Background Adaptive Faster R-CNN for semi-supervised convolutional object detection of threats in X-ray images

John B. Sigman, Gregory P. Spell, Kevin J Liang, and Lawrence Carin

Duke University, Durham, NC 27708, United States of America

ABSTRACT

Recently, progress has been made in the supervised training of Convolutional Object Detectors (*e.g.* Faster R-CNN) for threat recognition in carry-on luggage using X-ray images. This is part of the Transportation Security Administration's (TSA's) mission to ensure safety for air travelers in the United States. Collecting more data reliably improves performance for this class of deep algorithm, but requires time and money to produce training data with threats staged in realistic contexts. In contrast to these hand-collected data containing threats, data from the real-world, known as the Stream-of-Commerce (SOC), can be collected quickly with minimal cost; while technically unlabeled, in this work we make a practical assumption that these are without threat objects. Because of these data constraints, we will use both labeled and unlabeled sources of data for the automatic threat recognition problem. In this paper, we present a semi-supervised approach for this problem which we call Background Adaptive Faster R-CNN. This approach is a training method for two-stage object detectors which uses Domain Adaptation methods from the field of deep learning. The data sources described earlier are considered two "domains": one a hand-collected data domain of images with threats, and the other a real-world domain of images assumed without threats. Two domain discriminators, one for discriminating object proposals and one for image features, are adversarially trained to prevent encoding domain-specific information. Penalizing this encoding is important because otherwise the Convolutional Neural Network (CNN) can learn to distinguish images from the two sources based on superficial characteristics, and minimize a purely supervised loss function without improving its ability to recognize objects. For the hand-collected data, only object proposals and image features completely outside of areas corresponding to ground truth object bounding boxes (background) are used. The losses for these domain-adaptive discriminators are added to the Faster R-CNN losses of images from both domains. This technique enables threat recognition based on examples from the labeled data, and can reduce false alarm rates by matching the statistics of extracted features on the hand-collected backgrounds to that of the real world data. Performance improvements are demonstrated on two independently-collected datasets of labeled threats.

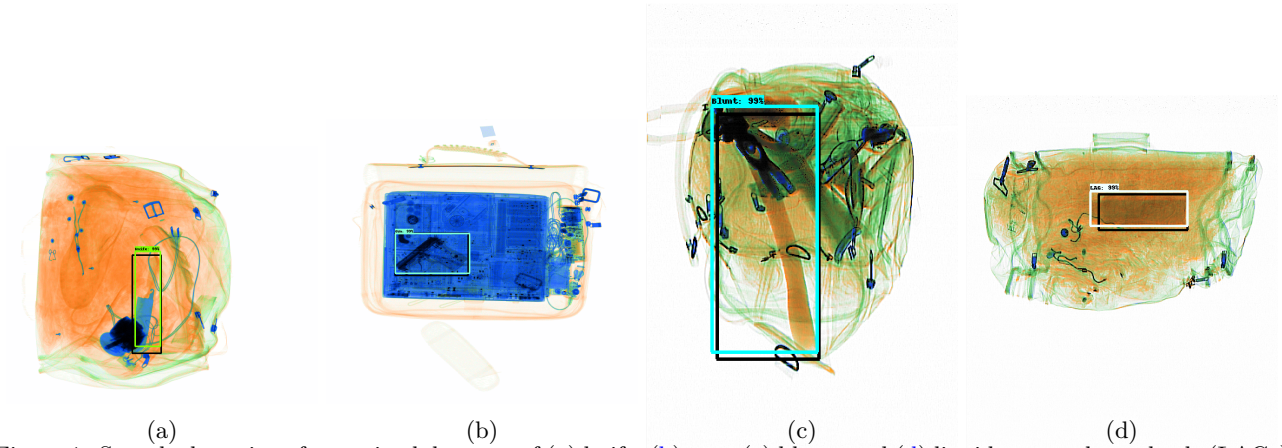
Keywords: Deep Learning, Threat Recognition, Security Screening, Convolutional Neural Networks, Machine Learning, Computer Vision, Object Detection

1. INTRODUCTION

Personal baggage security checkpoints consist of X-ray scanners and human operators, and their purpose is to prevent harm by capturing threatening objects and weapons. Modern X-ray scanner systems use sophisticated technology to construct internal views of bags or belongings, but these systems are still reliant on human operators to identify and locate threats. Bags can be highly cluttered environments, and owing to the transmissive nature of X-ray sensing, objects layer on top of each other in views. Some examples of threat objects in bags can be seen in Figure 1. Operators must be vigilant to catch all of a diverse, constantly evolving, but relatively rare set of prohibited items; all while maintaining high passenger throughput. In order to reduce the cognitive load on these human threat screeners, we seek automated solutions for threat screening.

In the field of computer vision, identifying and localizing objects in a scene is called object detection. Recently, convolutional neural networks (CNNs)[1] have resulted in major improvements in performance, with many modern object detection models[2–6] utilizing CNNs as a key component. Given their excellent performance on benchmark

Further author information:
John B. Sigman, E-mail: john.sigman@gmail.com



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

Figure 1: Sample detections from mixed datasets of (a) knife, (b) gun, (c) blunt, and (d) liquids, aerosols, and gels (LAGs) threats. Images scanned with a laboratory prototype not in TSA configuration.

datasets such as Cityscapes[7] and MS COCO,[8] these models are now being considered for certain real world settings, including baggage screening at airport checkpoints. However, the baggage scan images are different from the kinds of images in common object detection benchmarks. In particular, while common benchmark datasets are constructed to have a high number of object instances per image, threat items at checkpoints occur comparably infrequently. For example, the United States Transportation Security Administration (TSA) caught 4,432 firearms in carry-on baggage at US airport checkpoints in 2019, but against a backdrop of a billion passengers.[9]

This paper concerns an application of convolutional object detection using unlabeled imagery to improve threat detection in X-ray images. Unlike the expensive staging, imaging, and labeling of threat data, real-world scans can be readily and cheaply collected from functioning airport checkpoints. These real-world data are referred to as the Stream-of-Commerce (SOC). Modern deep neural network techniques are extremely data hungry, showing log-linear improvements with dataset size, *i.e.* roughly linear performance increases with exponentially increasing numbers of examples.[10] A technique which could train with these vast quantities of unlabeled images to improve threat detection is therefore highly valuable. Using such images would not be intended to improve recall (otherwise known as probability of detection). Instead, by exposing the model during training to a broader variety of real-world backgrounds we could reduce false alarm rates. However, strictly supervised use of unlabeled imagery as negative examples (the SOC) contains risks. Distinguishing patterns in the data (images) between the SOC and the labeled dataset could be discovered by a sufficiently flexible neural network, resulting in brittle predictions due to the difference in object occurrences between these two sets. In our case, this means that, if used naïvely, the network can recognize domain differences between hand-labeled data and SOC data. Once the model has learned to distinguish the domains, it can encode that images from the SOC all contain no objects and suppress detections, undermining the goal of real-world threat detection. In order to overcome this capacity of a Convolutional Neural Network (CNN) to distinguish domains, we will use techniques from domain adaptation.[11,12]

In this paper, our contribution is to demonstrate the real-world use of a variant of Domain Adaptive Faster R-CNN[13] that accounts for target shift between the image domains. Our Background Adaptive Faster R-CNN makes a strong assumption about the prior probability of targets in the target domain, particularly that images from this domain have no foreground (threat) objects. First we will give a brief introduction to Unsupervised Domain Adaptation (UDA) and its use in object detection. Then, we will describe some related works in the field of automatic threat detection in X-ray images, focusing on recent work in modern deep convolutional approaches. Next, we will describe how our method is used to enable learning from the threatless real-world data (the SOC). In the results section, we will show some improvements in performance using our technique to a baseline Faster Regions with Convolutional Neural Networks (Faster R-CNN)[5] approach. These positive results demonstrate successful use of the SOC data and indicate great promise for future applications of automatic threat recognition to airport security checkpoints.

2. BACKGROUND

In this section, we will describe the concepts and theory from recent works of machine learning which are applied in Background Adaptive Faster R-CNN (BA Faster R-CNN). Specifically, we will introduce the concepts of adversarial training, target shift, and covariate shift from the UDA literature.

2.1 Unsupervised Domain Adaptation

In the most basic formulation of adversarial domain adaptation,[11,14] we have access to a labeled dataset $\mathcal{D}^S$ from a source domain and an unlabeled dataset $\mathcal{D}^T$ from a target domain, and we wish to train a classifier that performs well on test data from the target domain without using any labels from $\mathcal{D}^T$. We will consider two types of shifts between data domains, defining them now as they pertain to image classification.[12,15] First, target shift refers to unequal label prior probabilities between two distributions, $p(y) \neq p'(y)$. The other type is covariate shift, which refers to a difference between the conditional distributions of an image given its class category: $p(x|y) \neq p'(x|y)$. One way to improve performance on the target dataset is to train a domain-invariant classifier through matching the marginal distribution of features extracted from the data x of either dataset $\mathcal{D}^T$ or $\mathcal{D}^S$. The marginal distribution of features, and not a class-specific conditional feature distribution, is used because it only requires the data and not the labels. Domain Adaptive Neural Network (DANN)[11] introduced a method for learning domain-invariant features by using an adversarial loss via domain discriminator. If the marginal distribution of features h extracted from target data are referred to by $p^T(h)$, and the marginal distribution of features extracted from source data are $p^S(h)$, then the domain-invariant features are achieved when:

$$p^S(h) = p^T(h) \tag{1}$$

Because these features depend on learnable parameters in a CNN, stating that probability distributions are equal is not a static description of the data, but implies the matching of these distribution by learning a suitable feature extractor using adversarial training.[11,13,14] As shown previously,[12] if the prior probabilities of the target domain classes were known, $p^T(y)$, then an adapted feature extractor for the target domain can be learned by matching the distributions:

$$p^T(h) = q^S(h) = \sum_c p^S(h|y=c)p^T(y=c) \tag{2}$$

Where c is one of the classes in $\mathcal{D}^T$ or $\mathcal{D}^S$, and q^S is a model distribution which is the class-conditional weighted average of source features, weighted according to their incidence in the target domain. This is not done under normal conditions because *unsupervised* domain adaptation implies that we do not have $p^T(y=c)$.

2.2 Unsupervised Domain Adaptation in Object Detection

These concepts were extended to object detection in a self-driving car setting[13] by matching the distributions of CNN features of the overall images and the proposed objects, demonstrating that an object detection model trained on mostly clear weather day images can still perform well on night, foggy, or inclement weather settings. As in DANN, the method assumed that only marginal feature distributions needed to be matched to produce domain-invariant features. In this case, this was a safe assumption, because the target dataset of cloudy images was synthetically produced by transforming the source dataset of sunny images, guaranteeing equal likelihood of object occurrences in images from both sets. This simple treatment of the object occurrences in the two data sources will not work for threat recognition in X-ray images. An individual image from the SOC has a low prior probability of containing a threat while labeled data were all staged with threats.

3. RELATED WORKS

This paper describes an application of recent developments in computer vision to the problem of automated threat detection in luggage, but this problem has been studied for many years. Early work using machine learning for X-ray image classification relied on hand-crafted features such as Difference of Gaussians (DoG) and scale-invariant feature transform (SIFT).[16] These hand-crafted features were then fed to a traditional classifier such as a Support Vector Machine (SVM),[17] leveraging an approach known as Bag-of-Visual-Words.[18–22]

The first application of deep learning algorithms to images of X-ray baggage involved manually cropping regions of x-ray images and classifying the crops according to different categories of firearms and knives, as well as classes of camera and laptop.[23] This was accomplished via transfer learning, in which a pre-trained CNN was fine-tuned for the specific datasets of X-ray baggage. Deep object detection algorithms have also been investigated for use in X-ray baggage scans.[24–27] In these investigations, a range of CNN feature extractors have been examined, including VGG,[28] Inception V2,[29] and ResNet.[30] Furthermore, a variety of CNN-based detection algorithms are adapted for X-ray baggage: Faster R-CNN,[5] R-FCN,[31] and YOLOv2.[32] We encourage readers to see a recent survey[33] for a more thorough overview of these efforts. A subset of these works[26,27] specifically sought to incorporate deep learning techniques into the security systems used by the Transportation Security Administration (TSA) at U.S. airport checkpoints. In many settings, acquiring unlabeled data is significantly easier than labeled data. For X-Ray baggage scanner threat detection settings, this is especially the case, as acquiring labeled data often also requires assembling the threat-containing bags and scanning them. Thus, semi-supervised approaches have drawn considerable interest as an efficient way to leverage more data. Another approach is threat image projection (TIP), which digitally adds threat objects into a bag using various synthetic methods,[34] this is possible here due to the transmissive nature of X-ray, .

4. METHODS

4.1 Data

Consider two data-generating processes $p^{\mathrm{SOC}}(x, y)$ and $p^{\mathrm{HC}}(x, y)$, which model the appearance of objects, $y = \{(c, b)\}$, in an image x. In this notation, the set y contains objects described by c, a categorical random variable indicating class, and $b \in \mathbb{R}^4$ describing the coordinates of the object's bounding box. Let $\mathcal{D}^{\mathrm{SOC}}$ be the set of data samples from $p^{\mathrm{SOC}}(x, y)$, and let $\mathcal{D}^{\mathrm{HC}}$ be the set of data samples from $p^{\mathrm{HC}}(x, y)$. Data from $p^{\mathrm{SOC}}(x, y)$ are our Stream-of-Commerce (SOC) data, gathered from checkpoints, and are the target domain. Data from $p^{\mathrm{HC}}(x, y)$ are our "Hand-Collected" source domain data, which were collected by subcontractors staging threats in realistic context during TSA Contract HSTS04-16-C-CT7020.[26,27]

By design, p^{SOC} and p^{HC} have a target shift between them. Namely, p^{SOC} contains little to no threat objects, $p^{SOC}(|y| > 0) = 0$, and every image in p^{HC} contains a threat object, $p^{HC}(|y| > 0) = 1$. We can make the following statements of target shift (Equation 3) and covariate shift (Equation 4):

$$p^{\mathrm{SOC}}(y) \neq p^{\mathrm{HC}}(y) \tag{3}$$

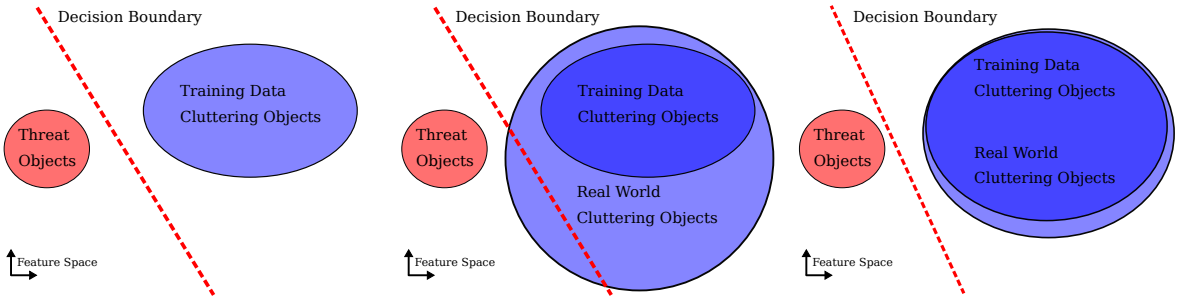
$$p^{\mathrm{SOC}}(x|y) \neq p^{\mathrm{HC}}(x|y) \tag{4}$$

We seek a method for learning an object detection model which addresses these disparities to more effectively find threats in the target domain.

4.2 Supervised Auxiliary Negative Training

A naïve way to learn from the objectless images is to treat them identically as the labeled examples in the training set. This implicitly treats all regions of the SOC images as background,[35] and tends to not work in practice. First, if the number of labeled images is small, adding too many unlabeled images dilutes the dataset, resulting in a large class imbalance that may drown out the learning signal for positive samples. Down-weighting negative samples emphasizes the learning signal for the actual objects of interest, but also reduces the value of the unlabeled set. Hard negative mining, or the focal loss[36] focus learning on the hardest background examples, but makes the assumption that the unlabeled data is from the same distribution as labeled dataset.

In this specific case, $\mathcal{D}^{\mathrm{SOC}}$ and $\mathcal{D}^{\mathrm{HC}}$ have some inherent differences which we cast as covariate shift. We hypothesize that these differences could include but not be limited to: idiosyncratic reconstruction noise on the periphery of images, signatures from the individual laboratory prototypes used to stage threat data, or effects from the close queuing of bags in scans which occurs in the real world and not in staged data collections. While these might not seem like meaningful differences, a sufficiently expressive feature extractor can pick up on these discrepancies and rapidly learn to suppress any detections in SOC images. While this reduces false positives, it could reduce the model's effectiveness when detecting threats in the real world, because it was only ever shown real-world data without threats.



(a) (b) (c)

Figure 2: Conceptual diagram included to demonstrate the intuition of this approach for anomaly/threat detection. (a) Separation of threats and background when training with labeled data. (b) Application of the supervised learned threat detector when applied to heterogenous real-world data. The set of "Real World Cluttering Objects" crosses over the decision boundary to imply increased false alarm rates in the real world. (c) Using adversarial training for the feature extractor to match the statistics of the backgrounds with the real-world data decreases false alarms while still learning robust features.

We have conducted this experiment and have observed empirically that the Faster R-CNN model is able to shatter metrics on the train *and test* SOC images. That is, while having some level of expected generalization error when evaluated on the test set of hand-labeled data, the model can predict without any false alarms across the test set of SOC data. These domain-dependent differences between the hand-collected data and the SOC are not a problem when the object detection model is trained with only labeled data, possibly only slightly increasing generalization error. However, they could be catastrophic if the model is allowed to learn to discriminate domains, which occurs when using supervised learning on negative examples from the SOC.

4.3 Adversarial Training and Background Matching

Our proposed method, Background Adaptive Faster R-CNN (BA Faster R-CNN), is a training procedure applied to the Faster R-CNN model,[5] which selectively applies the unsupervised domain adaptive Faster R-CNN[13] losses. Its intended use is in an anomaly/threat detection situation, with two separate datasets, such as $\mathcal{D}^{\text{HC}}$ and $\mathcal{D}^{\text{SOC}}$. Images from both sets train the Faster R-CNN loss for threat detection, and images from both are used simultaneously in adversarial domain adaptation. The method exposes the model to a wider variety of backgrounds than those in $\mathcal{D}^{\text{HC}}$ and through adversarial distribution matching, the CNN has barriers preventing it from encoding which dataset each object proposal comes from based on image signatures unrelated to the objects in a bag. A conceptual diagram is shown in Figure 2.

We will borrow previous notation[13] to describe probability distributions of features which are part of the Faster R-CNN algorithm. $p(B, I)$ is a marginal probability distribution of features extracted from feature map pixels I within bounding box B. $p(I)$ is a marginal probability distribution of the extracted feature map pixel, when each pixel is treated as an independent observation.

Extending previous work,[13] we address a particular case of target shift between domains in addition to covariate shift. As described in Equation 2, if we know the prior probability of labels in the target domain, we can perform adversarial domain adaptation by matching target features with class-conditional source features, weighted according to their incidence in the target domain. This requires matching the model distributions $q(B, I)$ and $q(I)$ to their respective distribution of features in the target $p^{SOC}(B, I)$ and $p^{SOC}(I)$:

$$p^{SOC}(B, I) = q(B, I) = \sum_c p^{HC}(B, I|c)p^{SOC}(c) \tag{5}$$

$$p^{SOC}(I) = q(I) = \sum_c p^{HC}(I|c)p^{SOC}(c) \tag{6}$$

The class-conditional distributions of instance features $p(B, I|c)$ and image features $p(I|c)$ refer to features corresponding to the locations of ground truth classes c, which can be one of many threat classes or background.

Under the assumption of no threats in the SOC, the class prior probability is one for background and zero for any foreground class.

$$p^{SOC}(c) = \begin{cases} 1 & c = \text{background} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Substituting Equation 7 into Equations 5 and 6 gives:

$$p^{SOC}(B, I) = p^{HC}(B, I | c = \text{background}) \tag{8}$$

$$p^{SOC}(I) = p^{HC}(I | c = \text{background}) \tag{9}$$

Despite a target shift between the two domains, we can still build from previous techniques[13] so long as we only match the background portions of the hand-collected data to the SOC. We achieve background instance matching (Equation 8) by first obtaining candidate object proposals in the standard way via Faster R-CNN. For object proposals from $\mathcal{D}^{\text{HC}}$, we designate as background any proposal for which the Intersection Over Union (IOU) with a ground truth threat object is below a certain threshold (in our experiments, 0.01). These background proposals are fed to an adversarial domain discriminator for object instances, along with all proposals made from $\mathcal{D}^{\text{SOC}}$.

For background image matching (Equation 9), we first anti-crop the features of images from $\mathcal{D}^{\text{HC}}$ according to the ground truth boxes of threats. By "anti-cropping", we mean masking out feature map pixels inside ground truth bounding boxes. Only the labeled image feature map pixels corresponding to background are used to train the adversarial pixel domain discriminator, alongside all of the feature map pixels from $\mathcal{D}^{\text{SOC}}$. A third domain adaptation loss, a consistency regularization,[13] enforces that the two domain classifiers predict the same domain. This loss is implemented as the ℓ_2 distance between the domain classifier outputs for an image.

5. RESULTS

In this section we give results for the application of BA Faster R-CNN to images of threats in baggage. These results are not meant to be compared across datasets, only within them. While each scan of a single bag produces multiple views (each an image), we calculate precision/recall scores and mean Average Precision (mAP) on a single-image basis, deliberately excluding the sizable impact that multiple views of a threat per bag has on detection rates for simplicity. This impact is outside the scope of our paper; our purpose is to indicate results of applying BA Faster R-CNN to the problem of threat detection in carry-on luggage. While we show here some improvements in precision and recall of using BA Faster R-CNN on the labeled data collected in TSA Contract HSTS04-16-C-CT7020, its real purpose is to improve generalization performance when tested on held-out images from the SOC. Due to national security concerns and data sensitivity, we omit these results.

5.1 Datasets

Results in Section 5 were trained with datasets of over 6,000 labeled images for Dataset A and over 19,000 labeled images for Dataset B. A training set of over 35,000 SOC images were used for Dataset A and over 70,000 SOC images were used for Dataset B. Performance was evaluated with over 1,100 labeled images from each set, and 500 SOC images from each set were used as a representative sample to estimate false alarm rates.

5.2 Labeled Data Performance

For our performance metrics, we will show improvements in the precision/recall metric on a held out labeled set of images from two different vendors, Dataset A and Dataset B. For our experiments, we used the Tensorflow Object Detection implementation,[37] which we extended for X-ray images and object detection domain-adaptive components. All experiments were conducted with ResNet101[30] as the feature extractor and Faster R-CNN[5] as the meta-architecture. For all experiments, we use domain adaptive loss weighting[13] $\lambda = 0.1$ and Gradient Reversal Layer (GRL) weight[11,13] 0.1.

We ran three different experiments on each dataset. First, we trained a supervised Faster R-CNN baseline. Then, we included only the instance loss component, which matches feature distributions of extracted proposals

as in Equation 8. Finally, we trained a model using instance losses and image losses. Image losses are used to match the extracted feature pixel distributions, as in Equation 8. When both image and instance losses are used, we also include the consistency loss.[13]

5.2.1 Dataset A

Dataset A saw performance increases for nearly all labeled data classes when evaluated on the held-out set. These precision/recall metrics are graphed in Figure 3, and summarized in Table 1. Average Precision (AP) on Knives data was only improved by using all three loss terms, while AP was the best for the LAGs class when used with supervised learning.

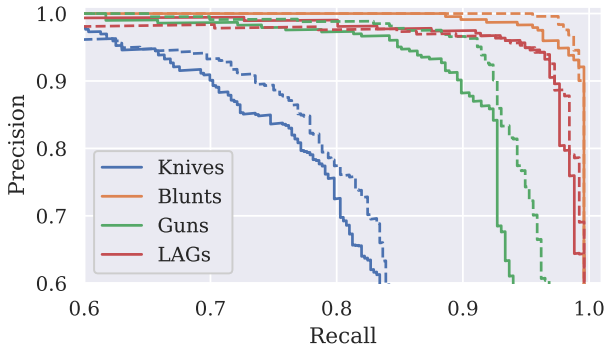


Figure 3: Precision/Recall for Dataset A, treating all views as independent images. The solid lines correspond to the purely Supervised model, and the dashed lines correspond to Semi-Supervised use of BA Faster R-CNN when all three domain-adaptive loss terms are applied. These results are summarized in Table 1

	Loss term			Threat Class			
Model Version	Instance	Image	Consistency	Knives	Blunts	Guns	LAGs
Faster R-CNN, Figure 3: (—)				0.832	0.989	0.931	**0.980**
+Match instances	✓			0.829	0.989	0.948	0.978
+Match instances and images, Figure 3: (– –)	✓	✓	✓	**0.839**	**0.992**	**0.955**	0.978

Table 1: Tabulated APs on labeled data from Dataset A.

5.2.2 Dataset B

Figure 4 shows precision/recall curves for the labeled held-out set from Dataset B. There was an increase in the mAP metric for all labeled data classes using background domain adaptation. Guns and LAGs did slightly worse using all three domain-adaptive losses than when only applied to instance losses.

	Loss term			Threat Class			
Model Version	Instance	Image	Consistency	Knives	Blunts	Guns	LAGs
Faster R-CNN, Figure 4: (—)				0.894	0.977	0.985	0.936
+Match instances	✓			0.908	0.980	**0.987**	**0.956**
+Match instances and images, Figure 4: (– –)	✓	✓	✓	**0.914**	**0.982**	0.986	0.951

Table 2: Tabulated APs on labeled data from Dataset B.

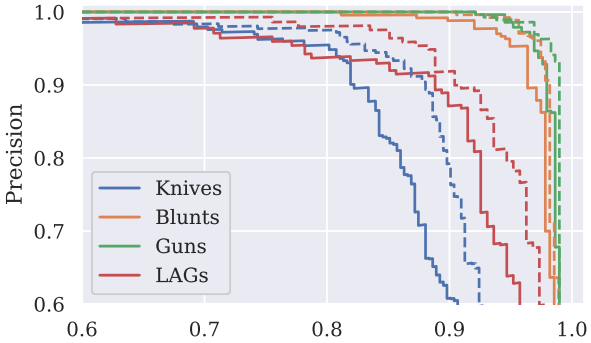


Figure 4: Precision/Recall for Dataset B, treating all views as independent images. The solid lines correspond to the purely Supervised model, and the dashed lines correspond to Semi-Supervised use of BA Faster R-CNN when all three domain-adaptive loss terms are applied. These results are summarized in Table 2

6. CONCLUDING REMARKS

This paper describes Background Adaptive Faster R-CNN, a technique for using real-world unlabeled data to improve threat detection in X-ray images. As in a typical Unsupervised Domain Adaptation (UDA) setup, two datasets are used. The first of these is a source dataset, which is small and contains labeled examples of objects. The second dataset is from our target domain, and contains many more images without labels. By matching the class conditional feature distributions of the background in the labeled data to the marginal distribution in the unlabeled data, we can use UDA techniques despite target shift between the two datasets.

This technique was required in order to use large quantities of real-world data because of the well-known tendency of deep convolutional feature extractors to learn brittle features if these are enough to separate the training data. In our case, this meant that with supervised use of large quantities of real-world SOC data, then the feature extractor need only recognize that the image was from the SOC to achieve low losses. Instead, we penalize the encoding of domain-specific information via two adversarial domain discriminators, which require the model learn from cluttering objects in the SOC.

We demonstrate BA Faster R-CNN on two independently-collected datasets. In this application, the target threat classes are guns, knives, blunt objects, and liquid, aerosols, or gels. For the X-ray baggage datasets that we examine, we find that BA Faster R-CNN succeeds in improving precision/recall performance for most threat classes in the labeled portion of the dataset. It is possible for BA Faster R-CNN to reduce false alarm rates in the real world, but we omit such results here.

Acknowledgments

This work was funded by the Transportation Security Administration (TSA) under Contract #HSTS04-16-C-CT7020 We thank our sponsors in the Department of Homeland Security as well as our collaborators in the X-ray sensing industry.

REFERENCES

1. Y. Lecun, B. Boser, J. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation* , 1989.
2. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Conference on Computer Vision and Pattern Recognition* , 2014.
3. R. Girshick, "Fast R-CNN," *International Conference on Computer Vision* , 2015.
4. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision* , 2016.
5. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems* , 2015.

6. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Conference on Computer Vision and Pattern Recognition* , 2016.
7. M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," *Conference on Computer Vision and Pattern Recognition* , 2016.
8. T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dolí, "Microsoft COCO: Common Objects in Context," *European Conference on Computer Vision* , 2014.
9. "TSA Year in Review: 2019," *https://www.tsa.gov/blog/2020/01/15/tsa-year-review-2019* , 2020.
10. C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," *International Conference on Computer Vision* , 2017.
11. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-Adversarial Training of Neural Networks," *Journal of Machine Learning Research* , 2015.
12. Y. Li, S. Dai, L. Carin, and D. Carlson, "On Target Shift in Adversarial Domain Adaptation," *AISTATS* , 2019.
13. Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool, "Domain Adaptive Faster R-CNN for Object Detection in the Wild," *Conference on Computer Vision and Pattern Recognition* , 2018.
14. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *Advances In Neural Information Processing Systems* , 2014.
15. I. Redko, N. Courty, R. Flamary, and D. Tuia, "Optimal transport for multi-source domain adaptation under target shift," *arXiv preprint arXiv:1803.04899* , 2018.
16. D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *In Proceedings of the International Conference on Computer Vision (ICCV)*, **99**(2), pp. 1150–1157, 1999.
17. C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning* **20**(3), pp. 273–297, 1995.
18. M. Baştan, M. R. Yousefi, and T. M. Breuel, "Visual Words on Baggage X-Ray Images," *Computer Analysis of Images and Patterns.* , pp. 360–368, 2011.
19. M. Bastan, W. Byeon, and T. Breuel, "Object Recognition in Multi-View Dual Energy X-ray Images," in *In Proceedings of the British Machine Vision Conference (BMVC)*, January 2013.
20. D. Turcsany, A. Mouton, and T. P. Breckon, "Improving Feature-Based Object Recognition for X-Ray Baggage Security Screening Using Primed Visual Words," in *In Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, February 2013.
21. D. Mery, E. Svec, and M. Arias, "Object Recognition in Baggage Inspection Using Adaptive Sparse Representations of X-ray Images," *Image and Video Technology* , pp. 709–720, 2016.
22. M. E. Kundegorski, S. Akçay, M. Devereux, A. Mouton, and T. P. Breckon, "On Using Feature Descriptors as Visual Words for Object Detection within X-ray Baggage Security Screening," in *In Proceedings of the International Conference on Imaging for Crime Detection and Prevention (ICDP)*, November 2016.
23. S. Akçay, M. E. Kundegorski, M. Devereux, and T. P. Breckon, "Transfer Learning Using Convolutional Neural Networks for Object Classification within X-ray Baggage Security Imagery," in *In Proceedings of the IEEE International Conference on Image Processing (ICIP)*, September 2016.
24. S. Akcay and T. P. Breckon, "An Evaluation of Region Based Object Detection Strategies within X-ray Baggage Security Imagery," in *In Proceedings of the IEEE International Conference on Image Processing (ICIP)*, September 2017.
25. 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 Trans. Info. Forens. Sec.* **13**(9), pp. 2203–2215, 2018. doi: 10.1109/TIFS.2018.2812196.
26. K. J. Liang, G. Heilmann, C. Gregory, S. Diallo, D. Carlson, G. Spell, J. Sigman, K. Roe, and L. Carin, "Automatic Threat Recognition of Prohibited Items at Aviation Checkpoints with X-Ray Imaging: a Deep Learning Approach," *Proc SPIE, Anomaly Detection and Imaging with X-Rays (ADIX) III* , 2018.
27. K. J. Liang, J. B. Sigman, G. P. Spell, D. Strellis, W. Chang, F. Liu, T. Mehta, and L. Carin, "Toward Automatic Threat Recognition for Airport X-ray Baggage Screening with Deep Convolutional Object Detection," *arXiv preprint arXiv:1912.06329* , 2019.

28. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *In Proceedings of the International Conference on Learning Representations (ICLR)*, May 2015.
29. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *In Proccedings of the International Conference on Machine Learning (ICML)*, 2015.
30. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
31. J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *Advances In Neural Information Processing Systems* , 2016.
32. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *Conference on Computer Vision and Pattern Recognition* , 2017.
33. S. Akcay and T. Breckon, "Towards Automatic Threat Detection: A Survey of Advances of Deep Learning within X-ray Security Imaging," *arXiv preprint arXiv:2001.01293* , 2020.
34. N. Bhowmik, Q. Wang, Y. F. A. Gaus, M. Szarek, and T. P. Breckon, "The Good, the Bad and the Ugly: Evaluating Convolutional Neural Networks for Prohibited Item Detection Using Real and Synthetically Composited X-ray Imagery," *arXiv preprint arXiv:1909.11508* , 2019.
35. Y. Yang, K. J. Liang, and L. Carin, "Object detection as a positive-unlabeled problem," *arXiv preprint arXiv:2002.04672* , 2020.
36. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *International Conference on Computer Vision* , 2017.
37. J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *Conference on Computer Vision and Pattern Recognition* , 2017.