Tackling the X-ray cargo inspection challenge using machine learning

Nicolas Jaccard[a], Thomas W. Rogers[a,c], Edward J. Morton[b], and Lewis D. Griffin[a]

[a]Dept. of Computer Science, University College London, UK
[b]Rapiscan Systems Ltd., Victoria Business Park, Stroke-on-Trent, UK
[c]Dept. of Security and Crime Sciences, University College London, UK

ABSTRACT

The current infrastructure for non-intrusive inspection of cargo containers cannot accommodate exploding commerce volumes and increasingly stringent regulations. There is a pressing need to develop methods to automate parts of the inspection workflow, enabling expert operators to focus on a manageable number of high-risk images. To tackle this challenge, we developed a modular framework for automated X-ray cargo image inspection. Employing state-of-the-art machine learning approaches, including deep learning, we demonstrate high performance for empty container verification and specific threat detection. This work constitutes a significant step towards the partial automation of X-ray cargo image inspection.

Keywords: Cargo screening, X-ray imaging, Classification, Machine Learning, Deep Learning, Threat Image Projection, TIP

1. INTRODUCTION

In 2013, global cargo container traffic at ports surpassed 650 million TEU (twenty-foot equivalent unit, the size of a standard container).[1] Growth of the global shipping network increases security risks as any container could potentially be used by malicious actors to smuggle restricted or prohibited items across borders. The methods employed by criminals include concealment of undeclared goods amongst legitimate cargo or in the fabric of the container itself (e.g. floor, refrigeration unit).[2] The detection and prevention of this type of criminal activity is particularly challenging as it must be done with minimal disruption to the flow of commerce. Indeed, it is not feasible to subject a significant fraction of containers to a physical inspection, a time-consuming and costly process. Instead, the detection of undeclared goods relies on a combination of statistical risk analysis (e.g. based on shipping information), non-invasive X-ray imaging, and physical inspection as a last resort.

The inspection of X-ray cargo images is a challenging visual search task for security officers;[3] the images tend to be very cluttered, often with other objects whose appearance closely resemble that of the targets of interest. This issue is compounded by the large variety of objects that are transported in cargo containers, making it difficult for operators to learn a complete catalogue of visual appearances. In addition, and unlike optical images of natural scenes, imaged objects appear translucent and overlap,[4–6] which complicates their interpretation. As such, an impractically large contingent of suitably trained security officers would be required to inspect all the images that could be acquired from the increasing number of container transactions. Machine vision and learning have been successfully employed to automate similar search tasks in other fields. However, their application to X-ray images, outside of biomedical applications, appear to be mostly limited to image pre-processing and enhancement methods.[7–10]

In this paper we propose a machine learning framework for tackling the X-ray cargo inspection challenge. The framework has two distinct module types; (i) the verification of whether containers are empty or not, and (ii) the detection of specific threat items. *Empty verification* acts to filter images that do not need inspection by the specific object detectors, that is those 20% of containers in the stream-of-commerce (SoC) that are definitely

Further author information: (Send correspondence to Lewis D. Griffin)
Lewis D. Griffin: E-mail: L.Griffin@cs.ucl.ac.uk, Telephone: +44 20 3108 7107

Anomaly Detection and Imaging with X-Rays (ADIX), edited by
Amit Ashok, Mark A. Neifeld, Michael E. Gehm, Proc. of SPIE Vol. 9847,
98470N · © 2016 SPIE · CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2222765

empty. It is also used to raise a flag when a declared-as-empty container is found to contain cargo, since it is likely that the container is being used in shipping fraud or to smuggle contraband.

All detected non-empty containers are sent to the specific threat item detection modules. We focus on two examples; cars and 'small metallic threats'. Thus, we cover both large-scale (cars occupy most of a container) and small-scale ('small metallic threats' occupy only a very small region of the image) threat detection. For convenience, we refer to cars as threats and indeed they are frequently involved in criminality, either when they are disassembled or undeclared to avoid duty, or when they have been stolen. Please note, we use the term 'small metallic threats' as we do not wish to make our research results easily discoverable by keyword searching. However, the threats in question are similar in form to hand drills.

In the next section we discuss related work in cargo and baggage imagery. In Sec. 3 we give an overview of the imaging modality and the collected dataset used in training and testing our methods. In Sec. 4 we present the proposed framework and a description of each module, as well as a discussion of the performance of the detection module.

2. RELATED WORK

There is a small, but steadily growing, body of publications on X-ray image processing for security purposes; mainly on threat detection in baggage, with a smaller number on the classification of X-ray cargo images. The sparsity of publications on cargo is presumably due to the relative novelty of the imaging technology and the difficulty of obtaining large datasets for training and validating algorithms. We have previously reported on our *car detection* module[11] and our *empty verification* module.[12] Other researchers have focussed on methods for *empty verification*[13–15] and *manifest verification*.[4,16]

On cargo, published classification methodologies have mostly employed devised metrics with little application of machine learning. For *empty verification*, Chalmers *et al.*[13,14] extract the container and generate "a histogram of cargo region data values" and compare "key metrics (e.g. the minimum, mean, standard deviation, and maximum cargo data values) with a reference set generated from a known empty container". Orphan *et al.*[15] report an accuracy 97.2% (with a 0.4% false negative rate) for SoC *empty verification*, by segmenting the image (floor/walls/roof) and applying a rule-based object detection algorithm. No attempt is made to extend the method to small, difficult loads, as we do in this work.

Manifest verification is a multi-class classification problem, where containers are classified according to their Harmonized System code (HS - broad category of cargo type e.g. live animals & animal products, or vegetable products). Tuszynski *et al.*,[16] form a model for each HS code by taking the median image grey-level histogram and average absolute deviation. They use a weighted city block distance to compare a given test image to each HS code model. This approach yields an overall accuracy of 48% given a false positive rate of 5%. Zhang *et al.*[4] achieve slightly better performance, by using a visual codebook based on a bank of Leung-Malik filters, and showing that this outperforms scale-invariant feature transform (SIFT). That approach leaves out image examples where the container is less than half-filled with goods. In this work, we include and encourage the machine learning to be robust to such examples. We also extend testing to very small synthetic loads representative of adversarial attempts to smuggle small amounts of contraband.

Research activities for *'small metallic threat' detection* in baggage security are based on three imaging modalities; (i) single-view X-ray imaging (single view),[17] (ii) multi-view X-ray imaging,[18,19] and (iii) Computed Tomography (CT).[20–23] Classification performance typically improves with the number of views, due to the additional information available. The state-of-the-art for *'small metallic threat' detection* ranges from 89% detection and 18% false positives using a single view,[17] to 97.2% detection and 1.5% false positives in full CT imagery.[20] For cargo, *'small metallic threat' detection* is much more challenging than for single-view baggage, since the threat is physically very small relative to a 20 ft container. Thus it can appear in many more distinct positions within the image than it can in baggage, which can lead to high false alarm rates without advanced methods. Additionally, threats are often shielded by dense cargo such that they are often not visible without intensity manipulations of the image.

The consensus, amongst the baggage community, is that X-ray image data is more challenging than visible spectrum data, and that direct application of methods used frequently on natural images (such as SIFT, RIFT,

HoG) does not always perform well.[24] However, performance can be improved by exploiting the characteristics of X-ray images; by augmenting multiple views, using a coloured material image,[25] or using simple (gradient) density histogram descriptors.[20,22] In this work, we only have access to a single view, and the high photon energy regime required for cargo imaging makes it difficult to form coloured material images.[10] It has also been widely reported that texture descriptors perform poorly for baggage images which are fairly unvarying in this respect.[24–26] In contrast, the amount of visible texture in cargo X-ray images does differ significantly between images. Medium to low density cargo (such as tyres, and machinery) is highly textured, while high density cargo (such as barrels of oil) has a uniform texture.

To our knowledge, no attempts have been made to use Deep Learning in the classification of X-ray images of cargo or baggage. However, Bastan[25] discusses some of the potential difficulties when learning features using deep convolutional neural networks (CNNs), including out-of-plane rotations and varying image size. Bastan suggests that CNNs have a slight in-plane rotation invariance due to pooling but that data augmentation is required to handle the "severe rotation problem". We are faced with similar problems in cargo. We handle varying image sizes by using sliding windows and taking the *maximum* of their scores to make a classification for the whole image. We do not explicitly tackle the out-of-plane rotation problem, instead we rely on a dataset captured with a sufficient number of poses of the threat objects.

3. STREAM-OF-COMMERCE DATASET

The stream-of-commerce (SoC) dataset consists of $\sim 1.2 \times 10^5$ cargo images collected over a period of one year. The images were captured using a Rapiscan Eagle®R60, which is a state-of-the-art single-view transmission X-ray portal system capable of imaging rail-hauled containers moving at 60 km/h. This is achieved using a two-dimensional imaging array and by warping images to account for small accelerations of the train as it moves through the scanner. Since the scanner can operate at high speed, it is able automatically to capture images of containers and their contents without disrupting the flow of commerce.

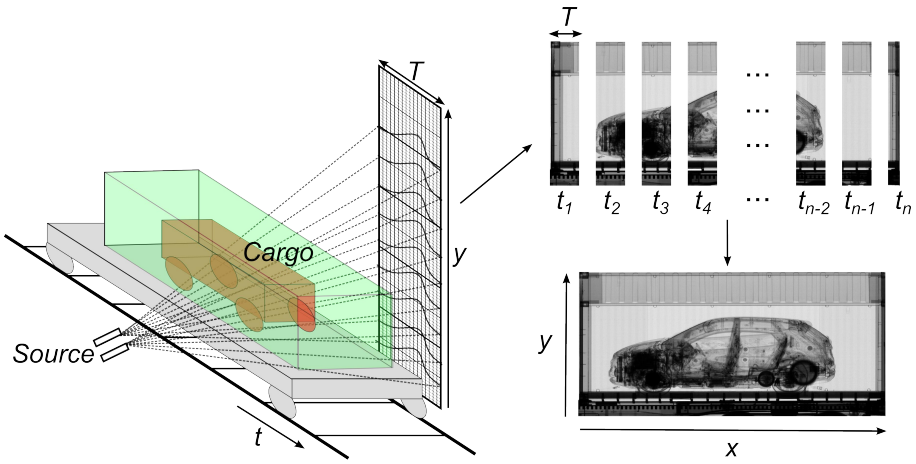

Figure 1. Illustration of the Rapiscan Eagle®R60 rail scanner. The scanner operates in portal mode. The imaging array has both vertical and horizontal extent, and at each time t a small patch of width T is imaged. These patches are stitched together over time to form the full X-ray image of the container. Source variation[27] leads to vertical stripes (of width T) in the image.

4. A FRAMEWORK FOR AUTOMATED CARGO INSPECTION

The proposed framework for automated cargo inspection consists of four modules: i) pre-processing, ii) image synthesis, iii) *empty verification*, iv) *car detection*, and v) *'small metallic threat' detection* (Fig. 2). *Empty verification* confirms that declared-as-empty containers really are empty and raises an alarm if not. Concealment of objects in declared-as-empty containers is a known tactic to avoid taxation or to smuggle prohibited items. 'Fake' declared-as-empty containers are characterised by the presence of cargo, which may be challenging to

ascertain due to inter-container appearance variation (e.g. geometry, damage, support structures). In addition, small object sizes and their concealment within the fabric of the container significantly increase the difficulty of the task. Only images classified as non-empty by this module are passed onto specific threat item detector modules, thus significantly reducing their workload given that around 20% of cargo containers are shipped empty.[28]

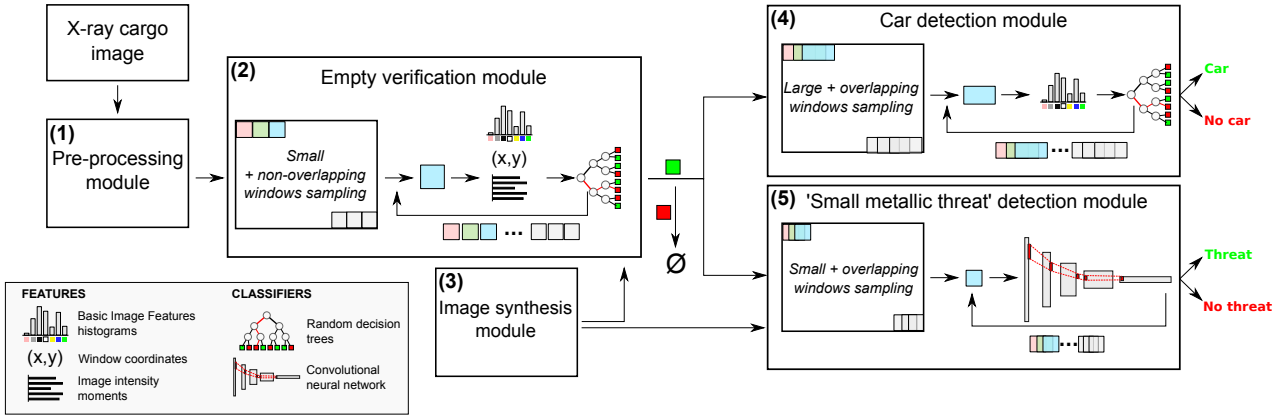

Figure 2. The framework for automated X-ray cargo image analysis includes the following modules: *empty verification*, *car detection*, and *'small metallic threat' detection*.

The first specific threat detection module determines whether X-ray cargo images contain at least one car. This module can be used to deter criminal activities involving cars transported in cargo containers, including tax evasion, export fraud, and trafficking.[29–31] Cars were chosen as a first target for the development of object detection algorithms as they are suitably difficult to detect, e.g. when disassembled or shielded by other cargo. The second threat item detection module targets 'small metallic threats', which pose serious and concrete security concerns. These threats usually occupy a very small part of the image and may be shielded by dense cargo, making them almost invisible to the Human eye. Unlike cars, 'small metallic threats' can be placed in any orientation, which adds further complexity. Both detection modules are proofs of concept and the presented algorithms can be trained to detect different threats or contraband, such as cigarettes, currency, or bottles of alcohol.

While machine learning techniques are employed across the framework, the specific algorithm used is customised for the module task. The following sections describe these algorithms in details.

4.1 Pre-processing module

Raw X-ray cargo images are first put through a pre-processing module, which (i) removes vertical or horizontal black image stripes due to faulty imaging detectors or due to source misfire, (ii) corrects for variations in the source intensity and the sensor responses,[27] (iii) sets isolated anomalous pixel values to the median of their neighbourhood, and (iv) extracts the container region from the image by performing a binary segmentation of the pixel intensity values. An example of pre-processing is shown in Fig. 3.

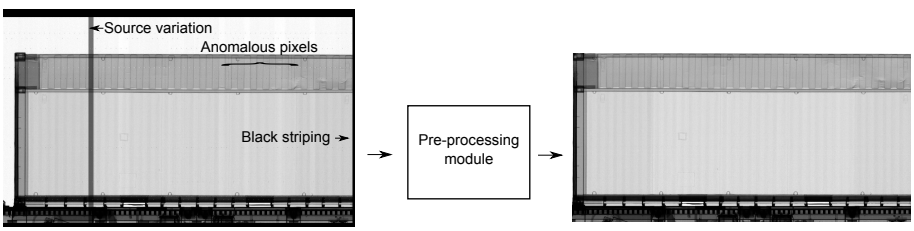

Figure 3. Example of a raw image (left) and the resultant image after pre-processing (right).

4.2 Image synthesis module

The X-ray image capture process approximately obeys the Beer-Lambert law

$$I_{xy} = I_0 \exp\left(-\int \mu_{xy}(z)dz\right), \tag{1}$$

where I_0 is the beam intensity, x are coordinates along the image horizontal, y are coordinates along the image vertical, z is coordinates along the photon path through the scene which has attenuation coefficient $\mu_{xy}(z)$.

Consider a scenario of an object placed inside a cargo container, the integral can be split into separate attenuation contributions from the container (C) and the object (O),

$$\begin{aligned} I_{xy} &= I_0 \exp\left(-\int_{\mathrm{C}} \mu_{xy}(z)dz\right) \exp\left(-\int_{\mathrm{O}} \mu_{xy}(z)dz\right), &(2)\\ &= I_0 C_{xy} O_{xy}, &(3)\end{aligned}$$

thus by estimating the contribution from the container $C_{xy} \in [0,1]$ and beam intensity I_0, one can obtain an attenuation mask $O_{xy} \in [0,1]$ for the object by computing $I_{xy}/(I_0 C_{xy})$. Now, we can project the object mask O_{xy} into other images by pixel-wise multiplication.

For cargo images, background estimation is often straightforward since the container is mostly uniform in the y-direction, and so a small background patch above the object can be interpolated over the object ROI to obtain C_{xy} as shown in Fig. 4(1). In more difficult cases, an object patch can be manually delineated to form a *threat mask*, and other non-background objects can be delineated to form an *ignore mask*. A simple background removal is achieved by (i) computing the mean of the patch pixels that are not in the *threat mask* or the *ignore mask*, (ii) divide each pixel in the threat mask by this mean, (iii) set all other pixels to 1. This process is shown in Fig. 4(2).

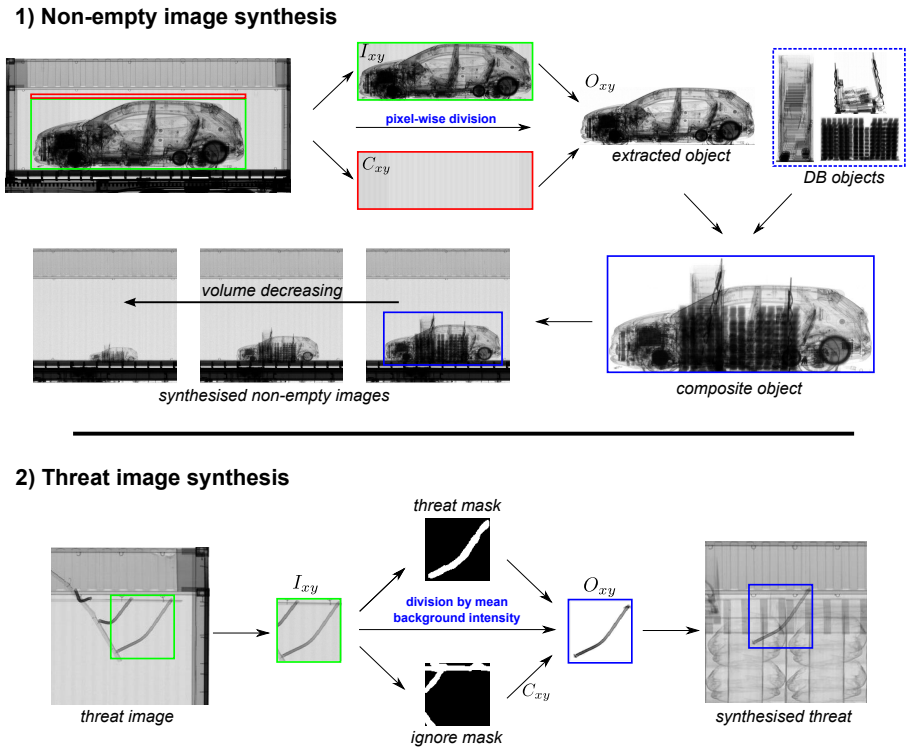

Figure 4. Illustration of the work-flow for synthesising (1) non-empty images with an inserted composite object with decreasing volume, and (2) a 'small metallic threat' image.

When training machine learning classifiers, it is often useful to robustify classification by adding realistic noise and variation to the training data. There are several ways in which this can be achieved;

1. *Object volume scaling:* scaling the object volume V by a factor ν, such that $V \rightarrow \nu V$. This is equivalent to scaling the in-plane area of the object by a factor $v^{2/3}$ and also scaling the out-of-plane thickness by a factor $v^{1/3}$ so that $O_{xy} \rightarrow (O_{x'y'})^{v^{1/3}}$, where $\{x', y'\}$ are the pixel indices inside the new $\nu^{2/3}$-scaled in-plane area.

2. *Object density scaling:* scaling of the object density ρ by a factor p, i.e. $\rho \rightarrow p\rho$. This implies that $O_{xy} \rightarrow (O_{xy})^p$.

3. *Object flips*: objects can be flipped in the x or y directions to increase appearance variation.

4. *Object composition:* given multiple extracted object attenuation masks, we can form a composite object by pixel-wise multiplying the masks together. We can additionally, scale the volume and density of the objects, magnify and rotate them, or translate them in x and y relative to each other, to further increase variation.

5. *Addition of noise:* patch intensity is scaled by a relative factor randomly picked between $\pm 5\%$.

6. *Background variation:* the object can be projected onto a wide range of different types of background.

We use image synthesis for training in both *empty verification* and *'small metallic threat' detection*. To be precise, for *empty verification* we use (1) and (2) to manipulate the difficulty of the loads so that the system can be trained on loads that are more difficult than the SoC. Objects are extracted from SoC images to form an object database. During image synthesis multiple objects are selected and then varied and composed using (1)-(4). Finally, the composite object is projected into an empty SoC image. It is important to include a large number and variety of different empty containers (6), and to project at a range of different (x, y) positions.

For *'small metallic threat' detection*, we use steps (5) and (6). A variety of different 'small metallic threats' are extracted from staged examples, where the threats are purposefully placed into containers for data collection. These examples are then projected onto a range of different SoC cargo images. The SoC cargo attenuation ranges from very high and thus very dark in the image, to very low and thus almost transparent in the image. Cargo texture can range from slowly varying and almost uniform (e.g. pallets of paper) to varying with high frequency (e.g. complex machinery).

The image synthesis module can also be considered a threat image projection (TIP) module useful for testing and training both human operators and machine learning algorithms.

4.3 Empty verification module

For *empty verification*, we consider a binary classification problem; is the container "empty" (negative class) or "non-empty" (positive class)? This problem is complicated by (i) the visible and inhomogeneous appearance of the container within the image, (ii) parts of the container, such as the corners and ribbing, locally appear similar to loads, (iii) the size and type of the container varies between images, (iv) detritus such as empty packaging, strapping and garment rails are often left in empty containers, and (v) cargo loads have a broad range of possible appearances i.e. anything that can be licitly or illicitly carried by container.

For *empty verification* we use histograms of oriented Basic Image Features (oBIFs) as a texture descriptor. Basic Image Features (BIFs) are a scheme for the classification of each pixel of an image into one of seven categories depending on local symmetries.[32] These categories are: "flat" (no strong symmetry), slopes (e.g. gradients), blobs (dark and bright), lines (dark and bright), and saddle-like. Oriented BIFs (oBIFs) are an extension of BIFs to include the quantized orientation of rotationally asymmetric features.[33] The computation of oBIFs relies on two parameters: σ dictates the scale of the features to detect (e.g. feature scale increases from fine to coarse as the value of σ increases) and γ the threshold below which a pixel is considered "flat". oBIF histograms encode a compact representation of images and have been applied successfully in many machine vision

tasks. Implementations for BIFs and oBIFs in `MATLAB` and `Mathematica` are available online.[34] After computing oBIFs, we efficiently construct 23-bin histograms using the integral histogram method.[35]

In our approach, we split the image into a grid of 96×96 pixel windows. Within each window, we compute (i) oBIFs at scales $\sigma = \{0.7, 1.4, 2.8, 5.6, 11.2\}$ and threshold $\gamma = 0.1$, (ii) window intensity moments up to 2nd order, and (iii) the coordinates of the window centre (x, y) within the image. The feature vector for each window is fed into a Random Forest (RF) classifier. A *window score* is computed based on the fraction of trees in the RF ensemble that vote that the window is "non-empty". To make a decision for the whole image, we take the *maximum* of the window scores and compare to a threshold.

Each feature type plays a distinct role in the window descriptor;

1. *oBIF histograms:* encode textural information for windows at different locations in the container. The container has a distinct, ordered, structure, and the smaller scale oBIFs, and their quantized orientation, help to encode this information. Due to the large variation in appearance of possible loads, it would be very difficult for the RF to learn all the different small-scale structures, thus large scale oBIFs prove useful, since they encode information about the larger scale image structure, rather than getting stuck in the intricate small-scale details. For example, we find that the large-scale minima-like BIF-type is assigned high importance by the RF since smaller loads tend to appear as dark blobs at larger scales.

2. *Intensity moments:* encode information about image intensity and its spatial distribution; information which is lost in the computation of oBIFs.

3. (x, y): allow the RF to implicitly learn the range of appearances at different locations in the container. For example, it is able to learn that empty windows near the top of the image should have roof-like appearance and that windows towards the bottom should have floor-like appearance. It does so without explicitly being told where the roof or floor are. Thus, we can avoid segmenting the container into separate regions (e.g. floor, walls, and roof) and training a separate classifier on each image region. We measure the x-coordinate from the nearest container end to account for its reflectional symmetry about the container centre. This effectively halves the number of x-locations that the RF needs to learn.

In the stream-of-commerce (SoC) dataset, it is relatively straightforward to classify containers, since cargo usually occupies a large proportion of the container. Performance on the SoC dataset does not adequately evaluate the ability of the classifier on more difficult scenarios, such as where criminals smuggle a small amount of contraband inside an otherwise empty container. To this end, we have developed a method of synthesising realistic examples from the SoC data (Sec. 4.2). Synthesising images in this way allows us to (i) obtain ground truth labels for individual windows which is useful in training and testing the classifier, (ii) control the volume and density of the load so that we can measure the performance of the system at different levels of difficulty, and (iii) to test on examples that are more difficult than those in the SoC.

The system was trained on synthesised non-empty examples and SoC empty examples. The densities and volumes of the training objects ranged from $0.2\,\mathrm{g/cm^3}$ to $1.75\,\mathrm{g/cm^3}$ and $0.001\,\mathrm{m^3}$ to $1.5\,\mathrm{m^3}$, respectively. In total 2.8×10^5 windows, with balanced empty and non-empty classes, were used to train the classifier. We randomly sample $11 \approx \sqrt{122}$ (where 122 is the dimensionality of the feature vector) variables at each split, and use 500 trees in the ensemble selected by finding the point where the out-of-bag error (versus number of trees) plateaus.

The system achieves state-of-the-art performance on the SoC dataset of 99.3% detection given 0.7% false positives (this outperforms Orphan *et al.*[15] who report an accuracy of 97.2% and a 0.4% FNR). This performance implies that the classifier has not significantly over-fitted to the synthetic examples, and that the synthetic examples are sufficiently realistic. Examples of SoC detections are shown in Fig. 5(1a-1d). The blue windows indicate where a window score exceeds the threshold tuned for *image classification*, and are a way of visualising where the classifier is highly confident that a load is present.

The system was also tested on synthesised images containing difficult objects. Fig. 5(1-4) show example results for SoC and synthesised images. Performance can be assessed for any object volume and density, but it is useful to pick those that are analogous to realistic smuggling scenarios, such as smuggling cocaine in an otherwise empty container. By choosing a density similar to cocaine ($\rho = 1.2\,\mathrm{g/cm^3}$) it is possible to vary the

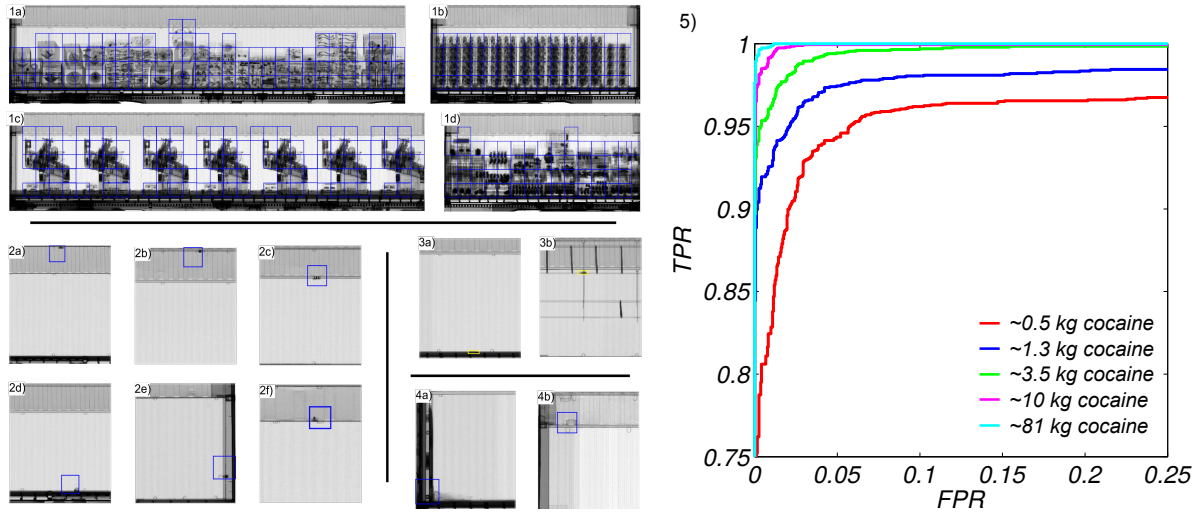

Figure 5. 1. Example true positives on SoC data. 2,3: Synthetic images with object of similar volume and density to 1 L of water. 2: True positives. 3: False negatives, missed because of alignment with container structures. 4: False positives; a is detritus, b is damage. 5: ROC curves for different masses of cocaine.

object volume and determine receiver operating characteristic (ROC) curves for different masses of cocaine, as shown in Fig. 5(5). As the cocaine mass gets smaller, the task becomes more challenging, and the ROC curve gets pushed towards lower true positive and false positive rates. For 1.25 kg of cocaine, the system achieves 93.0% detection, while triggering 1% false alarms on empty containers. Cocaine smugglers typically smuggle between 8 kg and 514 kg of cocaine,[2] so the system performance in this context is at a useful level.

4.4 Car detection module

The *car detection* module (Fig. 2) classifies the input image as "car" (contains at least one car, positive class) or "non-car" (does not contain a car, negative class). Rectangular 350×1050 pixel sub-windows are densely sampled with a 40 pixel stride. Window features are computed and fed to a RF classifier, which assigns a score to each window. The image score is then computed as the *maximum* of the window scores. Binary classification of the image is finally obtained by comparing the image score with a threshold value.

Image feature computation is a key aspect of this classification scheme. The simplest type of features evaluated for the classification of windows as "car" or "non-car" were histograms of raw pixel intensities that encode the distribution of intensity within a window. These histograms can thus be used to discriminate between windows of different contrasts, e.g. between container background and industrial machinery. Intensity histograms are convenient features due to their simplicity and efficient computation by the integral histogram method.[35] We also evaluated oBIFs as texture features for classification.

The dataset included 79 car images, for a total of 192 cars, and 20,000 non-car images. Due to the relatively low number of car images, leave-one-out cross-validation (LOOCV) was employed to evaluate the generalisation of the method to unseen car images. A hold-out validation scheme using all car images and 10,000 non-car images enabled the determination of the false alarm rate on previously unseen images. Performance was summarised by computing the H-measure, an alternative to the traditional area-under-the-curve (AUC) that was shown to better accommodate unbalanced datasets.[36]

Using intensity features only, the best performance obtained was an H-measure close to 0.9. In contrast, oBIFs histograms computed at scales $\sigma = \{0.7, 1.4, 2.8, 5.6, 11.2\}$ and $\gamma = \{0.011, 0.1\}$ resulted in a much improved H-measure of 0.991, which corresponds to a 100% *car detection* rate for a false alarm of only 0.41%. Images of cars on their own (Fig. 6.a), adjacent to other cargo (Fig. 6.b), and obscured by other cargo (Fig. 6.c and d) were thus correctly detected. These results demonstrate the potential of the proposed scheme for object detection in X-ray cargo images. Indeed, the level of performance achieved makes it suitable for implementation in the field.

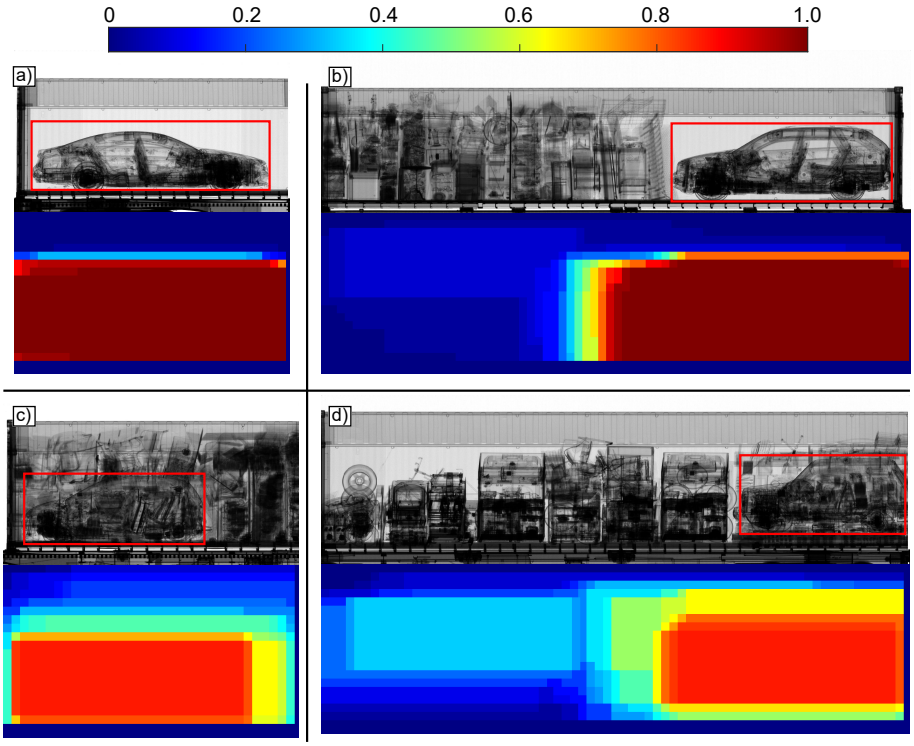

Figure 6. Examples of car classification outputs. For each panel, the top half is the raw X-ray transmission image overlaid with the manually annotated region of interest in red, the bottom half is a visualization of the classification scores at various location of the image (blue and red indicate high confidence that the region is background and car, respectively.

4.5 'Small metallic threat' detection module

The detection of 'small metallic threats', even more so than *car detection* and *empty verification*, is a challenging machine vision task for the following reasons: i) variations in size (i.e. from the equivalent of a mobile phone to that of large industrial tools), ii) unconstrained orientation, iii) large number of models and manufacturers, iv) lack of discriminative visual features (i.e. their appearance is close to that of legitimate cargo and container structures). The classification scheme employed in the *car detection* and *empty verification* modules (classification by RF based on various fixed features, including oBIFs) performed poorly for this task. Instead, a different approach was taken with the implementation of a classification scheme based on convolutional neural networks (CNNs).

CNNs are part of a family of representation-learning algorithms commonly referred to as Deep Learning.[37] As the name suggests, representation-learning consists in determining a representation of the data that optimise the task (classification or regression). Thus, instead of relying on generic features such as oBIFs, CNNs attempt to learn task-optimal features from labelled example images. It is well-known that for artificial neural networks to perform well on image data, it is necessary to process small pixel neighbourhoods rather than individual pixels.[38] Connections between the layers of a network should thus be from a neighbourhood of pixels to another, which preserves the two-dimensional nature of images. Moreover, the weights dictating these connections should be the same across the entire image, as it is likely that features performing well in one location will also do so everywhere else (i.e. translation invariance). In CNNs, those notions of weight sharing and local connectivity are enforced by introduction of convolutional layers where the mappings from inputs to ouputs are computed based on the convolution operator.[39] Together with convolutional layers, rectified linear units and regularisation techniques (e.g. dropout[40]) define the building blocks of CNN architectures that are becoming the state-of-the-art in many applications (e.g. face verification), even surpassing human performance for some tasks such as large-scale classification of natural images.[41]

Even more so than other machine learning approaches, deep learning requires a very large number of examples

Table 1. 'Small metallic threat' patch classification performance (only considering backgrounds of medium to low density). CNNs are compared to our oBIFs + RF scheme.

Method	H-measure	FPR for 90% TPR [%]
oBIFs + RF	0.63	21.0
CNN (9-layer)	0.89	0.92
CNN (19-layer)	**0.91**	**0.80**

in order to obtain high performance and avoid overfitting issues. Due to the rarity of 'small metallic threats', it is not possible to rely on SoC images for training. One potential solution would be the acquisition of images in a controlled setting where threats would be concealed within cargo in order to re-create as accurately as possible the conditions of real smuggling occurrences. However, this process is too time-consuming and costly to generate a sufficiently large number of training examples. A solution to this data bottleneck is the generation of synthetic examples that closely mimic real data. This strategy has been successfully employed to train deep networks to recognise text[42] or predict the trajectories of falling wooden blocks.[43] Using the technique described in Sec. 4.2, an arbitrary number of threat images can be generated from a library of threat objects and randomly sampled SoC backgrounds, making it possible to train CNNs to detect 'small metallic threats' in X-ray cargo images.

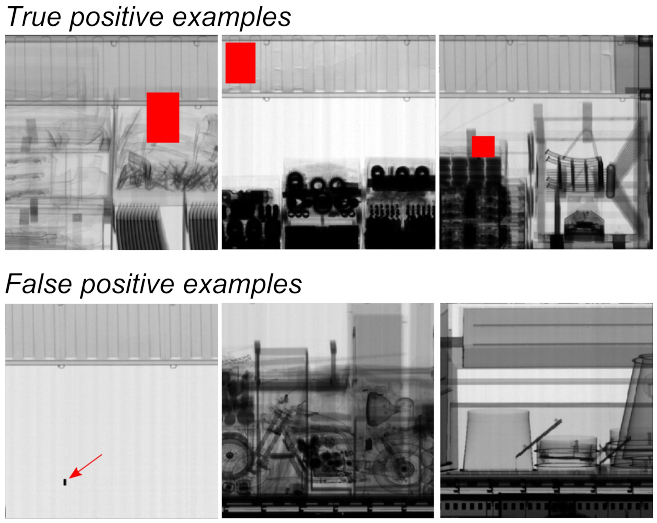

Figure 7. Example true and false positive detections by the *'small metallic threat' detection* module. Red rectangles obfuscate the precise nature of the 'small metallic threat' we have studied. The arrow points to an imaging artefact that raised a false alarm.

As for *car detection* and *empty verification*, the classification scheme for *'small metallic threat' detection* is window-based (Fig. 2). 512×512 pixel windows are densely sampled from the input image with a 32 pixel stride, before being down-sampled to 256×256 pixels to make the training of the CNN computationally tractable. The CNNs evaluated are 9-layer and 19-layer architectures with small 3×3 filters that were shown to produce state-of-the-art results in other applications.[44] The networks are trained from scratch based on 12,000 SoC patches for the negative class (no threat) and 12,000 synthetic threat patches for the positive class. Due to the preliminary nature of this work, only patch classification performance will be presented here, not whole-image classification. In addition, challenging adversarial examples where threats are concealed by very dense cargo are not considered.

Both the 9-layer and 19-layer CNNs outperformed significantly our scheme based on oBIFs and RF for 'small metallic threat' patch classification (Table 1). The deeper CNN architecture performed the best, achieving a false positive rate of 0.80% for a detection rate of 90%. The proposed scheme correctly detected threats concealed within legitimate cargo of low to medium density (Fig. 7).

5. CONCLUSION

The inspection of X-ray cargo images is a challenging visual search task akin to finding a needle in a haystack. The majority of images that an operator is tasked with inspecting do not contain an anomaly, thus wasting time and money. In this report, we presented a modular framework based on modern machine vision and learning techniques that aims to assist security officers by partially automating the inspection process. By automatically sifting through large numbers of images, the proposed system would enable security officers to focus their attention on images that are likely to be anomalous, thus easing the inspection time constraint and making sure that the security infrastructure can be scaled-up as necessary with the ever-increasing volume of commerce.

We demonstrated state-of-the-art performance for three tasks: i) *car detection*, ii) *empty container verification*, and iii) *'small metallic threat' detection*. In order to achieve these results, it was necessary to introduce several new methods such as location-specific appearance learning by providing the classifier with window coordinates, and threat image projection for the generation of *de-novo* synthetic training examples. To our knowledge, the proposed *'small metallic threat' detection* module is the first application of convolutional neural networks to X-ray cargo images.

Despite these achievements, it is clear that much remains to be done. Future efforts will be focused on obtaining larger and more diverse datasets as well as on refining the algorithms described in this contribution. For instance, it is likely that the method that was used to infer an image class based on a collection of window scores, essentially by thresholding the maximum window score, was not optimal. Similarly, the window-based scheme could potentially be replaced by per-pixel predictions using fully convolutional neural networks.[45]

Acknowledgement

The authors acknowledge the use of the UCL Legion High Performance Computing Facility (Legion@UCL), and associated support services, in the completion of this work. Funding for this work was provided by EPSRC Grant no. EP/G037264/1 as part of UCL's Security Science Doctoral Training Centre, and Rapiscan Systems Ltd.

REFERENCES

[1] The World Bank, "World development indicators." Data file available at: http://data.worldbank.org/indicator/IS.SHP.GOOD.TU/ (2016).

[2] European Commission, "Good practice guide for sea container control, ch. 6 concealment methods." Available at: http://ec.europa.eu/taxation_customs/elearning/demo/container/library/GPG/chapter_6_Concealment_methods.pdf (2002).

[3] Wolfe, J. M., Brunelli, D. N., Rubinstein, J., and Horowitz, T. S., "Prevalence effects in newly trained airport checkpoint screeners: trained observers miss rare targets, too.," *Journal of vision* **13**, 33 (jan 2013).

[4] Zhang, J., Zhang, L., Zhao, Z., Liu, Y., Gu, J., Li, Q., and Zhang, D., "Joint Shape and Texture Based X-Ray Cargo Image Classification," in [*2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*], 266–273, IEEE (June 2014).

[5] Baştan, M., Yousefi, M. R., and Breuel, T. M., "Visual words on baggage x-ray images," in [*Proceedings of the 14th International Conference on Computer Analysis of Images and Patterns - Volume Part I*], *CAIP'11*, 360–368, Springer-Verlag, Berlin, Heidelberg (2011).

[6] McDaniel, F. D., Doyle, B. L., Vizkelethy, G., Johnson, B. M., Sisterson, J. M., and Chen, G., "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).

[7] Abidi, B., Page, D., and Abidi, M., "A Combinational Approach to the Fusion, De-noising and Enhancement of Dual-Energy X-Ray Luggage Images," in [*2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*], **3**, 2–2, IEEE (2005).

[8] Woodell, G., Rahman, Z.-u., Jobson, D. J., and Hines, G., "Enhanced images for checked and carry-on baggage and cargo screening," in [*Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005, Proceedings, Part II*], Carapezza, E. M., ed. (Sept. 2005).

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

[10] Ogorodnikov, S. and Petrunin, V., "Processing of interlaced images in 410 MeV dual energy customs system for material recognition," *Physical Review Special Topics - Accelerators and Beams* **5**, 104701 (Oct. 2002).

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

[12] Rogers, T. W., Jaccard, N., Morton, E. J., and D., G. L., "Detection of cargo container loads from x-ray images," in [*The IET Conference on Intelligent Signal Processing (ISP 2015)*], (2015).

[13] Chalmers, A., "Automatic high throughput empty ISO container verification," *Proc. SIE* **6540**(65400Z), 1–4 (2007).

[14] Chalmers, A., "Cargo identification algorithms facilitating unmanned/unattended inspection at high throughput terminals," in [*Proc. SPIE*], **6736**, 67360M–67360M–6 (2007).

[15] Orphan, V. J. et al., "Advanced γ ray technology for scanning cargo containers," *Appl. Radiat. Isot.* **63**, 723–732 (2005).

[16] Tuszynski, J., Briggs, J. T., and Kaufhold, J., "A method for automatic manifest verification of container cargo using radiography images," *J. Transp. Secur.* **6**(4), 339–356 (2013).

[17] Riffo, V. and Mery, D., "Automated Detection of Threat Objects Using Adapted Implicit Shape Model," *IEEE Trans. Syst. Man Cybern.* , 1–11 (2015).

[18] Mery, D. et al., "Detection of regular objects in baggage using multiple X-ray views," *Insight Non-Destructive Test. Cond. Monit.* **55**(1), 16–20 (2013).

[19] Riffo, V. and Mery, D., "Active X-ray Testing of Complex Objects," *Insight - Non-Destructive Test. Cond. Monit.* **54**(1), 28–35 (2012).

[20] Flitton, G., Mouton, A., and Breckon, T. P., "Object classification in 3D baggage security computed tomography imagery using visual codebooks," *Pattern Recognit.* **48**(8), 1–11 (2015).

[21] Mouton, A. et al., "3D object classification in baggage computed tomography imagery using randomised clustering forests," *IEEE Int. Conf. Image Process.* , 5202–5206 (2014).

[22] Flitton, G., Breckon, T. P., and Megherbi, N., "A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery," *Pattern Recognit.* **46**(9), 2420–2436 (2013).

[23] Flitton, G., Breckon, T., and Megherbi Bouallagu, N., "Object Recognition using 3D SIFT in Complex CT Volumes," in [*Procedings Br. Mach. Vis. Conf. 2010*], 11.1–11.12 (2010).

[24] Bastan, M. et al., "Visual Words on Baggage X-Ray Images," in [*Comput. Anal. Images*], (Figure 1), 1–7 (2011).

[25] Batan, M., "Multi-view object detection in dual-energy X-ray images," *Mach. Vis. Appl.* **26**(7), 1045–1060 (2015).

[26] Schmidt-hackenberg, L. et al., "Visual cortex inspired features for object detection in X-ray images," in [*Int. Conf. Pattern Recognit.*], (Icpr), 2573–2576 (2012).

[27] Rogers, T. W., Ollier, J., Morton, E. J., and Griffin, L. D., "Reduction of wobble artefacts in images from mobile transmission x-ray vehicle scanners," in [*Imaging Systems and Techniques (IST), 2014 IEEE International Conference on*], 356–360 (Oct 2014).

[28] Dong, J.-X. and Song, D.-P., "Container fleet sizing and empty repositioning in liner shipping systems," *Transportation Research Part E: Logistics and Transportation Review* **45**(6), 860 – 877 (2009).

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

[30] Clarke, R. V. and Brown, R., "International trafficking in stolen vehicles," *Crime and Justice* , 197–227 (2003).

[31] Clarke, R. V. and Brown, R., "International Trafficking of Stolen Vehicles," in [*International Crime and Justice*], Natarajan, M., ed., ch. 16, 126–132, Cambridge University Press (2010).

[32] Griffin, L. D., Lillholm, M., Crosier, M., and van Sande, J., "Basic image features (bifs) arising from approximate symmetry type," in [*Scale Space and Variational Methods in Computer Vision*], Tai, X.-C., Mrken, K., Lysaker, M., and Lie, K.-A., eds., *Lecture Notes in Computer Science* **5567**, 343–355, Springer Berlin Heidelberg (2009).

[33] Newell, A. J. and Griffin, L. D., "Natural Image Character Recognition Using Oriented Basic Image Features," in [*2011 International Conference on Digital Image Computing: Techniques and Applications*], 191–196, IEEE (Dec. 2011).

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

[35] Porikli, F., "Integral Histogram : A Fast Way to Extract Histograms in Cartesian Spaces," (2005).

[36] Hand, D. J. and Anagnostopoulos, C., "A better Beta for the H measure of classification performance," (Feb. 2012).

[37] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature* **521**(7553), 436–444 (2015).

[38] LeCun, Y. et al., "Generalization and network design strategies," *Connections in Perspective. North-Holland, Amsterdam* , 143–55 (1989).

[39] Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y., "What is the best multi-stage architecture for object recognition?," in [*Computer Vision, 2009 IEEE 12th International Conference on*], 2146–2153, IEEE (2009).

[40] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research* **15**, 1929–1958 (jan 2014).

[41] He, K., Zhang, X., Ren, S., and Sun, J., "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," (feb 2015).

[42] Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A., "Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition," (jun 2014).

[43] Lerer, A., Gross, S., and Fergus, R., "Learning Physical Intuition of Block Towers by Example," (mar 2016).

[44] Simonyan, K. and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," (sep 2014).

[45] Long, J., Shelhamer, E., and Darrell, T., "Fully convolutional networks for semantic segmentation," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 3431–3440 (2015).