$ in the feature map, and $W$ is the horizontal resolution of $F_{avg}$.

**Columns difference and artifact removal.** We take the absolute difference between columns, $\mathcal{F}_{diff}$, to observe the rapid change in magnitude between column, and to identify the region of interest as follows:

$$\mathcal{F}_{diff}(i) = |F_{c(i+1)} - F_{ci}|. \tag{3}$$

However, the vertical and horizontal bars appear as artifacts in the scanned image horizontally and vertically due to the scanner or image processing steps [21]. Such artifacts can cause spikes in $\mathcal{F}_{diff}$ shown in Figure 7. We use a median filter to filter our matrix:

$$\mathcal{F}_{med} = \Phi(\mathcal{F}_{diff}, n), \tag{4}$$

where $\Phi(x, n)$ indicates median filter of $x$ with kernel $n$. We normalize $\mathcal{F}_{diff}$ using min-max normalization to derive $\mathcal{F}_{norm}$ as the final column difference matrix:

$$\mathcal{F}_{norm}(i) = \frac{F_{med}(i) - min(\mathcal{F}_{med})}{max(\mathcal{F}_{med}) - min(\mathcal{F}_{med})}, \tag{5}$$

where $\mathcal{F}_{norm} = \{F_{n1}, F_{n2}, ...F_{d(W-1)}\}$ indicates the normalized horizontal change in magnitude of the feature map.

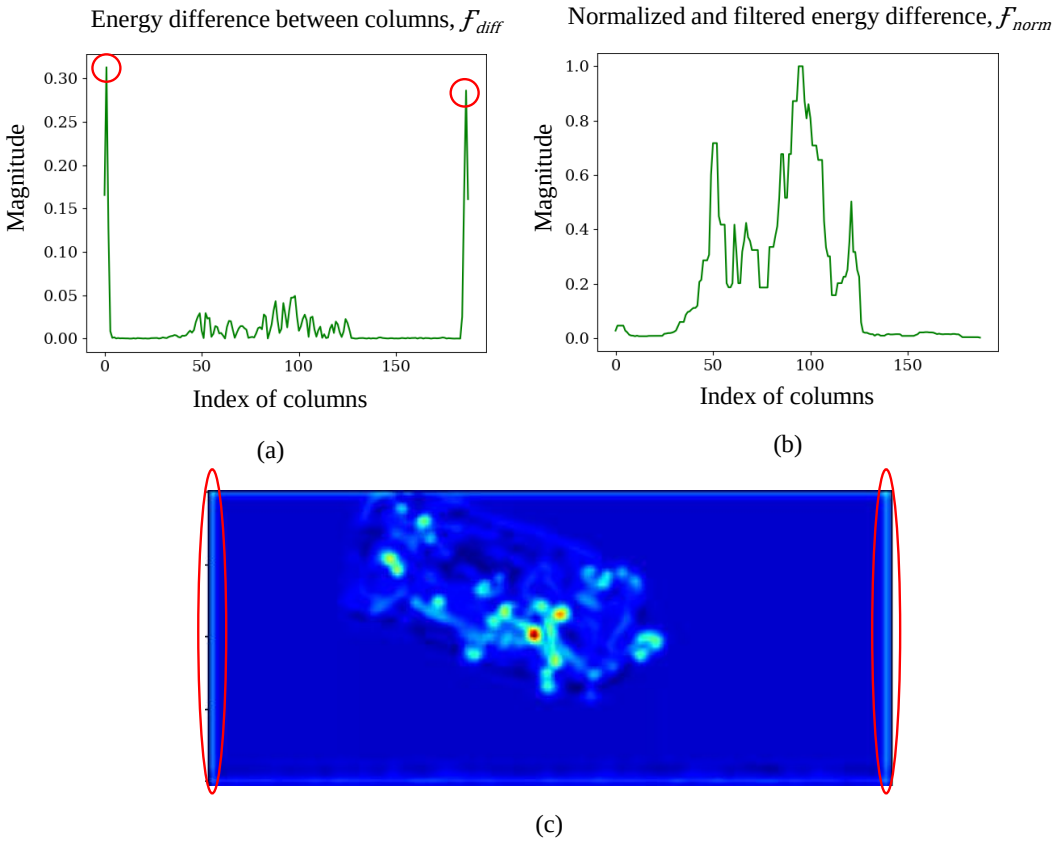


**Figure 7.** Result of median filtering and normalizing process. (**a**) is the visualization of $\mathcal{F}_{diff}$, the energy difference between columns of feature map. (**b**) is the visualization of $\mathcal{F}_{norm}$, the result of normalization and filtering process of $\mathcal{F}_{diff}$. (**c**) is the feature map, and the spikes labeled by the red circles caused by X-ray artifacts in $\mathcal{F}_{diff}$ can be seen in (**a**,**c**), which are removed by the median filter.

**Threshold-based cropping.** We determine a threshold $k$ based on $\mathcal{F}_{norm}$ to efficiently crop the image without removing salient objects. If $F_{ni} > k$, the rate of change of column $i$ is higher than the threshold, and column $i$ indicates the start of regions of interest. $\mathcal{F}_{norm}$ is scanned from left to right and vice versa to determine the horizontal boundary of the cropping region. We repeat the process vertically to decide the vertical boundary for the region of interest (RoI). After that, the original image will be cropped according to the RoI on the feature map.

## 5. Experiments

### *5.1. Experiment Setup*

**Baseline and dataset.** We use some of the most common single-stage detectors with relatively good detection performance, namely SSD [32], RetinaNet [31], and RFB Net [34] to set up the default baseline model for the cropping procedure. The dataset used for benchmarking is the SIXray-D dataset. We follow the PASCAL VOC 2007 dataset structure to split the train-test set. The original SIXray dataset contains 8820 positive images. We randomly split the dataset with a ratio of 90/10 for train/test, and reserve the test set for testing only to provide an unbiased evaluation of the model's performance. We use the recommended training parameters of SSD-512, RetinaNet, and RFB-512 from publicly available codes in PyTorch [56]. We set the training epochs to 200, and the training takes around 1–2 days to complete for each model.

For TDC, we use the baseline with the best result from the baseline benchmark to perform image cropping. The baseline is used in the image cropping and detection tasks for the test dataset. We use two configurations on the baseline: one is the default configuration, which resizes the input image to 512 $\times$ 512, the other is a dynamic input configuration that takes in arbitrary sizes image. We try to use the dynamic input model to increase the detection performance at the cost of model runtime. We use a median filter of kernel 9 to remove the X-ray artifacts and set a baseline threshold $k$ of 0.5. We believe a 50% difference between the energy of columns/rows of feature map can indicate the starting of the region of interest, then we tune it to optimize the cropping performance based on two criteria, (1) to achieve the best detection mAP and (2) to prevent over-cropping and cutoff objects of interest. We discover that ranging the threshold $k$ from 0.15 to 0.3 achieves the best performance satisfying both criteria, and $k$ being 0.15 has the highest detection performance. Thus, $k$ is set at 0.15 for both horizontal and vertical cropping.

To compare the performance of TDC with both conventional and deep learning-based image cropping, we choose two methods: the first is Canny edge detection [40] based cropping implemented using Python 3 [57], and the second method is the aesthetic-based cropping proposed by Peng et al. [24]. For the problem of cropping off white space and preserving the objects of interest, Canny edge detection is a simple and powerful unsupervised method to detect the saliency region. For the deep learning-based cropping, as there is no task-driven based image cropping and almost all of the currently available methods focus on improving the aesthetic aspect of the cropped image, we choose the method that provides the highest aesthetic scores and is available to the public. We follow the recommended parameters and use the pre-trained 512 $\times$ 512 model for the aesthetic-based cropping method [24] using TensorFlow [58]. All experiments are conducted on a single NVIDIA GTX 1080Ti GPU. Figure 8 displays some results from TDC where the background of images is cropped off and the main content is preserved.

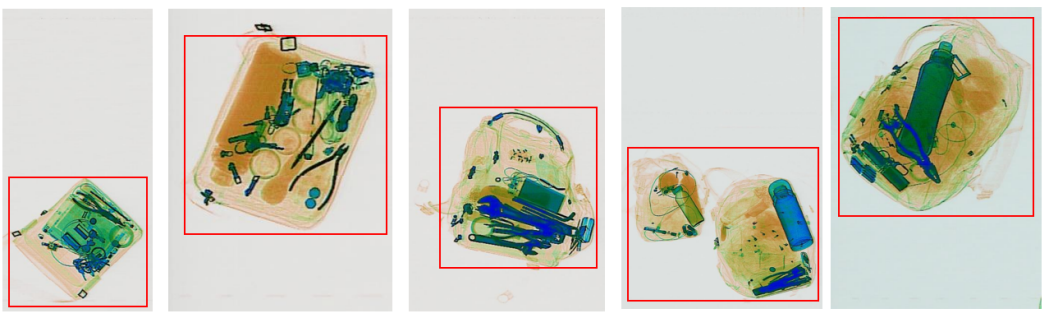

**Figure 8.** Results from TDC using RFB Net backbone. The red bounding boxes mark the output after cropping when using $k$ = 0.15.

### *5.2. SIXray-D Benchmarking*

In this section, we benchmark the detection performance on the SIXray-D dataset using mean Average Precision (mAP) from PASCAL VOC 2007 metric [59] with Intersection

over Union (IoU) of 0.5. Table 3 summarizes the experiment results on the SIXray-D test set with 836 images.

**Table 3.** Detection Average Precision on each class and mean Average Precision (mAP) on different models and test datasets. **Red** indicates the best and *blue* indicates the second best performance.

| Method | Pliers | Gun | Wrench | Scissors | Knife | Mean |
|---|---|---|---|---|---|---|
| SSD | 87.03 | 96.31 | 84.73 | 84.04 | 82.51 | 86.92 |
| RetinaNet | 82.73 | 84.51 | 75.69 | 79.95 | 74.64 | 81.50 |
| RFB | 88.78 | 96.13 | 85.92 | 84.73 | 83.22 | 87.76 |
| RFB + Edge [40] based crop | 88.79 | 95.85 | 86.12 | *86.04* | *83.93* | 88.16 |
| RFB + Aesthetic crop [24] | *89.43* | *96.32* | *86.17* | 85.48 | 83.43 | *88.38* |
| RFB + TDC | **89.52** | **96.63** | **86.19** | **87.57** | **84.37** | **88.86** |

Based on Table 3, RFB Net performs better than RetinaNet and SSD Net. Both SSD and RFB use the same reduced VGG-16 backbone for feature extraction. However, by replacing Conv2D layers in SSD with the RF module to capture features better [34], RFB Net achieves slightly higher mAP than SSD on the test set. Although using a ResNet, which is a deeper backbone [31], RetinaNet cannot match the performance of the above two models. The difference in mAP could come from the difference in architecture and activation function. RetinaNet does not have RF modules and uses FocalLoss (sigmoid activation) [31] while RFB uses Multibox loss (softmax activation) [34]. The softmax activation forces sum of the classification outputs to 1, which is more suitable in multi-classification. Alternatively, sigmoid activation is better for binary classification, hence for a dataset with multiple classes like SIXray-D, softmax in RFB Net is the better loss function. Through the experiment, the RFB Net is the best option for the cropping process baseline.

### *5.3. Cropping Performance Assessment*

In this experiment, we further evaluate the effectiveness of the proposed TDC module from two different perspectives: (1) Detection accuracy improvement using a fixed-size RFB Net; and (2) reduction of runtime using dynamic input RFB model when applying TDC on the SIXray-D dataset.

#### 5.3.1. Fixed-Size Model

For a fixed-size model, it resizes the input images into 512 × 512. It does not retain the objects' aspect ratios, but the inference time is faster than the dynamic shape input model. Table 3 summarizes the detection performance of RFB net on original and cropped SIXray-D datasets. From Table 3, TDC surpasses other cropping methods and improves the average detection mAP by 2 to the original dataset, which can be credited to the enlargement of small contraband objects such as Scissors and Pliers. While TDC can efficiently crop off the area with X-ray artifacts and redundant objects, the conventional Canny-edge [40] based cropping and deep learning-based aesthetic cropping [24] struggle to do so. Figure 9 illustrates several outputs of the three cropping methods on the SIXray-D dataset. An observation can be made that only TDC can detect the X-ray bar artifacts, the straps of the bags, and the mouse cursors on the image as noise, while the other two methods fail to recognize them, hence leading to less efficient cropping and less space reduced.

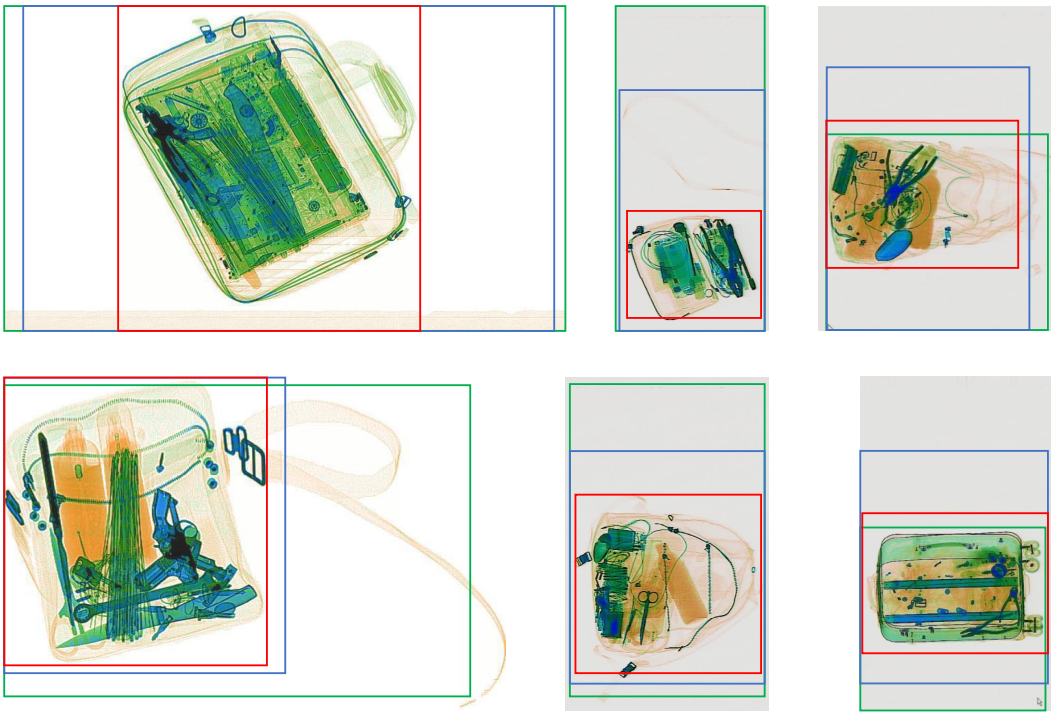

**Figure 9.** Comparison of cropping output between different methods: Green bounding boxes indicate the outputs from Canny edge [40]-based cropping; Blue bounding boxes indicate the outputs from aesthetic-based cropping [24]; Red bounding boxes indicate the outputs from our proposed TDC module.

#### 5.3.2. Dynamic Shape Input Model

As the RFB Net reuses the architecture of SSD, anchor boxes are introduced in the convolutional layers [32]. While these boxes are static in the original network, the anchor boxes are re-calculated for every single image in the dynamic shape model. Hence, this will drastically increase the runtime of the detection model. When using dynamic input shape, our goal is to achieve a better detection mAP at the cost of slower runtime. By reducing the input size through feature-based cropping, the time for anchors generation will decrease, and subsequently, the overall inference time will reduce. Table 4 summarizes the detection result when applying different cropping methods on the SIXray-D dataset and using dynamic input shape RFB Net. In this experiment, TDC proves to be the best cropping method with the highest mAP and lowest runtime compared to the other methods. The average inference time on the Dynamic RFB + TDC is decreased by 8% compared to the original set. The overall mAP is almost the same across all the cropping methods, with TDC being 0.3 mAP higher. This can be credited to the removal of the redundant background area, in which TDC surpasses other approaches as shown in Figure 9.

**Table 4.** Detector performance on SIXray-D datasets using dynamic shape input RFB Net (Dynamic RFB) after applying different cropping methods. **Red** indicates the best results and *blue* indicates the second best results.

| Method | Pliers | Gun | Wrench | Scissors | Knife | Mean | Runtime (s) ↓ | Runtime Reduction (%) ↑ |
|---|---|---|---|---|---|---|---|---|
| Dynamic RFB | *90.83* | **98.67** | 87.26 | 91.65 | 83.01 | 90.28 | 2.394 | N/A |
| Dynamic RFB + Canny edge [40]-based crop | 89.84 | 97.93 | 88.20 | 90.80 | **84.69** | 90.29 | 2.271 | 5.13 |
| Dynamic RFB + Aesthetic crop [24] | 90.52 | 98.36 | **88.76** | 89.31 | *83.90* | *90.37* | *2.221* | *7.23* |
| Dynamic RFB + TDC | **91.07** | *98.54* | *88.51* | **92.32** | 82.78 | **90.60** | **2.192** | **8.44** |

### *5.4. Ablation Study*

In this section, we assess the effect on detection accuracy using features extracted from different convolutional layers in the backbone VGG model of the RFB Net and summarize the result in Figure 10. We use the 2nd, 9th and 15th layers to represent the low, medium, and deep feature layers' outputs. An observation can be made that using the features from the 9th convolutional layer for the TDC module gives the best detection result. The lower layers only highlight contraband objects and make the algorithm cut off the important background and potential objects of interest. Meanwhile, the deep layers' features provide little change between columns and rows of the image as the whole baggage is highlighted, thus making the cropping algorithm ineffective. By using the middle convolutional layer, we can preserve the crucial background and cut off the unwanted redundant spaces in the process.

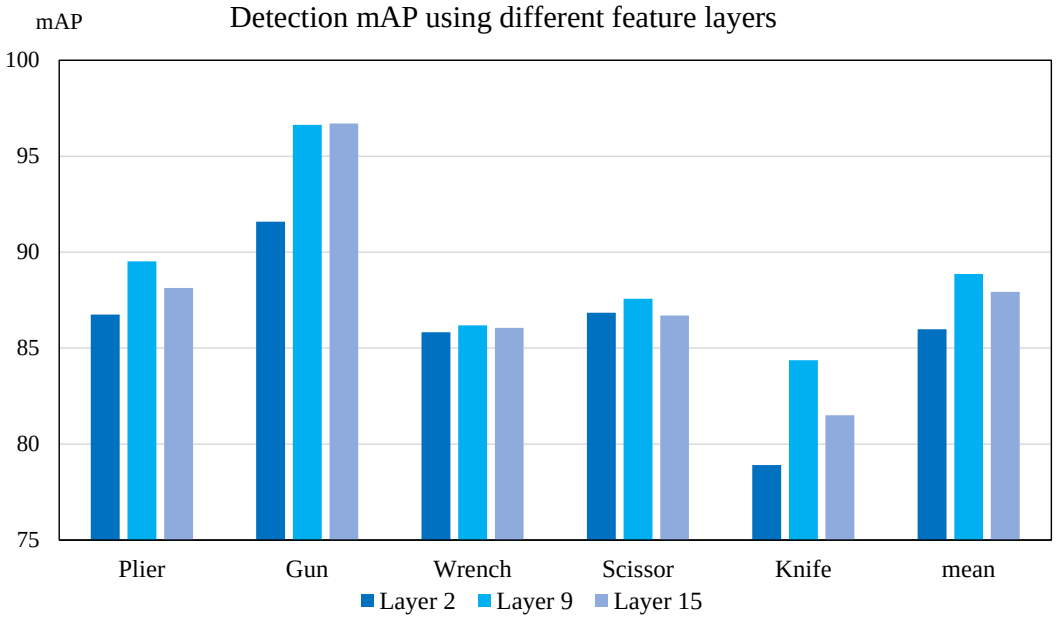


**Figure 10.** Ablation study: detection AP over each class (and the mean) of SIXray-D dataset using the RFB + TDC detection pipeline. The feature outputs are varied from the layer 2 to 15 of the RFB Net, which are used to crop the X-ray images using TDC, thus generating different detection results.

## 6. Conclusions

In this work, we attempted to tackle the practical challenges in X-ray image detection tasks. We proposed a fully-annotated SIXray-D dataset with completed positive samples with annotation boxes to benchmark the X-ray image detection tasks. We proposed the TDC module, a novel task-driven image cropping method that can effectively improve the X-ray image detection pipeline. We conducted extensive experiments showing that our proposed TDC-based X-ray detection scheme can reduce the run time for the dynamic shape input RFB net and increase the mAP for the fixed-size counterpart. The proposed TDC scheme is simply based on feature extraction without additional assumptions on the specific detector, thus it can be extended to other single-stage deep detection models as well. We plan to further investigate the effect of maintaining salient objects' aspect ratios and apply the loss to control the threshold of the cropping method in future work.

**Author Contributions:** Conceptualization, H.D.N.; methodology, H.D.N.; software, H.D.N. and R.C.; validation, H.D.N. and B.W.; data curation, R.C. and H.Z.; writing—original draft preparation, H.D.N.; writing—review and editing, B.W., R.C. and H.D.N.; supervision, B.W. and A.C.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** This work is carried out at the Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University (NTU), Singapore and the author would like to acknowledge the Vingroup Science and Technology Scholarship Program for Overseas Study for Master's and Doctorate Degrees managed by VinUniversity for the graduate research scholarship.

**Data Availability Statement:** The publicly available SIXray-D dataset was analyzed in this study. This data can be found here: https://www.ntu.edu.sg/rose/research-focus/datasets (accessed on 13 August 2021) Note that we only contribute the extra annotations for the detection task. The images are all from the existing SIXray dataset (https://github.com/MeioJane/SIXray) (accessed on 13 August 2021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chavaillaz, A.; Schwaninger, A.; Michel, S.; Sauer, J. Expertise, automation and trust in X-ray screening of cabin baggage. *Front. Psychol.* **2019**, *10*, 256. [CrossRef] [PubMed]
2. Arcúrio, M.S.; Nakamura, E.S.; Armborst, T. Human factors and errors in security aviation: An ergonomic perspective. *J. Adv. Transp.* **2018**, *2018*, 5173253. [CrossRef]
3. Bolfing, A.; Halbherr, T.; Schwaninger, A. How image based factors and human factors contribute to threat detection performance in X-ray aviation security screening. In *Symposium of the Austrian HCI and Usability Engineering Group*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 419–438.
4. Mendes, M.; Schwaninger, A.; Michel, S. Can laptops be left inside passenger bags if motion imaging is used in X-ray security screening? *Front. Hum. Neurosci.* **2013**, *7*, 654. [CrossRef] [PubMed]
5. Abidi, B.R.; Zheng, Y.; Gribok, A.V.; Abidi, M.A. Improving weapon detection in single energy X-ray images through pseudocoloring. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2006**, *36*, 784–796. [CrossRef]
6. Chen, Z.; Zheng, Y.; Abidi, B.R.; Page, D.L.; Abidi, M.A. A combinational approach to the fusion, de-noising and enhancement of dual-energy x-ray luggage images. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops, San Diego, CA, USA, 21–23 September 2005; p. 2.
7. Singh, M.; Singh, S. Optimizing image enhancement for screening luggage at airports. In Proceedings of the CIHSPS 2005. Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, Orlando, FL, USA, 31 March–1 April 2005; pp. 131–136.
8. Chan, J.; Evans, P.; Wang, X. Enhanced color coding scheme for kinetic depth effect X-ray (KDEX) imaging. In Proceedings of the 44th Annual 2010 IEEE International Carnahan Conference on Security Technology, San Jose, CA, USA, 5–8 October 2010; pp. 155–160.
9. Liu, Z.; Li, J.; Shu, Y.; Zhang, D. Detection and recognition of security detection object based on YOLO9000. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018; pp. 278–282.
10. Akcay, S.; Breckon, T.P. An evaluation of region based object detection strategies within x-ray baggage security imagery. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 1337–1341.
11. Cui, Y.; Oztan, B. Automated firearms detection in cargo x-ray images using RetinaNet. In *Anomaly Detection and Imaging with X-Rays (ADIX) IV*; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 10999, p. 109990P.
12. Morris, T.; Chien, T.; Goodman, E. Convolutional neural networks for automatic threat detection in security X-Ray images. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 285–292.
13. Wei, Y.; Tao, R.; Wu, Z.; Ma, Y.; Zhang, L.; Liu, X. Occluded prohibited items detection: An x-ray security inspection benchmark and de-occlusion attention module. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 138–146.
14. Mery, D.; Riffo, V.; Zscherpel, U.; Mondragón, G.; Lillo, I.; Zuccar, I.; Lobel, H.; Carrasco, M. GDXray: The database of X-ray images for nondestructive testing. *J. Nondestruct. Eval.* **2015**, *34*, 42. [CrossRef]
15. Caldwell, M.; Griffin, L.D. Limits on transfer learning from photographic image data to X-ray threat detection. *J. X-Ray Sci. Technol.* **2019**, *27*, 1007–1020. [CrossRef] [PubMed]
16. Rogers, T.W.; Jaccard, N.; Protonotarios, E.D.; Ollier, J.; Morton, E.J.; Griffin, L.D. Threat Image Projection (TIP) into X-ray images of cargo containers for training humans and machines. In Proceedings of the 2016 IEEE International Carnahan Conference on Security Technology (ICCST), Orlando, FL, USA, 24–27 October 2016; pp. 1–7.
17. Miao, C.; Xie, L.; Wan, F.; Su, C.; Liu, H.; Jiao, J.; Ye, Q. Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 2119–2128.
18. Kannojia, S.P.; Jaiswal, G. Effects of varying resolution on performance of CNN based image classification: An experimental study. *Int. J. Comput. Sci. Eng.* **2018**, *6*, 451–456.

[CrossRef]
19. Luke, J.J.; Joseph, R.; Balaji, M. Impact of Image Size on Accuracy and Generalization of Convolutional Neural Networks. 2019. Available online: https://www.researchgate.net/profile/Mahesh-Balaji/publication/332241609_IMPACT_OF_IMAGE_SIZE_ON_ACCURACY_AND_GENERALIZATION_OF_CONVOLUTIONAL_NEURAL_NETWORKS/links/5fa7a715299bf10f732fdc1c/IMPACT-OF-IMAGE-SIZE-ON-ACCURACY-AND-GENERALIZATION-OF-CONVOLUTIONAL-NEURAL-NETWORKS.pdf (accessed on 20 December 2021).
20. Sabottke, C.F.; Spieler, B.M. The effect of image resolution on deep learning in radiography. *Radiol. Artif. Intell.* **2020**, *2*, e190015. [CrossRef] [PubMed]
21. Shetty, C.M.; Barthur, A.; Kambadakone, A.; Narayanan, N.; Kv, R. Computed radiography image artifacts revisited. *Am. J. Roentgenol.* **2011**, *196*, W37–W47. [CrossRef] [PubMed]
22. Zhang, Y.; Yu, H. Convolutional neural network based metal artifact reduction in x-ray computed tomography. *IEEE Trans. Med. Imaging* **2018**, *37*, 1370–1381. [CrossRef] [PubMed]
23. Wang, W.; Shen, J. Deep cropping via attention box prediction and aesthetics assessment. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2186–2194.
24. Lu, P.; Zhang, H.; Peng, X.; Jin, X. An end-to-end neural network for image cropping by learning composition from aesthetic photos. *arXiv* **2019**, arXiv:1907.01432.
25. Cho, D.; Park, J.; Oh, T.H.; Tai, Y.W.; So Kweon, I. Weakly-and self-supervised learning for content-aware deep image retargeting. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017, pp. 4558–4567.
26. Wang, Y.S.; Tai, C.L.; Sorkine, O.; Lee, T.Y. Optimized scale-and-stretch for image resizing. In *ACM SIGGRAPH Asia 2008 Papers*; ACM: New York, NY, USA, 2008; pp. 1–8.
27. Tao, R.; Wei, Y.; Jiang, X.; Li, H.; Qin, H.; Wang, J.; Ma, Y.; Zhang, L.; Liu, X. Towards real-world X-ray security inspection: A high-quality benchmark and lateral inhibition module for prohibited items detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10923–10932.
28. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2016; pp. 2818–2826.
29. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2016; pp. 770–778.
31. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
32. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer; Berlin/Heidelberg, Germany, 2016; pp. 21–37.
33. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 2117–2125.
34. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.
35. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012; Kwon, Y.; TaoXie; Fang, J.; imyhxy; Michael, K.; et al. *Ultralytics/yolov5: V6.1—TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference*; 2022; doi:10.5281/zenodo.6222936. [CrossRef]
36. Chen, D.J.; Hsieh, H.Y.; Liu, T.L. Adaptive image transformer for one-shot object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12247–12256.
37. Avidan, S.; Shamir, A. Seam carving for content-aware image resizing. In *ACM SIGGRAPH 2007 Papers*; ACM: New York, NY, USA, 2007; p. 10-es.
38. Wu, J.; Xie, R.; Song, L.; Liu, B. Deep feature guided image retargeting. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, NSW, Australia, 1–4 December 2019; pp. 1–4.
39. Lin, S.S.; Yeh, I.C.; Lin, C.H.; Lee, T.Y. Patch-based image warping for content-aware retargeting. *IEEE Trans. Multimed.* **2012**, *15*, 359–368. [CrossRef]
40. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, 679–698. 1986.4767851. [CrossRef]
41. Baştan, M.; Yousefi, M.R.; Breuel, T.M. Visual words on baggage X-ray images. In *International Conference on Computer Analysis of Images and Patterns, Proceedings of the 14th International Conference, CAIP 2011, Seville, Spain, 29–31 August 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 360–368.
42. Zhang, N.; Zhu, J. A study of x-ray machine image local semantic features extraction model based on bag-of-words for airport security. *Int. J. Smart Sens. Intell. Syst.* **2015**, *8*. Available online: https://pdfs.semanticscholar.org/3bf2/5c94c1b87a7ac4731c237a17bc8cf4ba0ac2.pdf (accessed on 30 December 2021). [CrossRef]
43. Bastan, M.; Byeon, W.; Breuel, T.M. Object Recognition in Multi-View Dual Energy X-ray Images. *BMVC* **2013**, *1*, 11. Available online: https://projet.liris.cnrs.fr/imagine/pub/proceedings/BMVC-2013/Papers/paper0131/abstract0131.pdf (accessed on 30 December 2021).

44. Schmidt-Hackenberg, L.; Yousefi, M.R.; Breuel, T.M. Visual cortex inspired features for object detection in X-ray images. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 2573–2576.
45. Mery, D. Automated detection in complex objects using a tracking algorithm in multiple X-ray views. In Proceedings of the CVPR 2011 WORKSHOPS, Colorado Springs, CO, USA, 20–25 June 2011; pp. 41–48.
46. Mery, D.; Riffo, V.; Zuccar, I.; Pieringer, C. Automated X-ray object recognition using an efficient search algorithm in multiple views. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, OR, USA, 23–28 June 2013; pp. 368–374.
47. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
48. Liang, K.J.; Heilmann, G.; Gregory, C.; Diallo, S.O.; Carlson, D.; Spell, G.P.; Sigman, J.B.; Roe, K.; Carin, L. Automatic threat recognition of prohibited items at aviation checkpoint with x-ray imaging: A deep learning approach. In *Anomaly Detection and Imaging with X-Rays (ADIX) III*; International Society for Optics and Photonics: Bellingham, WA, USA, 2018; Volume 10632, p. 1063203.
49. Sigman, J.B.; Spell, G.P.; Liang, K.J.; Carin, L. Background adaptive faster R-CNN for semi-supervised convolutional object detection of threats in x-ray images. In *Anomaly Detection and Imaging with X-Rays (ADIX) V*; International Society for Optics and Photonics; Bellingham, WA, USA, 2020; Volume 11404, p. 1140404.
50. Liu, J.; Leng, X.; Liu, Y. Deep convolutional neural network based object detector for X-ray baggage security imagery. In Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 4–6 November 2019; pp. 1757–1761.
51. Dumagpi, J.K.; Jeong, Y.J. Pixel-Level Analysis for Enhancing Threat Detection in Large-Scale X-ray Security Images. *Appl. Sci.* **2021**, *11*, 10261. [CrossRef]
52. Akcay, S.; Kundegorski, M.E.; Willcocks, C.G.; Breckon, T.P. Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2203–2215. [CrossRef]
53. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
54. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Springer; Berlin/Heidelberg, Germany, 2014; pp. 818–833.
55. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *arXiv* **2014**, arXiv:1411.1792.
56. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
57. Van Rossum, G.; Drake, F.L. *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, USA, 2009.
58. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: tensorflow.org (accessed on 30 December 2021).
59. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. Available online: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html (accessed on 30 December 2021).