Anomaly Detection for Security Imaging

J T. A. Andrews[a,b,c,*], N. Jaccard[a], T. W. Rogers[a,c], T. Tanay[a], L. D. Griffin[a]

[a] *Department of Computer Science, University College London, UK*

[b] *Department of Statistical Science, University College London, UK*

[c] *Security Science Doctoral Research Training Centre, University College London, UK*

Abstract

Existing approaches to automated security image analysis focus on the detection of particular classes of threat. However, this mode of inspection is ineffectual when dealing with mature classes of threat, for which adversaries have refined effective concealment techniques. Furthermore, these methods may be unable to detect potential threats that have never been seen before. Therefore, in this paper, we investigate an anomaly detection framework, at X-ray image patch-level, based on: (i) image representations, and (ii) the detection of anomalies relative to those representations. We present encouraging preliminary results, using representations learnt using convolutional neural networks, as well as several contributions to a general-purpose anomaly detection algorithm based on decision-tree learning.

Keywords: Anomaly detection; representation-learning; machine learning; cargo screening; X-ray imaging; security imaging

1. Introduction

Non-intrusive inspection systems are increasingly used to scan intermodal freight shipping containers, at national ports, to ensure cargo conformity with customs regulations. Initially, each container is risk assessed based on shipping information such as origin, destination, and manifest [9, 29]. If the risk is deemed sufficiently high the container is imaged, typically by non-intrusive X-ray radiography. Finally, on the basis of the X-ray image, a human operator must make a decision, as to whether a container then necessitates physical inspection.

[*] Corresponding author
Email address: jerone.andrews@cs.ucl.ac.uk (J T. A. Andrews)
URL: http://compass.cs.ucl.ac.uk (J T. A. Andrews)

These processes aim to minimise the number of false searches, whilst maximising the number of true searches, thus facilitating the detection of suspicious cargoes with negligible interference to the flow of commerce. However, due to the large number of containers being transported yearly, the number of X-ray transmission images to be visually inspected remains high. Moreover, the heterogeneity within and between the X-ray images provides an appreciable visual challenge to human operators, exacerbated by overlapping, translucent cargo [4, 6, 31].

Existing approaches to automated security image analysis focus on the supervised[1] detection of particular classes of threat [11, 12, 14, 15, 32]. However, this mode of inspection is ineffectual when dealing with mature classes of threat, for which adversaries have refined effective concealment techniques [8]. Furthermore, these methods may be unable to detect potential threats that have never been seen before in Stream-of-Commerce (SoC) data. To detect these concealed or unseen types of threat, customs officers often observe anomalies of shape, texture, weight, feel or response to perturbation. Inspired by the practice of customs officers, we are developing algorithms to discover visual anomalies, in X-ray images, that are historically atypical with respect to expected patterns.

In this work, we investigate an anomaly detection (AD) framework[2], at X-ray image patch-level, for the automated discovery of anomalous regions. The framework consists of two main components: (i) image representations, and (ii) the detection of anomalies relative to those representations.

The development of discriminative representations is problematic, since we have no a priori knowledge of the underlying, generating distribution of anomalies. Therefore, we pursue representations that have been optimised for a related, very general, task, on similar[3] data, which have been found to be useful in previous works [2, 7, 20, 26, 30]. The representations, for each patch, are then scored using a forest of random-split trees (FRST), a general-purpose machine learning AD algorithm. Moreover, we propose to randomly rotate the representation space prior to constructing each tree in the FRST, as well as an alteration to the way in which samples are scored, which empirically improves performance.

The remainder of this paper is structured as follows. We start by reviewing related work on supervised and unsupervised approaches to cargo security anal-

[1] In supervised machine learning labelled training data is required, with each training sample consisting of an input object and a desired output label. The learning algorithm then infers a function from the labelled data, which can then be used to map novel input samples to an output label.

[2] Note, our AD system is strictly constructed on *normal* data only. In this manner, as opposed to supervised learning, the task of the learner is to deduce a function that describes the latent structure of the *normal* data.

[3] Similar in the sense that they share the same input space, i.e. pixel space.

ysis, in Section 2. In Section 3, we move on to describing the dataset used in this work. In Section 4, details of our anomaly detection framework are given. Preliminary results are presented in Section 5.

2. Related work

At present, there is a scarcity of work directly related to automated AD in cargo security imagery [23]. Most have focused wholly on supervised learning for the purpose of automated contents verification or automated threat detection.

2.1. Automated contents verification

Automated contents verification makes an assessment as to whether the contents of a cargo container are in agreement with its paired shipping manifest. Research in this area typically spans empty cargo verification [21, 22] and manifest verification [28, 31].

Empty cargo verification is by and large addressed as a supervised binary-classification problem. Orphan et al. [21] approach this task by first segmenting an image into different regions and then employing an unnamed rule-based algorithm for object detection to categorise images as either *empty* or *non-empty*. More recent scholarship, by Rogers et al. [22], utilise fixed geometric representations [19], image moments, and spatial coordinates of sampled windows for classification.

Manifest verification expands empty cargo verification to a multi-class classification setting, with the aim of determining to which of a set of Harmonised Commodity Description and Coding System (HS) codes[4] a new cargo container belongs. Tuszynski et al. [28] construct HS code models based on simple intensity histograms of the images, which are then used to compute the Manhattan distance between a new observation and each model. A significant limitation of their system lies in its inability to locate small incongruous objects within otherwise properly declared cargo. Zhang et al. [31] address the lack of texture-based representations, used by Tuszynski et al. [28], by employing a Leung-Malik filter bank to build a visual bag-of-words codebook to classify cargo images into 1 of 22 categories.

2.2. Automated threat detection

With reference to automated threat detection, Jaccard et al. [12, 13] undertake the detection of small metallic threats (SMTs) by utilising automatically learnt

[4] Each HS code specifies a broad class of cargo type.

convolutional neural network (CNN) representations, trained on an augmented dataset, with genuine threat items projected into SoC images. In other work, Jaccard et al. [11, 14, 15] present methods for classifying image sub-windows as either *car*[5] or *non-car*, comparing intensity histograms, fixed geometric representations, pyramid histogram of visual words (PHOW) representations, and CNN representations.

2.3. Anomaly detection

Some works have studied the problem of AD. For instance, Zheng & Elmaghraby [32] attempt to detect anomalous regions within images of vehicles by computing the window-wise correlation between a new test image of a vehicle to historical windows of the same vehicle. However, the anomalous regions flagged seldom correspond to actual threats. More recently, we have used AD techniques, based on representations derived from a sparse auto-encoder [1], as well as transfer learnt CNN representations [2], to perform empty container verification on down-sampled images of containers. Impressively, we achieve 99.2% [1] on this task, which is comparable to the supervised learning result reported by Rogers et al. [22].

3. Dataset

The dataset used consists of 1.2×10^5 SoC X-ray transmission images of intermodal freight shipping containers, obtained from a Rapiscan Eagle®R60 rail car scanner. The scanner images individual rail cars moving at speeds of up to $60\,\mathrm{kmh}^{-1}$. Image dimensions depend on the freight container size (typically 20 ft or 40 ft wide) and range from 1290×850 to 2570×850 pixels. Each image is stored as a 16-bit greyscale array with a resolution of $5.6\,\mathrm{mm}$ pixel^{-1} in the x-axis. Furthermore, all images vary due to small differences in freight containers and their fixtures, while cargo images also differ in the cargo.

3.1. Image preprocessing

In this work, we will assume that each SoC image has been preprocessed prior to any representation extraction or classification [14, 22]. This involves: (i) removal of black stripe artefacts caused by faulty detectors or source misfires; (ii) normalisation based on air attenuation values to correct for variations in source intensity and sensor responses; (iii) replacement of isolated erroneous pixels by the median intensity value of their local neighbourhoods; (iv) and, binary segmentation to extract the container region from the image. An example of preprocessing is shown in Fig. 3.1.

[5] Aronowitz et al. [3] observe that cars are commonly involved in export fraud and tax evasion schemes.

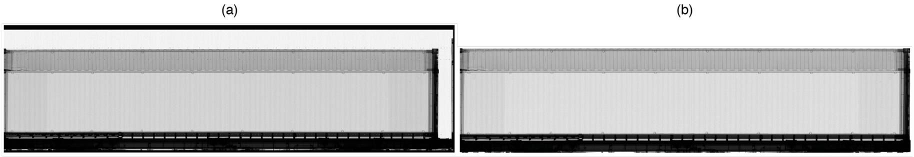

Figure 1: Example of preprocessing: (a) the raw SoC X-ray transmission image, and (b) the SoC X-ray transmission image having been preprocessed .

3.2. Generation of normal samples

We randomly extract 2×10^5 image patches of size 256×256 pixels from the SoC dataset. These patch-level images form the basis of our training set of *normal* samples. See Fig. 3.2 for an example of the types of patches extracted.

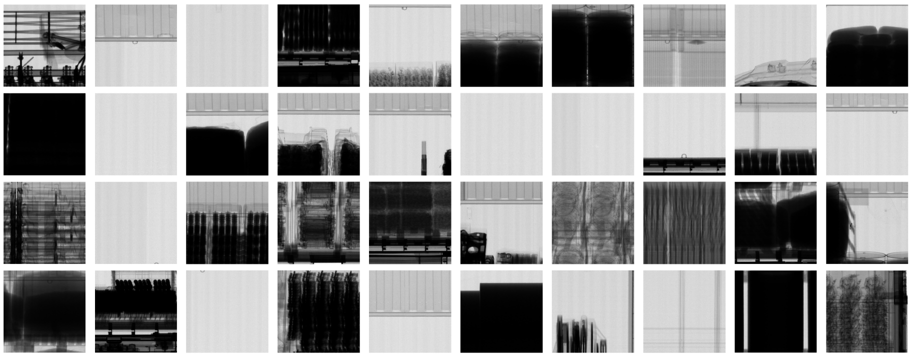

Figure 2: Example of 40 randomly sampled patches from the SoC dataset.

3.3. Generation of anomaly samples

For the purposes of evaluating our AD framework, small metallic threat patch images were synthesised using a threat image projection framework proposed by Rogers et al. [24]. A full description may be found in [13, 15, 22]. In precis, the method capitalises on the estimated multiplicative characteristics of X-ray transmission imagery, and can project 1 of 700 available X-rays of SMTs into a different image. What is more, the authors show that there is no apparent visual disparity between an actual threat and a synthetically generated threat, as illustrated in Fig. 3.3.

Thus, for assessment, SoC image patches of size 256×256 were randomly sampled and SMT instances were projected into half of them. Note, there is no overlap between the training set of *normal* samples and the testing set of *normal* and *anomaly* samples, with respect to the SoC image backgrounds and the projected SMTs.

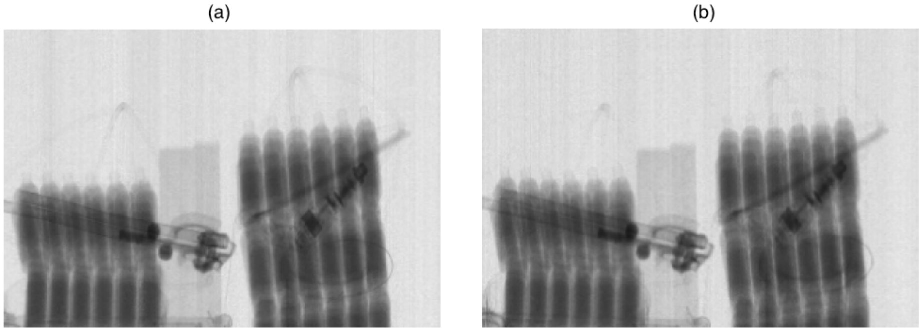

Figure 3: Example of threat image projection of an electric drill and a pipe bender: (a) corresponds to a real threat image, and (b) corresponds to a threat image projection.

4. Anomaly detection framework

Our framework is based on two integral components: (i) image representations, and (ii) the detection of anomalies relative to those representations. We detail these constituents below.

4.1. Image representations

Below we outline our approaches to representation extraction for the image patches, which are either intensity-based or CNN-based. These representations will be the basis on which our AD algorithm is built.

4.1.1. Intensity representations

As a baseline *vanilla* approach, each input image has its intensity values and its log-intensity values separately encoded as 256-bin histograms, which are then concatenated to give a 512-dimensional vector representation.

4.1.2. Convolutional neural network representations

Convolutional Neural Networks (CNNs) obviate the dependency on human representation-engineering, by offering a versatile method for automatically discovering multiple levels of representation, within raw data, that are extremely adept when utilised in tasks such as classification [16]. Remarkably, these representations are often generic enough that they can be transferred to other vision tasks and still achieve highly competitive results [7, 20, 26, 30]. Furthermore, we have found CNN representations to be useful for anomaly detection [2].

ImageNet. We utilise the publicly available pre-trained 19-layered CNN model, namely VGG-VD-19 [27], to transfer learn representations. The network was trained for the image classification task ILSVRC [25] using

1.2×10^6 colour images of 1×10^3 diverse object categories. High-level image representations are then extracted from the first fully-connected layer, $FC1$.

Cars. Representations were extracted from the first fully-connected layer, $FC1$, of a pre-trained 19-layered CNN model [14]. The network was trained to classify patches of the same SoC dataset, aforementioned in Section 3, as either *car* or *non-car*.

Auxiliary. We formulate an auxiliary task so as to transfer learn representations. We learn representations from scratch, by training a 19-layered CNN, based on the architecture reported in Jaccard et al. [13], to make distinctions within the *normal* class of SoC patches. The CNN was tasked with determining whether or not two patches were the same. For this training, the CNN was presented with two patches: (i) labelled +1 if the two patches were the same (90% overlap), and (ii) a label of −1 if the two patches were different (0% overlap). See Fig. 4 for an example of patch pairs. Representations, to be used for AD, were extracted from the fully-connected layer.

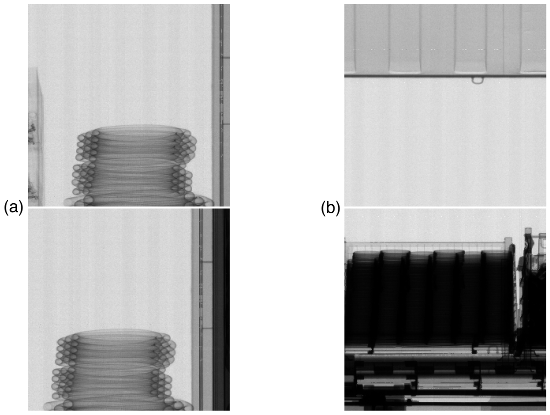

Figure 4: Example pairs used for CNN training on an auxiliary task: (a) two patches from the SoC dataset with 90% overlap given the label +1, and (b) two patches with 0% overlap given the label −1.

4.2. Forest of random-split trees anomaly detector

Our aim is to infer a function that describes the hidden structure of the *normal* data. The method used here constructs a set of random binary decision trees to form a forest of random-split trees (FRST), originally proposed by Liu et al. [17, 18]. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a set of N *normal* training samples, where $\mathbf{x} = (q_1, \ldots, q_D)$ and $q_i, i = 1, \ldots, D$ are its elementary continuous-valued attributes. Each binary tree randomly splits the data at each node, with atypical samples tend to be found in shallower leaves of the tree, whereas typical data in deeper leaves. Ergo, a FRST is constructed under the working assumption that anomalies are *rare* and *dissimilar*, since samples that are more readily separated from the main body, by randomly chosen criteria, give rise to higher *anomaly* scores.

This construction is different to standard supervised decision tree learning, where each tree is built by selecting the attribute that best splits a labelled set of training samples. The FRST uses no target labels, and chooses each attribute and split completely at random.

4.2.1. Forest of random-split trees training

Random-split tree construction. A node t in a tree T is either a leaf (terminal) node with no children, or an internal node with precisely two children (t_L, t_R), with each child connected to its parent by an edge e. For growing a tree, at each node t, a random attribute $q_{i(t)}$ and a random split value $p_t \in \left[\min(q_{i(t)}), \max(q_{i(t)})\right]$ are selected, for some attribute index $i(t)$. The boolean test $q_{i(t)} < p_t$, at node t, determines which training samples go to t_L or t_R. Top-down binary recursion continues until no further splits are possible, that is, until all samples have been partitioned into their own leaf node.

Our first *contribution*: in the case of randomly selecting an attribute $q_{i(t)}$, at node t, with more than one sample, if $\min(q_{i(t)}) = \max(q_{i(t)})$, we randomly send half of the samples to t_L and the other half to t_R. This is different to the original algorithm [17, 18], where if $\min(q_{i(t)}) = \max(q_{i(t)})$ then the samples cannot be divided and either t_L or t_R is empty and the other is sent all of the samples. We do this to ensure that each tree has $2N - 1$ nodes, where $N - 1$ nodes are internal and N nodes are leaves, and therefore control the memory requirement.

Forest of random-split tree construction. A FRST is an ensemble of M random-split trees, $F = \{T_1, \ldots, T_M\}$, with each tree T constructed on a random sub-sample $X' \subset X$ selected without replacement, such that $N' = |X'| < |X| = N$.

The free parameters (M, N') control the FRST. Unless stated otherwise, we adopt $M = 1000$ and $N' \approx N(1 - 1/\exp(1))$ in our experiments.

Randomly rotated FRST. Our second *contribution* is a randomly rotated version of the FRST algorithm:

For each tree, in a FRST, we first select a random sub-sample X'. Prior to any splitting, a proper rotation matrix R is generated using Householder QR decomposition [10], such that it is an orthogonal $D \times D$ matrix with $\det(R) = 1$. The representation space, for each tree, can then be rotated, using R, to give a unique coordinate system, which has been shown to increase diversity in supervised ensembles of trees [5]. The additional diversity is a by-product of the rotated tree, as opposed to its unrotated counterpart, having a very different orientation and vastly dissimilar data partitions.

Note, for computational efficiency, if $D > 1000$ then we rotate a random attribute subset of size $D' = 1000$, with each tree utilising a different random subset for each rotation.

4.2.2. Forest of random-split trees testing

The output of an anomaly detector is used to provide a method in which to measure the extent to which an unseen test sample is regarded as an *anomaly*. Thus, the output should be in the form of a ranked list, which can be attained by computing the depth of each test sample:

Depth. The depth $d_T(\mathbf{x})$ of a sample $\mathbf{x}$ is equal to the number of edges it has to traverse from the root node to end in a leaf node, in tree T. In the case of a FRST, we take the average depth of a sample $\mathbf{x}$ over all trees in the ensemble: $\frac{1}{M}\sum_{j=1}^{M} d_{T_j}(\mathbf{x})$.

Samples with lower depth values (easier to separate from the training data) are more likely to be anomalies, than samples with higher depth values (harder to separate from the training data).

Depth tweak. Our final *contribution* is an alteration to the way in which depths are computed:

For each tree T, if a test sample $\mathbf{x}$, at current node t (internal or leaf) of current depth d_t, has a value of $q_i > \max\left(q_{i(t)}\right)$, then generate a random real number $r \in \left[\min\left(q_{i(t)}\right), q_i\right]$. If $r > \max\left(q_{i(t)}\right)$, then stop splitting and assign the test sample a final depth of $d_T(\mathbf{x}) = d_t + 1$, otherwise continue descending the tree.

Conversely, if a test sample $\mathbf{x}$, at current node t (internal or leaf) of current depth d_t, has a value of $q_i < \min\left(q_{i(t)}\right)$, then generate a random real number $r \in \left[q_i, \max\left(q_{i(t)}\right)\right]$. If $r < \min\left(q_{i(t)}\right)$, then stop splitting and assign the test sample a final depth of $d_T(\mathbf{x}) = d_t + 1$, otherwise continue descending the tree.

The intuition behind this is as follows: if $q_i \notin \left[\min(q_{i(t)}), \max(q_{i(t)})\right]$, then the further away it lies from the interval the less likely it is to be similar to the training points that travelled through that node, and as such the depth tweak is more likely to terminate the traversal of the test sample.

4.2.3. Example: pinched annulus

To qualitatively demonstrate the merits of: (i) randomly rotating the representation space of each tree, and (ii) the use of the depth tweak, we generate a pinched annulus dataset of 1×10^3 samples. The results of the different combinations are shown in Fig. 5. First, it is readily seen that by rotating the representation space the axis-alignedness of the contour lines diminishes significantly; resulting in a much smoother decision boundary (green curve). Second, when the depth tweak is applied the decision boundary becomes tighter, as it penalises points with attribute values $q_i \notin \left[\min(q_{i(t)}), \max(q_{i(t)})\right]$ that lie far away. Finally, the combination of the two gives a decision boundary that is both smoother and tighter.

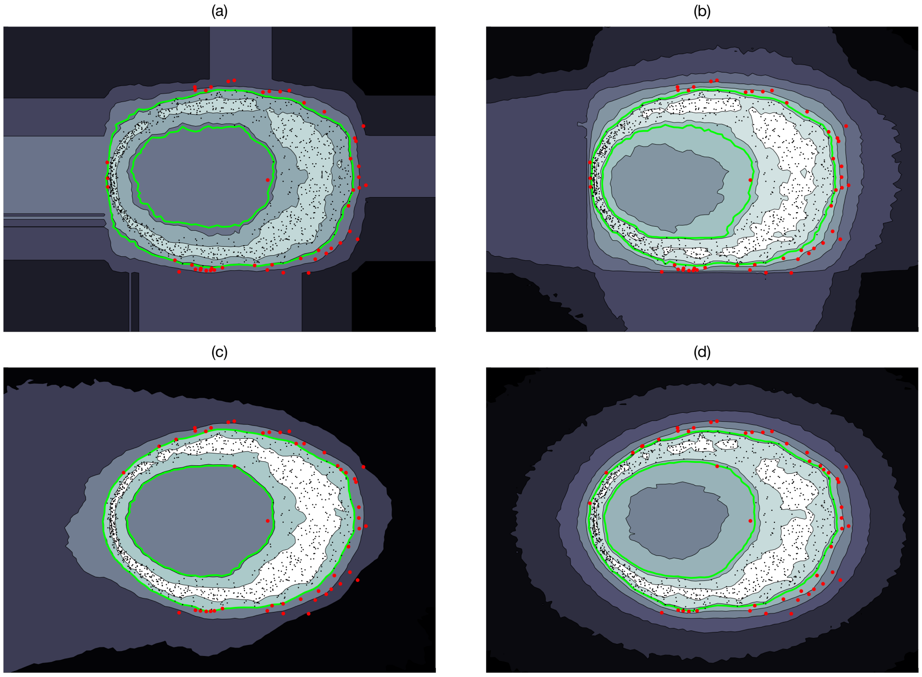

Figure 5: Contour plots of the different FRST algorithm variants trained on a 2-dimensional training set of 1000 samples: (a) FRST; (b) FRST with depth tweak; (c) randomly rotated FRST; and (d) randomly rotated FRST with depth tweak. Brighter regions correspond to higher average depth values, and darker regions to lower average depth values. Regions not contained within the green curves indicate regions below the 5^{th} percentile of the training set depths, with black dots representing training samples above the 5^{th} percentile and red dots representing training samples below the 5^{th} percentile.

4.3. Performance metric

For each FRST, trained on the various image representations (Section 4.1), we obtain the area under the receiver operating characteristic (AUC) attained on the fixed test set. The receiver operating characteristic is computed such that as the discrimination threshold is varied, the errors on the *normal* and *anomaly* data are calculated. In our case, the AUC $\in [0, 1]$ is equal to the probability that the detector will give a randomly selected *normal* sample a higher depth value than a randomly selected *anomaly* sample.

5. Results

Table 1 displays the preliminary AUC results for the different patch-level representations, using the FRST as an anomaly detector. We also compared the performance of the FRST with random rotations and the depth tweak (rr-dt), to the standard FRST without these options (none).

The AUC, using each representation, is improved when the feature values are randomly rotated and the depth tweak is applied. Whilst, these are only preliminary results, we believe that our contributions to the original FRST algorithm are indeed of benefit.

With respect to the representations, the intensity histograms offer an unreliable representation and perform just above chance. However, this is to be expected, since one cannot reasonably expect simple intensity features to be rich enough encodings for such a difficult and complex task. The auxiliary and car representations are the first and second best performers, respectively. Their performance is likely due to the CNNs being trained on SoC X-ray transmission images, whereas ImageNet is trained on natural images which stem from a very different distribution. Nonetheless, the ImageNet representation does not lag too far behind. Consequently, we are encouraged by the results given by the CNN representations, since they appear to provide far richer representations for the task of AD than the intensity histogram approach.

Table 1: AUC performance on the AD task using different representations. Note, under the heading **Method**: rr-dt denotes random rotation and depth tweak, using the FRST, whereas none denotes no random rotation or deth tweak.

Representation	**Method**	**AUC**
Intensity	none	0.5039
Intensity	rr-dt	0.5076
ImageNet	none	0.6205
ImageNet	rr-dt	0.6270
Cars	none	0.6261
Cars	rr-dt	0.6307
Auxiliary	none	0.6163
Auxiliary	rr-dt	**0.6365**

6. Conclusion

In this work, we presented an anomaly detection framework, at X-ray image patch-level, for the automated discovery of anomalous regions based on several different patch representations. Significantly, our improvements to the FRST algorithm have proven to be beneficial, particularly when used in combination with high-level CNN representations.

Future work will focus on how to improve these CNN representations: by evaluating different architectures and auxiliary tasks, so as to better define the concept of *normality* by learning how to improve discrimination between different types of *normal* patches. Moreover, the patch-level results will also then be fused into an overall anomaly heat map of the entire container, to facilitate human inspection. In addition, we will evaluate unsupervised methods for automated FRST free parameter (M, N) selection.

Acknowledgements

This work was supported by the Department for Transport, the Engineering and Physical Sciences Research Council (EPSRC) under CASE Award Grant 157760, and Rapiscan Systems.

References

[1] Andrews, J. T., Morton, E. J., & Griffin, L. D. (2016). Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing*, *6*, 21.

[2] Andrews, J. T. A., Tanay, T., Morton, E. J., & Griffin, L. D. (2016). Transfer representation-learning for anomaly detection. In *ICML 2016, Anomaly Detection Workshop*.

[3] Aronowitz, A. A., Laagland, D., & Paulides, G. (1996). *Value-added tax fraud in the European Union*. Kugler publications.

[4] Baştan, M., Yousefi, M. R., & Breuel, T. M. (2011). Visual words on baggage x-ray images. In *International Conference on Computer Analysis of Images and Patterns* (pp. 360–368).

[5] Blaser, R., & Fryzlewicz, P. (2016). Random rotation ensembles. *Journal of Machine Learning Research*, *17*, 1–26.

[6] Chen, G. (2005). Understanding X-ray cargo imaging. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, *241*, 810–815.

[7] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML* (pp. 647–655).

[8] European Commission (2002). Concealment methods. *Good Practice Guide for Sea Container Control*, .

[9] European Commission (2002). Container specifications. *Good Practice Guide for Sea Container Control*, .

[10] Householder, A. S. (1958). Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, *5*, 339–342.

[11] Jaccard, N., Rogers, T. W., & Griffin, L. D. (2014). Automated detection of cars in transmission x-ray images of freight containers. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on* (pp. 387–392). IEEE.

[12] Jaccard, N., Rogers, T. W., Morton, E. J., & Griffin, L. D. (2015). Using deep learning on X-ray images to detect threats. *In: Proceedings Cranfield Defence and Security Doctoral Symposium*, (pp. 1–12).

[13] Jaccard, N., Rogers, T. W., Morton, E. J., & Griffin, L. D. (2016). Automated detection of smuggled high-risk security threats using deep learning. *arXiv preprint arXiv:1609.02805*, .

[14] Jaccard, N., Rogers, T. W., Morton, E. J., & Griffin, L. D. (2016). Detection of concealed cars in complex cargo X-ray imagery using deep learning. *CoRR, abs/1606.08078*.

[15] Jaccard, N., Rogers, T. W., Morton, E. J., & Griffin, L. D. (2016). Tackling the X-ray cargo inspection challenge using machine learning. *In: Proceedings SPIE*, *9847*, 98470N–98470N–13.

[16] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

[17] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). IEEE.

[18] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*, 3.

[19] Newell, A. J., & Griffin, L. D. (2011). Natural image character recognition using oriented basic image features. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on* (pp. 191–196). IEEE.

[20] Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1717–1724).

[21] Orphan, V. J., Muenchau, E., Gormley, J., & Richardson, R. (2005). Advanced γ ray technology for scanning cargo containers. *Applied radiation and Isotopes*, *63*, 723–732.

[22] Rogers, T., Jaccard, N., Morton, E., & Griffin, L. (2015). Detection of cargo container loads from x-ray images, .

[23] Rogers, T. W., Jaccard, N., Morton, E. J., & Griffin, L. D. (2016). Automated x-ray image analysis for cargo security: Critical review and future promise. *Journal of X-Ray Science and Technology*, (pp. 1–24).

[24] Rogers, T. W., Jaccard, N., Protonotarios, E. D., Ollier, J., Morton, E. J., & Griffin, L. D. (2016). Threat image projection (tip) into x-ray images of cargo containers for training humans and machines. In *Proceedings IEEE Inter national Carnahan Conference on Security Technology, to appear*.

[25] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252.

[26] Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 806–813).

[27] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, .

[28] Tuszynski, J., Briggs, J. T., & Kaufhold, J. (2013). A method for automatic manifest verification of container cargo using radiography images. *Journal of Transportation Security*, *6*, 339–356.

[29] U.S. Customs and Border Protection (). Container Security Initiative. Source: https://www.cbp.gov/sites/default/files/documents/csi_brochure_2011_3.pdf. Accessed: 17-10-2016.

[30] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision* (pp. 818–833). Springer.

[31] Zhang, J., Zhang, L., Zhao, Z., Liu, Y., Gu, J., Li, Q., & Zhang, D. (2014). Joint shape and texture based X-ray cargo image classification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop* (pp. 266 – 273).

[32] Zheng, Y., & Elmaghraby, A. (2013). A vehicle threat detection system using correlation analysis and synthesized x-ray images. In *SPIE Defense, Security, and Sensing* (pp. 87090V–87090V). International Society for Optics and Photonics.