Automatic threat recognition of prohibited items at aviation checkpoints with x-ray imaging: a deep learning approach

Kevin J Liang[a], Geert Heilmann[b], Christopher Gregory[c], Souleymane O. Diallo[c], David Carlson[a], Gregory P. Spell[a], John B. Sigman[a], Kris Roe[c], and Lawrence Carin[a]

[a]Duke University, Durham, NC 27708, United States of America
[b]Smiths Detection, Im Herzen 4, 65205 Wiesbaden, Germany
[c]Smiths Detection, 2202 Lakeside Blvd, Edgewood, MD 21040, United States of America

ABSTRACT

The Transportation Security Administration safeguards all United States air travel. To do so, they employ human inspectors to screen x-ray images of carry-on baggage for threats and other prohibited items, which can be challenging. On the other hand, recent research applying deep learning techniques to computer-aided security screening to assist operators has yielded encouraging results. Deep learning is a subfield of machine learning based on learning abstractions from data, as opposed to engineering features by hand. These techniques have proven to be quite effective in many domains, including computer vision, natural language processing, speech recognition, self-driving cars, and geographical mapping technology. In this paper, we present initial results of a collaboration between Smiths Detection and Duke University funded by the Transportation Security Administration. Using convolutional object detection algorithms trained on annotated x-ray images, we show real-time detection of prohibited items in carry-on luggage. Results of the work so far indicate that this approach can detect selected prohibited items with high accuracy and minimal impact on operational false alarm rates.

Keywords: Deep Learning, Threat Recognition, Security Screening, Convolutional Neural Networks, Machine Learning, Computer Vision, Object Detection

1. INTRODUCTION

It is the responsibility of the Transportation Security Administration (TSA) to ensure the safety of the traveling public within the US, including the over 2.5 million passengers passing through American airports each day.[1] As such, before boarding an airplane, every passenger must pass through a security checkpoint, where the TSA screens carry-on baggage and personal belongings for dangerous and prohibited items. To facilitate screening, the TSA employs dual-energy multi-view x-ray scanners produced by Smiths Detection and other vendors that provide a non-intrusive internal view of bags. These scanners produce images color-coded to show material properties that are then displayed on screens for human Transportation Security Officers (TSOs) to examine. Bags or bins containing potential threats are removed for further inspection.

The current Concept of Operations (CONOPs) requires the TSO to visually inspect each image to pick out threats, which can be a challenging task. In this context, threats refer to items prohibited by the TSA, which can vary widely, including (but not limited to) firearms, sharps, blunt weapons, precursors, and explosives. Not only do these items come in many different and evolving forms, they are also often packed in cluttered bag environments, and many can be confused with benign objects of similar material properties or shapes. Certain threats are quite rare as well, requiring TSOs to maintain vigilance over long shifts. Moreover, in order to maintain high passenger throughput at the checkpoints, TSOs must make their decisions quickly.

While challenging for humans alone, computer algorithms that analyze scans alongside human operators may boost overall performance. Current scanners already implement algorithms that calculate material properties from dual-energy multi-view scans, automatically highlighting objects or regions that might contain explosives

Further author information:
Kevin J Liang: E-mail: kevin.liang@duke.edu
Christopher Gregory: E-mail: christopher.gregory@smiths-detection.com

or other prohibited items. It is of TSA interest to extend this automatic detection capability to support operator detection of other prohibited item classes, such as firearms or sharps. Such an algorithm must be accurate enough to be trusted, have a low enough false alarm rate so as to not be a distraction, and be fast enough to not slow down current operations–we aim for a rough cutoff of about a second per bag.

Within the greater field of computer vision, the task of localizing and classifying objects is a canonically studied problem, commonly termed "object detection." In this context, localization refers to determining the location of an object within an image, often by producing the coordinates of a box that tightly bounds it (a "bounding box"); classification refers to the selection of one of a pre-determined number of class labels for each such object. Locating and identifying threat objects is exactly what TSOs at security checkpoints do every day, so developing the ability to do so automatically is of value. In recent years, the emergence of deep learning, a subfield of machine learning, has resulted in an unprecedented leap in the performance of object detection models. In particular, methods based on convolutional neural networks[2–7] have resulted in algorithms that have proven effective at detecting a wide range of object classes.[8–13]

The bulk of object detection research has focused on datasets of natural images,[14–16] but x-ray scans of baggage possess certain unique aspects. X-ray scans are produced by transmission (photons pass completely through the target), as opposed to reflections off of surfaces. This means that individual items can appear superimposed on top of each other, while also appearing in any orientation. Additionally, multiple x-ray detectors are positioned within the scanner to provide multiple views of the same object, unlike single-perspective natural images typically considered in object detection. Nonetheless, deep learning has already demonstrated some success for x-ray image security screening.[17–19] The work shown in this paper, however, is part of the first effort to incorporate deep learning in real-world systems at US airport security checkpoints. The goal of this work is to detect and automatically highlight prohibited items in bags as well as determine the performance of these methods on datasets collected for this study. Although the list of classes to detect for this research effort is quite extensive, discussion will be limited to the detection of firearms (e.g. guns) and sharps (e.g. knives).

In this paper, we describe methods and present results from an ongoing research effort funded by the TSA to develop a deep learning based Automatic Threat Recognition (ATR) system for airport checkpoint scanners. In Section 2, we describe (1) the Smiths Detection platform used to collect the x-ray images and (2) the data collection and labeling protocols. In Section 3, we introduce five deep object detection models which we apply to the labeled data, as well as metrics for evaluating system performance and the hardware set-up of the deployable prototype. For results (Section 4), we show object detection metrics on firearms and sharps datasets, and we demonstrate improvement by combining detection results from the multiple views of a single scan. We also describe a test to extrapolate how such a prototype system might perform when deployed in the field, showing threat detection alongside false alarm rates.

2. DATA COLLECTION

2.1 Smiths Detection X-ray System

Data used for the training and testing of the ATR methods were collected by Smiths Detection personnel using a Smiths Detection host x-ray system. The platform is a cabinet x-ray security system that contains four separate pairs of 160 KeV x-ray sources and detector arrays, arranged opposite each other around a tunnel. Passenger bags and other belongings are carried through the tunnel by a conveyor belt at a rate of 240 mm/sec. Each x-ray beam is collimated to a narrow width for optimal resolution, while the x-ray line detectors are arranged linearly; radiation emitted from the sources passes through the objects being scanned before reaching the detectors. The system is dual-energy: the x-ray detectors measure intensity at a high and low energy band. Image slices of a given view are assembled to produce two grayscale images corresponding to the two energies (see Figures 1a and 1b), yielding eight images in total across the four views. All training and evaluation were performed on these low and high energy images.

Although outside the scope of this paper, for purposes of illustration it is helpful to show color images, which make certain material characteristics easier to see with the human eye. The images that are displayed to the TSOs at airport checkpoints are post-processed color images that fuse the high and low energy scans to estimate the effective atomic number, Z-effective, of materials along the x-ray path from source to detector; Figure 1c

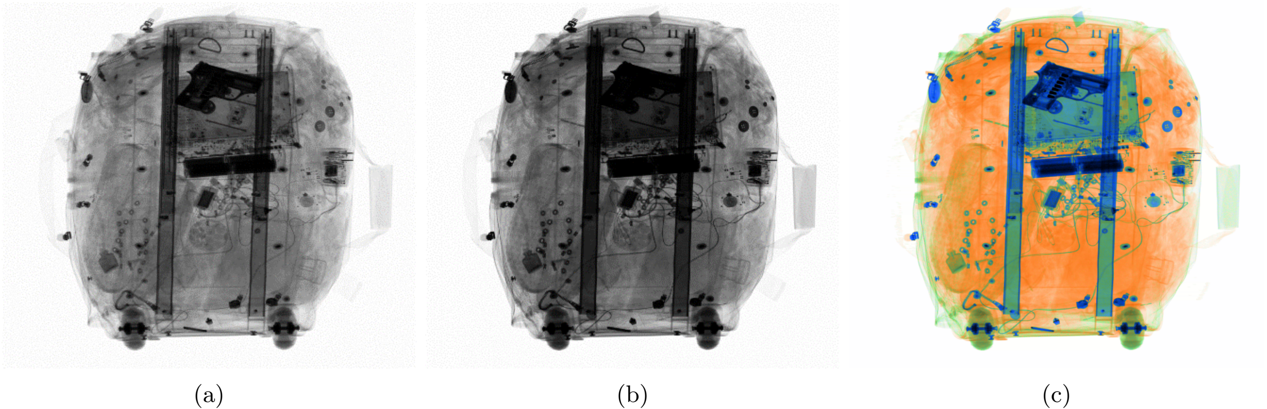

(a) (b) (c)

Figure 1: (a) High energy. (b) Low energy. (c) Color-mapped Material lumination. Various image types generated by the Smiths Detection host x-ray system. Images scanned with a laboratory prototype not in TSA configuration.

shows the color image corresponding to the high and low images in Figures 1a and 1b, respectively. The colors have been selected to correspond to three different Z-effective bands. Orange corresponds to materials of low Z-effective (e.g. organic materials), green corresponds to materials of medium Z-effective (e.g. ceramics), and finally blue corresponds to materials of high Z-effective (e.g. metals). The brightness (or intensity) of the color image reflects the absorption (and thus provides information on relative thickness) of the materials. The process for computing the colors is proprietary and uses a periodic calibration process to assure consistent results.

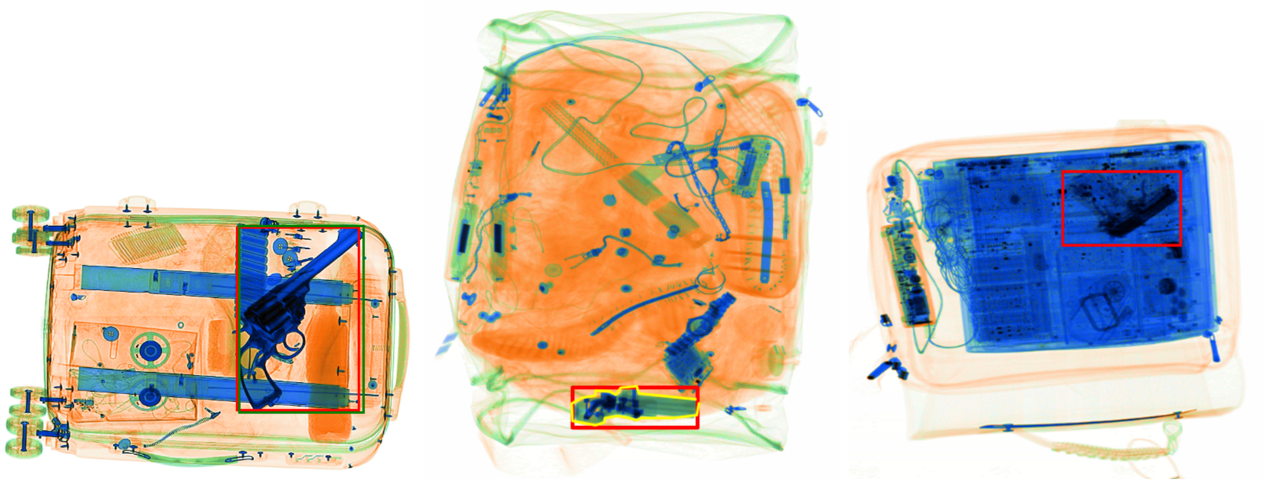
While visually helpful to humans, it was found that the material color-mapped RGB images do not provide any noticeable benefit to the deep learning algorithms. We conjecture that this is because the color mapping is simply a transformation of the high and low energy images, and deep learning is capable of effectively learning its own representations. Because of these considerations, we use the high and low energy images for training the model, but use color-mapped material lumination when presenting results.

2.2 Images and Labeling

Two types of image data were used in the training of the ATR algorithm: non-threat and threat. The non-threat data, referred to as Stream-of-Commerce (SOC) data, were acquired over multiple days from real traffic at airport checkpoints. These data are generally assumed not to contain any threats and represent negative examples for training and false alarm evaluation. Positive training examples with threats were collected on multiple occasions at the Transportation Security Laboratory (TSL) and at a Smiths Detection laboratory. For this investigation, a total of 37 hand guns, 92 pocket knives, and 20 other mixed sharps were scanned in approximately 100 different pre-packed bags. Guns and sharps of several sizes were considered. Over the course of a few days, 2022 scans of guns, 1350 scans of pocket knives, and 706 scans of other mixed sharps were acquired.

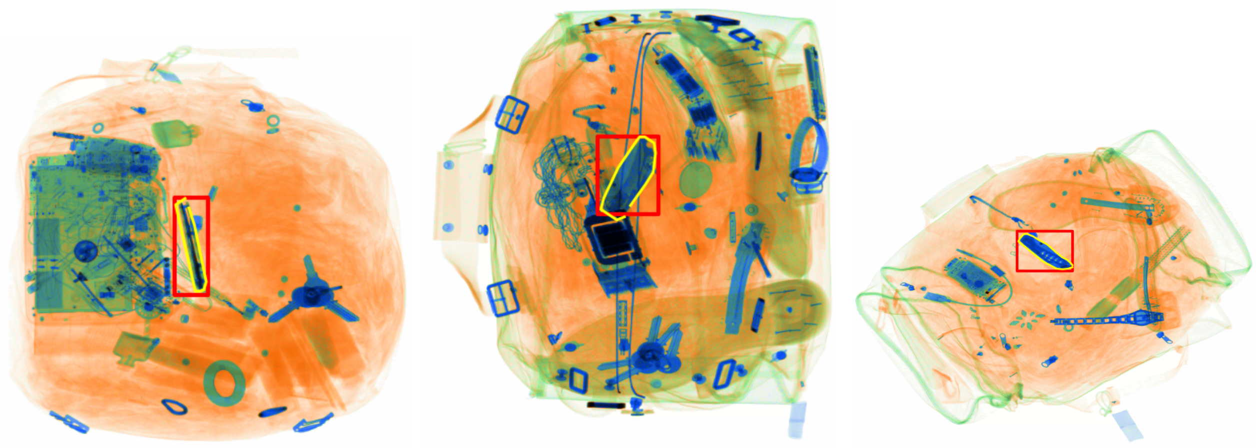
A systematic process was performed to acquire the data. Sets of bags were filled with contents typical of passenger luggage, and a threat item was packed into each. These bags were then scanned in multiple positions and orientations with the x-ray system. Upon completion of a number a scans, the threat was typically moved to a new bag in the bag sequence, and the scanning process was repeated. Two measures were implemented to avoid overfitting on specific bags. First, bag contents were periodically altered: this involved adding materials to the bag to inject clutter into the scene or partially obscure the threat object in at least one view. Second, the bag sets were rotated so that no single bag was scanned excessively throughout the data collection. To ensure that the threat objects were scanned in multiple perspectives, their location and orientation within the bags were also varied. This occurred while moving the threats from one bag to the next during data collection. For instance, a small gun that was laid flat (horizontally) in the center of Bag 1 may be placed along the front of Bag 2 in a vertical orientation.

Most object detection algorithms are supervised, meaning that they require training data with labels in the format of the expected output. Within this context, this means that both image scans of threats as well as threat localizations need to be collected. Labeling of the training set was performed manually using a Smiths



(a) (b) (c)

Figure 2: Example bounding boxes of firearms in red. Images scanned with a laboratory prototype not in TSA configuration.



(a) (b) (c)

Figure 3: Example bounding boxes of sharps in red; segmentation outlines also visible in yellow. Images scanned with a laboratory prototype not in TSA configuration.

Detection proprietary utility to draw an outline around each threat object. These outlines were then used to create a binary mask of the threat object's location. Since the methods explored in this paper rely on bounding boxes rather than pixel-wise segmentations, the min and max coordinates of these masks were used to generate bounding boxes to train the model; see Figures 2 and 3 for examples.

3. METHODS

3.1 Models

Virtually every state-of-the-art object detection algorithm begins with some form of Convolutional Neural Network (CNN) operating as a feature extractor, followed by some form of specialized architecture for producing bounding box coordinates and classifications. The primary difference between these algorithms pertains to this latter part. While there have been many object detection models that have been proposed in recent years, we focus on a few popular models with high performance: Faster Regions with Convolutional Neural Networks (Faster R-CNN),[10] Single Shot MultiBox Detector (SSD),[11] and Region-based Fully Convolutional Networks (R-FCN).[12]

For the convolutional feature extractor, many CNNs have been developed, primarily designed for image classification; however, it has been shown that such CNNs do indeed also work well for object detection[8] and that there is a positive correlation between image classification and object detection performance.[13] However, classification accuracy is not the only consideration; networks with larger numbers of parameters may have higher representational power, but are often computationally slower as a result (see Table 1 for a comparison). Since maintaining high throughput of bags at airport checkpoints is of interest, the algorithm cannot take an arbitrarily long time to deliberate.

CNN	Top-1 Accuracy	Number of parameters
MobileNet[7]	71.1	3,191,072
Inception V2[4]	73.9	10,173,112
ResNet-101[5]	76.4	42,605,504
Inception ResNet V2[6]	80.4	54,336,736

Table 1: Top-1 accuracy on ImageNet classification and model size for the CNNs used in experiments.[13]

Each of the aforementioned object detection models, termed meta-architectures, along with several different CNNs have been implemented in TensorFlow[20] as part of Google's Object Detection API,[13] and we leverage these implementations as the basis for our experiments. Models vary from approximately 5 to 50 frames per second, depending on the CNN and detection meta-architecture. Note, the model must process all four views and perform additional overhead to acquire and display the results, but even the slowest of these models still meets our target of delivering results within one second.

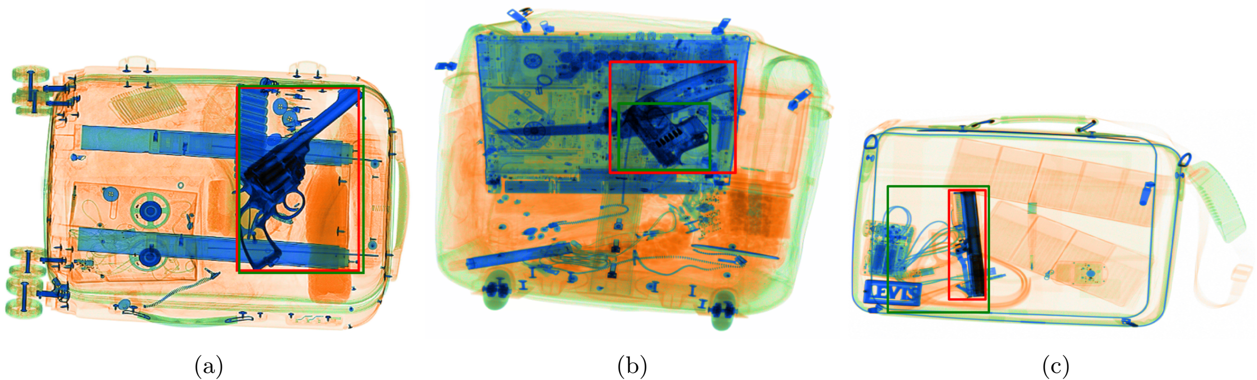
We train each model entirely on data from a single class (firearms or sharps) and report results individually. Each model can be run independently on a particular scan to look for threats of a particular category. However, the models were originally developed for multi-class discrimination, and nothing prevents us from doing the same. Training separate models was due to the order in which data was collected and annotated.

3.2 Evaluation Metrics

In order to quantify model performance, we borrow several common metrics from the object detection and information retrieval literature:

- *Intersection over Union (IOU)*: IOU is the ratio calculated by the intersection (overlap) of two sets divided by their union. A value of 0 implies no overlap, while a value of 1 means that the two sets are equal (Intersection = Union). See Figure 4 for examples.
- *True Positive (T_p)*: Defined as a correctly classified box that has an IOU above a threshold (commonly 0.5) with a ground truth box[*].
- *False Positive (F_p)*: A proposed bounding box that either misses the classification or is not tight enough to achieve an IOU above the set threshold.
- *False Negative (F_n)*: A ground truth object that was not properly bounded and classified.
- *Precision* ($\frac{T_p}{T_p+F_p}$): The proportion of proposed bounding boxes (algorithm-produced detections) that are correct.
- *Recall* ($\frac{T_p}{T_p+F_n}$): The proportion of objects (ground-truth) that were correctly detected by the algorithm.

[*] Often object detection algorithms produce many bounding boxes with high IOU with the ground truth object. While these are all technically correct, this kind of output is not as desirable, so often only one true positive is allowed per ground truth. Non-maximal suppression[21] is typically used to cut down the number of "repeat" detections for a single object.



(a) (b) (c)

Figure 4: Ground truth boxes shown in green; generated detection boxes shown in red. Images scanned with a laboratory prototype not in TSA configuration. (a) IOU = 0.92, alarm is above IOU threshold, resulting in a T_p. (b) IOU = 0.43, alarm is too big, resulting in a F_p and a F_n. (c) IOU = 0.31, alarm is too small, resulting in a F_p and a F_n.

- *Average Precision (AP)*: The area under the curve (AUC) of the precision-recall curve for a single class.
- *mean Average Precision (mAP)*: The mean of the APs across all classes. For a single class problem (as we consider here), the mAP is equivalent to the class AP.
- *Percent Correctly Localized (CorLoc)*: Percentage of positives correctly identified as a T_p by the model. Equivalently, the recall value for the point on the precision-recall curve at which we choose to operate.

The latter two are reported for the quantitative results to compare performances. Additionally, precision-recall curves are shown to give a sense of performance for various operating thresholds.

3.3 Multi-View Evaluation

Many x-ray systems have several x-ray detector lines, providing orthogonal views of a bag. As alluded to in Section 2.1, the Smiths Detection host x-ray system used for data collection provides four views, meaning each scan results in four images. While these can be treated independently during ATR, this ignores spatial correlations of objects between views (see Figure 5). In the field, TSOs do not ignore threats which appear in only a single view, so we can consider the performance of the ATR if we apply an OR-gate to detections in scans. This means a T_p is only required in one out of four scans, while all F_ps are counted the same as in single-view evaluation.

Multi-view performance is estimated through two perspectives: object detection and real-world threat recognition. For each, the deep learning object detection algorithm is first run on all views of each scan independently. Performance is then scored accordingly, depending on the metric. Deep learning object detection literature is primarily concerned with the concept of information retrieval: there are ground truth objects in each image, and the goal is to tightly bound as many as possible while avoiding false positives and mislabeled objects. mAP and CorLoc capture these objectives. For the implications of ATR in the field, we choose to analyze detection and

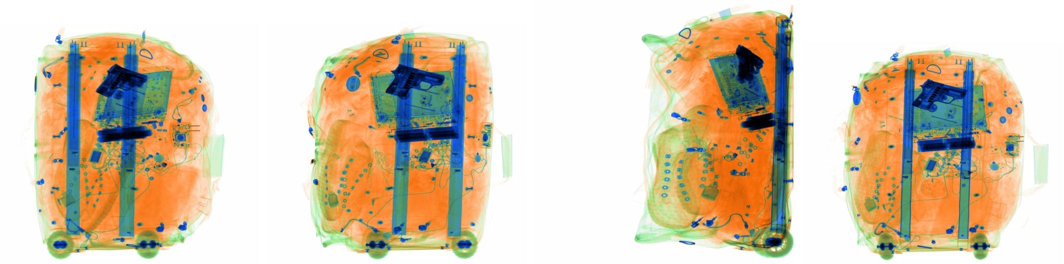


Figure 5: Four views generated during a scan. The Y-axis is the conveying direction. Images scanned with a laboratory prototype not in TSA configuration.

false alarm rates on a per bag basis. In Section 4.2 we discuss results of multi-view to object detection metrics, and in Section 4.3, we show simulated real-world performance and the corresponding false alarm rates.

3.4 Hardware Implementation

The object detection models are trained on labeled data with a NVIDIA Titan X GPU. After training has converged, learned model weights are saved for inference. Training and evaluation of images can be conducted with a pre-scanned and labeled dataset on any workstation set-up with enough computational power. Results shown throughout the rest of this paper were evaluated offline on a held-out test set, for speed and reproducibility.

The end goal, however, is to deliver a system that can be deployed to airport security checkpoints. Therefore, initial laboratory prototypes pipe scanned images to a GPU and computer bolted to the exterior of the x-ray scanner to compute threat locations, which are then sent back and projected on top of the color images on the main display. As part of this effort, such a prototype system has been demonstrated to TSA sponsors in a setting and manner consistent with anticipated evaluation and possible field use.

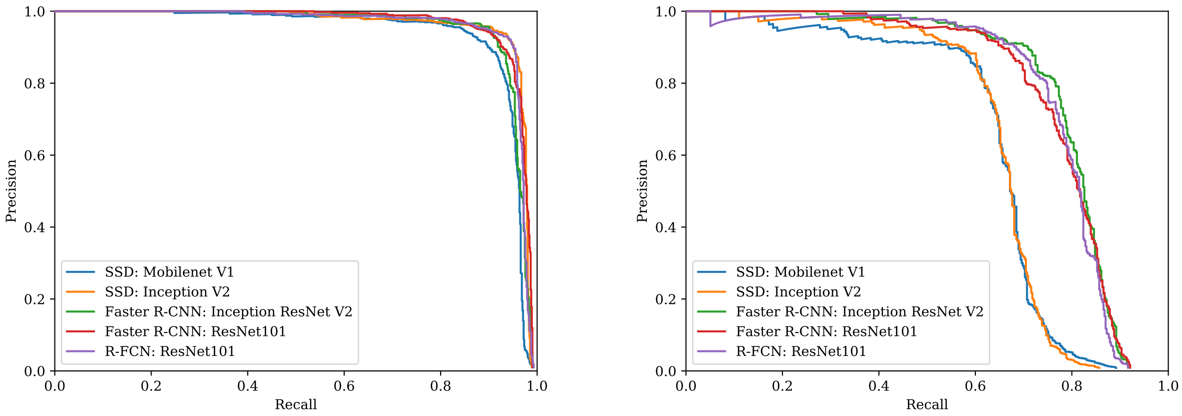
4. RESULTS

4.1 Object Detection Evaluation: Single View

ATR performance was first evaluated using object detection metrics (mAP, CorLoc), treating all four views generated from a single bag as independent images. Images were randomly shuffled into a roughly 70:10:20 training:validation:test split at the bag level, ensuring views of the same bag ended up in the same split.

4.1.1 Firearms

With a single view, all tested object detection models do well on a dataset of 2022 firearms scans (see Figure 6a and Table 2). An SSD model with a MobileNet V1 convolutional feature extractor, designed for speed and compactness, has the lowest mAP and CorLoc, but still achieves 0.9393 and 0.9295 respectively. Faster R-CNN with ResNet101 achieves the highest mAP of 0.9644, while R-FCN with ResNet101 has the highest CorLoc of 0.9550.



(a)

(b)

Figure 6: Precision-recall curve of each objection model on (a) firearms and (b) sharps data. Curves approaching the upper right corner imply better performance.

	Firearms		Sharps	
Model	mAP	CorLoc	mAP	CorLoc
SSD: MobileNet V1[7,11]	.9393	.9295	.6463	.6872
SSD: Inception V2[4,11]	.9621	.9514	.6575	.6828
Faster R-CNN: Inception ResNet V2[6,10]	.9546	.9478	**.8003**	**.7775**
Faster R-CNN: ResNet101[5,10]	**.9644**	.9514	.7863	.7621
R-FCN: ResNet101[5,12]	.9591	**.9550**	.7884	.7753

Table 2: Performance of each object detection model on firearms and sharps, treating all views as independent images. Best values for each class category and metric are bolded.

4.1.2 Sharps

In Figure 3, we show results of single-view evaluation for all 5 object detection models on a dataset of 20 mixed open and closed blades in baggage. The dataset contains 274 scans of fixed-blade knives, 210 scans of pocket knives, 74 scans of scissors, and 148 scans of tools with sharp edges. The ground truths for all of these sharps were used to train a single-class sharps detector and then tested on held-out test scans.

Sharps pose several challenges different from firearms. By visually comparing Figures 2 and 3, it can be seen that knives provide a much smaller profile for the algorithm to find. Furthermore, depending on the sharp's orientation and surroundings, some views may be uninformative if the x-ray beam hits a knife edge-on or traverses through the handle and down the blade; knives, with their smaller size and thin aspect ratio, are also easily obscured by common opaque objects, such as metal ribbing or bottles. As such, we expect performance on par with firearms to require additional effort, and both Figure 6b and Table 2 illustrate this. In particular, it appears that the SSD models do much worse than the other models. We hypothesize that this is due to the smaller number of knives training samples and higher variation in aspect ratios of bounding boxes relative to firearms being exacerbated by the way learned variables are arranged in the SSD architecture. More investigation is necessary to draw any concrete conclusions.

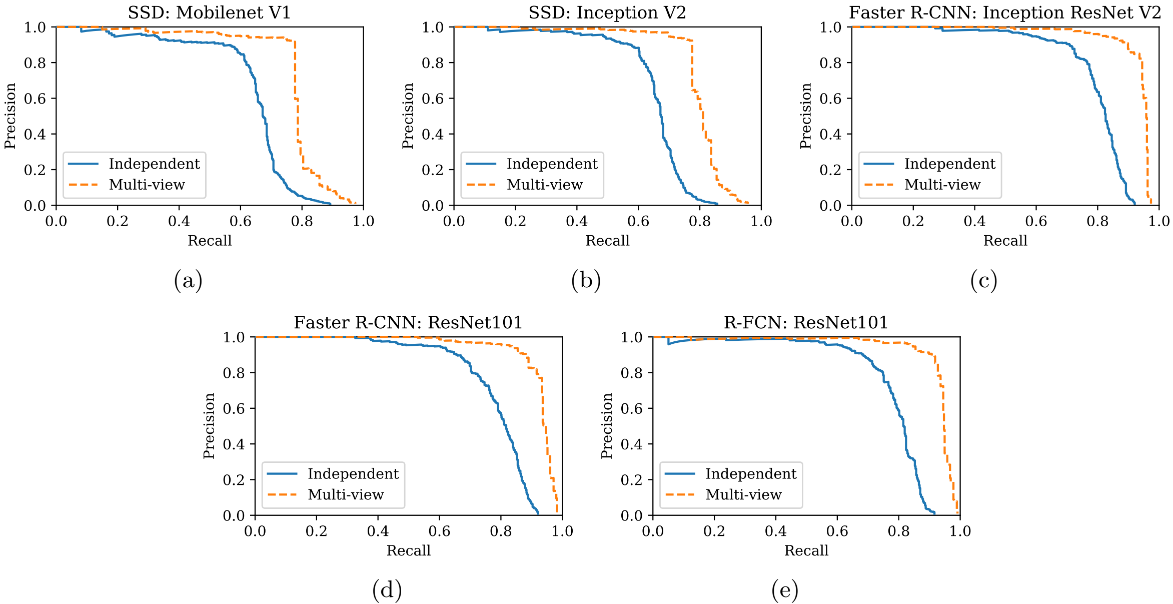
4.2 Object Detection Evaluation: Multi-View

At this time, ATR performance on firearms is much better than sharps. Part of the lower performance on sharps may be due to uninformative views of the threat being more likely for thinner objects. However, as noted in Section 2.1, the Smiths Detection host x-ray system provides four different angles of the same object, potentially offering clearer perspectives of an object occluded in one such view. By using the information combination scheme (OR-gate) outlined in Section 3.3, we see a substantial jump in performance (Figure 7, Table 3). A similar scheme can be employed for firearms, but we omit this here because of the already excellent performance with a single view.

Figure 7 shows the gain in precision-recall for multi-view evaluation applied to the same results which were presented with single-view evaluation in Section 4.1.2, while Table 3 shows the gain in mAP by considering multi-view evaluation. For the largest network, the Faster R-CNN with Inception ResNet V2, the mAP increased by 19.2% to .9347.

	mAP-Independent Views	mAP-Multi-View
SSD: MobileNet V1[7,11]	.6425	**.7852**
SSD: Inception V2[4,11]	.6541	**.8020**
Faster R-CNN: Inception ResNet V2[6,10]	.7836	**.9347**
Faster R-CNN: ResNet101[5,10]	.7837	**.9283**
R-FCN: ResNet101[5,12]	.7976	**.9383**

Table 3: Performance on knives, incorporating all views as described in Section 3.3. Better performance for each model bolded.



(a) (b) (c)

(d) (e)

Figure 7: Performance gain of the precision-recall curve (orange dotted line) for each model by incorporating all four views for the sharps dataset. Each of the blue lines correspond to one of the colored lines in Figure 6b, which were computed using single-view evaluation. Curves further up and to the right indicate better performance.

4.3 Detection vs False Alarm Evaluation

As proof-of-concept and to demonstrate feasibility, a trained Faster R-CNN with ResNet101 was evaluated on a combined test set comprised of 260 pocket knife scans, 1300 firearm scans, and 15000 real SOC scans, which are assumed to not contain firearms or sharps. For this experiment, we considered detection and false alarm rates at the bag level. An alarm satisfying the T_p criteria in Section 3.2 in any of the four views resulted in the entire bag being labeled as a detection. False alarm rate was calculated as the proportion of bag scans where any of the four views had a F_p bounding box produced by the object detection algorithm; if there are multiple false alarm bounding boxes produced, the scan still only counts as a single false alarm. In some sense, this approach is more amenable to the security checkpoint application, as a single alarm in any views should be cause for a TSO to investigate further. For the data collected in this study, the ATR captures 91.9% of views containing firearms at a 1% false alarm rate and 89.8% of sharps at a 3% false alarm rate with single-view evaluation. However, by utilizing all four views, the ATR algorithm is able to capture 95.5% of bags containing firearms at a 1% false alarm rate and 94.0% of bags containing sharps at a 3% false alarm rate.

5. CONCLUDING REMARKS

Despite major advances in threat detection for aviation security, there are still challenges regarding both operator and algorithm efficiency detecting many prohibited items. Of special interest are weapons such as firearms and knives, which continue to be frequently encountered by TSOs. Results of applying deep learning techniques to this setting both in offline computer testing and live demonstrations indicate that this approach can detect prohibited items with high accuracy, minimal false alarm rates, and no adverse impact to passenger throughput, especially for firearms detection. A simple multi-view evaluation has also been demonstrated to improve performance, especially for items like sharps, which may be difficult to spot in just a single view. A more sophisticated algorithm that combines information between views may do even better, but we leave this to future work.

Ultimately, a goal of the TSA is to lessen the burden on the TSOs by instituting a computer-aided CONOPs where TSOs only have to review on alarm, or even transition to an entirely automated system. To achieve this, a wider range of prohibited items are being targeted in future work with a similar approach. These items, including blunt weapons, precursors, and flammable liquids, come in almost infinitely many variations in size, shape, texture, and materials, but given the promise shown by deep learning in this initial study, Smiths Detection and the TSA are well positioned to bring ATR systems able to detect such classes to airport checkpoints in the near future.

5.1 Acknowledgments

This work has been funded by the Transportation Security Administration (TSA) under Contract #HSTS04-16-C-CT7020.

REFERENCES

[1] "Air Traffic By The Numbers." https://www.faa.gov/air_traffic/by_the_numbers/.

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation* **1**(4), pp. 541–551, 1989.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[4] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, pp. 448–456, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *International Conference on Learning Representations Workshop*, 2016.

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[9] R. Girshick, "Fast R-CNN," in *Proceedings of the International Conference on Computer Vision*, 2015.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *European Conference on Computer Vision*, pp. 21–37, Springer, 2016.

[12] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in *Advances in Neural Information Processing Systems*, pp. 379–387, 2016.

[13] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision* **88**(2), pp. 303–338, 2010.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision* **115**(3), pp. 211–252, 2015.

[17] T. W. Rogers, N. Jaccard, E. J. Morton, and L. D. Griffin, "Automated X-ray Image Analysis for Cargo Security: Critical Review and Future Promise," *Journal of X-ray Science and Technology* **25**(1), pp. 33–56, 2017.

[18] S. Akcay and T. P. Breckon, "An Evaluation of Region Based Object Detection Strategies within X-ray Baggage Security Imagery," in *IEEE International Conference on Image Processing*, pp. 1337–1341, IEEE, 2017.

[19] S. Akcay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, “Using Deep Convolutional Neural Network Architectures for Object Classification and Detection within X-ray Baggage Security Imagery,” *IEEE Transactions on Information Forensics and Security*, 2018.

[20] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *Software available from tensorflow.org*, 2015.

[21] A. Neubeck and L. Van Gool, “Efficient Non-Maximum Suppression,” in *International Conference on Pattern Recognition*, **3**, pp. 850–855, IEEE, 2006.