DCNNs: A Transfer Learning comparison of Full Weapon Family threat detection for Dual-Energy X-Ray Baggage Imagery

Ashley Williamson[1]
ashley.williamson@hull.ac.uk
Patrick Dickinson[2]
pdickinson@lincoln.ac.uk
Tryphon Lambrou[2]
tlambrou@lincoln.ac.uk
John C. Murray[1]
john.murray@hull.ac.uk*

[1] Department of Computer Science and Technology
University of Hull
Hull, UK
(* Corresponding Author)

[2] Department of Computer Science
University of Lincoln
Lincoln, UK

Abstract

Recent advancements in Convolutional Neural Networks have yielded super-human levels of performance in image recognition tasks [13, 25]; however, with increasing volumes of parcels crossing UK borders each year, classification of threats becomes integral to the smooth operation of UK borders. In this work we propose the first pipeline to effectively process Dual-Energy X-Ray scanner output, and perform classification capable of distinguishing between firearm families (Assault Rifle, Revolver, Self-Loading Pistol, Shotgun, and Sub-Machine Gun) from this output. With this pipeline we compare recent Convolutional Neural Network architectures against the X-Ray baggage domain via Transfer Learning and show ResNet50 to be most suitable to classification - outlining a number of considerations for operational success within the domain.

1 Introduction

Dual-Energy X-Ray scanning systems are ubiquitous in border security applications, and pose a substantial challenge for automation - requiring trained officers for successful operation. These technologies are employed for a wide range of logistical solutions for passenger, commercial, industrial baggage and parcel services. With an ever increasing volume of parcels, systems are put under pressure to classify complex contents in shorter time-spans for detection of threats.

In recent years, significant advancements have been made in the field of Object Classification and Detection, specifically through the yearly ImageNet(ILSVRC) competition [25]. Whilst ILSVRC is designed for general object classification, there has been little work applying such advancements specifically to the security domain.

Existing work towards Dual-Energy X-Ray baggage object detection focuses on traditional feature extraction, segmentation, enhancement, and detection algorithms to facilitate human

operators in the interrogation of baggage imagery. Turcsany *et al* [31] demonstrate a Visual Bag-of-Words model applied to 2D pseudo-colour images using DoG, DoG+SIFT, and DoG+Harris feature representations, with expansions [5] on such work focusing on the use of SURF [6] and SVM Classifiers - yielding improved classification results due to a large diverse dataset. In addition, Flitton *et al* [11] propose 3D Computed Tomography (CT) imagery solutions extending on 2D methods via a combination of 3D Feature Descriptors - Density Histogram(DH), Density Gradient Histogram(DGH), SIFT, and Rotation Invariant Feature Transform(RIFT). Kechagias-Stamatis *et al*[17] outline a proposed pipeline relying on local feature extraction via SURF features, utilising soft and hard clustering. Further work has looked at enhancing image output as a means of improving object detection [7]. Akçay and Breckon [2] compare transfer learning within the domain of X-Ray Threat Detection on a limited-scope dataset comprised of disparate threats with various mechanisms such as Sliding Window CNN, and recent region proposal-based architectures concluding these approaches to be superior to hand-crafted features. Akçay *et al*. [4] continues this work - outlining datasets labelled Dbp_2 and Dbp_6 for firearm-not-firearm and mutli-class firearm/threat classification respectively - whereby classification and detection mechanisms are compared for both these datasets and classification is performed on Full-Firearm vs Operational Benign (FFOB) and Firearm Parts vs Operational Benign (FPOB); confirming application of Convolutional Neural Networks to outperform hand-crafted features. However [2, 4] include objects such as guns, knives, laptops as 'threat' objects when performing classification.
Akçay *et al*. [3] compare the depth of representation freezing, when transfer learning, against accuracy with a pre-trained AlexNet[18] model, showing benefits when freezing layers 1-3. To the best of our knowledge, we are the first to consider various Deep Convolutional Neural Network models, including more recent models, for the application of transfer learning to this problem via a direct-from-scanner approach - where our dataset preprocessing enables us to produce classification directly from X-Ray Scanner Output, on a dataset constructed of 5 similar firearms of distinct families.

1.1 Convolutional Neural Networks & Transfer Learning

Deep Convolutional Neural Networks have been applied to a host of domains since their inception, including Video classification [16], Reinforcement Learning [19], Natural Language Processing [10], and in recent years have surpassed human-level performance in image recognition tasks [13, 26].
These networks provide a means of deeper image representation, where initial layers represent basic image features such as edges or boundaries, with further layers providing more abstract representations such as faces; dependent upon the training dataset [32]. These representations are then combined with fully-connected layers to weight which features contribute towards the correct classification of a given class - often utilising softmax to provide class probability outputs.

Successful classification typically relies on substantial numbers of training examples to learn from, with ILSVRC containing upwards of 14 millions images over 1000 classes - providing sufficient information to train CNNs from scratch. Evolution of Neural Network architectures are producing more accurate classification accuracies on ILSVRC challenges, yet for domains where training examples are scarce, or expensive to obtain, training from scratch can be problematic or may lack sufficient data to adequately produce a model. Transfer Learning [23] exploits the innate ability of CNNs to produce feature abstraction, and

applies this to a new domain not originally trained on, the *target* domain. This technique has become popular across difficult training domains, and has been shown to work within detection scenarios [21, 27]. Transfer Learning involves taking the weights of a given architecture, trained to a high degree of accuracy on an *existing* domain, and initialising a new model with those same weights for a different domain, the *target*. This approach significantly reduces training times by bootstrapping learning, and on occasion, prohibiting backpropagation into the earlier layers, focusing only on the final layers - fine-tuining. A variation upon this approach freezes a sub-set of the convolutional layers, enabling fine-tuning of the mid to high-level features [22]. Chollet [3] states that training required 3 days on the original ILSVRC-2012 dataset, utilising 60 K80 GPUs; additionally Simonyan and Zisserman [28] reported 3-4 weeks of training on NVidia Titan Black GPUs depending on the variant of their architecture used. With Transfer Learning we can re-use the knowledge of these original domains, and adapt them for Dual-Energy X-Ray Imagery within fractions of the time; when compared against training a CNN from random initialisation.

2 Experimental

2.1 Dataset

We utilise a novel dataset provided by the Home Office's Centre for Applied Science and Technology (CAST), consisting of false-colour images of baggage items, where higher atomic weights are represented via blue hues, corresponding to metallics, and orange hues represent lower atomic weights, such as organic material; with greens being a mix of organic and in-organic materials (See Figure 1). Data is comprised of fullweapon examples only, and represents the following classes: *assault rifle*, *revolver*, *self-loading pistol*, *shotgun*, and *sub-machine gun* with 2160 positive examples across all classes; containing 450, 450, 450, 360, and 450 examples per class, respectively. Each image belongs to an *imagegroup*, where members of an *imagegroup* correspond to the same physical baggage being scanned from multiple viewpoints; these include top-down, side-view, and $\pm$ 45 oblique, dependent upon manufacturer. These are split into training and testing example sets, whereby no imagegroup is bisected, with 70-30 ratio maintaining class distribution consistent across the set boundary. It is worth noting that no selective filtering is done upon the dataset to remove erroneous images, examples of which include distortion or empty images during image acquisition. Image labels were provided as-is from CAST via metadata related to each file. Final training set contains 1524 fullweapons, with 318, 318, 318, 252, 318 examples over respective classes; with the testing set containing 132, 132, 132, 108, 132 examples respectively.

Prior works have only sought to address a binary gun-not-gun problem, or a 6-class multi-object problem. Our dataset includes more difficult cases where differences between classes represent fundamental differences between specific gun families; whereby overlap of features will be commonplace. In addition, our dataset includes significantly fewer examples for this task.

To our knowledge, we are the first to consider sub-classes of firearm classification in this context, specifically in an end-to-end manner.

Algorithm 1 Maximal information bounding

Require:

$function\ inRange(i,l,u)-$ produces a 0, 1 output if a given pixel lies between the lower-bound, l, and the upper-bound, u.

$matrix\ J_n-$ unit matrix of nxn, composed of values 1.

$function\ hsv(im_{bgr})-$ converting im_{bgr} into HSV Colour Space.

$function\ boundingRect(mask)-$ calculate minimum up-right bounding rectangle of non-zero elements of *mask*.

$function\ centroid(mask)-$ calculate the centroid of the given mask.

$function\ padd(image,top,bottom,left,right)-$ Pads the provided image with whitespace, by the amount specified in the given four directions.

$B_{min} = (90,\ 100,\ 100)$
$B_{max} = (180,\ 255,\ 255)$
$images = \{im_0,\ im_1,\ \ldots,\ im_N\}$
$c = \{c_0,\ c_1,\ \ldots,\ c_N\}$
$meanWindow = [0,0]$
$count = 0$
for $im_{hsv} \leftarrow hsv(im_{bgr}) \in images$ **do**
    $mask_{hsv} \leftarrow inRange(im_{hsv}, B_{min}, B_{max})$
    $morphMask_{hsv} \leftarrow (mask_{hsv} \ominus J_3) \bullet J_{10}$
    $c_{im_{hsv}} \leftarrow centroid(morphMask_{hsv})$
    $bRect \leftarrow boundingRect(morphMask_{hsv})$
    $meanWindow \mathrel{+}= \frac{1}{counter+1} \cdot (bRect - meanWindow)$
    $counter = counter + 1$
end for
for $im_{hsv} \leftarrow hsv(im_{bgr}), c_{im_{hsv}} \in images, c$ **do**
    $bounds_x \leftarrow (c[0], c[0] + meanWindow[0])$
    $bounds_y \leftarrow (c[1], c[1] + meanWindow[1])$
    $padded_{hsv} \leftarrow padd($
        $image = im_{hsv},$
        $top = \lfloor \frac{meanWindow[1]}{2} \rfloor,$
        $bottom = \lceil \frac{meanWindow[1]}{2} \rceil,$
        $left = \lfloor \frac{meanWindow[0]}{2} \rfloor,$
        $right = \lceil \frac{meanWindow[0]}{2} \rceil$
    $)$
    $final \leftarrow im_{hsv}[bounds_x[0] : bounds_x[1], bounds_y[0] : bounds_y[1]]$
    $save(resize(final, \frac{1}{2}))$
end for

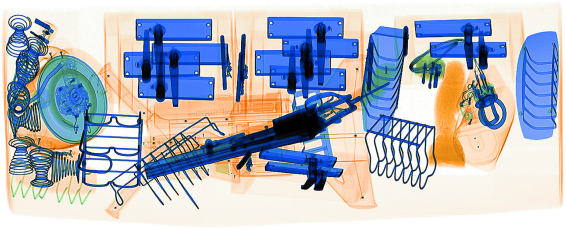
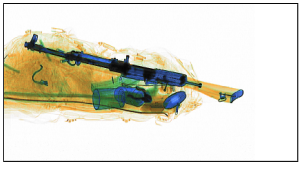
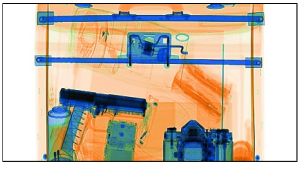


Figure 1: Example of false colour representation of Dual-Energy X-Ray Imagery.

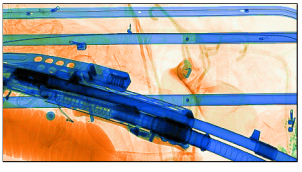


(a) Assault Rifle

(b) Revolver



(c) Self-Loading Pistol



(d) Shotgun

(e) Sub-Machine Gun

Figure 2: Training example images after maximal information windowing from 5 full-weapon categories. Prior to shorter-side cropping.

2.1.1 Preprocessing

Preprocessing consists of taking an output image from an X-Ray Scanner and processing it ready for interpretation by the Convolutional Neural Network. The same steps taken here apply for construction of the training dataset, as well as preprocessing of new images for inference only. Preliminary HSV slicing, between $(H = 90,\ S = 100,\ V = 100)$ and $(H = 180,\ S = 255,\ V = 255)$, to highlight high *Effective Atomic Weight* (Z_{eff}) values, is performed to segment metallic responses; positive threats within the dataset have high metallic components. Secondly, morphology operations reduce any smaller erroneous responses, as well as emphasise and focus on the primary cluster of high-response; representing the actual threat. From this we denote centroid locations, and bounding boxes of responses in order to calculate a *mean response window*, for which our network will be shaped to. The intuition behind our approach is that high metallic responses will contain the maximal information from the sample, and thus creating a minimum bounding box around these responses will result in the highest likelihood of threat detection contributing to learning. This process is outlined in Algorithm 1. As Convolutional Neural Networks containing fully-connected layers require a fixed input size, it is important to choose an appropriate input size; we chose the mean window response as an indication of aspect ratio - later resizing by $\frac{1}{2}$ to reduce

memory usage and complexity for processing. Examples of preprocessing output can be seen in Figure 2.

As data provided consists of Multi-View Dual-Energy X-Ray images of baggage, it is important to ensure that those images which represent the same physical specimen be grouped such that they entirely lie within either the training set, or the test set; due to high similarity between images of the same image group. Therefore we employ an image group split mechanism as a means of ensuring our training-test split is as close to ideal as possible - 70-30 training-testing split. We maintain class balance over the sets via this process, such that the distribution amongst classes pre-split is as close as possible to the post-split, whilst still adhering to imagegroup boundaries.
After splitting, the training set contains 1524 fullweapons, with 318, 318, 318, 252, 318 examples over respective classes; with the testing set containing 132, 132, 132, 108, 132 examples respectively. To utilise this dataset with the original networks we perform shorter-side cropping to the two modes of input dimension, 224x224, or 299x299, when feeding the network.

2.2 Framework

To enable a direct comparison, an evaluative framework was developed which encapsulates each specific network, acting as an interface for standard training/testing operations. These include *building*, *training*, *testing*, *loading*, and *saving* each network. Tensorflow [1], and Keras [8] were used to realise this framework, with Keras providing a substantial number of the models with existing weights trained from the ILSVRC domain. AlexNet [18] was originally under the Caffe framework [15], with the architecture obtained from Tensorflow/Models Github [30] for Tensorflow with a conversion of the original weights being provided by Michael Guerzhoy [12]. All other pre-trained weights were provided via Keras implementations.
Models selected for comparison include AlexNet [18], VGG19 [28], ResNet50 [14], InceptionV3 [29], and Xception [9]. We use colloquial nomenclature to enable reproducibility and linking between implementation and theory; where VGG19 is equivalent to VGG Model D, and ResNet50 is a Residual Network of length 50.

2.3 Training

Each model is built following the architecture outlined by their respective implementations, whereby we perform shorter-side cropping of either 224x224 or 299x299, centrally resizing to the target dimensions. We re-implement a standard *top-layer* on-top of each convolutional neural network for the given classification task, consisting of ReLU [20] activation functions, terminated by a softmax output. We apply a stop mechanism between the convolutional layers and the redefined *top-layers* preventing any gradient calculation being propagated backwards and modifying the weights of the earlier layers of the networks; facilitating faster learning by reducing the number of trainable parameters calculated.

From the Model definition we use a Stochastic Gradient Descent Optimiser with $lr = 1^{-3}$, $momentum = 0.9$, and $decay = 1^{-4}$, with $batchsize = 64$ for all models. Batching is done by randomly sampling from the given set, without replacement. Each epoch represents a processing of all batches from the dataset. AlexNet model parameters [12] are loaded via

TensorFlow, with top-layer weights and biases being initialised via truncated normal distribution with $\mu = 0.0$, and $\sigma = 0.001$. Remaining models are initialised using ImageNet weights provided by Keras for Convolutional Layers, with custom top-layer weights being randomly initialised via *glorot uniform distribution*(Xavier uniform distribution), and zero initialised bias units - as default.

Whilst training we use early stopping, such that if k consecutive epochs loss value does not improve (minimise) we halt training and return the model with the lowest loss. We denote an $upperlimit = 3000$ as our absolute upper-bound on number of epochs to train, and use $k = 50$ for stopping.

Each model is trained on dual Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz CPUs, 8x16GB Samsung DDR4 Registered DIMMs @ 2667 MT/s, with a single NVidia Titan XP GPU with 3840 CUDA cores, running TensorFlow 1.4.0-rc1 compiled from source. Models were trained in parallel, each with their own dedicated card; however sharing system resources for CPU and RAM.

3 Results & Discussion

Average Inference time per image was calculated based on 500 iterations of the test-set, obtaining the time over all epochs, averaging over this sum, followed by division of number of test samples within an epoch to obtain the average image response time. For Sensitivity and Specificity calculations, these were conducted on a one-vs-all approach for each model, calculated from the generated confusion matrices of each model.

Threat Detection algorithms do not work alone, and are typically part of a larger system; This, in combination with an increasing volume of parcels and baggage being processed by X-Ray scanning equipment, places a large emphasis on minimising processing time whilst maintaining accuracy for successful operation. In order to evaluate the usefulness of each network tested, for the domain of Threat Detection for Dual-Energy X-Ray systems, we propose the consideration of the following criteria: **a)** retrainability, **b)** high accuracy, **c)** reduced parameters, and **d)** low inference.

The ability of deep learning to learn complex visual problems combined with a reduced time-response to retraining are advantageous in a domain where the threat landscape is ever-changing. The system must be robust to these introductions, and be able to quickly be redeployed promptly following identification and acquisition of new threat information.

Detection of threats at border control has a direct impact upon the safety of the population, the ability for the approach to classify weapons to a high accuracy is important and should be considered safety critical; with misclassification or omission resulting in severe consequences.

Reduced parameter count enables more images to be processed simultaneously by a single GPU, prompting larger scan volumes due to reduced number of operations required. Fewer parameters by the model need to be stored on the GPU, more room is freed up to dedicate to data processing. In addition, fewer trainable parameters directly influences the time taken to train the model sufficiently, as fewer gradients need to be calculated in a single backpropagation pass. If the model can be initialised and ran utilising less GPU memory, it directly results in cheaper implementation costs; using existing consumer-grade hardware within scanning equipment.

As threat detection solutions do not operate in isolation but in tandem, low inference times are essential to ensure that the impact of the threat detection pipeline as a whole is not im-

peded; if classification cannot be performed in a timely manner this can cause reduction in throughput of border control and distribution centres, and overall disruption.

From the overall results (See table 1) it can be seen that newer architectures have a trend towards fewer parameters with the most recent, Xception, leading in this category. The architecturally simpler networks of AlexNet, and VGG19 lend themselves to lower training times, due in-part to their low inference times allowing higher throughput. We found consecutive stopping criteria to be the most effective when applying transfer learning, as the loss function was relatively smooth - PQ Early-Stopping [24] was designed with more noisy functions in mind, and was therefore not beneficial in our scenario and thus discarded. With our previously defined stopping criteria (See section 2.3) mechanism we achieve a best training of 26.3 minutes for VGG19, with Xception taking 814.9 minutes of training. Whilst the overall stopping time for ResNet50 is denoted as 111.47 minutes (See table 1), the test-set accuracy plateaus relatively quickly (See figure 3) showing the reported training time as an upper-bound, where highest-accuracy models, are saved and output significantly earlier in the training process. With reference to Figure 3, training time for ResNet50 can be shown to be comparable to VGG19, with both models having similar inference times of 4.7 and 4.2 respectively. Of models tested, AlexNet yields the lowest accuracy of 77.51% with the latest models, InceptionV3 and Xception, performing with 81.13%, and 84.43% respectively. Surprisingly the larger, more simple, VGG19 network out-performs these within this domain with 88.68%. Overall ResNet50 produces the highest test-set accuracy of 91.04%, a 2.36% improvement over VGG19. Of these networks both VGG19 and ResNet50 boast a low BER per-class with a low of 5.01% and 3.35% respectively; other models produced BER typically between 10 - 20%. Further metrics from each model can be seen in tables 3, 4, 5, 6, and 7 - with reference to table 2 for a reference key for the class id.

Table 1: CNN architectures with Parameters, Training Times (hours), and Average Inference Times (ms) over 500 test-set runs, and test-set accuracy.

Model Name	Number of Parameters	Transfer Training Time (minutes)	Average Inference Time Per Image (ms)	Test-set Accuracy (%)
AlexNet [23]	111,443,342	70.40	**1.35**	77.51
VGG19 [23]	55,704,649	**26.3**	4.70	88.68
Resnet50 [23]	23,597,961	111.47	4.2	**91.04**
InceptionV3 [23]	21,813,033	370.1	6.27	81.13
Xception [1]	**20,871,729**	814.9	8.54	84.43

Table 2: Lookup table mapping Class ID to Full Weapon Category

Class ID	0	1	2	3	4
Category	Assault Rifle	Revolver	Self-Loading Pistol	Shotgun	Sub-Machine Gun

Table 3: AlexNet per class classification metrics - each class is treated as a one-vs-all approach.

Class	TP	TN	FP	FN	Sens (%)	Spec (%)	Acc (%)	BER(%)
0	110	470	34	22	83.33	93.25	91.20	11.71
1	90	462	42	42	68.18	91.67	86.79	20.08
2	97	489	15	35	73.48	97.02	92.13	14.75
3	86	516	12	22	79.62	97.72	**94.65**	**11.32**
4	110	464	40	22	83.33	92.06	90.26	12.30

Table 4: VGG19 per class classification metrics - each class is treated as a one-vs-all approach.

Class	TP	TN	FP	FN	Sens (%)	Spec (%)	Acc (%)	BER(%)
0	122	491	13	10	92.42	97.42	**96.38**	5.08
1	113	496	8	19	85.60	98.41	95.75	7.99
2	109	497	7	23	82.58	98.61	95.28	9.41
3	96	504	24	12	88.89	95.45	94.33	7.83
4	124	484	20	8	93.94	96.03	95.60	**5.01**

Table 5: ResNet50 per class classification metrics - each class is treated as a one-vs-all approach.

Class	TP	TN	FP	FN	Sens (%)	Spec (%)	Acc (%)	BER(%)
0	125	497	7	7	94.70	98.61	97.80	**3.35**
1	109	492	12	23	82.58	97.62	94.50	9.90
2	121	488	16	11	91.67	96.83	95.75	5.75
3	102	521	7	6	94.44	98.67	**97.96**	3.44
4	122	489	15	10	92.42	97.02	96.07	5.28

Table 6: InceptionV3 per class classification metrics - each class is treated as a one-vs-all approach.

Class	TP	TN	FP	FN	Sens (%)	Spec (%)	Acc (%)	BER(%)
0	118	488	16	14	89.39	96.83	95.28	**6.89**
1	96	481	23	36	72.73	95.44	90.72	15.92
2	100	479	25	32	75.76	95.04	91.04	14.60
3	95	513	15	13	87.96	97.16	**95.60**	7.44
4	107	463	41	25	81.06	91.87	89.62	13.54

Table 7: Xception per class classification metrics - each class is treated as a one-vs-all approach.

Class	TP	TN	FP	FN	Sens (%)	Spec (%)	Acc (%)	BER(%)
0	119	484	20	13	90.15	96.03	94.81	**6.91**
1	108	489	15	24	81.82	97.02	93.87	10.58
2	105	481	23	27	79.55	95.44	92.14	12.51
3	94	516	12	14	87.04	97.73	**95.91**	7.62
4	111	475	29	21	84.09	94.24	92.14	10.83

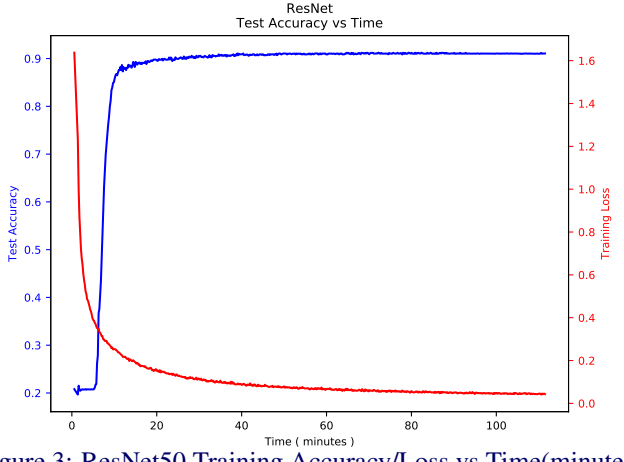


Figure 3: ResNet50 Training Accuracy/Loss vs Time(minutes)

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[2] Samet Akcay and Toby P. Breckon. An evaluation of region based object detection strategies within x-ray baggage security imagery. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1337–1341. IEEE, sep 2017. ISBN 978-1-5090-2175-8. doi: 10.1109/{ICIP}.2017.8296499. URL http://ieeexplore.ieee.org/document/8296499/.

[3] 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 *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1057–1061. IEEE, 2016.

[4] Samet Akcay, Mikolaj E. Kundegorski, Chris G. Willcocks, and Toby P. Breckon. Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. *IEEE Transactions on Information Forensics and Security*, 13(9):2203–2215, sep 2018. ISSN 1556-6013. doi: 10.1109/{TIFS}.2018.2812196. URL https://ieeexplore.ieee.org/document/8306909/.

[5] Muhammet Baştan, Mohammad Reza Yousefi, and Thomas M Breuel. Visual words on baggage x-ray images. In *Computer analysis of images and patterns*, pages 360–368. Springer, 2011.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006.

[7] Zhiyu Chen, Yue Zheng, Besma R Abidi, David L Page, and Mongi A Abidi. A combinational approach to the fusion, de-noising and enhancement of dual-energy x-ray luggage images. In *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference On*, pages 2–2. IEEE, 2005.

[8] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357*, 2016.

[10] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[11] 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.

[12] Michael Guerzhoy. Alexnet implementation + weights in tensorflow. http://www.cs.toronto.edu/~guerzhoy/tf_alexnet/, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[16] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[17] Odysseas Kechagias-Stamatis, Nabil Aouf, David Nam, and Carole Belloni. Automatic x-ray image segmentation and clustering for threat detection. In *Target and Background Signatures III*, volume 10432, page 104320O. International Society for Optics and Photonics, 2017.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-ne pdf.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[20] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[21] Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, and Stefan Winkler. Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pages 443–449. ACM, 2015.

[22] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[23] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[24] Lutz Prechelt. Early stoppingâĂŤbut when? In *Neural Networks: Tricks of the Trade*, pages 53–67. Springer, 2012.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. URL http://arxiv.org/abs/1409.0575.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015.

[27] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[30] TensorFlow. Tensorflow models alexnet. https://github.com/tensorflow/models/blob/11733fcafdb148878052c47dda0e4b9e76736700/tutorials/image/alexnet/alexnet_benchmark.py, 2017.

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

[32] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.