applied sciences

MDPI

Article

Pixel-Level Analysis for Enhancing Threat Detection in Large-Scale X-ray Security Images

Joanna Kazzandra Dumagpi * and Yong-Jin Jeong

Department of Electronics and Communications Engineering, Kwangwoon University, Seoul 01897, Korea; yjjeong@kw.ac.kr

* Correspondence: jkmdumagpi@kw.ac.kr

Abstract: Threat detection in X-ray security images is critical for preserving public safety. Recently, deep learning algorithms have begun to be adopted for threat detection tasks in X-ray security images. However, most of the prior works in this field have largely focused on using image-level classification and object-level detection approaches. Adopting object separation as a pixel-level approach to analyze X-ray security images can significantly improve automatic threat detection. In this paper, we investigated the effects of incorporating segmentation deep learning models in the threat detection pipeline of a large-scale imbalanced X-ray dataset. We trained a Faster R-CNN (region-based convolutional neural network) model to localize possible threat regions in the X-ray security images on a balanced dataset to maximize detection of true positives. Then, we trained a DeepLabV3+ model to verify the preliminary detections by classifying each pixel in the threat regions, which resulted in the suppression of false positives. The two models were combined in one detection pipeline to produce the final detections. Experiment results demonstrate that the proposed method significantly outperformed previous baseline methods and end-to-end instance segmentation methods, achieving mean average precisions (m*AP*s) of 94.88%, 91.40%, and 89.42% across increasing scales of imbalance in the practical dataset.

Keywords: X-ray; threat detection; semantic segmentation; instance segmentation; deep learning; security

check for updates

Citation: 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. https://doi.org/10.3390/app112110261

Academic Editor: Sungho Kim

Received: 14 October 2021
Accepted: 29 October 2021
Published: 1 November 2021

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

Copyright: © 2021 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 imaging is widely used for securing public spaces [1]. Developing algorithms that provide aid to human inspectors to accomplish the monotonous and nontrivial task of detecting threats in X-ray security images is of utmost importance. Recently, deep learning has become the most dominant method used in the field of automatic threat detection in X-ray security images [2]. The most common approach is to adopt deep learning models trained on natural images and apply these models to X-ray security images. However, X-ray security images do not have as many rich features as natural images. For instance, X-ray security images have limited color range, lower contrast, and poor texture. The most prominent distinguishing feature of X-ray security images is the visibility of overlap between objects, which is a challenge when adopting deep learning models because object overlap aggravates intra-class variations [3]. The pixels in X-ray security images provide insight into the state of overlap between objects since each pixel corresponds to the radiation intensity that is attenuated by all overlapping objects [4]. Darker regions in X-ray security images signify higher attenuation, which could be caused by several overlapping objects or non-overlapping objects made up of higher density materials. Similarly, lighter regions in the image mean lower attenuation. Thus, an intuitive approach to handle overlapping objects is to analyze the X-ray security images in a pixel-wise manner. Pixel-level deep learning method continues to be a popular approach used in the related field of medical X-ray imaging [5]. Pixel-level analysis can also bring similar improvements to efficiency and reliability not only limited to X-ray security applications but also encompassing other related X-ray imaging fields, such as structural materials inspection [6].

In natural images, image segmentation is the task domain concerned with the pixel-wise analysis of images, which is further broken down into tasks, namely, semantic segmentation and instance segmentation [7]. Semantic segmentation aims to classify each pixel in the image into either of the available classes of objects. Instance segmentation combines object localization and pixel-wise classification such that it results in the delineation of each object instance in the image. With regards to threat detection in X-ray security images, instance segmentation would seem to be the appropriate task domain that allows pixel-level localization of each potential threat. However, this task cannot be directly adopted to X-ray security images because it restricts that each pixel is classified into only one object instance. Figure 1 shows instance segmentation in natural images, which do not exhibit overlap. Instead, we observe occlusion, wherein the objects in front fully cover parts of the objects behind. Yet, for X-ray security images with overlapping objects, a pixel can belong to as many object instances that are overlapping, as shown in Figure 2. Thus, a more specific task domain, called object separation, is required to address the issue of overlap. Object separation can be thought of as a multi-class and multi-label version of instance segmentation. This task domain was first introduced in [8] as a method of separating potentially overlapping objects in X-ray security images, then assigning the correct pixel values according to the estimated atomic number of each object. The first part of the object separation task can detect prohibited items with non-organic material properties, such as weapons, through their shape, texture, and other visual features. The second part can detect organic prohibited items, despite the lack of visual features, such as explosives and illegal drugs, by estimating their material composition. Thus, the complete object separation task domain can be used as a general solution to detecting prohibited items in X-ray security images. Deep learning can be leveraged in the first part of the object separation task since annotations can be readily attainable. However, the second part of object separation requires additional information about the material properties of each object in the image, which is currently not available in the public domain.



(**a**) (**b**) (**c**)

Figure 1. Instance segmentation. (**a**) Input image. (**b**) Binary masks for each target object. (**c**) Isolated objects using the binary masks (for visualization).



(**a**) (**b**) (c)

Figure 2. Object separation. (**a**) Input image. (**b**) Binary masks for each target object. (**c**) Isolated objects using the binary masks (for visualization).

As a response to the growing interest in computer vision research on X-ray security images, Miao et al. [3] recently published SIXray, which is the largest database of X-ray security images that closely mirrors the practical scenario. The dataset demonstrates the visual complexities of X-ray security images, including overlapping of objects, but more importantly, the dataset places emphasis on the issue of imbalance as well. In practice, the data distribution of X-ray security images is highly skewed towards the majority class or

negative samples, i.e., images that do not contain threats, since threat objects are rarely encountered during security screening compared to normal objects. Due to the overrepresentation of the majority class, conventional models that are trained with imbalance datasets are extremely biased at always predicting the majority class. However, this becomes a major problem because the misclassification cost of the minority class or positive samples, i.e., images containing at least one threat, is significantly higher [9]. A straightforward approach in balancing an imbalanced dataset would be to either oversample the minority class or under-sample the majority class. Oversampling the minority class is not only inefficient when the majority class is magnitudes of times larger but has also been consistently shown to be inferior to other approaches. Consequently, under-sampling the majority class leads to a loss of a considerable amount of information depending on the scale of imbalance, which results in a surge of false positives as the conventional model tries to fit objects from unseen negative samples to any of the designated threat objects. Yet, it is of utmost importance to keep the false positives low if the detection algorithm is to be used as an automation tool in security systems to aid human inspectors [10].

In this paper, we investigated the impact of pixel-level analysis when it is integrated into the threat detection pipeline to address the first part of the object separation task in a large-scale and imbalanced X-ray security image dataset, also referred to as a practical X-ray dataset. We trained a Faster-RCNN [11] to localize possible threat regions on a balanced subset of the dataset to maximize the detection of the true positives. We then trained a DeepLabV3+ [12] to classify each pixel in the possible threat regions such that the regions without any pixel-level predictions are discarded as false positives. The two models were combined in one detection pipeline to produce the final predictions. Both models were selected after an exhaustive evaluation of the state-of-the-art object detection and semantic segmentation models, respectively.

The main contributions of this paper are as follows:

1. Reintroducing object separation as a unique task domain for X-ray security images;
2. Using segmentation as a mechanism to address class imbalance problem in practical X-ray security dataset;
3. Exhaustive evaluation of detection and segmentation deep learning models on the SIXray dataset; and
4. Development of a two-stage threat detection model that decouples detection and segmentation in order to maximize the use of available annotation.

Experiment results showed that pixel-level analysis significantly improved the threat detection performance, wherein the proposed method achieved a mean average precision of 94.88%, 91.40%, and 89.42%, across increasing imbalance ratios, outperforming previous baseline methods and an end-to-end instance segmentation method by a large margin.

The rest of the paper is organized as follows: Section 2 briefly reviews the related works. Section 3 provides information on the imbalanced dataset. Section 4 defines the evaluation metrics. Section 5 discusses the methodology of the proposed approach. Section 6 describes the details of the experiments and analysis of the results. Section 7 presents the conclusions of this study.

2. Related Works

In this section, we review prior studies that used pixel-level analysis as a threat detection approach in X-ray security images and the previous works that attempted to solve the class imbalance problem in large-scale X-ray security images.

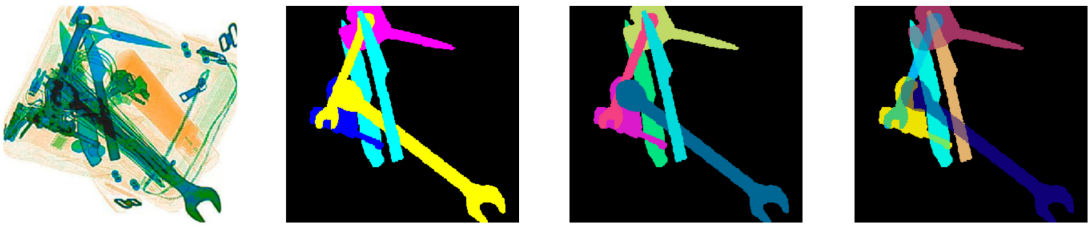
As the first to introduce the concept of object separation, the work in [6] is also the first to propose a solution to the problem. The authors exploited the log space additivity of pixels in X-ray security image sets. However, their method demands a few images from different views of the same target objects, which may not be the case in most practical port security operations. Furthermore, their approach resulted in extremely inaccurate segmentation, and therefore they instead focused on estimating the material properties using atomic numbers. Thus, their work cannot be directly applied in the detection of

other threat objects such as weapons. Since then, no work has attempted to solve the object separation task partially or completely. Alternatively, we briefly reviewed recent pixel-level approaches used in X-ray security images. The scarcity of works in this field is due to the difficulties in acquiring pixel-level annotation [2]. The study conducted in [13] evaluated classical machine learning techniques to segment objects in X-ray security images on the basis of color and later classify them into two broad classes, i.e., organic or inorganic. The work in [14] used a two-stage segmentation method for the broader task of intra-object anomaly detection. A closely related approach used semantic segmentation [15] to detect threat objects in X-ray images. However, semantic segmentation is not compatible with object separation since it cannot distinguish between instances.

Apart from publishing SIXray, Miao et al. initiated the research for threat detection in imbalanced X-ray security images by introducing the class-imbalance hierarchical refinement (CHR) approach [3], wherein a convolutional neural network (CNN), which approximates a function that removes weakly related elements from the feature map, is implemented in multiple levels of the convolution operation hierarchy. In our previous work [16], we explored the effect of filtering the initial image-level predictions using a generative adversarial network (GAN) [17] to learn the underlying distribution of negative samples so that we can designate the positives samples as anomalies when they deviate from the learned distribution. In this way, we can train a classifier on a balanced ideal dataset and suppress the induced false positives by identifying anomalies. Moreover, we also studied the effect of enlarging the pool of positive samples through image synthesis using different image-generating GANs [18]. We found that generating synthetic images by combining isolated threat objects and negative samples causes the model to disassociate the threat objects to the backgrounds distinct to positive samples, resulting in a better generalization and suppression of false positives for object-level threat detection. The deep feature fusion method in [19] combines early fusion by concatenating extracted features at the earlier stages of the classification model and late fusion by taking the weighted sum of losses calculated at different stages of the classification model. These components are integrated into a dual branch network with the aim of exploiting low-level spatial features and alleviating bias caused by the imbalanced data. A closely related approach is the tensor pooling method proposed in [20], where the authors train an instance segmentation network to classify pixels of threat objects using preprocessed inputs, wherein the image contours are extracted at different orientations to encode tensor representations. However, the authors did not consider the issue of overlap in X-ray security images and, hence, were not concerned with separating or isolating threat objects as a threat detection approach. Furthermore, their method requires that each variant of the threat category must be labeled, which exponentially increases the cost of annotation.

3. Dataset

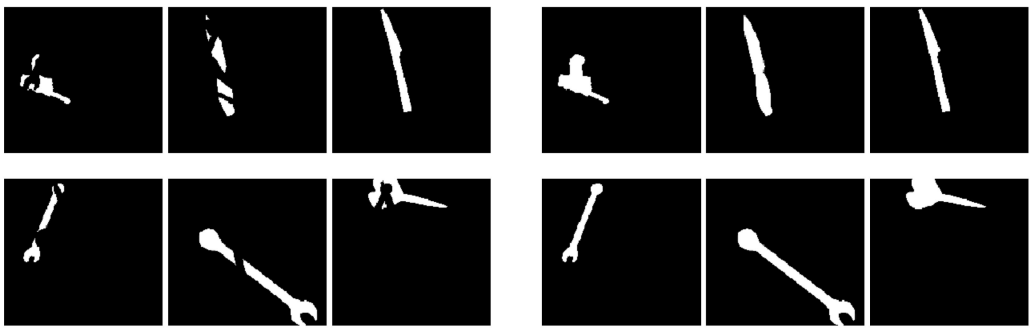
SIXray [3] is the largest publicly available benchmark dataset for X-ray security images comprised of more than 1 million images, wherein more than 8000 of which are labeled for containing at least one of the following objects: gun, knife, wrench, pliers, and scissors. The dataset is further split into subsets that correspond to the increasing ratio of imbalance between positive samples and negative samples: SIXray10, SIXray100, and SIXray1000. Even so, the dataset only consists of image-level and object-level annotations. For the task of object separation, pixel-level annotations are required. Figure 3 shows three different versions of ground truth masks for the tasks of semantic segmentation, instance segmentation, and object separation. In semantic and instance segmentation, occlusion restricts each pixel to a single semantic group and object instance, respectively, whereas object separation allows pixels to be classified under multiple object instances that describe the area of overlap between objects. Thus, one of the defining attributes of the object separation task is the ground truth labels used in supervised training.



(**a**) (**b**) (**c**) (**d**)

Figure 3. Pixel-level annotations for different tasks. (**a**) Input image with overlapping objects. (**b**) Semantic segmentation ground truth labels. (**c**) Instance segmentation ground truth labels. (**d**) Object separation ground truth labels.

Figure 4 illustrates the difference between the binary masks of each object instance for the tasks of instance segmentation and object separation, correspondingly. Suppose that the task is instance segmentation; objects at the bottom-most level of the stack will tend to lose more, if not all, pixel-wise labels to objects above it, even though its entirety can still be visually recognized in the image. It is apparent that training a model with these masks would lead to a model ignoring overlaps between objects and not have sufficient information for the objects at the bottom of the stack. Hence, we manually labelled the images such that all the pixels that describe an instance are included in the ground truth mask. However, given the high cost of annotating the entire dataset, we first randomly sampled 2500 images from the positive samples used in the training subsets and presented the initial results of our investigation with the aim of encouraging future research in this task domain to create more labeled datasets. The distribution of instances across all five categories for the pixel-level labeled dataset is presented in Table 1.



(**a**) (**b**)

Figure 4. Isolated binary ground truth labels of each object instance for (**a**) instance segmentation and (**b**) object separation.

Table 1. Distribution of instances from the subset.

	Total	Gun	Knife	Wrench	Pliers	Scissors
Instances	4826	1376	871	909	1321	349

Comprehensive data exploration of the dataset was presented in [3,21]. Still, further examination of the dataset revealed that there was a considerable portion of the negative samples that were incorrectly labeled as negative despite noticeably containing threat objects. Some of the erroneous samples are shown in Figure 5.

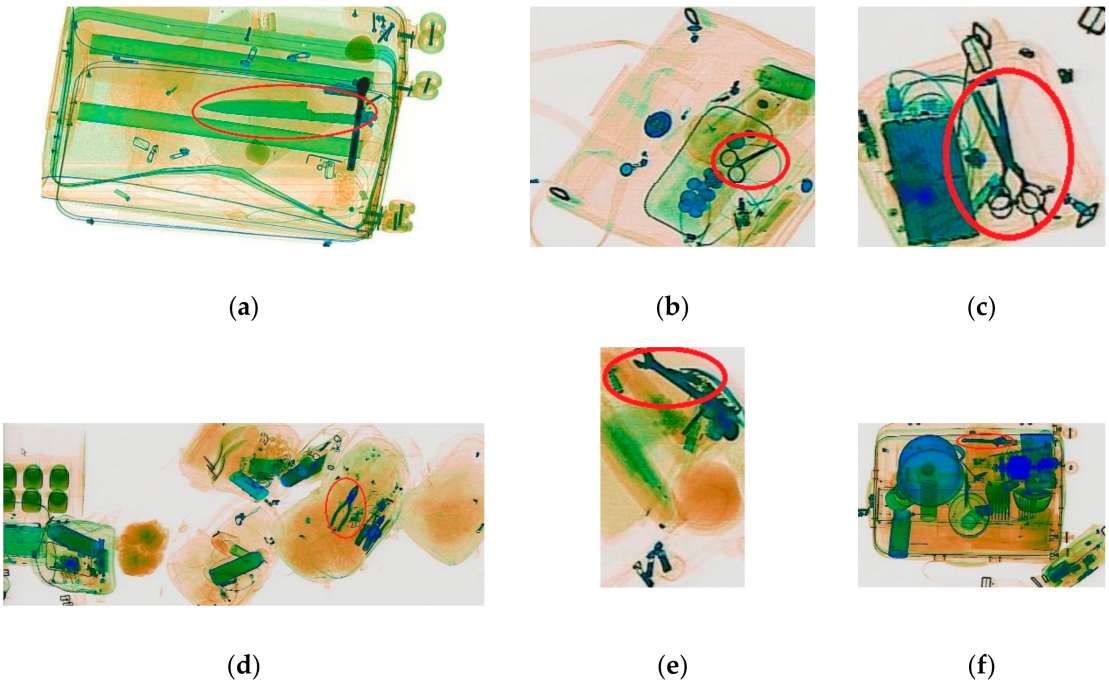


Figure 5. Mislabeled negative samples. Images are incorrectly labeled negatives despite containing (**a**) knife, (**b**) scissors, (**c**) scissors, (**d**) pliers, (**e**) wrench, and (**f**) wrench.

Noisy or corrupted labels are not uncommon in the field of machine learning. It is reported that real-world datasets include 8% to 38.5% of corrupted labels [22]. As a result, an entire field of methods for training models robust to these noisy labels has emerged, which is beyond the scope of this paper. We found that the SIXray dataset contains only about 1% mislabeled negative samples, as illustrated in Figure 6. Still, including mislabeled negative samples in evaluating algorithms is counterintuitive to the goal of security applications. In practice, we want these samples to be predicted as positives, but during evaluation, they are considered otherwise. Furthermore, using ranking metrics such as classification m*AP* adds further confusion since the predictions are sorted in descending order on the basis of the confidence scores. The best threat detection algorithms will predict these mislabeled negative samples as positive with very high confidence scores, which will cause a significant deterioration in the performance metric. Hence, in our evaluations, we removed the mislabeled negative samples from the test datasets to reveal the true strength of the algorithms. The proportion of mislabeled negatives is small enough to not affect the ratio of imbalance in the subsets. For comparison with previous works, we did not remove the mislabeled negative samples in the dataset we used for training our models, and we reevaluated previous works using the test datasets without the mislabeled data. Additionally, we have provided the list of the mislabeled negative samples as well as the pixel-level annotations of the data subset (https://github.com/jodumagpi/Xray-ObjSep-v1, accessed on 8 October 2021).

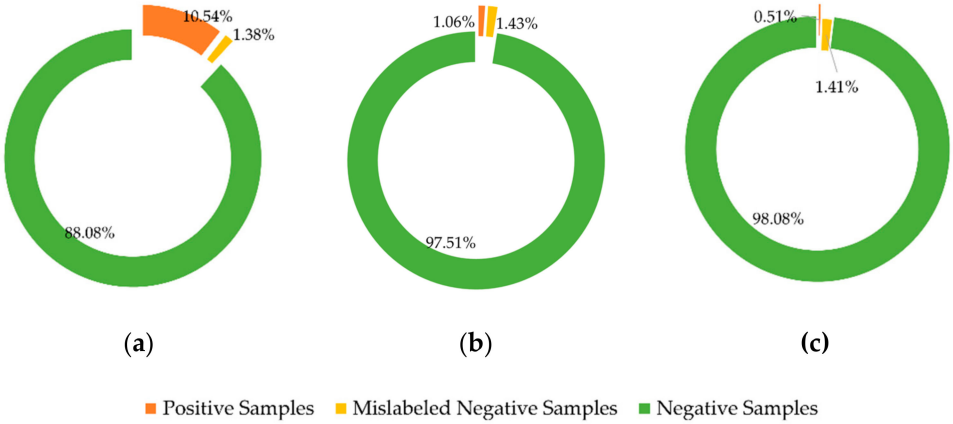


Figure 6. The proportion of positive samples, mislabeled negative samples, and the rest of the negative samples for the (**a**) SIXray10, (**b**) SIXray100, and (**c**) SIXray1000 subsets.

4. Evaluation Metrics

To select the detection model, we evaluated the performance using average precision (*AP*) [23], as defined in Equation (1).

$$AP = \sum_{n=0} (r_{n+1} - r_n)\ p_{interp}(r_{n+1}) \tag{1}$$

where r is the recall, and p_{interp} is the interpolated precision given by $p_{interp}(r_{n+1}) = \max_{\widetilde{r} \geq r_{n+1}} p(\widetilde{r})$, wherein p is the precision at $\widetilde{r}$, and n includes all the recall points. *AP* is also defined as the area under the precision-recall curve. The m*AP* is the average of the *AP*s calculated for each class of threat objects. This is one of the most used metrics for evaluating detection and classification performance.

Moreover, the segmentation model is chosen on the basis of its performance on various evaluation metrics, namely, intersection-over-union (*IoU*), dice coefficient (*DC*), precision, and recall, as defined in Equations (2)–(5).

$$IoU = \frac{|A\ \cap B|}{|A\ \cup B|} \tag{2}$$

$$DC = \frac{2|A\ \cap B|}{|A| + |B|} \tag{3}$$

where A and B represent the target and predicted segmentation masks, respectively. *IoU*, also known as the Jaccard index, is calculated for each class of threat objects, and the average is reported as the mean *IoU* (m*IoU*). *DC* is another metric that has become increasingly popular in reporting the performance of modern segmentation algorithms, which bears some resemblance to the *IoU* metric.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

where *TP*, *FP*, and *FN* are the proportions of the pixels in the masks that are considered true positives, false positives, and false negatives, respectively. Precision and recall can be calculated for each class and as the cumulative of all classes.

As per the benchmark standard set in [3], we again used the m*AP*, defined in Equation (1), to evaluate our proposed method against previous works. We took the *AP* on each of the classes by ranking the classification predictions by their confidence

scores, then took the average (m*AP*) across all the classes to report the overall performance of the models.

5. Proposed Method

In this section, we discuss the details of the model selection experiments and the description of the complete threat detection pipeline.

5.1. Model Selection

Training an end-to-end object separation model requires all image-level, object-level, and pixel-level annotation for each data entry. Yet, this kind of dataset is non-existent in the field of X-ray security images. Choosing to train an end-to-end model with a limited dataset has the downside of throwing away the data with incomplete annotations. Instead, we created a threat detection pipeline wherein the localization of the object instances and the prediction of segmentation masks are separated into two different tasks handled by two different models. This way, all the samples in the dataset with only image-level and object-level annotations are used to train the first model, i.e., the object detection model, and the limited samples with additional pixel-level annotations are used to train the second model, i.e., the segmentation model. Combining these models in one pipeline results in an object separation model that can localize each object instance and delineate all the pixels belonging to that instance while also maximizing the utilization of the available annotations. Consequently, selecting the appropriate models to use for the two tasks are conducted using the experiments discussed in the following subsections.

5.1.1. Detection Models

In selecting the best detection model, we evaluated four of the most popular state-of-the-art approaches.

Faster R-CNN [11] belongs to the family of two-stage object detection models known as region-based CNNs. The first stage of the algorithm is known as the region proposal network (RPN), where areas in the image that are suspected of containing the target objects are extracted and passed down to the second stage of the algorithm, which then classifies the objects contained in the extracted area. The localization is improved by considering the prediction of the bounding boxes as a regression task, where the difference between the coordinates of the real bounding boxes and the coordinates of the proposed regions are considered. This model is designed to be the fastest and most accurate compared to its previous iterations [24,25].

You Only Look Once (YOLOv3) [26] belongs to the one-stage object detectors that do not require a region proposal as part of the algorithm and, instead, analyze a dense sampling of the possible locations. In YOLO, the input image is divided into grid cells, and each cell forecasts a fixed number of bounding boxes, called priors, along with the confidence scores. All these predictions are simultaneously made using a single CNN, making it one of the fastest algorithms suitable for real-time applications. In this paper, we adopted the third version of the algorithm, which is designed to detect small objects better with the help of the incorporation of shortcut connections.

Single-shot Multibox Detector (SSD) [27] is also a one-stage algorithm that performs detection at every pyramidal layer of the CNN to focus on objects of varying scales. Instead of dividing the input into grids, SSD predicts the offset of the default boxes, also known as anchor boxes, for every point in the feature map. Receptive fields of feature maps differ at every level in the pyramid of the CNN. Earlier layers tend to have fine-grained feature maps, and later layers tend to have coarse-grained feature maps. Since the anchor boxes have fixed sizes relative to their corresponding cell, predictions at later layers capture larger objects in the image, while predictions at the earlier layers capture smaller objects in the image.

RetinaNet [28] is another one-stage algorithm that bears a resemblance to SSD in the sense that it also uses the concept of detection at each level of pyramidal layers of the

CNN. Yet, they also attach a feature pyramid network (FPN) [29], which concatenates later feature maps with earlier feature maps to gain stronger representations. Furthermore, they introduce a different loss function, called focal loss, with the aim of putting more weight on hard samples, i.e., samples that are often misclassified.

We only used positive samples to evaluate the effectiveness of each detection model in detecting threats when they are present in the image. We divided the data into training and validation sets. We used the same backbone network for all the models, i.e., ResNet-50 [30], and trained them all for 60,000 iterations. Table 2 shows the per-category *AP* and the m*AP* of the detectors on the validation set. Although per-category results are not consistent, the overall evaluation favors Faster R-CNN as the most accurate detector. Moreover, RetinaNet already has the advantage of having an FPN attached to its backbone network, which all the other models did not have. This may have been the reason why among the other one-stage detectors, it achieved the closest overall performance to the Faster R-CNN approach. On the basis of this result, we selected Faster R-CNN as our object detection model.

Table 2. Detection mean average precision (%) on the validation set.

Method	Mean	Gun	Knife	Wrench	Pliers	Scissors
Faster R-CNN	**86.71**	96.24	**82.92**	86.79	82.05	**85.53**
YOLO	82.84	93.04	75.84	85.26	79.87	80.19
SSD	84.32	95.31	78.90	85.41	80.10	81.91
RetinaNet	85.68	**97.06**	70.33	**91.61**	**86.18**	83.21

5.1.2. Segmentation Models

To determine the best segmentation model for the task, we evaluated six of the most popular state-of-the-art approaches.

FCN [31] is one of the earliest approaches to pixel-level classification. It takes the conventional CNN architecture and replaces the fully connected layers with a convolutional layer by arguing that the dense layers can be thought of as doing 1×1 convolutions. The final convolution layer is then up-sampled using deconvolution to learn non-linear up-sampling and produce a feature map that has a similar size as the input, wherein each pixel corresponds to the predicted classes.

U-Net [32] builds on top of FCN in that it is also comprised of symmetric encoding and decoding fully convolutional layers that form a U-shape. Additionally, it adds skip-connections between corresponding encoding and decoding layers to reinforce the information that is lost during downsampling.

PSPNet [33] introduces the pyramid pooling module, which concatenates the feature maps from the layers of the backbone model to capture global context, which is important in providing indications on the distribution of the segmentation classes across the image. Moreover, it uses an auxiliary loss applied at the input of the pooling module to serve as intermediate supervision during the training.

DeepLabV3 [34] adds several techniques to achieve a finer delineation of objects, such as atrous convolutions and spatial pyramid pooling (SPP). SPP is used to capture multi-scale context by applying multiple pooling layers that divide the feature maps of the last convolutional layer into fixed spatial bins and concatenating the output vector to be fed to the subsequent fully connected layer. As this increases the complexity of the model, atrous convolutions, also known as dilated convolutions, are used so that the receptive fields of filters are larger, thereby incorporating a larger context without increasing the number of parameters.

PAN [35] proposes two new modules to the segmentation framework, namely, feature pyramid attention (FPA) and global attention up-sample (GAU). GAU guides the low-level features by combining them with a global context extracted from performing global average pooling to the high-level features. FPA learns better representation by combining global pooling and spatial pyramid attention on the output of the backbone model.

MA-Net [36] also applies attention mechanism in its architecture specifically by introducing two new blocks: point-wise attention block (PAB) and multi-scale fusion attention block (MFAB). PAB captures the spatial dependencies between pixels in the global view, and MFAB captures the channel dependencies between any feature map by multi-scale semantic feature fusion.

We only used the pixel-level annotated data to evaluate the segmentation models. We used the detection model selected from the previous model selection experiment to extract patches or regions in the samples that were to be used to train the segmentation models. We added background patches, i.e., regions that do not contain threats, by running negative samples through the chosen detection model. The resulting dataset was then divided into training and validation sets. We also used the same backbone network for all the models, i.e., ResNet-50 [30], and trained them all for 60,000 iterations. Table 3 shows the results on the validation set using various evaluation metrics. We selected DeepLabV3 as our segmentation model since it outperformed the other approaches on most of the evaluation metrics.

Table 3. Segmentation performance on the validation set using various evaluation metrics (%).

Method	m*IoU*	DC	Precision	Recall
UNet	83.32	87.64	94.16	87.04
PSPNet	81.54	86.71	94.68	85.05
MANet	80.81	84.46	95.87	83.69
PANet	82.28	86.62	96.38	84.82
FCN	84.42	88.48	**96.48**	86.86
DeepLabV3	**86.77**	**90.54**	93.89	**91.52**

5.2. Full Pipeline

The final threat detection pipeline is illustrated in Figure 7. We preprocessed all images using [37], wherein we cropped out the unnecessary air space in the image, leaving only the relevant information in the data. The cropped image was fed to the detection model, which predicted the locations of the predicted threat objects. In our final pipeline, we improved the backbone of the Faster R-CNN model by attaching FPN. The predicted regions were cropped out of the image and then fed to the segmentation model, which predicted the classification of each pixel in the region. In our final pipeline, we improved the segmentation model by using the next iteration of the algorithm, called DeepLabV3+ [12]. This latest version of the algorithm introduces atrous depthwise convolution that combines the ideas from atrous convolution and depthwise separable convolution [38] to drastically reduce the network parameters while maintaining, or even achieving better, performance. Furthermore, we also enhanced the model used in the encoder by using ResNext-50 [39] architecture with squeeze-and-excitation (SE) blocks [40]. We made the pixel level predictions threshold to be at 95% so that only the pixels with strong detections were considered. The regions without a pixel-level annotation were discarded. In this sense, the segmentation model acts as a verification model that ensures all predicted regions contain threat objects.

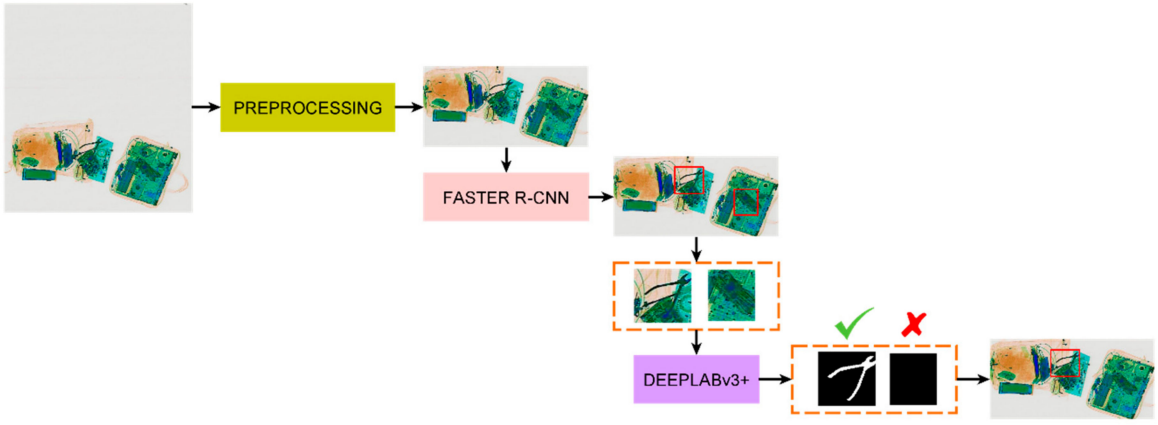


Figure 7. Full detection pipeline.

6. Experiment Results and Discussion

In this section, we discuss the experiment setup and the analysis of the results.

6.1. Experiment Setup

We trained our models using the predefined subsets of the SIXray dataset. SIXray10, SIXray100, and SIXray1000 have the ratio of negative samples over positive samples at 1:10, 1:100, and 1:1000, respectively. All the images in the training subsets were used to train the detection model, while only the pixel-level annotated images that are in the training subsets were used to train the segmentation model. We generated the patches used to train the segmentation model as described in the previous section. The spatial dimensions of all the patches were resized to 192×192.

We trained Faster-RCNN for 80,000 iterations with a batch size of 2 using stochastic gradient descent (SGD) [41] with a base learning rate of 0.001, which is linearly decayed by a factor of 0.1 after the 30,000th and 50,000th iterations, and a momentum of 0.9. We trained DeepLabV3+ for 250 epochs with a batch size of 32 using Adam optimizer [42] with a base learning rate of 0.001, which is linearly decayed by a factor of 0.1 after the 75th, 150th and 200th epochs. For further comparison, we also trained Mask R-CNN [43], a state-of-the-art end-to-end instance segmentation framework, using all of the samples in the training set with complete annotations. Mask R-CNN was trained with the same backbone and configurations as Faster R-CNN.

6.2. Experiment Results

Tables 4–6 show the quantitative results of the experiments on the predefined test datasets. We compared our approach to the baseline classification method, CHR [3]; our previous classification method, GBAD [16]; and an end-to-end instance segmentation method, Mask-RCNN [43]. The removal of the mislabeled negative samples caused a substantial performance boost in the reevaluation of the previous works showing that their true strength was obscured by the quality of the dataset. Still, results revealed that image-level approaches drastically fell behind object-level and pixel-level approaches. Moreover, it is especially concerning how image-level approaches cannot cope with increasing ratios of imbalance. In contrast, both Mask R-CNN and our proposed approach showed more robustness to the imbalance, proving that localized methods such as object-level and pixel-level approaches can drastically enhance the performance of threat detection models, which is of utmost importance in security applications. Furthermore, our approach enjoys an even larger boost in performance compared to Mask R-CNN thanks to the uncoupling of the detection and segmentation task, which allowed for the use of the entire training set.

Table 4. Classification mean average precision (%) for SIXray10.

Approach	Mean	Gun	Knife	Wrench	Pliers	Scissors
CHR	79.93	99.40	87.84	66.15	85.34	60.94
GBAD	84.75	99.72	90.06	76.80	86.02	71.13
Mask R-CNN	87.74	98.76	83.32	81.44	92.06	83.10
Proposed	**94.88**	**99.79**	**95.02**	**90.91**	**96.62**	**92.04**

Table 5. Classification mean average precision (%) for SIXray100.

Approach	Mean	Gun	Knife	Wrench	Pliers	Scissors
CHR	64.54	98.6	81.09	46.69	62.72	33.61
GBAD	66.14	95.74	85.25	54.87	53.10	41.74
Mask R-CNN	86.55	98.74	77.72	78.35	89.78	77.11
Proposed	**91.40**	**99.46**	**90.24**	**86.88**	**94.37**	**86.06**

Table 6. Classification mean average precision (%) for SIXray1000.

Approach	Mean	Gun	Knife	Wrench	Pliers	Scissors
CHR	47.21	96.68	70.18	22.82	16.86	29.52
GBAD	47.50	90.70	77.89	20.09	27.06	21.78
Mask R-CNN	82.24	98.02	82.91	75.98	76.80	77.48
Proposed	**89.42**	**99.03**	**88.95**	**85.18**	**89.81**	**84.12**

The multi-label detection confidence of two randomly sampled images from the positive and negative samples is shown in Table 7. Detection confidence of 100% conveys that the model is certain that the particular threat is present in the image, while detection confidence of 0% conveys that the model is certain that the image does not contain the particular threat. Analysis of the detection confidence will enable us to gain more insight into the overall m*AP* performance of each approach. m*AP* relies on the ranking of the detection confidence on each target class. Thus, a false positive with high detection confidence and a false negative with low detection confidence causes significant degradation in the performance. Meanwhile, a true positive with high detection confidence and a true negative with low detection confidence is desired for optimum performance. For the positive sample, i.e., Figure 8a, CHR [3] correctly detected the knife with relatively high confidence, but the rest of the threat classes also had considerably high detection confidence, especially the gun class. On the other hand, GBAD [16] also correctly detected the knife with 100% certainty and had extremely low detection confidence for the wrench, pliers, and scissors classes, but, unfortunately, it also had 100% certainty for the presence of a gun. From this sample alone, we might expect that GBAD performs worse on the gun and better on the rest of the target classes when compared to CHR. Indeed, we see this reflected in Table 5. The filtering mechanism of Mask R-CNN [43] and our proposed method allowed for the complete suppression of potential false positives to 0%. Both segmentation approaches correctly detected the knife and did not make any other false detections. Still, our approach boasts higher detection confidence for the same detected object. For the negative sample, i.e., Figure 8b, both CHR and GBAD incorrectly detected a gun object in the image with high confidence and, for the rest of the classes, the latter method continued to have extremely low detection confidence. It can be observed that GBAD had a tendency to have extreme detection confidence for any case, which may have been the reason for its marginal improvement over CHR. Meanwhile, Mask R-CNN was not able to suppress the incorrect detection of scissors, unlike our approach, which was able to correctly suppress any potentially wrong detection. It can be observed that

the high prediction confidence on true positives and the consistent suppression of false positives caused the significant performance improvement of our proposed approach over the other methods.

Table 7. Multi-label detection confidence (%) for randomly sampled images shown in Figure 8.

Image	Approach	Gun	Knife	Wrench	Pliers	Scissors
a	CHR	24.0	98.30	0.06	0.00006	0.003
	GBAD	100	100	0.0000007	0.000002	0.0000002
	Mask R-CNN	0	97.99	0	0	0
	Proposed	0	99.98	0	0	0
b	CHR	99.97	0.03	0.000004	0.02	0.00005
	GBAD	100	0.0000005	0.000006	0.000005	0.00000008
	Mask R-CNN	0	0	0	0	93.33
	Proposed	0	0	0	0	0



(a) **(b)**

Figure 8. Randomly sampled images from the SIXray100 test set. The images were drawn from the pool of (**a**) positive and (**b**) negative samples.

Figures 9–11 shows the qualitative results of the experiment. Figure 9a shows how our approach was able to accurately segment the detected threat objects even though there being overlap between three objects, i.e., knife, wrench, and gun. However, despite being able to produce segmentations for all the correct detections, our approach is shown in Figure 9b to have had a hard time in accurately delineating overlapping objects when most of them had high-density material properties. Still, for most of the non-overlapping objects, our approach was able to correctly verify the detections.

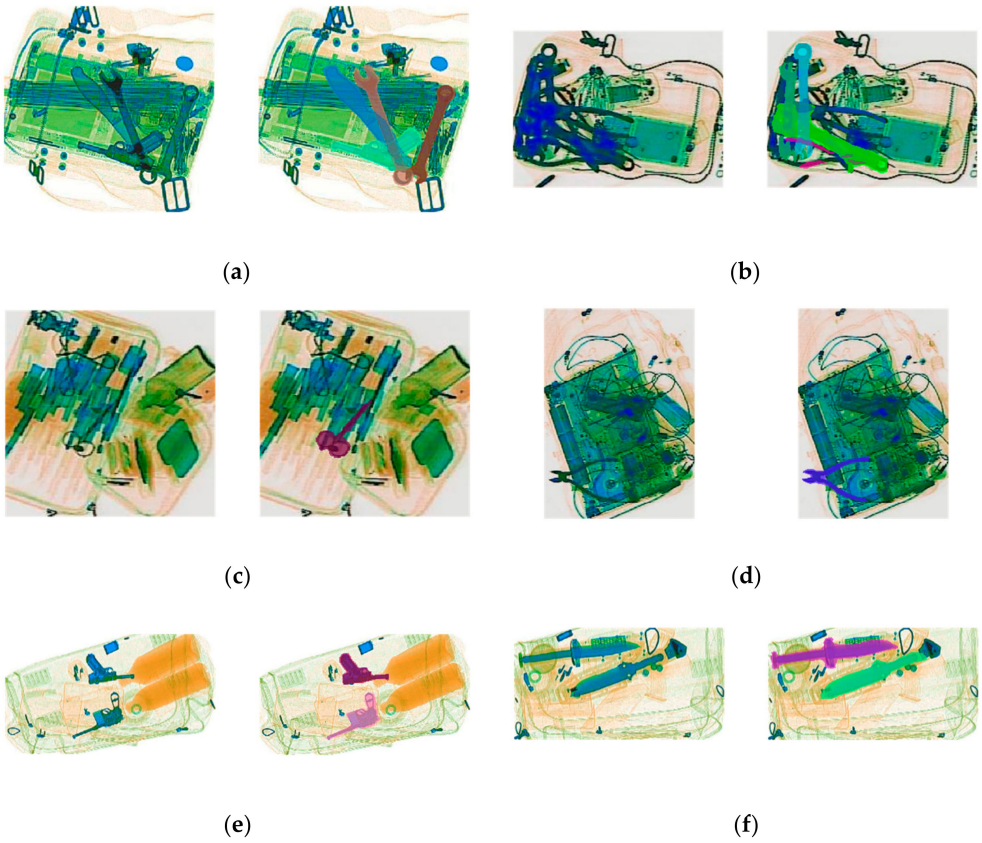


Figure 9. Exemplar images of verified detections. Images on the right are the inputs and images on the left are the outputs with overlayed masks. The examples range from overlapping threat objects, with (**a**) low densities and (**b**) high densities, to non-overlapping objects such as (**c**) scissors, (**d**) pliers, (**e**) guns, and (**f**) knives.

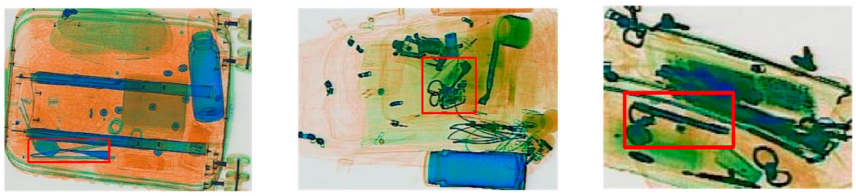
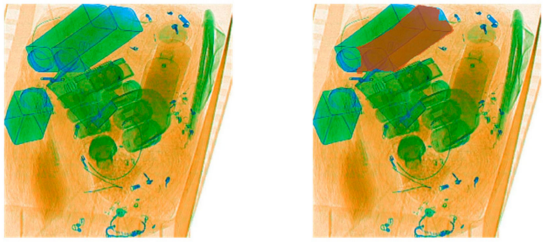


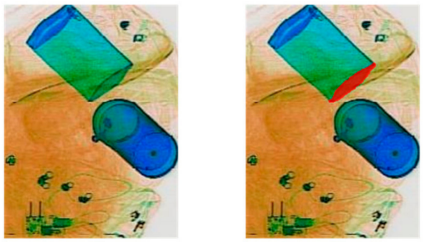
Figure 10. Exemplar images of suppressed detections. Wrong detections for (**a**) knife, (**b**) scissors, and (**c**) another pair of scissors.

Figure 10 shows the false positive detections that were suppressed by our approach. These were regions in the negative samples that were predicted by Faster R-CNN to have threat objects. Since DeepLabV3+ did not predict any segmentation mask for these regions, they were not included in the final prediction output. We observed an overwhelming number of false positives for the scissors class, which was mostly squashed by the second stage of our approach. It may also be the case that the other approaches also produced too many unsuppressed false positive predictions for the scissors class, causing it to suffer the most degradation compared to other classes. Lastly, Figure 11 shows some of the failure cases of our approach. In these images, Faster R-CNN wrongly predicted suspected regions to which DeepLabV3+ generated segmentation masks. We observe that most of the errors came from wrong localization and verification of suspected knives and wrenches. This may have been due to the bland features of the objects in these classes, which can be easily mistaken as one of the elongated metals in the baggage. In contrast, the guns class

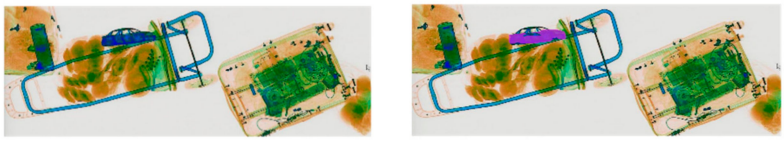
benefitted from having very strong visual features in that it was able to consistently be predicted with high accuracy, regardless of the imbalance.



(**a**)



(**b**)



(**c**)

Figure 11. Exemplar images of unsuppressed detections. Images are wrongly predicted to contain (**a**) knife, (**b**) knife, and (**c**) gun. Images on the right are the inputs, and images on the left are the outputs with overlayed masks.

6.3. Ablation Study

We examined each module of our threat detection pipeline to determine their individual contribution to the final algorithm performance. Table 8 shows the performance of four different cases represented by the different configurations of the threat detection pipeline. First, we only considered the detection model (Det) without the preprocessing and verification by the segmentation model. Then, we added the preprocessing algorithm to the detection model (Crop + Det) but still omitting the segmentation. Next, we combined detection and segmentation (Det + Segm) but removed the preprocessing algorithm. Finally, we merged all the modules into one threat detection pipeline (Crop + Det + Segm). Table 8 shows the results of the evaluation using SIXray100 subset. As demonstrated in our previous work [37], we can achieve a marginal yet valuable improvement by simply cropping out irrelevant areas in the image, such as the air gaps/spaces, which is very common in X-ray security images. Cropping the images to expose only relevant information and extract more features mostly enhances the detection model's ability to detect more objects with higher confidence. Due to being trained on a balanced training subset, the detection model is forced to fit the vast amount of unseen data from the negative samples to the more familiar targets in the positive samples. While this increases the detection of true positives, it also inadvertently increases the detection of false positives. This issue proves to be the main limiting factor and is thereby rectified by integrating the segmentation model to verify the initial predictions.

Table 8. Classification mean average precision (%) of the ablation study using SIXray100.

Approach	Mean	Gun	Knife	Wrench	Pliers	Scissors
Det	84.70	98.39	82.89	79.88	90.58	71.77
Crop + Det	86.11	98.76	85.22	81.75	91.05	73.77
Det + Segm	90.54	99.22	90.04	85.19	93.45	84.82
Crop + Det + Segm	**91.40**	**99.46**	**90.24**	**86.88**	**94.37**	**86.06**

7. Conclusions

In this paper, we investigated the impact of integrating pixel-level analysis in the threat detection pipeline of a large-scale and imbalanced X-ray security image dataset. We reintroduced the object separation as a unique task domain for analyzing X-ray security images and address the first part of the task, which aimed to accurately delineate target object instances from X-ray images, including pixels that were shared by overlapping objects. We developed a straightforward and effective object separation pipeline composed of a detection model for localizing possible threat regions and a segmentation model for verifying the existence of threats in the predicted regions. We chose the appropriate detection and segmentation models by extensively evaluating current state-of-the-art deep learning models.

Our empirical results show that object-level and pixel-level approaches significantly outperformed image-level approaches for threat detection in X-ray security images. Our approach outperformed the baseline classification and end-to-end instance segmentation methods by up to 26.86% and 4.85%, respectively, on the subset that closely mirrored a practical scenario. Furthermore, our approach consistently outperformed the previous works across all the subsets with increasing scales of imbalance. These preliminary results support the claim that our intuitive approach works effectively without the need to build an entirely new framework or device a complicated data processing technique. Object separation has the potential to advance research in automated threat detection in X-ray security images as well as other X-ray imaging applications. However, exploration in this field is limited due to the lack of pixel-level annotations of X-ray security image datasets. We believe that this study would encourage researchers to create more high-quality labeled datasets and develop more sophisticated approaches to fully solve the object separation task, which can help in the generalization of the detection of all prohibited items such as weapons, explosives, and drugs. In our future research, we intend to create an end-to-end object separation framework.

Author Contributions: Conceptualization, J.K.D. and Y.-J.J.; methodology, J.K.D. and Y.-J.J.; software, J.K.D.; validation, Y.-J.J.; formal analysis, J.K.D. and Y.-J.J.; investigation, J.K.D. and Y.-J.J.; resources, J.K.D.; data curation, J.K.D.; writing—original draft preparation, J.K.D.; writing—review and editing, J.K.D. and Y.-J.J.; visualization, J.K.D.; supervision, Y.-J.J.; project administration, Y.-J.J.; funding acquisition, Y.-J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Kwangwoon University and by the MISP Korea, under the National Program for Excellence in SW (2017-0-00096) supervised by IITP.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Mery, D.; Saavedra, D.; Prasad, M. X-ray Baggage Inspection with Computer Vision: A Survey. *IEEE Access* **2020**, *8*, 145620–145633. [CrossRef]
2. Akcay, S.; Breckon, T. Towards Automatic Threat Detection: A Survey of Advances of Deep Learning within X-ray Security Imaging. *arXiv* **2020**, arXiv:2001.01293.

3. 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 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2114–2123. [CrossRef]
4. Mery, D. Computer vision for x-ray testing: Imaging, systems, image databases, and algorithms. In *Computer Vision for X-ray Testing*; Springer: Cham, Switzerland, 2015; pp. 1–12. ISBN 978-3-319-20746-9. [CrossRef]
5. Hesamian, M.H.; Jia, W.; He, X.; Kennedy, P. Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. *J. Digit. Imaging* **2019**, *32*, 582–596. [CrossRef]
6. Wu, S.C.; Xiao, T.Q.; Withers, P.J. The imaging of failure in structural materials by synchrotron radiation X-ray microtomography. *Eng. Fract. Mech.* **2017**, *182*, 127–156. [CrossRef]
7. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, 1–22. [CrossRef]
8. Heitz, G.; Chechik, G. Object separation in x-ray image sets. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2093–2100. [CrossRef]
9. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [CrossRef]
10. Hättenschwiler, N.; Sterchi, Y.; Mendes, M.; Schwaninger, A. Automation in airport security X-ray screening of cabin baggage: Examining benefits and possible implementations of automated explosives detection. *Appl. Ergon.* **2018**, *72*, 58–68. [CrossRef] [PubMed]
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
12. Chen, L.-C.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Computer Vision—ECCV, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 833–851. [CrossRef]
13. Chouai, M.; Merah, M.; Sancho-Gómez, J.-L.; Mimi, M. A Machine Learning Color-Based Segmentation for Object Detection within Dual X-Ray Baggage Images. In Proceedings of the 3rd International Conference on Networking, Information Systems & Security, Marrakech, Morroco, 31 March–2 April 2020; Association for Computing Machinery: New York, NY, USA, 2020. [CrossRef]
14. Bhowmik, N.; Gaus, Y.F.A.; Akcay, S.; Barker, J.W.; Breckon, T.P.; Akçay, S.; Barker, J.W.; Breckon, T.P. On the Impact of Object and Sub-Component Level Segmentation Strategies for Supervised Anomaly Detection within X-ray Security Imagery. In Proceedings of the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 986–991. [CrossRef]
15. An, J.; Zhang, H.; Zhu, Y.; Yang, J. Semantic Segmentation for Prohibited Items in Baggage Inspection. In *Intelligence Science and Big Data Engineering, Visual Data Engineering*; Cui, Z., Pan, J., Zhang, S., Xiao, L., Yang, J., Eds.; Springer International Publishing: Cham, Switzerland, 2019; Volume 106, pp. 495–505. [CrossRef]
16. Dumagpi, J.K.; Jung, W.Y.; Jeong, Y.J. A new GAN-based anomaly detection (GBAD) approach for multi-threat object classification on large-scale x-ray security images. *IEICE Trans. Inf. Syst.* **2020**, *E103D*, 454–458. [CrossRef]
17. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
18. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-Attention Generative Adversarial Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 7354–7363.
19. Xu, Y.; Wei, J. Deep Feature Fusion Based Dual Branch Network for X-ray Security Inspection Image Classification. *Appl. Sci.* **2021**, *11*, 7485. [CrossRef]
20. Hassan, T.; Akçay, S.; Bennamoun, M.; Khan, S.; Werghi, N.; Akcay, S.; Bennamoun, M.; Khan, S.; Werghi, N. Tensor Pooling Driven Instance Segmentation Framework for Baggage Threat Recognition. *Neural Comput. Appl.* **2021**. [CrossRef]
21. Dumagpi, J.K.; Jeong, Y.J. Evaluating gan-based image augmentation for threat detection in large-scale xray security images. *Appl. Sci.* **2021**, *11*, 36. [CrossRef]
22. Song, H.; Kim, M.; Park, D.; Shin, Y.; Lee, J.-G. Learning from Noisy Labels with Deep Neural Networks: A Survey. *arXiv* **2020**, arXiv:2007.08199.
23. Everingham, M.; Van Gool, L.; Williams, C.K.I.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
24. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
25. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
26. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
27. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.Y.; Berg, A.C. SSD: Single shot MultiBox detector. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37. [CrossRef]

28. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Doll´ar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef]
29. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; et al. Feature Pyramid Networks for Object Detection. In Proceedings of the Pattern Recognition and Computer Vision (PRCV), Guangzhou, China, 23–26 November 2017; pp. 2117–2125. [CrossRef]
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, CA, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
31. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *arXiv* **2014**, arXiv:1605.06211.
32. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241. [CrossRef]
33. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239. [CrossRef]
34. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
35. Li, H.; Xiong, P.; An, J.; Wang, L. Pyramid Attention Network for Semantic Segmentation. *arXiv* **2018**, arXiv:1805.10180.
36. Fan, T.; Wang, G.; Li, Y.; Wang, H. MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation. *IEEE Access* **2020**, *8*, 179656–179665. [CrossRef]
37. Dumagpi, J.K.; Jung, W.; Jeong, Y. KNN-Based Automatic Cropping for Improved Threat Object Recognition in X-Ray Security Images. *J. IKEEE* **2019**, *23*, 1134–1139. [CrossRef]
38. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
39. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. [CrossRef]
40. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [CrossRef]
41. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
42. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. *arXiv* **2015**, arXiv:1412.6980.
43. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef]