DETECTING PROHIBITED ITEMS IN X-RAY IMAGES: A CONTOUR PROPOSAL LEARNING APPROACH

Taimur Hassan[1] *Meriem Bettayeb*[1] *Samet Akçay*[2,3] *Salman Khan*[4]
Mohammed Bennamoun[5] *Naoufel Werghi*[1]

[1]Khalifa University, [2]Durham University, [3]COSMONiO AI
[4]Inception Institute of Artificial Intelligence, [5]University of Western Australia

ABSTRACT

X-ray baggage screening plays a vital role in aviation security. Manual inspection of potentially anomalous items is challenging due to the clutter and occlusion within X-ray scans. Here, we address this issue by presenting an object-boundaries driven framework for the automated detection of suspicious items from X-ray baggage scans. Rather than recognizing objects directly from the X-ray images, our two-stage detection approach first extracts contour-based proposals using a novel cascaded structure tensor technique and subsequently passes the candidate proposals to a single feed-forward convolutional neural network for recognition. Thorough experimentation on GDXray and SIXray datasets demonstrates that the proposed model achieves a mean area under the curve of 0.9878, outperforming the existing renown state-of-the-art object detection frameworks.

***Index Terms*—** X-ray images, Baggage screening, Image analysis, Structure Tensor, Convolution Neural Network

1. INTRODUCTION

Potential threats concealed in baggage have become a prime security concern all over the world. According to a recent report, approximately 1.5 million passengers are searched every day in the United States against weapons and other dangerous items [1]. The manual detection of such items in each baggage is a cumbersome and time-consuming process. Therefore, automated and reliable baggage screening system is of most interest. X-ray images, however, are quite different from natural photographs as they lack texture, making conventional object detection methods not suitable on them [2]. In general, screening objects and anomalies from baggage X-ray scans is a challenging task, especially when the objects are tightly packed to each others, leading to heavy occlusions. Several methods for detecting objects in X-ray imagery have been proposed in the literature. We provide a representative list of the main approaches, and we refer the reader to the work of [3] for an exhaustive survey. The very first approaches used handcrafted features such as key-point descriptors [2], and bag of visual words [4,5]. Afterwards, with the advent of deep learning paradigm, a variety of end-to-end methods have been proposed as an alternative aiming at more robust, scalable, and reproducible solutions. Akçay et al. [6] used a pre-trained GoogleNet [7] model for object classification from X-ray baggage scans. They tested their approach on an in-house dataset comprising cameras, laptops, guns, gun components and ceramic knives. Dhiraj et al. [8] used YOLOv2 [9], Tiny YOLO [10] and Faster R-CNN [11] models to detect guns, shuriken, razor-blades and knives from baggage X-ray scans. Akçay et al. [12] compared various networks for the object classification and detection with X-ray imagery. They concluded that AlexNet [13] as a feature extractor with support vector machines (SVM) perform better than other machine learning (ML) methods. Recently, Miao et al. [14] provided one of the most challenging X-ray imagery datasets (named SIXray) together with a framework dubbed deep class-balanced hierarchical refinement (CHR) for detecting suspicious items. Gaus et al. [15] evaluated transfer learning approaches and their transferability to prohibited items detection, using Faster R-CNN [11], Mask R-CNN [16] and RetinaNet [17]. Moreover, adversarial learning is also employed in the past for baggage threat detection [18–20].

Despite the progress accomplished so far, accurately recognising overlapping items is still a challenge. One of the main reasons is that the existing approaches employ proposal generation strategies specifically designed for colour images. X-ray scans, however, are remarkably different as they lack textural details that characterise grayscale and coloured images. In this work, rather than considering region-based or keypoint-based techniques to generate multiple-scale regions, our approach derives proposals according to the hierarchy of the regions defined by the contours. This approach is based on the insight that shape information is the most reliable cue in X-ray scans. For instance, suspicious 3D-printed items such as guns show only their outlines in X-ray scans. We, therefore, propose to utilise the contour information from X-ray images to generate object candidates. We incarnate this proposed approach in an original framework blending a robust contour-based proposals extraction, and a single feed-forward CNN model for object recognition. Compared to competitive

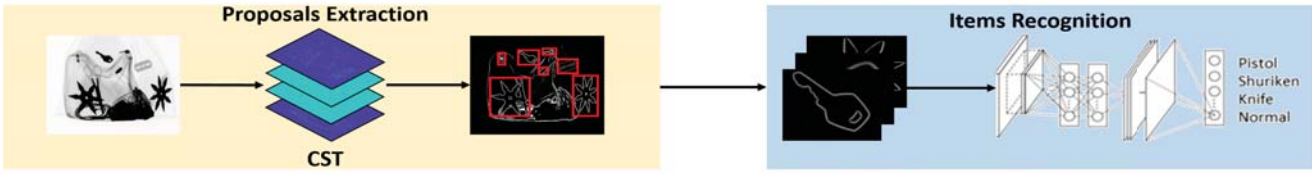

Fig. 1: Block diagram: After adaptive histogram equalization the X-ray image is passed thorough the CST to generate series of transition maps. From these, contour-based proposals are derived, and passed afterwards to a pretrained ResNet50 model for recognition.

methods, our system exhibits following characteristics: 1) The extraction of object proposals performed through a novel Cascaded Structure Tensor (CST) technique, which analyzes the image transitions in series of tensors derived from the X-ray scans. 2) The proposed framework alleviate the class imbalance problem by performing the training on a balanced set of proposals of regular and suspicious items rather than on the set of scans containing the imbalanced ratio of normal and irregular objects. 3) The robustness of our approach to occlusion and highly cluttered scenarios.

2. PROPOSED METHOD

As depicted in the block diagram in Fig. 1, our first step is to enhance the image contrast via an adaptive histogram equalization [21]. Afterwards, a series of tensors are generated by the CST framework, where each tensor contains transition maps of the targeted objects from different orientations. By using these tensors, the object proposals are automatically extracted, and passed to a ConvNet object recognition module employing a pre-trained ResNet model.

CST Framework: Our CST framework is based on the concept of structured tensors [22], which is, in its simplest form, a 2×2 symmetric positive-semi definite matrix defined by the outer products of image gradients at each pixel. It reflects the predominant orientations of the changes (contained in the gradients) within a specified pixel's neighbourhood. In our approach, given a set of η orientations $\theta_1, \cdots, \theta_\eta$ we define a general multi-oriented block-structured tensor as $\nu \times \nu$ matrix

$$\begin{bmatrix} \phi * \bigtriangledown^2_{\theta_1}(I) & \phi * \bigtriangledown_{\theta_1} \bigtriangledown_{\theta_2}(I) & \cdots & \phi * \bigtriangledown_{\theta_1} \bigtriangledown_{\theta_\eta}(I) \\ \phi * \bigtriangledown_{\theta_2} \bigtriangledown_{\theta_1}(I) & \cdots & \cdots & \phi * \bigtriangledown_{\theta_2} \bigtriangledown_{\theta_\eta}(I) \\ \vdots & \cdots & \ddots & \vdots \\ \phi * \bigtriangledown_{\theta_1} \bigtriangledown_{\theta_\eta}(I) & \cdots & \cdots & \phi * \bigtriangledown^2_{\theta_\eta}(I) \end{bmatrix}$$

where ϕ is a Gaussian smoothing function, and $\bigtriangledown_{\theta_k}(I)$ is the image gradient in the θ_k direction. Being symmetric, the above block-structured tensor contains $\mathcal{N} = \eta(\eta+1)/2$ unique tensors. Each tensor $\mathcal{T}_k, k = 1, ..., \mathcal{N}$, is a second-moment matrix representing transition intensity maps, whereby the predominant transition is defined by the eigenvector associated to its maximum eigenvalue. These tensors can exploit transitional variations across cluttered objects. But, rather than deriving edges from the transition maps in a single pass, we propose an iterative extraction scheme, whereby we repeat the following operations: compute the tensors, extract the edge details associated to the most predominant transition, and apply non-maximal suppression of the detected edges at each iteration. Our motivation here is that the object boundaries in the X-ray scans do not show the same level of intensity (compare for instance the blade and the shuriken in Fig.1), hence extracting them together will undermine the effectiveness of contours detection from this type of modality. At each iteration, we compute the tensors $\mathcal{T}_{k=1,...,\mathcal{N}}$; then we determine the tensor $\mathcal{T}_\mathcal{M}$ having the maximum predominant transition, by selecting the one with the largest eigenvalue. In the next stage, we perform binarization followed by morphological enhancement on $\mathcal{T}_\mathcal{M}$ to remove the unwanted blobs and noisy artefacts. Afterwards, we extract labeled contours using connected component analysis [23]. From each labelled contour, we generate a bounding box representing a candidate proposal. Finally, the detected contours are suppressed from the current scan so that the CST can pick the other less predominant transitions during the next iterations. The suppression is performed by setting the pixels of the inner area of the contour by the mean value of the candidate scan. These steps are reiterated until no more object proposals can be extracted. The detailed algorithm of our CST framework is reported in Algorithm-1.

Algorithm 1 CST framework

Input: X-ray scan Ξ
Output: Set of object proposals $\mathcal{P}$

Initialize:
    HasObjects $\leftarrow$ True
While HasObjects == True
Compute the tensors $\mathcal{T}_{k=1..\mathcal{N}}$
Determine the predominant tensor $\mathcal{T}_\mathcal{M}$
Extract the set of contours $\mathcal{C}$ from $\mathcal{T}_\mathcal{M}$
if $\mathcal{C} == \emptyset$ **then**
    HasObject = False
end if
for each labeled contour $p \in \mathcal{C}$ **do**
    Compute its bounding box β
    Object proposal $\mathcal{O}_p \leftarrow$ crop($\mathcal{T}_\mathcal{M}$, β)
    Add $\mathcal{O}_p$ to $\mathcal{P}$
    Suppress $\mathcal{C}$ from Ξ
end for
endWhile

Object recognition: After extracting the object proposals, these are passed to the ResNet50 for recognition. The ResNet50 exhibits good performance in catering the vanishing gradient problem through the residual blocks [24]. We employ ResNet50 in fine-tuning mode, whereby we replace the final classification layer with our custom layer for the recognition of proposals within this application. We do not freeze the rest of the layers so that they also get updated during the training phase to recognize the object proposals effectively. However, we use the original weights in the initialization phase for faster convergence. Note that the training set is composed of object proposals obtained with the CST framework, which generate 150 proposals on average per scan. This amplification in the number of training samples allows deriving balanced sets for normal and suspicious items, as will be described further in Section 3.

3. EXPERIMENTS

We evaluated the proposed framework against state-of-the-art methods on two different public datasets: the GRIMA X-ray Database (GDXray) [25] and Security Inspection X-ray Dataset (SIXray) [14] considering two gradient orientations $\theta = 0, \pi/2$, and using a variety of evaluation metrics. The GDXray contains a baggage group of 8,150 X-ray scans containing both occluded and non-occluded items with marked ground truths for handguns, razor blades, shuriken and knives. For a more in-depth evaluation of the proposed framework, we have refined this categorization by splitting the original handgun category into two classes, namely, pistol and revolver. We have also identified and annotated two new classes, i.e., the mobile phone class and the chip class, which represent all the electronic gadgets, including laptops. We adopted a training set in accordance with the protocol defined in [26], i.e., 400 scans from B0049, B0050 and B0051 series containing proposals for revolver (handgun), shuriken and razor blades, to which we added 388 more scans for the new categories (chip, pistol, mobile and knives). The SIXray contains 1,059,231 colour X-ray scans having 8,929 suspicious items which are classified into six groups, i.e. gun, knife, wrench, plier, scissor and hammer. To validate the performance of the proposed framework against the class imbalance problem, we used the same subsets reported in [14], in which the ratio of suspicious items and normal objects have been matched with real-world scenarios. Also, the ratio of 4 to 1 for training and testing has been maintained in accordance with [14]. For both GDXray and SIXray datasets, we have added a separate normal class to filter the proposals of miscellaneous and unimportant items, such as keys and bag zippers. The normal class is not considered in the evaluations since it is only added to prevent the misclassification of such miscellaneous items as suspicious. For the GDXray dataset, we compared our framework with the methods [27], [8], [28] and [29] as shown in Table 1 (top). Contrary to these methods, we accessed our framework for all the performance criteria shown in the table. The performance comparison is nevertheless indirect as the experiment protocol in each study differs, where we (as well as authors in [27]) followed the standards laid in [49], i.e., 400 images for training, i.e., 100 for razor blades, 100 for shuriken and 200 for handguns. However, [27] used 600 images for testing purposes (200 for each item) and considered only 3 items, whereas we considered seven items and used 7,362 scans for testing. The authors in [8] considered a total of 3,669 selective images having 1,329 razor blades, 822 guns, 540 knives, and 978 shurikens. To train Faster R-CNN, YOLOv2 and Tiny YOLO models, they picked 1,223 images from the dataset and augmented them to generate 2,446 more images. The work reported in [29] involved 18 images only while [28] reports a study that is based on non-ML methods where the authors conducted 130 experiments to detect razor blades within the X-ray scans. Note that the proposed framework has been evaluated in the most restrictive conditions as compared to its competitors where the true positive samples (of the extracted items) were only counted towards the scoring when they were correctly classified by the ResNet50 model as well. Therefore, if an item has been correctly extracted by the CST framework and was not correctly recognized by the ResNet50 model, we counted it as a misclassification for evaluation. Despite such restrictions, we were able to achieve 4.26% and 2.97% improvements in the precision and the F1 score, respectively. Overall, the performance of our framework exceeds or equates all the other competitive methods across all criteria except for the accuracy where it scores slightly less than [8]. For the SIXray dataset, we compared our system with the methods proposed in [14] and [15] (the only two frameworks which have been applied on SIXray dataset till date). These two works employed different CNN models, which we also reported in the comparison for completeness in Table 1 (bottom). Also, for a fair comparison with [14] and [15], we have trained the proposed framework on each subset of the SIXray dataset individually and reported the performance, where the hammer class is not considered as in [14]. Note also that the SIXray dataset is divided into three subsets to address the problem of class imbalance. These subsets are dubbed SIXray(10, 100, 1000). SIXray10 contains all 8,929 positive scans (having suspicious items) and ten times the negative scans (which do not contain any suspicious item). Similarly, SIXray100 has all the positive scans and 100 times the negative scans. SIXray1000 contains only 1000 positive scans and all the negative scans (1,050,302 in total), making it the most challenging subset for the class imbalance problem. The results obtained across the different criteria is an evidence that the proposed framework outperforms the state-of-the-art within object classification and localization tasks. In Fig. 2, we report the performance of our framework for recognizing individual suspicious items (the average precision), together with [14] and [15]. We can no-

Table 1: Performance comparison on GDXray (top) and SIXray dataset (bottom). '-' indicates metric not reported.

Criteria	Proposed (Generalized)	Proposed (GDXray)	Faster RCNN [8]	YOLOv2 [8]	Tiny YOLO [8]	$AISM_1$ [27]*	$AISM_2$ [27]*	SURF [27]	SIFT [27]	ISM [27]	[28]	[29]
Mean AUC	0.9878	**0.9934**	-	-	-	0.9917	0.9917	0.6162	0.9211	0.9553	-	-
Accuracy	0.9457	0.9833	**0.9840**	0.9710	0.89	-	-	-	-	-	-	-
Sensitivity	0.9428	0.9969	0.98	0.88	0.82	**0.9975**	0.9849	0.6564	0.8840	0.9237	0.89	0.943
Specificity	0.95	**0.9650**	-	-	-	0.95	**0.9650**	0.63	0.83	0.885	-	0.944
False Positive Rate	0.05	**0.035**	-	-	-	0.05	**0.035**	0.37	0.17	0.115	-	0.056
Precision	0.6985	**0.9714**	0.93	0.92	0.69	-	-	-	-	-	0.92	-
F1 Score	0.8025	**0.9836**	0.9543	0.8996	0.7494	-	-	-	-	-	0.9048	-

Criteria	Subset	ResNet50 (CST)	ResNet50 [30]	ResNet50 (CHR) [14]	DenseNet [31]	DenseNet (CHR) [14]	Inceptionv3 [32]	Inceptionv3 (CHR) [14]	[15]
Mean Average Precision	SIXray10	**0.9612**	0.7685	0.7794	0.7736	0.7956	0.7956	0.7949	0.86
	SIXray100	**0.9297**	0.5222	0.5787	0.5715	0.5992	0.5609	0.5815	-
	SIXray1000	**0.8894**	0.3390	0.3700	0.3928	0.4836	0.3867	0.4689	-
Localization Accuracy	SIXray10	**0.8254**	0.5140	0.5485	0.6246	0.6562	0.6292	0.6354	-
	SIXray100	**0.7786**	0.3405	0.4267	0.4470	0.5031	0.4591	0.4953	-
	SIXray1000	**0.7429**	0.2669	0.3102	0.3461	0.4387	0.3026	0.3149	-

tice that our framework achieves the best performance across the three subsets, with a significant improvement over [14] for the wrench, plier and scissor.

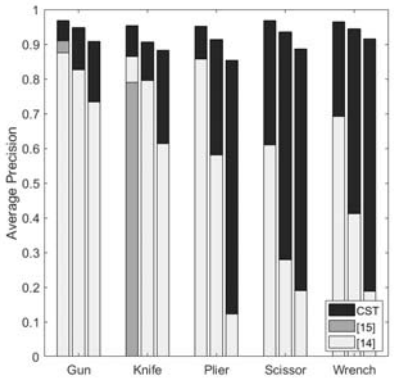

Fig. 2: Performance comparison of the proposed CST framework with [14] and [15] on SIXray subsets (left bar: SIXray10, middle bar: SIXray100, right bar: SIXray1000 for each item) where ResNet50 is used as a backbone. (CST and [14] are evaluated on all subsets while [15] only used SIXray10 subset).

In the last experiment, we compared the computational performance of CST with standard one staged (such as YOLOv2, RetinaNet) and two-staged detectors (such as R-CNN variants), as they have been widely used for threat detection in the past. All experiments were performed using MATLAB R2019a on an Intel i5-8400@2.8GHz processor with 16 GB DDR3 RAM and equipped with an NVIDIA RTX 2080 GPU. The results depicted in Table 2 show that our system scores the best average time performance in both training and testing, outperforming in particular YOLOv2. It is important to note that although YOLOv2 has significant improvements in computational performance over other two staged models, it has been designed to extract objects from rich textured photographs. Also, due to its spatial constraints, it does not detect small objects well.

Table 2: Time performance comparison of our proposed approach for training and testing. Average time is computed per image.

	Training(s)	Testing(s)
YOLOv2 [9]	712.72	0.025
RetinaNet [17]	927.52	0.073
R-CNN [33]	306000	134.75
Faster R-CNN [11]	19600	0.55
Proposed	**677.09**	**0.019**

4. CONCLUSION

This paper presents a novel framework for the automated detection of suspicious items from X-ray baggage scans. The proposed system is rigorously tested on two publicly available datasets and is thoroughly compared with existing state-of-the-art solutions based on various metrics. The extraction of object proposals, in the proposed framework, is based on a novel CST segmentation scheme to accurately represent the transitional patterns and robustly derive the object's contours. Furthermore, the proposed framework can detect multiple objects from X-ray scans, irrespective of the scan type and imbalanced classes. It is only based on a single feed-forward CNN model and does not require exhaustive searches and regression networks, due to which it is more time-efficient compared to the state-of-the-art two-staged CNN object detectors [11]. Future work will investigate the proposed system on normal rich-textured photographs and popular large scale publicly available datasets for autonomous object detection.

5. REFERENCES

[1] N. R. Council *et al.*, *Airline passenger security screening: new technologies and implementation issues*. National Academies Press, 1996.

[2] M. Bastan *et al.*, "Object recognition in multi-view dual energy X-ray images.," in *BMVC*, vol. 1, p. 11, 2013.

[3] S. Akçay *et al.*, "Towards automatic threat detection: A survey of advances of deep learning within X-ray security imaging," *arXiv*, vol. abs/2001.01293, 2020.

[4] M. Baştan *et al.*, "Visual words on baggage X-ray images," in *CAIP*, pp. 360–368, Springer, 2011.

[5] M. Baştan, "Multi-view object detection in dual-energy X-ray images," *Machine Vision and Applications*, vol. 26, no. 7-8, pp. 1045–1060, 2015.

[6] S. Akçay *et al.*, "Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery," in *IEEE ICIP*, pp. 1057–1061, 2016.

[7] C. Szegedy *et al.*, "Going Deeper with Convolutions," *arXiv:1409.4842v1*, September 2014.

[8] D. K. Jain *et al.*, "An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery," *Pattern Recognition Letters*, vol. 120, pp. 112–119, 2019.

[9] J. Redmon *et al.*, "YOLO9000: Better, Faster, Stronger," *arXiv:1612.08242*, December 2016.

[10] J. Redmon, "YOLO: Real-Time Object Detection," *URL: https://pjreddie.com/darknet/yolo/*, Accessed: February, 2020.

[11] S. Ren *et al.*, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *arXiv:1506.01497v3*, January 2016.

[12] S. Akçay *et al.*, "Using deep convolutional neural network architectures for object classification and detection within X-ray baggage security imagery," *IEEE transactions on information forensics and security*, vol. 13, no. 9, pp. 2203–2215, 2018.

[13] A. Krizhevsky *et al.*, "ImageNet Classification with Deep Convolutional Neural Networks," *in Advances in Neural Information Processing Systems*, January 2012.

[14] C. Miao *et al.*, "SIXray: A Large-scale Security Inspection X-ray Benchmark for Prohibited Item Discovery in Overlapping Images," in *IEEE CVPR*, pp. 2119–2128, 2019.

[15] Y. F. A. Gaus *et al.*, "Evaluating the transferability and adversarial discrimination of convolutional neural networks for threat object detection and classification within X-ray security imagery," *arXiv preprint arXiv:1911.08966*, 2019.

[16] K. He *et al.*, "Mask R-CNN," *arXiv:1703.06870v3*, January 2018.

[17] T.-Y. Lin *et al.*, "Focal Loss for Dense Object Detection," *arXiv:1708.02002v2*, February 2018.

[18] S. Akcay *et al.*, "GANomaly: Semi-supervised Anomaly Detection via Adversarial Training," in ACCV, 2018.

[19] S. Akcay *et al.*, "Skip-GANomaly: Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection," in IJCNN, 2019.

[20] Y. F. A. Gaus *et al.*, "Evaluation of a Dual Convolutional Neural Network Architecture for Object-wise Anomaly Detection in Cluttered X-ray Security Imagery," in IJCNN, 2019.

[21] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*, pp. 474–485, Academic Press Professional, Inc., 1994.

[22] M. Kass and A.Witkin, "Analyzing oriented patterns," *Computer Graphics and Image Processing*, vol. 37, pp. 363–385, 1987.

[23] H. Samet and M. Tamminen, "Efficient component labeling of images of arbitrary dimension represented by linear bintrees," *IEEE transactions on pattern analysis and machine intelligence*, vol. 10, no. 4, pp. 579–586, 1988.

[24] S. Khan *et al.*, "A guide to convolutional neural networks for computer vision," *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, 2018.

[25] M. Murguía and J. L. Villaseñor, "Estimating the effect of the similarity coefficient and the cluster algorithm on biogeographic classifications," in *Annales Botanici Fennici*, pp. 415–421, JSTOR, 2003.

[26] D. Mery *et al.*, "GDXray: The database of X-ray images for nondestructive testing," *Journal of Nondestructive Evaluation*, vol. 34, no. 4, p. 42, 2015.

[27] V. Riffo and D. Mery, "Automated detection of threat objects using adapted implicit shape model," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 4, pp. 472–482, 2015.

[28] V. Riffo and D. Mery, "Active X-ray testing of complex objects," *Insight-Non-Destructive Testing and Condition Monitoring*, vol. 54, no. 1, pp. 28–35, 2012.

[29] D. Mery, "Automated detection in complex objects using a tracking algorithm in multiple X-ray views," in *IEEE CVPR 2011 WORKSHOPS*, pp. 41–48, 2011.

[30] K. He *et al.*, "Deep residual learning for image recognition," in *IEEE CVPR*, pp. 770–778, 2016.

[31] G. Huang *et al.*, "Densely connected convolutional networks," in *IEEE CVPR*, pp. 4700–4708, 2017.

[32] C. Szegedy *et al.*, "Rethinking the inception architecture for computer vision," in *IEEE CVPR*, pp. 2818–2826, 2016.

[33] R. Girshick *et al.*, "Rich feature hierarchies for accurate object detection and semantic segmentation," *in IEEE CVPR*, 2014.