Object Detection Based on CNNs: Current and Future Directions

Long Chen, Abdul Hamid Sadka, Junyu Dong, and Huiyu Zhou

1 Introduction

The goal of object detection is to learn a visual model for concepts such as cars and use this model to localize these concepts in an image. As shown in Fig. 1, given an image, object detection aims at predicting the bounding box and the label of each object from the defined classes in the image. This requires the ability to robustly model invariants against illumination changes, deformations, occlusions and other intra-class variations. Among a number of vision tasks, object detection is one of the fastest moving areas due to its wide applications in surveillance [1, 2] and autonomous driving [3, 4].

L. Chen · H. Zhou
School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK
e-mail: lchen15@qub.ac.uk; h.zhou@ecit.qub.ac.uk

A. H. Sadka (✉)
Department of Electronic and Computer Engineering, Brunel University, London, UK
e-mail: abdul.sadka@brunel.ac.uk

J. Dong
Department of Computer Science and Technology, Ocean University of China, Qingdao, China
e-mail: dongjunyu@ouc.edu.cn

J. M. Alja'am et al. (eds.), *Recent Trends in Computer Applications*,
https://doi.org/10.1007/978-3-319-89914-5_2

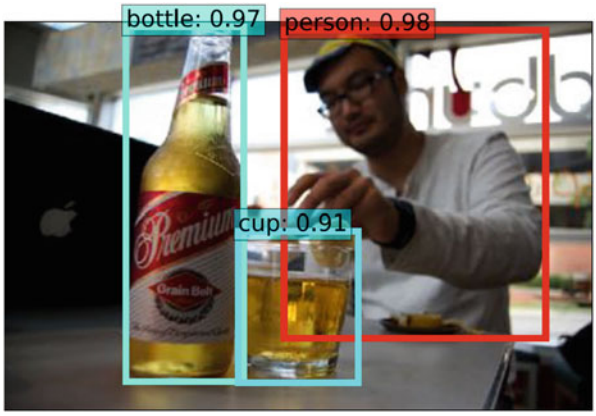

Fig. 1 Bounding boxes and labels with corresponding class probabilities predicted by detectors

2 From Handcrafted Features to Deep CNNs Methods

2.1 Handcrafted Features

Before deep CNNs, convolutional neural networks [5], were introduced, the progress on various visual recognition tasks had been considerably based on the use of handcrafted features, such as SIFT [6] and HOG [7]. Handcrafted features can be broadly divided into three categories:

1. Interest Point Detection. These methods use certain criteria to select pixels, edges and corners as well-defined local texture features. Among them, Sobel, Prewitt, Roberts, Canny and LoG (Laplacian of Gaussian) are typical edge detection operators [8–11], while Harris, FAST (Features from Accelerated Segment Test), CSS (Curvature Scale Space) and DOG (Difference of Gaussian) are typical corner detection operators [6, 12, 13]. Interest point detection methods usually have a certain geometric invariance which can be found at a small computational cost.
2. Methods based on local features. These methods mainly extract local features, which are different from global features such as colour histograms, which are ideal for dealing with partial occlusion of target objects. Commonly used local features include Scale-Invariant Feature Transform (SIFT) [6], HOG (Histogram of oriented gradient) [7], Haar-like [14] and Local Binary Pattern [15, 16]. Local features are informative, unique, with strong invariance and distinguishability. But the calculation is generally complicated, and local features are further developed to have better representations in recent years.
3. Methods based on multi-feature combination. A combination of interest point and local feature extraction methods can be used to handle the deficiency of using a single feature to represent target objects. DPM (Deformable Part-based

Model) [17] is an effective multi-feature combination model which has been widely applied to the object detection task and has achieved good performance, such as pedestrian detection [14, 16], face detection [15, 18] and human pose estimation [19]. In [20], three prohibitive steps in the cascade version of DPM were accelerated, which greatly improved the detection speed.

The characteristics of handcrafted features are largely dependent on experience and environments, where most of the test and adjustment workloads are undertaken by the user, which is time-consuming. In contrast, an important viewpoint in the deep learning theory, which has drawn much attention in recent years, is that handcrafted descriptors, as the first step in a visual system, tend to lose useful information. Directly learning task-related feature representation from raw images is more effective than handcrafted features [21].

For object detection tasks, handcrafted features based systems have become a dominant paradigm in the literature before deep CNNs were introduced. If we look at system performance on the canonical visual recognition task, PASCAL VOC object detection [22], it is acknowledged that certain progress has been made during 2010–2012, by building ensemble systems and employing variants of successful methods. Recently, Convolutional Neural Networks (CNNs) [5] have produced impressive performance improvements in many computer vision tasks since 2012, such as image classification, object detection and image segmentation. CNNs witnessed its frequent use in the 1990s (e.g., [5]), but then became less used, particularly in computer vision, with the powerful impact of support vector machines (SVMs) [23]. In 2012, Krizhevsky et al. [24] rekindled interests in CNNs by showing substantially high image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [25]. Their success resulted from training a large CNN on 1.2 million labelled images, together with a few twists on [5] (e.g., 'dropout' regularization). The significance of deep CNNs methods will be introduced in the following section.

2.2 Deep Learning Approaches

Convolutional Neural Networks [5] is the first successful method in deep learning approaches. The key difference between CNNs-based and conventional approaches is that in the former, the feature representation is learned instead of being designed by the user. These recent successes were built upon the powerful deep features that are learned from large-scale datasets, which accompany accurate annotations with the drawback that a large number of training samples are required for training the classifier. Among many variants of the CNNs-based approaches, they can be roughly divided into two streams: region proposal-based methods and proposal-free methods.

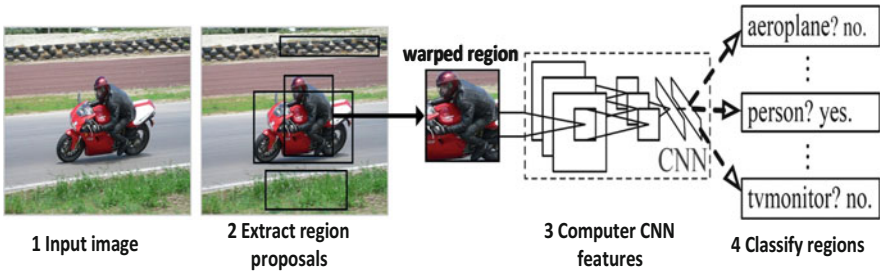

Fig. 2 The overview of the R-CNN detection system. (1) Input an image, (2) extracts region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs

2.2.1 Region Proposal-Based Methods

The dominant paradigm in modern object detection is the region proposal-based method. The pioneering work Selective Search [26] consists of two stages: The first stage generates a sparse set of candidate proposals that should contain all the objects while filtering out the majority of negative locations and the second stage classifies the proposals into foreground or background. R-CNN [27] upgrades the second-stage classifier to a convolutional network yielding large gains in accuracy and ushering in the modern era of object detection (shown in Fig. 2). R-CNN requires high computational costs while each proposal is processed by the CNNs separately. Fast R-CNN [28] improved efficiency by sharing computation and using RoI (Region of Interest) pooling [29] to efficiently generate features for object proposals. Region Proposal Networks (RPNs) integrate proposal generation with the second-stage classifier in a single convolution network, forming the Faster RCNN framework [30]. R-FCN [31] further improved efficiency and accuracy by removing fully connected layers while adopting position-sensitive score maps for the final detection. However, one problem with the region-based methods is that in order to process a large number of proposals, the computation in the second stage is usually costly. To accelerate the detection process, proposal-free methods have been proposed for real-time detection.

2.2.2 Proposal-Free Methods

Proposal-free methods aim to eliminate the region proposal stage and directly train a single-stage end-to-end detector. Without the region proposal stage, they have the potential to be faster and simpler, but have trailed the accuracy of two-stage detectors thus far. YOLO [32] used a single feed-forward convolutional network to directly predict object classes and locations. Compared with region-based methods, YOLO no longer requires a second per-region classification operation so that it is extremely fast. SSD [33] improved YOLO in several aspects, including (1) using

small convolutional filters to predict categories and anchor offsets for bounding box locations; (2) using pyramid features for prediction at different scales; (3) using default boxes and aspect ratios for adjusting varying object shapes. Those clever designs save considerable amounts of computation and perform much faster than Faster RCNN. The proposal-free detectors are usually easier to train with less computational efforts. However, such advantage is largely overwritten when the models are evaluated in benchmarks considering mean average precision (mAP) for high intersection-over-union (IoU) thresholds (e.g., KITTI car) since the two-stage methods are usually advantageous in performance. It achieved good results in datasets for the IoU threshold of 0.5. However, the performance drops significantly when we increase the bar for detection quality.

2.2.3 Fine-Tuning Strategy

When training supervised classifiers, we expect that there are sufficient labelled samples available for the target classes [1]. However, this requirement seems too demanding in some real-world applications. For example, many objects 'in the wild' follow a long-tailed distribution such that they do not occur frequently enough to collect and label a large set of representative exemplars to build the corresponding recognizers [34]. In addition, the labelling effort for many objects can be very expensive because the expert knowledge is required, for example, fine-grained bird recognition [35]. Under these circumstances, it is always expected to train effective classifiers with as few labelled samples as possible. Fine-tuning is one of the widely adopted paradigms to save efforts for labelling data in a supervised learning. It involves learning a generic feature representation on a large dataset of labelled images, and then specializing or fine-tuning the learned generic feature representation for a specific task at hand. Especially, in order to achieve good performance, most of the advanced object detection systems fine-tune classification networks that start from generic features learned on the ImageNet dataset using over a million labelled images and then specialize them for object detection tasks. Other approaches [36, 37] design specific backbone network structures for object detection, but still require pre-training the networks on the ImageNet classification dataset.

Fine-tuning object detectors from the pre-trained classification models has at least two advantages. First, there are many state-of-the-art deep models publicly available. It is convenient to reuse them for object detection. Second, fine-tuning can quickly generate the final model and requires much less instance-level annotated training data than the classification task. However, there are also critical limitations when adopting the pre-trained networks in object detection: (1) A limited structure design space. The pre-trained network models are mostly from the ImageNet-based classification task, which are usually very heavy—containing a huge number of parameters. (2) Learning bias. As both the loss functions and the category distributions between classification and detection tasks are different, this will lead to different searching/optimization spaces. Therefore, learning may be biased towards

a local minimum, which is not the best for the detection task. Model fine-tuning for the detection task can alleviate this bias to some extent but not fundamentally. (3) Domain mismatch. As known, fine-tuning can mitigate the gap due to different target category distributions. However, it is still a severe problem when the source domain (ImageNet) has a huge mismatch to the target domain such as depth images.

Finally, it is worth noting that some recent work attempts to train CNNs from scratch. The proposed approach has very appealing advantages over the existing pre-training solutions [38, 39]. In semantic segmentation, Jégou et al. [40] demonstrated that a well-designed network structure can outperform state-of-the-art solutions without using the pre-trained models. It extends DenseNets [39] to fully convolutional networks by adding an up-sampling path to recover the original resolution. Shen et al. [41] presented the Deeply Supervised Object Detector (DSOD), a framework that can learn object detectors from scratch and contribute to a set of design principles for training object detectors from scratch. Training CNNs from scratch is a promising future direction due to its wide applications, though not much work has been done in this area yet.

3 Current Research Directions

In the present section, we discuss current research directions. Current research has been focused on three principal directions for developing better object detection systems. The first direction relies on innovating the base architecture of the existing networks. It has been shown that using all the examples does not always lead to an optimal solution [42] and data selection is the key. So, another research direction focuses on how to better exploit the data itself. The third area of research is to use contextual reasoning, as it can be a rich source of information about an object identity, location and scale [31].

3.1 Excellent Base Architectures

Many innovative CNN structures have been proposed [24, 38, 43, 44]. Meanwhile, several regularization techniques have also been proposed to further enhance the model capabilities. Krizhevsky et al. proposed a new convolution neural network AlexNet [24], followed by a series of improved models, such as ZFNet [43], VGG [37], GoogLeNet [44] and ResNet [38], proposed by other researchers. Table 1 shows the performance comparison of the classical CNN model in the image classification task of ILSVRC. The error rate in the image classification task of ILSVRC is reduced every year. The image classification top-5 error rate is getting lower as the base architecture becomes increasingly deeper. Although these network architectures are designed for image classification tasks, people aim to

Table 1 Performance comparison of the classical CNN model in image classification task of ILSVRC

CNN architecture	Top-5 error rate (%)
AlexNet [24]	16.4
ZFNet [43]	14.8
VGG [37]	7.3
GoogLeNet [44]	6.7
ResNet [38]	3.57
Inception-ResNet-v2 [45]	3.08

solve one of the most fundamental questions—how to create more powerful feature representation.

Given those CNN models which have strong feature representation, applying them to the target detection task results in good detection accuracy. He et al. [38] proposed residual learning blocks with skip connections, which enable training very deep detection networks with more than 100 layers. Huang et al. [39] proposed DenseNets with dense layer-wise connections. Kim et al. [46] proposed PVANet for object detection, which consists of the simplified 'Inception' block from GoogleNet. Huang et al. [47] investigated various combinations of network structures and detection frameworks, and found that Faster R-CNN with Inception-ResNet-v2 [45] achieved the best performance. Lin et al. [48] designed a simple one-stage object detector called RetinaNet, named for its dense sampling of object locations in an input image. Its design features include an efficient in-network feature pyramid and the use of anchor boxes. Thanks to these excellent network structures, the accuracy of the object detection task has been greatly improved. Performance comparison of some object detection methods on public datasets can be seen in Table 2.

3.2 Hard Example Mining

Training data plays a critical role in machine learning. The data selection strategy along the training process could significantly impact the performance of the learned model. For detection datasets which contain an overwhelming number of easy examples and a small number of hard examples, automatic selection of these hard examples can make training more effective and efficient. Hard example mining is one technique of allowing the learning system to select the most informative samples to train the model. The underlying assumption in hard example mining is that the samples have different information and only a small portion of the samples can provide sufficient information for supervised learning. In fact, the information of each sample is different; therefore, if the most representative/informative samples are selected and labelled, even a few labelled samples can provide sufficient knowledge to construct effective classifiers. Hard example mining has existed for at least 20 years, which was first introduced in [49] in the mid-1990s (if not earlier) for training face detection models. Their key idea is to perform training on a sparse set of hard examples and prevent the vast number of easy negatives from overwhelming

Table 2 Performance comparison of some object detection methods on public datasets

Datasets	Methods	mAP(%)
VOC2007	Fast R-CNN (VGG16)	70.00
	Faster R-CNN (VGG16)	73.20
	Faster R-CNN (VGG16)[a]	78.80
	Faster R-CNN (ResNet)	76.40
	Faster R-CNN (ResNet)[a]	85.60
	YOLO	63.40
	YOLOv2(544 × 544)	78.60
	SSD300 (VGG16)	72.10
	SSD500 (VGG16)	75.10
	ION	79.20
	HyperNet (VGG16)	76.30
	R-FCN (ResNet-101)	79.50
	R-FCN (ResNet-101)[a]	83.60
	PVANET	83.80
VOC2012	Fast R-CNN (VGG16)	68.40
	Faster R-CNN (VGG16)	70.40
	Faster R-CNN (VGG16)[a]	75.90
	Faster R-CNN (ResNet)[a]	83.80
	YOLO	57.90
	YOLOv2(544 × 544)	73.40
	Fast R-CNN, YOLO	70.70
	SSD300 (VGG16)	70.30
	SSD300 (VGG16)[a]	79.30
	SSD500 (VGG16)	73.10
	SSD512 (VGG16)	78.50
	SSD512 (VGG16)[a]	82.20
	ION	76.40
	OHEM, Fast R-CNN (VGG16)[a]	80.10
	HyperNet (VGG16)	71.40
	R-FCN (ResNet-101)	77.60
	R-FCN (ResNet-101)[a]	85.00
	R-FCN, ResNet Ensemble[a]	88.40
	PVANET	82.50
	Faster R-CNN, PVANET[a]	84.20
MSCOCO2015(@[0.5–0.95])	Fast R-CNN (VGG16)	19.70
	Faster R-CNN (VGG16)[a]	21.90
	Faster R-CNN (ResNet)[a]	37.40
	SSD300 (VGG16)	20.80

(continued)

Table 2 (continued)

Datasets	Methods	mAP(%)
	SSD500 (VGG16)	24.40
	ION	33.10
	R-FCN (ResNet-101)	29.20
	R-FCN (ResNet-101)[a]	31.50
	YOLOv2	21.60
MSCOCO2015(@0.5)	Fast R-CNN (VGG16)	35.90
	Faster R-CNN (VGG16)[a]	42.70
	Faster R-CNN (ResNet)[a]	59.00
	SSD300 (VGG16)	38.00
	SSD500 (VGG16)	43.70
	ION	55.70
	R-FCN (ResNet-101)	51.50
	R-FCN (ResNet-101)[a]	53.20
	YOLOv2	44.00

For VOC2007 dataset, the training set is the union of VOC2007 trainval and VOC2012 trainval, the testing set is VOC2007 test; for VOC2012, the training set is the union of VOC2007 trainval, VOC2007 test and VOC2012 trainval, the testing set is VOC2012 test

[a]Indicates using the union of MS COCO dataset and PASCAL dataset as training set; @ [0.5–0.95] means AP (averaged precision over IoU thresholds between 0.5 and 0.95) defined in COCO metric

the detector during the training. This strategy leads to an iterative training algorithm that alternates between updating the detection model given the current set of examples, and then using the updated model to find new false positives to add to the training set. The process typically commences with a training set consisting of all the object examples and a small, random set of background examples.

Hard example mining has seen widespread use in object detection research. Hard example mining algorithms are commonly used when optimizing SVMs [17, 26, 27]. In this case, the training algorithm maintains a working set of examples and alternates between training an SVM on the working set, and updating the working set by removing some examples and adding others according to a specific rule [17]. The rule removes easy examples since they provide little information to update the current model. Conversely, the rule adds hard examples which can provide sufficient information to accelerate the network training. Applying this rule leads to a global SVM solution. Hard example mining has also been applied to a number of models including shallow neural networks [50], boosted decision trees [51] and deep CNNs [52–55]. In this kind, an algorithm usually starts with a dataset of positive examples and a random set of negative examples. The machine learning model is then trained on that dataset and subsequently applied to a larger dataset to harvest false positives. The false positives are then added to the training set and then the model is trained again.

3.3 Contextual Reasoning

Context is known to play an important role in visual recognition [56]. Using contextual reasoning, proxy tasks for reasoning and other top-down mechanisms can improve image representation for object detection. Sermanet et al. [57] used two contextual regions centred on each object for pedestrian detection. In [58], in addition to specific features, features from the entire image are used to improve region classification. He et al. [29] implemented context in a more implicit way by aggregating CNN features prior to classification using different sizes' pooling regions. More recently, [59] proposed to use ten contextual regions around each object with different crops. Shrivastava and Gupta [60] used segmentation as a way to contextually prime object detectors and provide feedback to initial layers. Bell et al. [61] used a skip network architecture and the features extracted from multiple layers of representation in conjunction with contextual reasoning. Other approaches include using top-down features for incorporating context and finer details [62, 63], which leads to improved detection results.

4 Open Problems and Future Directions

In the following, we outline the problems that we believe have not been addressed, or addressed only partially in the literature, and may become interesting and relevant research directions.

4.1 Scale Invariance

To handle different variations, such as occlusion and deformations, current CNNs-based classifiers and detectors usually use a data-driven strategy—collect large-scale datasets which have object instances under different conditions. For example, the COCO dataset [64] has more than 10K examples of cars under different occlusions and deformations. We hope that these examples capture all possible variations of a visual concept and the classifier can then effectively model invariances. For CNN-based object detectors, the variance in pose and appearance can be handled by the capacity of convolutional neural networks. However, the CNN does not inherently hold scale invariance.

In academic research, two techniques are introduced to address this problem: (1) Simple multi-scale testing on image pyramids can be used to avoid the problem and achieve good accuracy [27, 28, 30, 31]. However, multi-scale testing leads to heavy computational costs. (2) The second way is to fit a CNN model to multiple scales [33, 61, 65]. They either construct a stronger network structure by combining features from different depths of a network or directly predict objects at different

depths of a network. These attempts have been, to some extent, successful under this kind of problem, but they may also lead to an increase in model size and computation. So, further improvement is still required.

4.2 High Localization Accuracy

In many real-world applications, robustly detecting objects with high localization accuracy, namely to predict the bounding box location with high Intersection over Union (IoU) is crucial to the quality of service. For instance, in vision-based robotic arm applications, the process of generating robust and accurate operations in picking up an object is highly dependent on the object localization accuracy. In advanced driver assistance systems (ADAS), accurately localizing cars and pedestrians is also closely related to the safety of the autonomous actions.

R-CNN and its variants challenge the problem using a classification approach, and they employ regression as a post-processing stage to refine the localization of the proposed bounding boxes. Najibi et al. [66] modelled object detection as finding a path from a fixed grid to boxes tightly surrounding the objects, and slacked the regression process to several iterations for the reason that one step regression cannot handle the nonlinearity of the coordinates of bounding boxes. Gidaris et al. [67] proposed a novel object localization methodology that is based on assigning probabilities related to the localization task. Those probabilities provide useful information regarding the location of the object inside the search region and they can be exploited in order to infer its boundaries with high accuracy. Further improvements are required considering its importance in many practical applications.

4.3 Long-Tail Distribution

The ImageNet image classification dataset is a well-compiled dataset, in which objects of different classes have similar numbers of samples. In real applications, however, we will experience the long-tail distributions, where a small number of object classes appear very often but the others appear rarely. For object detection, some object classes such as persons have much more samples than the other object classes like sheep for both PASCAL VOC [22] and ImageNet [68] object detection datasets, as shown in Fig. 3. For deeply learned features, however, the feature learning will be dominated by the object classes with a large number of samples and the features are not good for object classes with fewer samples in the long tail. Therefore, the extreme class imbalance encountered during the training of detectors cannot learn discriminative features well for each category. Besides, the existence of many background samples makes the feature representation capture less intra-category variance and more inter-category variance (i.e., mostly between the object

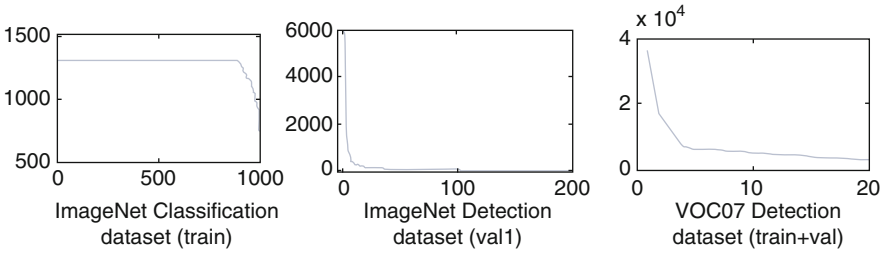

Fig. 3 The number of samples in *y*-axis sorted in decreasing order for different classes in *x*-axis on different datasets

category and background), causing many false positives between ambiguous object categories (e.g., classify horses as cows).

Long-tailed distributions of data have been studied in object detection [69], scene parsing [70], and zero-shot learning [71]. Ouyang et al. [69] investigated the factors that influence the performance in fine-tuning for object detection with long-tailed distributions of samples. Their analysis and empirical results indicate that classes with more samples will pose a great impact on feature learning. It is better to make the sample number more uniform across classes. In [70], much better super-pixel classification results were achieved by expanding the poor classes' samples. Bengio et al. [72] pointed out that poor classes can be beneficial for knowledge learned from semantically similar but richer classes. While in practice, other than learning the transfer features from richer classes, previous work mainly selects or simply replicates some of the data to avoid the potential long-tailed distribution problem. In [69], even if only 40% of positive samples are left out for feature learning, detection performance will improve slightly if the samples are uniform. The issue: To simply abandon part of the data, information contained in these identities may also be omitted. While some sampling heuristics may be applied, they are inefficient as the training procedure is still dominated by richer classes and there is room for further improvement.

5 Conclusion

Object detection is a key ability for most computers and robot vision systems. Although great progress has been observed in the last few years, we still notice that object detection has not been used much in many real-time applications where it could be of great help. Taking into account the speed of the object detection methods, while keeping the detection accuracy has gradually become the current research trend. Region-based methods have achieved good detection accuracy, but cannot satisfy the efficiency requirement in many practical applications. R-FCN is more computationally efficient than Faster R-CNN and well balanced in

detection accuracy and speed. Although proposal-free methods (such as YOLO) can achieve real-time performance, detection accuracy is a concern when compared to region-based methods. SSD improves YOLO by taking into account both detection accuracy and real-time requirements. Finally, we need object detection systems for robots that will explore areas that have not been seen by humans, such as deep sea or other planets, and the detection systems will have to learn new object classes as and when they progressively encounter more objects. In such cases, a real-time open-world learning ability will be critical.

References

1. Haritaoglu, I., D. Harwood, and L.S. Davis, W/sup 4: real-time surveillance of people and their activities. IEEE Transactions on pattern analysis and machine intelligence, 2000. 22(8): p. 809–830.
2. Collins, R.T., A.J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, and P. Burt, A system for video surveillance and monitoring. VSAM final report, 2000: p. 1–68.
3. Geiger, A., P. Lenz, C. Stiller, and R. Urtasun, Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research, 2013. 32(11): p. 1231–1237.
4. Dollár, P., C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. 2009. IEEE.
5. LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998. 86(11): p. 2278–2324.
6. Lowe, D.G. Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on. 1999. Ieee.
7. Dalal, N. and B. Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. 2005. IEEE.
8. Roberts, L.G., Machine perception of three-dimensional solids. 1963, Massachusetts Institute of Technology.
9. Pellegrino, F.A., W. Vanzella, and V. Torre, Edge detection revisited. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2004. 34(3): p. 1500–1518.
10. Marr, D. and E. Hildreth, Theory of edge detection. Proceedings of the Royal Society of London B: Biological Sciences, 1980. 207(1167): p. 187–217.
11. Canny, J., A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, 1986(6): p. 679–698.
12. Rosten, E., R. Porter, and T. Drummond, Faster and better: A machine learning approach to corner detection. IEEE transactions on pattern analysis and machine intelligence, 2010. 32(1): p. 105–119.
13. Harris, C. and M. Stephens. A combined corner and edge detector. In Alvey vision conference. 1988. Manchester, UK.
14. Papageorgiou, C.P., M. Oren, and T. Poggio. A general framework for object detection. In Computer vision, 1998. sixth international conference on. 1998. IEEE.
15. Ojala, T., M. Pietikäinen, and D. Harwood, A comparative study of texture measures with classification based on featured distributions. Pattern recognition, 1996. 29(1): p. 51–59.

16. Ojala, T., M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on. 1994. IEEE.
17. Felzenszwalb, P.F., R.B. Girshick, D. McAllester, and D. Ramanan, Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence, 2010. 32(9): p. 1627–1645.
18. Yan, J., Z. Lei, D. Yi, and S.Z. Li. Multi-pedestrian detection in crowded scenes: A global view. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. 2012. IEEE.
19. Yang, Y. and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. 2011. IEEE.
20. Yan, J., Z. Lei, L. Wen, and S.Z. Li. The fastest deformable part model for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.
21. Huang, K.-Q., W.-Q. Ren, and T. Tan, A review on image object classification and detection. Chinese Journal of Computers, 2014. 37(6): p. 1225–1240.
22. Everingham, M., L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman, The pascal visual object classes (voc) challenge. International journal of computer vision, 2010. 88(2): p. 303–338.
23. Hearst, M.A., S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, Support vector machines. IEEE Intelligent Systems and their applications, 1998. 13(4): p. 18–28.
24. Krizhevsky, A., I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. 2012.
25. Deng, J., A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei, Imagenet large scale visual recognition competition. (ILSVRC2012), 2012.
26. Uijlings, J.R., K.E. Van De Sande, T. Gevers, and A.W. Smeulders, Selective search for object recognition. International journal of computer vision, 2013. 104(2): p. 154–171.
27. Girshick, R., J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
28. Girshick, R. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision. 2015.
29. He, K., X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In European Conference on Computer Vision. 2014. Springer.
30. Ren, S., K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems. 2015.
31. Dai, J., Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems. 2016.
32. Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
33. Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C. Berg. Ssd: Single shot multibox detector. In European conference on computer vision. 2016. Springer.
34. Changpinyo, S., W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
35. Wah, C., S. Branson, P. Welinder, P. Perona, and S. Belongie, The caltech-ucsd birds-200-2011 dataset. 2011.
36. Redmon, J. and A. Farhadi, YOLO9000: better, faster, stronger. arXiv preprint arXiv:1612.08242, 2016.
37. Simonyan, K. and A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
38. He, K., X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

39. Huang, G., Z. Liu, K.Q. Weinberger, and L. van der Maaten, Densely connected convolutional networks. arXiv preprint arXiv:1608.06993, 2016.
40. Jégou, S., M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. 2017. IEEE.
41. Shen, Z., Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, DSOD: Learning Deeply Supervised Object Detectors from Scratch. arXiv preprint arXiv:1708.01241, 2017.
42. Takác, M., A.S. Bijral, P. Richtárik, and N. Srebro. Mini-Batch Primal and Dual Methods for SVMs. In ICML (3). 2013.
43. Zeiler, M.D. and R. Fergus. Visualizing and understanding convolutional networks. In European conference on computer vision. 2014. Springer.
44. Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
45. Szegedy, C., S. Ioffe, V. Vanhoucke, and A.A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In AAAI. 2017.
46. Kim, K.-H., S. Hong, B. Roh, Y. Cheon, and M. Park, PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection. arXiv preprint arXiv:1608.08021, 2016.
47. Huang, J., V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, and S. Guadarrama, Speed/accuracy trade-offs for modern convolutional object detectors. arXiv preprint arXiv:1611.10012, 2016.
48. Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár, Focal loss for dense object detection. arXiv preprint arXiv:1708.02002, 2017.
49. Sung, K.-K., Learning and example selection for object and pattern detection. 1996.
50. Rowley, H.A., S. Baluja, and T. Kanade, Neural network-based face detection. IEEE Transactions on pattern analysis and machine intelligence, 1998. 20(1): p. 23–38.
51. Dollár, P., Z. Tu, P. Perona, and S. Belongie, Integral channel features. 2009.
52. Shrivastava, A., A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
53. Simo-Serra, E., E. Trulls, L. Ferraz, I. Kokkinos, and F. Moreno-Noguer, Fracking deep convolutional image descriptors. arXiv preprint arXiv:1412.6537, 2014.
54. Loshchilov, I. and F. Hutter, Online batch selection for faster training of neural networks. arXiv preprint arXiv:1511.06343, 2015.
55. Wang, X. and A. Gupta. Unsupervised learning of visual representations using videos. In Proceedings of the IEEE International Conference on Computer Vision. 2015.
56. Torralba, A., Contextual priming for object detection. International journal of computer vision, 2003. 53(2): p. 169–191.
57. Sermanet, P., K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.
58. Szegedy, C., S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441, 2014.
59. Gidaris, S. and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In Proceedings of the IEEE International Conference on Computer Vision. 2015.
60. Shrivastava, A. and A. Gupta. Contextual priming and feedback for faster r-cnn. In European Conference on Computer Vision. 2016. Springer.
61. Bell, S., C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
62. Pinheiro, P.O., R. Collobert, and P. Dollár. Learning to segment object candidates. In Advances in Neural Information Processing Systems. 2015.

63. Shrivastava, A., R. Sukthankar, J. Malik, and A. Gupta, Beyond skip connections: Top-down modulation for object detection. arXiv preprint arXiv:1612.06851, 2016.
64. Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision. 2014. Springer.
65. Cai, Z., Q. Fan, R.S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In European Conference on Computer Vision. 2016. Springer.
66. Najibi, M., M. Rastegari, and L.S. Davis. G-cnn: an iterative grid based object detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
67. Gidaris, S. and N. Komodakis. Locnet: Improving localization accuracy for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
68. Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. 2009. IEEE.
69. Ouyang, W., X. Wang, C. Zhang, and X. Yang. Factors in finetuning deep model for object detection with long-tail distribution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
70. Yang, J., B. Price, S. Cohen, and M.-H. Yang. Context driven scene parsing with attention to rare classes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.
71. Norouzi, M., T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G.S. Corrado, and J. Dean, Zero-shot learning by convex combination of semantic embeddings. arXiv preprint arXiv:1312.5650, 2013.
72. Bengio, S. The battle against the long tail. In Talk on Workshop on Big Data and Statistical Machine Learning. 2015.