Detection of concealed cars in complex cargo X-ray imagery using deep learning

Nicolas Jaccard[1] **Thomas W. Rogers**[1,3] **Edward J. Morton**[2] **Lewis D. Griffin**[1*]

[1]Department of Computer Science, University College London, London, UK
[2]Rapiscan Systems Ltd., Stroke-on-Trent, UK
[3]Department of Security and Crime Sciences, University College London, London, UK

[*]Corresponding-author: l.griffin@cs.ucl.ac.uk

Abstract

Non-intrusive inspection systems based on X-ray radiography techniques are routinely used at transport hubs to ensure the conformity of cargo content with the supplied shipping manifest. As trade volumes increase and regulations become more stringent, manual inspection by trained operators is less and less viable due to low throughput. Machine vision techniques can assist operators in their task by automating parts of the inspection workflow. Since cars are routinely involved in trafficking, export fraud, and tax evasion schemes, they represent an attractive target for automated detection and flagging for subsequent inspection by operators. In this contribution, we describe a method for the detection of cars in X-ray cargo images based on trained-from-scratch Convolutional Neural Networks. By introducing an oversampling scheme that suitably addresses the low number of *car* images available for training, we achieved 100% *car* image classification rate for a false positive rate of 1-in-454. Cars that were partially or completely obscured by other goods, a modus operandi frequently adopted by criminals, were correctly detected. We believe that this level of performance suggests that the method is suitable for deployment in the field. It is expected that the generic object detection workflow described can be extended to other object classes given the availability of suitable training data.

1 Introduction

Non-Intrusive Inspection (NII) systems are routinely used at transport hubs to scan the content of cargo containers, and ensure their compliance with both the shipping manifest and transport regulations, without disrupting the flow of commerce [1]. NII systems use radiation such as fast neutrons, gamma-rays or most commonly X-rays to image containers [2, 3, 4]. Currently, X-ray transmission images are inspected by human operators who search for anomalies or discrepancies with the shipping manifest [5].

Despite ambitious plans to scan all cargo entering the United States [6], it is not feasible to image every container due to the ever-increasing international trade volumes [7], let alone visually inspect all images hypothetically produced in the process. Container targeting is routinely carried out by risk analysis based on information such as origin, destination, and declared content [8, 9]. This approach limits the number of containers to image to those deemed "high risk" and thus greatly reduces the impact on the flow of commerce. However, the number of images to manually inspect remains overwhelmingly high, a trend compounded by the recent deployment of high throughput X-ray scanners capable of imaging cargo transported by rail at speed.

The application of machine vision methods to X-ray cargo images present many advantages over manual inspection, including high throughput through automation, consistency, scalability, and resistance to corrupt operation. However, as discussed in related work (Sec. 3), there is little published work on automated cargo X-ray image processing, potentially due to the difficulty of obtaining suitably large labeled datasets [10].

This contribution describes highly accurate algorithms for the detection of cars in complex X-ray cargo imagery. The automatic and reliable detection of cars is highly desirable because they are routinely involved in export fraud, tax evasion schemes, and trafficking activities [11, 12, 13]. Two main challenges are addressed: i) detection of cars that were intentionally obscured by other goods in order to minimise risk of detection by imaging and physical inspection; and ii) minimise the false alarm rate on *non-car* images that frequently contain "car-like" patterns. The latter point is particularly important for deployment in the field as unjustified false alarms could lead to operators ignoring the output of the automated detection scheme, or discourage its use altogether.

Results for preliminary *car* image classification experiments were previously presented at a conference [14]. However, the dataset used for evaluation was small and did not contain challenging adversarial examples. Moreover, the classification scheme was based around fixed image features only (intensity values, BIFs, oBIFs). This new contribution explores, for the first time, the classification of large X-ray cargo images using CNNs (either trained from scratch on X-ray cargo imagery or pre-trained on natural images) and compare their performance with that of the aforementioned fixed features. The dataset used for evaluation is also much larger and contains adversarial examples where cars are partially or completely obscured by other goods. Furthermore, additional experiments were carried out to better characterise the resilience of the CNN classification scheme to obscuration.

The structure of this paper is as follows. First, the formation process and properties of X-ray transmission images are briefly introduced in section 2. In section 3, related work from the literature is discussed. The methods underpinning the experiments carried out in this study are then outlined in section 4 while results of performance evaluation experiments comparing Convolutional Neural Networks (CNNs) with other types of features for the detection of cars in stream-of-commerce (SoC) images are presented in section 5. Finally, findings, limitations, and avenues of future research are discussed in section 6.

2 X-ray transmission image formation and properties

The two main components of a typical X-ray cargo scanner are an X-ray source and an array of detectors located behind the scanned object (Fig. 1). The fan-shaped X-ray beam (large vertical and small horizontal spread) is matched by the tall, narrow geometry of the detector array. The fraction of photons absorbed or scattered by the container and its content, is determined by measuring the number of photons incident on the detectors. The signal attenuations measured at different locations on the detector array are then mapped to pixel values, forming an X-ray transmission image where low attenuation regions (e.g. air) and regions containing dense objects have high and low pixel values, respectively. Due to the narrow geometry of the detector array, this acquisition process has to be repeated multiples times by moving either the container (portal configuration) or the source and detector array (gantry configuration). Individual column images are then assembled to form the final image.

X-ray transmission images differ significantly from visible spectrum photographs. In general, X-ray images have a skewed perspective due to the position of the source relative to the detector and scanned object, contain partially overlapping translucent objects, are cluttered, and are highly noisy [15, 16, 3].

3 Related work

Methods have been described to enhance images for improved target detection by operators, and to fully replace operators with detection algorithms. For enhancements, Retinex filters, false colour mapping, and the fusion of images acquired at two different energies (dual-energy X-ray imaging, e.g. for material discrimination based on atomic number) have been explored [17, 18, 19, 20]. For automated detection, bag of words (BoW) representations classified using support vector machines (SVM) were used for the analysis of 2D and 3D X-ray scans of baggage containing objects of interests such as firearms and mobile phones [21, 16, 22]. These studies report impressive performance, which

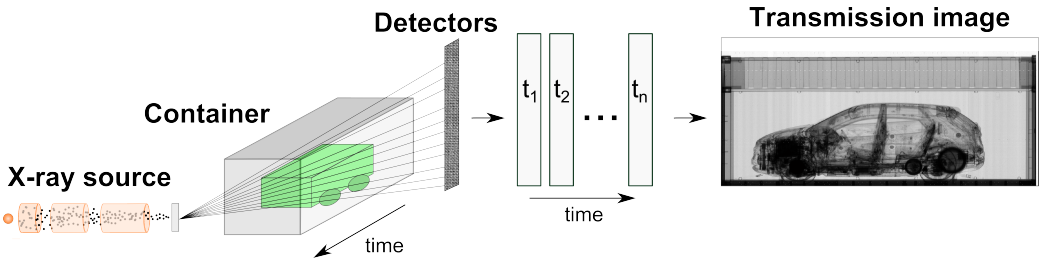

Figure 1: Illustration of the X-ray image formation and acquisition processes. Photons emitted by an X-ray source interact with a container and its content, leading to a signal attenuation measured by detectors placed behind the container. By moving the container or the detector, attenuations are determined spatially and are be mapped to pixel values to produce an X-ray transmission image

is in part made possible by the relatively constrained process of baggage scanning: scene dimensions and complexity are both bounded by the small dimensions of a bag. Multi-view (potentially volumetric), multi-energy, and high resolution imaging enable discriminating between threats and legitimate objects, with the latter being mostly identical across different baggage.

In contrast, the detection of threats and anomalies in X-ray cargo imagery is significantly more challenging. Scenes tend to be very large and complex with little constraints on the arrangement and packing of goods. Scanning is usually limited to a single view and the spatial resolution is much lower than in baggage, making it especially difficult to resolve and locate small anomalous objects. Moreover, a very high fraction of items packed in baggage are well-cataloged (e.g. clothing), whereas potentially anything can be transported in a container making it impractical to learn the appearance of frequent legitimate objects to facilitate the detection of threats. For these reasons, the performance reported for cargo imagery is usually low.

Zhang *et al.* [15] built a so-called "joint shape and texture model" of X-ray cargo images based on BoW extracted in superpixel regions. Using this model, images were classified into 22 categories depending on their content (e.g. car parts, paper, plywood). The results highlighted the challenges associated with X-ray cargo image classification, with only 51% of images being assigned to the correct category. In another effort to develop an automated method for the verification of cargo content in X-ray images, Tuszynski *et al.* [5] developed models based on the log-intensity histograms of images categorized into 92 high-level HS-codes (Harmonized Commodity Description Coding System). A city block distance was used to determine how much a new image deviates from training examples for the declared HS-code. Using this approach, 31% of images were associated with the correct category, while in 65% of cases the correct category was amongst the five closest matching models.

With around 20% of cargo containers being shipped empty, it would be of interest to automatically classify images as empty or non-empty in order to facilitate further processing (e.g. avoid processing empty images with object-specific detectors) and to prevent fraud. Rogers *et al.* [23] described a scheme where small non-overlapping windows were classified by a Random Forest (RF) based on multi-scale oriented Basic Image Features (oBIFs) and intensity moments. In addition, window coordinates were used as features so that the classifier would implicitly learn location-specific appearances. The authors reported that 99.3% of SoC non-empty containers were detected as such for a 0.7% false alarm rate and that 90% of synthetic images (where a single object equivalent to 1L of water was placed) were correctly classified as empty for 0.51% false alarms. The same problem was tackled by Andrews and colleagues [24] using an anomaly detection approach; instead of implementing the empty container verification as a binary classification problem, a "normal" class is defined (either empty or non-empty containers) and new images are scored based on their distance from this "normal" class. Features of markedly down-sampled images (32×9 pixel) were extracted from the hidden layers of an auto-encoder and classified by a one-class SVM, achieving 99.2% accuracy when empty containers were chosen as the "normal" class and non-empty instances were considered as anomalies.

Representation-learning is an alternative to classification based on designed features, whereby the image features that optimise classification are learned during training. CNNs, often referred to as deep learning, are representation-learning methods [25] that were recently shown to significantly

outperform other machine vision techniques in many applications, including large-scale natural image classification [26]. While most examples of applications to X-ray imagery to date have been limited to medical data [27], Akçay *et al.* [28] recently demonstrated the use of CNNs for baggage X-ray image classification. As there was insufficient training data to train a network from scratch, the authors fine-tuned a variant of the AlexNet architecture [29] that was pre-trained on ImageNet, a dataset of natural images. This approach significantly outperformed prior work in the field, indicating that features learned from natural images do indeed transfer, at least to a certain degree, to X-ray images.

To our knowledge, CNNs have not been applied to X-ray cargo imagery. In this contribution, we compare CNNs with other types of features and determine whether trained-from-scratch models (e.g. trained only on X-ray images) perform better than pre-trained networks.

4 Method

4.1 Dataset

X-ray transmission images of SoC cargo containers (typically 20 or 40 foot long) and tankers transported on railway carriages were acquired using a Rapiscan Eagle®R60 rail scanner equipped with a 6 MV linac source. Image dimensions vary between 1290×850 and 2570×850 pixel depending on the type of cargo and container size, with a pixel size of ≈ 6 mm pixel^{-1} in the horizontal direction. The raw images are greyscale with 16-bit precision.

For the purpose of this work, images containing *at least* one car (*car* images) are taken as the positive class and images not containing any car (*non-car* images) as the negative class. The dataset contains 79 *car* images for a total of 192 individual cars. *Car* images can be broadly divided into 5 categories: (i) a single car on its own in a small container (20 ft long), (ii) two cars in a large container (40 ft long), (iii) multiple cars stacked in a container, including one at an angle, (iv) a single car next to unrelated goods (no overlap), (v) one or two cars placed in-front or behind other goods (partial or complete occlusion). The specific car models and manufacturers were unknown, however based on visual appearances sedans, SUVs, compacts, and sports cars were present in the dataset.

Non-car images were randomly sampled from SoC images acquired over the course of several months. These images can be of cargo containers and tankers, with the first type being the most frequent. The nature of the cargo loads varies greatly from a container to another and include pallets of commercial goods, industrial equipment, household items, and bulk materials. Approximately 20% of the containers imaged were empty. *Non-car* images also include other types of vehicles such as vans, motorbikes, and industrial vehicles (e.g. tractors, bulldozers).

4.2 Image pre-processing

Prior to classification, X-ray transmission images were pre-processed as previously described by Rogers *et al.* [30, 23]. Black stripes resulting from source misfires or faulty detectors were first removed. Variations in the source intensity and sensor responses were corrected by column-wise pixel intensity normalisation based on air attenuation values, which are considered invariant. Erroneous isolated pixels (e.g. excessively bright or dark) were replaced by the median of their neighbourhood. For certain experiments, the log transform of images was also computed as it is frequently used to facilitate the detection of concealed items by operators and was also previously employed for the automated classification of cargo images by Tuszynski and colleagues [5].

4.3 Classification scheme

The detection of cars in X-ray images was implemented as a binary classification task (Fig. 2). A window-based approach was taken enabling i) to process optimally small sub-images for high classification performance as well as low computational time and memory consumption, and ii) to obtain approximate localisation of car-containing regions. Each window w_i, densely sampled from an image I, was classified and associated with a "car-likeness" score $p_{w,i}$. The image score p_I, which is indicative of the confidence that the image contains *at least* one car, was given by the maximum value of $p_{w,i}$ across all w_i of I. The image was classified as *car* if $p_I \geq t_{CAR}$, and *non-car* otherwise.

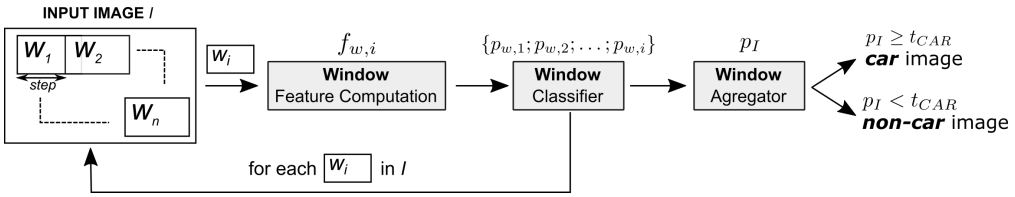

Figure 2: A window-based scheme for the classification of large X-ray cargo images. Windows are densely sampled from large input images and their features computed, based on which their "car-likeness" score is assigned by a window classifier. An image score is computed as the maximum window score across all windows of an image. Image class label (*car* or *non-car*) is obtained by thresholding of the image score.

t_{CAR} is a tunable threshold parameter that defines the balance between detection and false alarm rates.

Two types of windows were evaluated: square 512×512 pixel and rectangular 350×1050 pixel. The latter corresponded to the average size of cars in the training set and can be interpreted as a geometric prior. In all cases, windows were sampled with a stride of 32 pixels and 64 pixels for training and inference, respectively.

Heatmaps for classification visualisation were generated by mapping the mean window response at all image locations to pixel values. Such visualisations are essential to clarify the decision of the automated detection scheme and to enable verification by the operator before deciding whether further actions (e.g. physical inspection) are required.

Windows were classified by RF, SVM or logistic regression (for CNNs only) based on pixel intensity, fixed geometric image descriptors (BIFs), learned visual words (Pyramid Histograms Of Visual Words, PHOW), and features extracted from CNNs.

4.4 Window classification using Random Forest and Support Vector Machines

For this work, an open-source implementation of Random Forest for MATLAB was employed[1]. If not otherwise stated, classification was carried out using 40 trees, randomly sampling the square root of the total number of features at each split during tree building, and using equal weights for the two classes. For each window, the classifier outputs the "car-likeness" score $p_{w,i}$ computed as the fraction of trees voting for the *car* class.

Classification using linear SVMs was implemented using MATLAB's built-in functions. The box-constraint (or regularisation) parameters C and the kernel scale γ were tuned empirically. The "car-likeness" score $p_{w,i}$ was computed using a function that maps uncalibrated SVM scores to posterior probabilities. As proposed by Platt [31], a sigmoid was used as mapping function and parameters were estimated post-training using 10-fold cross validation.

In addition to RF and SVM, softmax was also used for classification using CNNs as described in section 4.6.

4.5 Feature computation

The simplest type of features assessed for *car* image classification was intensity values (Sec. 4.5.1). More advanced descriptors included oBIFs (fixed geometric features, sec. 4.5.2) and PHOW (learned visual words, sec. 4.5.3). CNNs for feature computation and classification are described in section 4.6.

4.5.1 Intensity features

Intensity features were encoded in multi-scale 256-bin histograms. Input images were blurred by convolution with a Gaussian kernel of standard deviation equal to 1, 2, 4, and 8. The resulting feature vector was 1024-dimensional. Histograms of intensity features were computed efficiently for a large number of windows using the integral histogram method described by Porikli [32].

[1] https://code.google.com/p/randomforest-matlab/ - Last accessed 31.05.2016

4.5.2 oriented Basic Image Features

BIFs encode textural information by classifying pixels of an image into one of seven categories according to local symmetry [33]. BIFs were computed based on the response to a bank of derivative-of-Gaussian (DtG) filters [33, 34]. The scale-normalized response s_{ij} to the ij-th DtG G_{ij} of scale σ_B is shown in equation 1.

$$s_{ij} = \sigma_B^{i+j} G_{ij} * I \tag{1}$$

Intermediate terms are then calculated pixel-wise: λ (equation 2) is the scale-normalised image Laplacian and γ (equation 3) is a measure of the variance over directions of the second directional derivative.

$$\lambda = s_{20} + s_{02} \tag{2}$$

$$\gamma = \sqrt{(s_{20} + s_{02})^2 + 4s_{11}^2} \tag{3}$$

The BIF value for a pixel is an integer between 1 and 7 given by the index of the largest of the following quantities: $\{\epsilon s_{00}, \sqrt{s_{10}^2 + s_{01}^2}, \lambda, -\lambda, \frac{\gamma+\lambda}{\sqrt{2}}, \frac{\gamma-\lambda}{\sqrt{2}}, \gamma\}$, with ϵ being a threshold parameter that dictates when a pixel is considered 'flat' (i.e. with no strong local structure), which is one type of BIF. The remaining six BIFs are slopes, dark blobs, bright blobs, dark lines, bright lines, and saddle-like (Fig. 3).

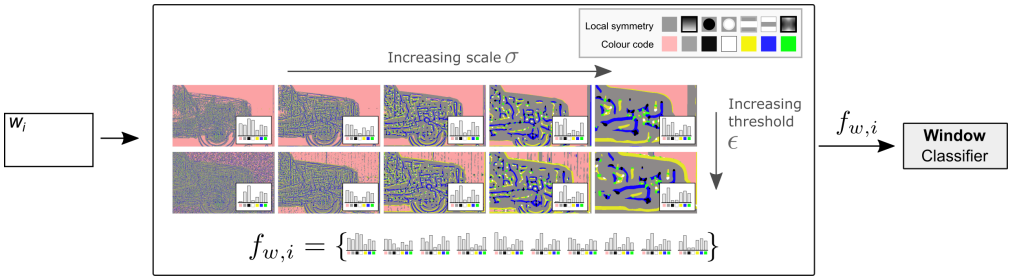

Figure 3: Computation of oriented Basic Image Features for window classification. oBIFs for the input window are computed at multiple scales and for different threshold values. Histograms for each combination of parameters are constructed and concatenated to produce the window feature vector. For clarity, orientation quantization is omitted from the schematic.

The BIF formulation can be extended by additionally determining the quantized orientation of rotationally asymmetric features [35]. This extended formulation, termed oriented Basic Image Features (oBIFs), has 23 features in total; with dark lines, light lines, and saddle-like types having 4 unpolarised orientations, while the slope type has 8 polarised directions. Implementations of both BIFs and oBIFs in MATLAB and Mathematica are available online [36].

oBIFs were computed at four scales (σ_B=$\{0.7, 1.4, 2.8, 5.6\}$) for two threshold parameters (γ=$\{0.011, 0.1\}$). oBIFs were encoded in histograms of 23 bins per scale and per threshold value, resulting in 184-dimensional feature vectors per window. As for intensity features, oBIFs histogram construction for multiple windows was carried out efficiently using the integral histogram method [32].

4.5.3 Pyramid Histograms Of visual Words

PHOW are a multi-scale extension of dense SIFT (Scale-Invariant Feature Transform) proposed by Bosch et *al.* [37, 38]. Whereas sparse SIFT approaches compute scale and rotation-invariant image descriptors based on local gradients at keypoint locations [39], dense SIFT features are computed for each pixel or on a regular grid with constant spacing [40]. The latter approach makes SIFT descriptors suitable for classification tasks where keypoints are not reliably detected or not consistent between the images considered, which is the case for X-ray cargo images.

PHOW computation (Fig. 4) consists of three steps : i) dense SIFT computation, ii) visual words quantization, and iii) spatial visual word histogram computation. SIFT descriptors were extracted

at each location of a regular grid with a step of 3 pixels. SIFT descriptors are spatial histograms of image gradient with 8 orientation bins and arranged in 4×4 spatial bins centred at each grid location, producing a 128-dimension feature vector per location. This extraction step was carried out at four different scales (4, 6, 8, and 10 pixels) by varying the dimensions of the spatial bins. Images were smoothed prior to computation, with Gaussian kernels of standard deviation equal to the scale divided by 6. Descriptors were then quantized into 300 visual words that were learned by k-means clustering of training image descriptors. A two level pyramid histogram of visual words (2×2 and 4×4 spatial bins) was constructed across all grid locations and scales, resulting in 6000-dimensional feature vectors for each window.

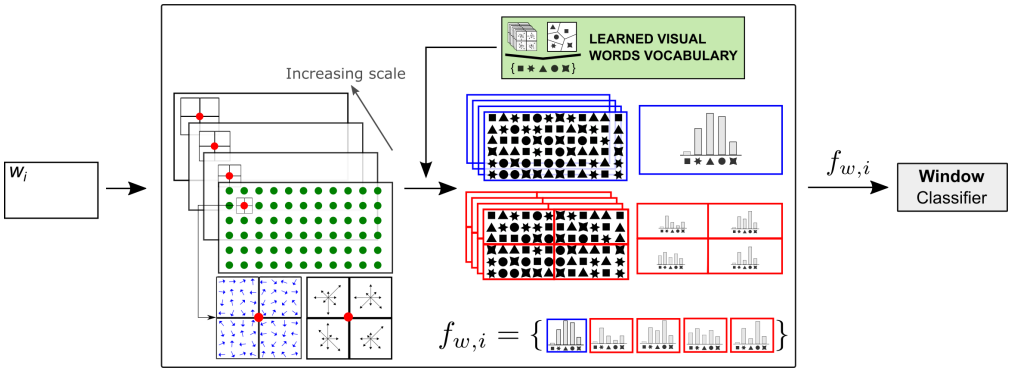

Figure 4: Computation of PHOW features for window classification. SIFT descriptors are extracted at multiple scales before being quantized into visual words. A two level pyramid histogram of visual words is the constructed across scales. The feature vector is obtained by concatenation of all individual visual word histograms.

4.6 Convolutional Neural Networks

CNNs were implemented using the MatConvNet library [41]. Two types of network were evaluated, both based on the very deep architectures proposed by Simonyan and Zisserman [42]. The first one is a 11-layer architecture (8 convolutional layers and 3 full-connected layers), while the second is a 18-layer architecture (16 convolutional layers and 3 fully-connected layers). In both cases, all filters in the convolutional layers had 3×3 dimensions. Details of the architectures can be found in supplementary materials. The networks were regularised by batch normalisation, whereby the mean and variance of layer inputs are fixed [43]. Batch normalisation performed significantly better than the conventional regularisation approach that uses dropout layers [44].

At the start of training, the learning rate was set to 10^{-4} and then to 10^{-5} when the validation error stopped decreasing. Weight decay was fixed at 5×10^{-4}. The average image computed over the training set was subtracted from all input images. When window classification was carried out solely based on CNNs, the "car-likeness" score $p_{w,i}$ was given directly by the output of the softmax classifier. In some experiments, features extracted from the first or second connected layers (FC1 and FC2, respectively) were classified using Random Forest or SVM classifiers as outlined in 4.4. Only 512×512 square windows were considered for classification using features extracted from CNNs. In order to make the memory footprint suitable for GPU processing, input images were first down-sampled to 256×256 pixels and converted to 8-bit precision.

In addition to models trained from scratch on windows sampled from X-ray cargo imagery, transfer learning was also evaluated. Window features extracted from the FC1 and FC2 layers of the VGG-VD-19 model [42] pre-trained on ImageNet were classified using Random Forest and SVM classifiers. As VGG-VD-19 expects 224×224 pixel RGB images as input, the grayscale channel of input X-ray images was replicated twice and downsampled, resulting in 3-channel 224×224 pixel images.

4.7 Car oversampling

While potentially millions of *non-car* windows examples can be sampled from the SoC dataset, there are only a total of 192 individual cars. Training a balanced classifier (i.e. 192 windows for each

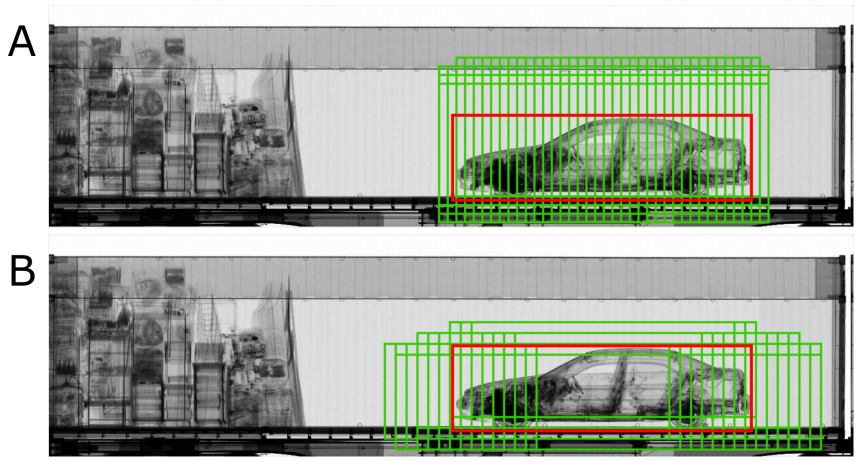

Figure 5: Example of *car* windows over-sampling. Windows in green are over-sampled and red windows indicate the user-annotated region of interest. Panels A and B show square window with $t_{ROI} = 0.5$ and rectangular windows with $t_{ROI} = 0.65$, respectively.

classes) would certainly lead to poor performance and generalisation. A similar outcome would be expected if a classifier was trained on a severely imbalanced dataset containing significantly more *non-car* examples. Such issues are frequently encountered in machine learning and more recently with CNNs where performance and generalisation is contingent on the availability of suitably large training datasets. Dataset augmentation by sampling random crops of input images at training was shown to significantly reduce CNN overfitting in large scale image classification tasks [29]. A similar approach was taken here.

Issues related to the scarcity of *car* window examples were alleviated by over-sampling of *car* regions at training. In addition to the user-defined ROI, partial *car* windows whose intersection with said ROI was greater than a t_{ROI} threshold value were also considered (Fig. 5). This approach had two advantages: i) it enabled training balanced classifiers with large number of examples, and ii) encouraged the classifier to be invariant to the position of the sampled windows in relation to the *car* ROI. t_{ROI} was set to 0.5 for square 512×512 windows (Fig. 5.A) and to 0.65 for 350×1050 rectangular window, increasing the number of *car* windows examples available at training by factors of ≈140 and ≈50, respectively (Fig. 5.B).

4.8 Performance evaluation

Performance was evaluated on the classification of entire images as *car* or *non-car* based on aggregated window scores. Two assumptions were made: (i) *non-car* images (negative class) were generally associated with lower p_I values (image score) than *car* images (positive class); and (ii) achieving high detection rate on *car* images was trivial but doing so while minimizing false alarms on *non-car* (e.g. high sensitivity, high specificity classification) is challenging. *Non-car* images were partitioned into disjoint training, validation, and test sets each comprising 10,000 SoC images.

The performance evaluation scheme was identical across all combinations of features and classifiers. Leave-one-out cross-validation (LOOCV) was used for the determination of p_I for *car* images due to the low number of examples of the positive class in the dataset. A classifier was trained using windows sampled from 78 *car* images and the *non-car* training set before being used to infer p_I for the left-out *car* image. The p_I for *non-car* validation images was computed using a classifier trained on all 79 *car* images and the same *non-car* training images. All free parameters, including t_{CAR}, were then tuned before repeating the process, with fixed parameters, using the *non-car* test images.

Combining the p_I values obtained for the negative class (hold-out on validation or test set) and positive class (LOOCV), performance metrics such as the area under the ROC curve (AUC) and the H-measure could be computed. The latter was introduced by Hand and Anagnostopoulos [45] to suitably accommodate imbalanced datasets, such as the one considered here, while also addressing issues related to the underlying cost function of the AUC metric. Like the AUC, the H-measure can

be computed without having to explicitly set a value for the threshold parameter (here t_{CAR}). A beta distribution with modes ($\pi_2 + 1$, $\pi_1 + 1$) is used as distribution of relative misclassification severities, where π_2 and π_1 are the relative frequencies of the positive and negative class, respectively. Details regarding the H-measure computation are given elsewhere [46] and implementations for most scientific computing packages are freely available[2]. The false positive rate (FPR) was computed by thresholding the test set p_I scores using the highest possible value for t_{CAR} (tuned individually for each experiment based on validation images) that still resulted in 100% *car* image classification accuracy.

During performance evaluation, dictionary learning for PHOW features and mean image computation for CNNs were carried out solely based on training images (e.g. new dictionaries were learned and new mean images were computed for each iteration of LOOCV).

4.9 Generation of synthetically obscured car examples

Synthetically obscured *car* images were generated by projecting *non-car* objects onto SoC *car* images. Due to the nature of the X-ray transmission image formation process, objects can be inserted into images by multiplication as previously described by Rogers *et al.* [23]. The process started with a raw *car* image. A first object was sampled from a database containing a total of 196 objects and placed at a random location in the container. The dimensions and density of the object were set to half and a third of that of a typical *car*, respectively. The newly generated synthetic image was then classified and the image score p_I computed. The mean relative attenuation of the *car* ROI was computed as the difference between the synthetic image and the raw image, divided by the raw image. This process was repeated, adding more and more objects, until the car was completely obscured (mean relative attenuation equal to one). Five different realisations of this experiment were combined to generate a plot of the image score versus mean relative attenuation.

5 Results

For each type of feature considered, the best car image classification results obtained across different combinations of pre-processing, window geometry and classifiers are presented in table 1. It was found that an approach combining multi-scale computation (scale=$\{1, 2, 4, 8\}$) and encoding using 256-bin histograms (though diminishing returns were observed from 32-bin upwards) was optimal for intensity features. Log-transforming windows prior to analysis was found to be detrimental but using rectangular windows (based on prior knowledge about car geometry) significantly improved performance over square windows (H-measure of 0.95 and 0.86, respectively). However, intensity features performed the worst when compared to other types of features with a false alarm rate above 5%; while the differences in intensity distribution between *car* and *non-car* windows might be a useful cue for classification, more advanced image descriptors such as PHOW and oBIFs were required to achieve satisfactory levels of performance.

PHOW features outperformed intensity features when using raw images as input and log-transforming windows led to a further two-fold decrease in false alarm rate to approximately 1%. Interestingly, oBIFs outperformed PHOW features even though the former do not rely on *ad-hoc* dictionary learning or a pyramidal scheme. Instead, oBIFs are fixed geometric descriptors computed independently at multiple scales. oBIFs results showed a ≈3-fold improvement in false alarm rate to 0.35% when compared to PHOW features. Using BIFs instead of oBIFs led to a marked degradation in performance, indicating that orientation quantisation was beneficial for classification. Log-transforming input windows also had a negative impact on classification using oBIFs, which was potentially caused by the lack of apparent texture and structure in these transformed images.

The best performance across all experiments, correct classification of all cars and a false positive rate of 0.22% (p_I = 0.990), was achieved using features extracted from the FC1 layer of a trained-from-scratch CNN when square input windows were log-transformed and classification was carried out using a random forest model. The 95% confidence interval for the detection rate, which was estimated by supplementing the results with a single artificial failure case, was [0.96, 100].

The 18-layer trained-from-scratch CNN outperformed the shallower 11-layer network in all cases, indicating that the former generalised well to unseen data despite significantly increased complexity.

[2] http://www.hmeasure.net/ - Last accessed 23.06.2016

Table 1: Performance for the detection of cars in X-ray cargo images. Only the best results for each type of features shown. +Log denotes that input images were log-transformed prior to features computation. R and S denote 1050×350 and 512×512 windows, respectively.

Features	Windows	Classifier	H-measure	FPR [%]
Intensity (4 scales)	R	RF	0.900	5.20
PHOW (4 scales) + Log	S	RF	0.977	1.05
oBIFs (4 scales, 2ϵ)	R	RF	0.992	0.35
CNN 11-layer + Log	S	SM	0.990	0.47
CNN 18-layer (FC1) + Log	**S**	**RF**	**0.995**	**0.22**
ImageNet VGG-VD-19 (FC2) + Log	S	SVM	0.993	0.34

The second-best result was obtained using a CNN pre-trained on the ImageNet dataset with no further fine-tuning, which suggests that features learned from natural images constitute a robust baseline for X-ray image classification.

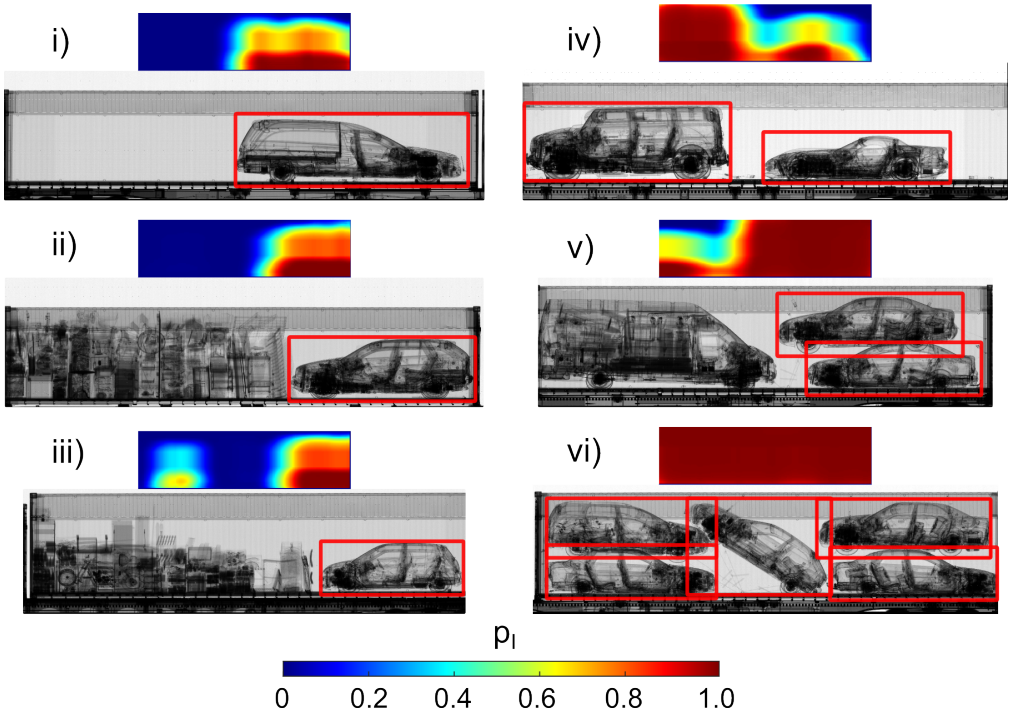

Figure 6: Classification outcome for **non-obscured** *car* images during leave-one-out cross-validation (previously unseen by classifier). For each example, raw X-ray transmission image (top, with additional red outlines indicating the location of cars) and the output of the classifier formatted as a heatmap (bottom) are shown.

Figure 6 shows representative examples of *car* image classification by the CNN scheme where individual cars are not obscured by other goods. Various scenarios are shown: single cars without other goods (Fig. 6.i), multiple cars without other goods (Fig 6.iv, v, and vi), car with other goods (Fig. 6.ii and iii), cars with other vehicles (Fig. 6.v), and cars at an angle (Fig. 6.vi). In all cases, cars were also suitably localised by the heat map generated during classification regardless of the model (e.g. sedan, coupe, station wagon, SUV) and dimensions. Regions of images that contained other unrelated cargo usually gave very little to no signal (Fig. 6.ii), with the exception of cases where said cargo also included semantically-related objects, such as motorbikes (Fig. 6.iii) or vans (Fig. 6.v). The CNN scheme also performed well for complex X-ray imagery in which cars were partially and completely obscured by other cargo (Fig. 7).

The vast majority of *non-car* images (97.82% of the test set) had $p_I \leq 0.5$ and are thus correctly classified using a naive $t_{CAR}=0.5$ threshold (Fig. 8). These images typically include empty containers

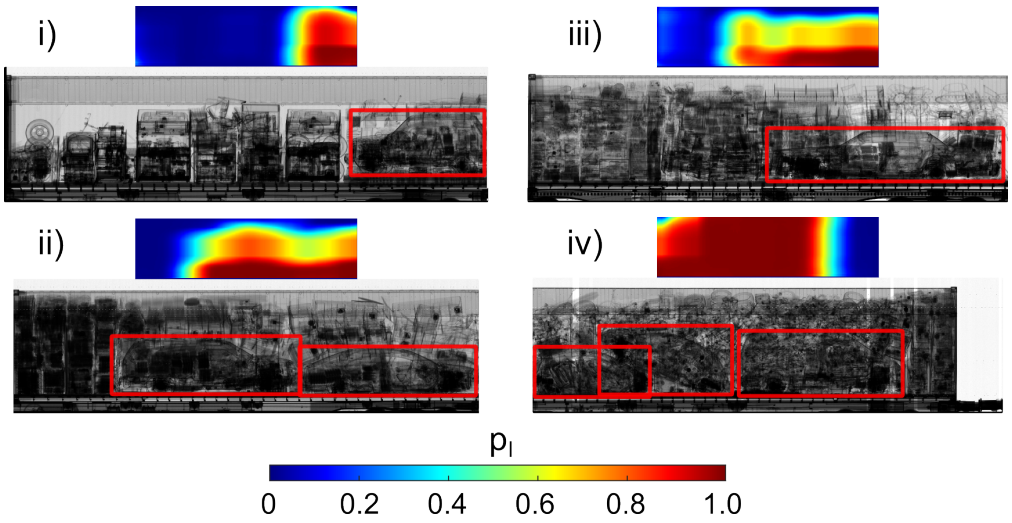

Figure 7: Classification outcome for **obscured** *car* images during leave-one-out cross-validation (previously unseen by classifier). For each example, raw X-ray transmission image (top, with additional red outlines indicating the location of cars) and the output of the classifier formatted as a heatmap (bottom) are shown.

(Fig. 8.i a), containers fully filled with bulk materials (Fig. 8.ii a), and containers with goods loaded onto pallets (Fig. 8.iv a). The intermediate image category ($0.5<p_I\leq 0.95$, 1.6% of the test set) was more challenging to classify due to the presence of uncommon high frequency structures. This includes industrial vehicles (Fig. 8.i b), containers sparsingly loaded with bulk materials (Fig. 8.ii b), and pallets containing objects that present *car*-like features. Images with $p_I>0.95$ represented 0.6% of the test set and a majority of those contained objects that are visually and semantically similar to cars, including motorbikes (Fig. 8.i c) and vans (Fig. 8.ii b). Indeed, if only considering false alarms that are not related to vehicles, the FPR for the CNN scheme decreases from 0.22% to just 0.08%.

Interestingly, the response for a given type of objects differed vastly depending on factors such as orientation, spatial arrangement, and fraction of space left empty in a container. For example, bulk materials were usually associated with very low image scores when uniformly loaded throughout the entire container but result in scores that tended to increase as more container background was visible (see Fig. 8 ii a-c). Similarly, images of tires, which are semantically related to cars, were given low scores when their arrangements inside the container was such that a majority only had their profile showing (Fig. 8.iii a). However, image score increased markedly as packing order decreased and tires adopted multiple orientations, including those typically seen in *car* images (Fig. 8.iii b and c).

The robustness of the proposed classification scheme to obscuration of cars was evaluated by generating synthetic adversarial images where other goods were projected into *car* images (Fig. 9). Up to a mean relative attenuation of 0.8, which corresponds to a visually very busy scene, the CNN scheme consistently classified the synthetic images as *car* (i.e. $p_I\geq 0.990$). This indicates good resilience to concealment strategies commonly used by criminals as shielding methods to provide this degree of attenuation would be difficult to devise in practice. The spatial arrangement of the obscuring objects also played a role as shown by distinct synthetic images with the same relative attenuation value resulting in different classification outcomes.

6 Conclusion

We described a scheme for the detection of cars in X-ray cargo imagery whereby densely sampled windows were classified using Convolutional Neural Networks (CNNs). The proposed approach significantly outperformed other methods such as classification of windows based on intensity features, Pyramids Histograms of Visual Words (PHOW), and oriented Basic Image Features (oBIFs). *Car* window oversampling alleviated issues associated with the low number of *car* examples and

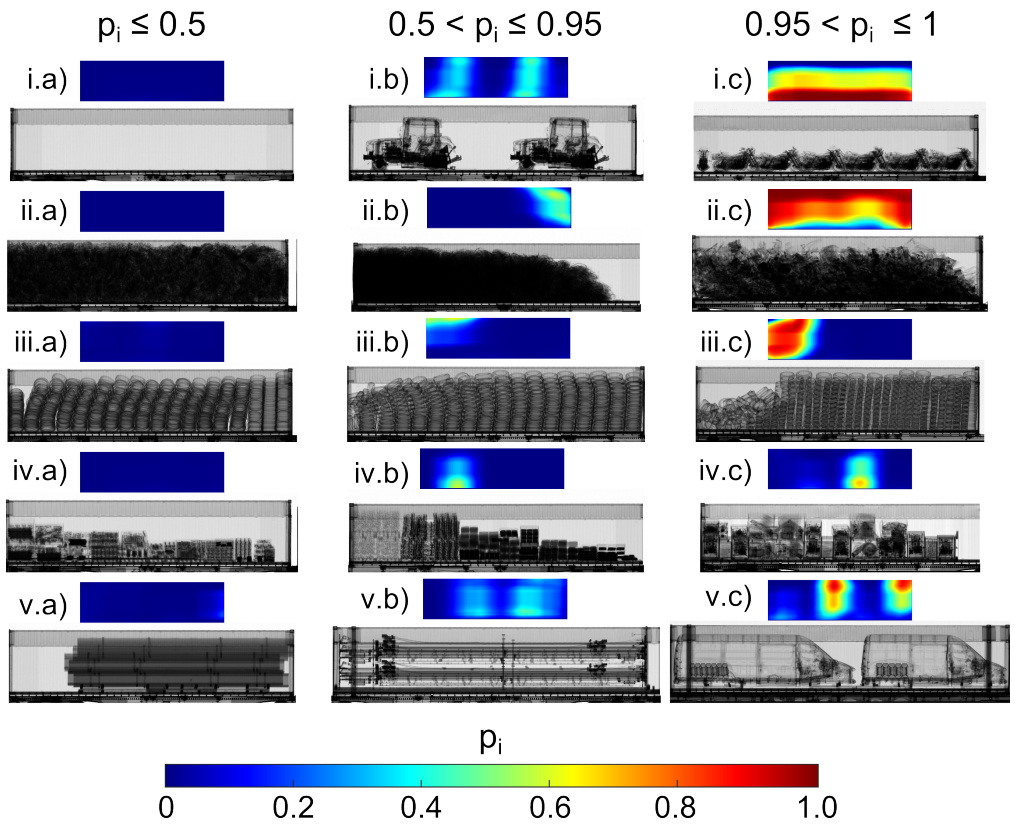

Figure 8: Classification outcome for SoC *non-car* images from the test set (previously unseen by the classifier). For each example, the X-ray transmission image (bottom) and the localisation heatmap (top) are shown.

enabled the training of networks from scratch instead of solely relying on pre-trained networks as done previously for baggage imagery [28].

All *car* images in the stream-of-commerce dataset were correctly classified as such, including cases where cars were partially or totally obscured by other goods. In an experiment based on synthetic images, the CNN scheme demonstrated a high degree of robustness to obscuration by accurately classifying images that could be deemed challenging for Human observers. This resilience might be made possible by the use of log-transformed input images; due to the nature of the X-ray image formation process, *car*-like structures are preserved in those transformed images and can still act as classification cues that CNNs excel at picking up, even for moderate to high relative attenuation values.

When trained from scratch, CNNs can thus suitably accommodate properties of X-ray imagery, such as translucency and multiplicative occlusion, that do not typically occur in natural images. Moreover, with fewer than 1-in-450 false alarms, it was shown that perfect sensitivity did not come at the cost of unsuitably low specificity. A large fraction of those false positives were comprised of images containing objects semantically related to cars.

On average using a MATLAB implementation running on a E5-1620 Intel Xeon 3.4 Ghz CPU, a Titan X GPU and 32 GB of RAM, image classification including pre-processing, sampling, CNN features computation, and classification using Random Forest took 2.6 seconds. It is likely that processing could be further improved by taking advantage of lower-level coding, smart batch processing of sampled windows, and multiple GPU setups.

Several limitations and areas for future work were identified. The images used in the experiments were all acquired using a single X-ray machine. Further investigations will be required to determine how the processing parameters and classifiers would generalise across different models, operating conditions (e.g. photon energy), and imaging protocols. In particular, differences in image geometry

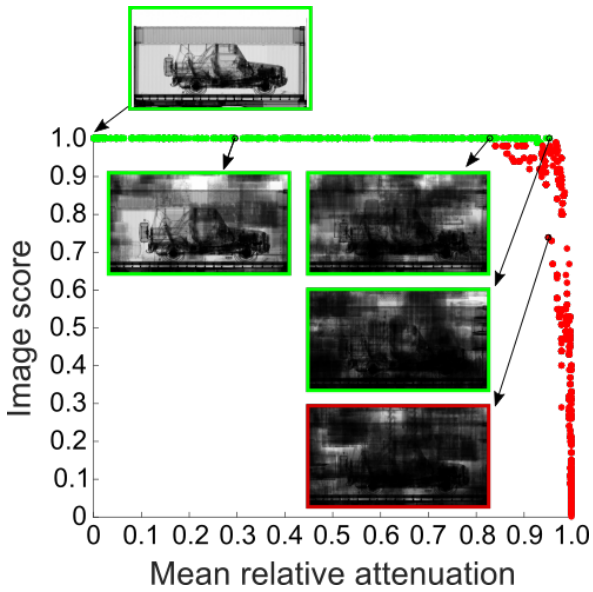

Figure 9: Evaluation of the robustness of the classification scheme to obscuration by other cargo. The green and red points indicate scores above and below the optimal threshold $p_I = 0.990$, respectively.

(e.g. perspective, warping) might require instrument-specific calibrations. It would also be beneficial to evaluate performance on a larger dataset of *car* images. However, while classification performance does increase with the number of *car* images used at training, it was found that said performance started to plateau and adding more training examples from 50 images onwards yielded diminishing returns.

The reported performance suggests that this approach could be deployed in the field to assist operators in the detection of fraud and crime related to the undeclared transport of cars in cargo containers. Due to its generic nature, this deep learning scheme could likely be used to detect many classes of objects in complex X-ray imagery, even when only a modest dataset of examples is available.

7 Acknowledgements

This work was funded by Rapiscan Systems Ltd. and through the EPSRC Grant no. EP/G037264/1 as part of UCL's Security Science Doctoral Training Centre.

8 References

[1] Ravi Sarathy. Security and the global supply chain. *Transportation journal*, 45:28–51, 2006.

[2] Nicholas G. Cutmore, Yi Liu, and James R. Tickner. Development and commercialization of a fast-neutron/x-ray Cargo Scanner. In *International Conference on Technologies for Homeland Security*, pages 330–336. IEEE, November 2010.

[3] Floyd Del McDaniel, Barney L. Doyle, Gyorgy Vizkelethy, Brant M. Johnson, Janet M. Sisterson, and Gongyin Chen. Understanding X-ray cargo imaging. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 241(1):810–815, 2005.

[4] Y Liu, BD Sowerby, and JR Tickner. Comparison of neutron and high-energy x-ray dual-beam radiography for air cargo inspection. *Applied Radiation and Isotopes*, 66(4):463–473, 2008.

[5] Jarosław Tuszynski, Justin T. Briggs, and John Kaufhold. A method for automatic manifest verification of container cargo using radiography images. *Journal of Transportation Security*, 6(4):339–356, July 2013.

[6] Kristin Archick. *US-EU cooperation against terrorism*. DIANE Publishing, 2010.

[7] Regina Asariotis, Hassiba Benamara, Jan Hoffmann, Azhar Jaimurzina, Anila Premti, José María Rubiato, Vincent Valentine, and Frida Youssef. *Review of maritime transport 2013 (UNCTAD/RMT/2013)*. United Nations Publication, Geneva, 2013.

[8] John King. The security of merchant shipping. *Marine Policy*, 29(3):235–245, May 2005.

[9] Stuart F. Weele and Jose E. Ramirez-Marquez. Optimization of container inspection strategy via a genetic algorithm. *Annals of Operations Research*, 187(1):229–247, February 2010.

[10] D Mery. Computer vision technology for X-ray testing. *Insight - Non-Destructive Testing and Condition Monitoring*, 56(3):147–155, March 2014.

[11] A. A. Aronowitz, D. C. G. Laagland, and G. Paulides. *Value-added Tax Fraud in the European Union*. Kugler Publications, 1996.

[12] Ronald V Clarke and Rick Brown. International trafficking in stolen vehicles. *Crime and Justice*, pages 197–227, 2003.

[13] Ronald V. Clarke and Rick Brown. International Trafficking of Stolen Vehicles. In Mangai Natarajan, editor, *International Crime and Justice*, chapter 16, pages 126–132. Cambridge University Press, 2010.

[14] Nicolas Jaccard, Thomas W. Rogers, and Lewis D. Griffin. Automated detection of cars in transmission X-ray images of freight containers. In *International Conference on Advanced Video and Signal Based Surveillance*, pages 387–392. IEEE, August 2014.

[15] Jian Zhang, Li Zhang, Ziran Zhao, Yaohong Liu, Jianping Gu, Qiang Li, and Duokun Zhang. Joint Shape and Texture Based X-Ray Cargo Image Classification. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 266–273. IEEE, June 2014.

[16] Muhammet Baştan, Mohammad Reza Yousefi, and Thomas M. Breuel. Visual words on baggage x-ray images. In *Conference on Computer Analysis of Images and Patterns*, CAIP'11, pages 360–368, Berlin, Heidelberg, 2011. Springer-Verlag.

[17] B.R. Abidi, D.L. Page, and M.A. Abidi. A Combinational Approach to the Fusion, De-noising and Enhancement of Dual-Energy X-Ray Luggage Images. In *Conference on Computer Vision and Pattern Recognition - Workshops*, volume 3, pages 2–2. IEEE, 2005.

[18] Glenn Woodell, Zia-ur Rahman, Daniel J. Jobson, and Glenn Hines. Enhanced images for checked and carry-on baggage and cargo screening. In Edward M. Carapezza, editor, *International Conference on Advances in Pattern Recognition*, September 2005.

[19] B.R. Abidi, Y. Zheng, A.V. Gribok, and M.A. Abidi. Improving Weapon Detection in Single Energy X-Ray Images Through Pseudocoloring. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 36(6):784–796, November 2006.

[20] S. Ogorodnikov and V. Petrunin. Processing of interlaced images in 4–10 MeV dual energy customs system for material recognition. *Physical Review Special Topics - Accelerators and Beams*, 5(10):104701, October 2002.

[21] Diana Turcsany, Andre Mouton, and Toby P. Breckon. Improving feature-based object recognition for X-ray baggage security screening using primed visualwords. In *International Conference on Industrial Technology*, pages 1140–1145. IEEE, February 2013.

[22] Greg Flitton, Andre Mouton, and Toby P Breckon. Object classification in 3d baggage security computed tomography imagery using visual codebooks. *Pattern Recognition*, 48(8):2489–2499, 2015.

[23] TW Rogers, N Jaccard, EJ Morton, and LD Griffin. Detection of cargo container loads from x-ray images. 2015.

[24] Jerone TA Andrews, Edward J Morton, and Lewis D Griffin. Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing*, 6(1):21, 2016.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. IEEE, 2015.

[27] C Cernazanu-Glavan and S. Holban. Segmentation of Bone Structure in X-ray Images using Convolutional Neural Network. *Advances in Electrical and Computer Engineering*, 13(1):87–94, 2013.

[28] Samet Akçay, Mikolaj E Kundegorski, Michael Devereux, and Toby P Breckon. Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery. In *Proceeding of the International Conference on Image Processing, IEEE*. IEEE, 2016.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[30] Thomas W Rogers, James Ollier, Edward J Morton, and Lewis D Griffin. Reduction of wobble artefacts in images from mobile transmission x-ray vehicle scanners. In *International Conference on Imaging Systems and Techniques*, pages 356–360. IEEE, 2014.

[31] John C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.

[32] Fatih Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 829–836. IEEE, 2005.

[33] L. D. Griffin, M. Lillholm, M. Crosier, and J. van Sande. Basic image features (bifs) arising from approximate symmetry type. In Xue-Cheng Tai, Knut Mørken, Marius Lysaker, and Knut-Andreas Lie, editors, *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 343–355. Springer Berlin Heidelberg, 2009.

[34] L. D. Griffin, P. Elangovan, A. Mundell, and D.C. Hezel. Improved segmentation of meteorite micro-ct images using local histograms. *Computers & Geosciences*, 39:129–134, 2012.

[35] Andrew J. Newell and Lewis D. Griffin. Natural Image Character Recognition Using Oriented Basic Image Features. In *International Conference on Digital Image Computing: Techniques and Applications*, pages 191–196. IEEE, December 2011.

[36] L. D. Griffin et al. Basic Image Features (BIFs) implementation. Available at: `https://github.com/GriffinLab/BIFs`, 2015.

[37] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification via plsa. In *Computer Vision–ECCV 2006*, pages 517–530. Springer, 2006.

[38] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[39] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[40] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[41] Andrea Vedaldi and Karel Lenc. Matconvnet-convolutional neural networks for matlab. *arXiv preprint arXiv:1412.4564*, 2014.

[42] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. sep 2014.

[43] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. feb 2015.

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, jan 2014.

[45] David J. Hand and Christoforos Anagnostopoulos. A better Beta for the H measure of classification performance. February 2012.

[46] David J Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine learning*, 77(1):103–123, 2009.