Object Detection using Deep Learning: A Review

Biponjot Kaur  
Computer Science and Engineering  
UIET, Panjab University  
Chandigarh, India  
biponjotkaur@gmail.com

Sarbjeet Singh  
Computer Science and Engineering  
UIET, Panjab University  
Chandigarh, India  
sarbjeet@pu.ac.in

ABSTRACT

Object detection is one of the most critical and challenging tasks in computer vision. It is the process of finding objects belonging to some predefined categories and determining their location in an image or video. This paper reviews deep learning-based object detection models. The paper discusses some benchmark datasets. The performance evaluation of different detectors on different datasets based on mean Average Precision (mAP) is reviewed. Object detection is used in different fields in different forms. Applications of object detection like pedestrian detection, autonomous driving, face detection, etc., are presented. Finally, the future scope is discussed to work on new techniques for object detection.

KEYWORDS

Object Detection; Deep Learning Performance Evaluation; Benchmark Datasets

1. Introduction

Object detection is currently being widely used in different areas because of its numerous applications. It can be used to detect objects of different kinds like a person, an animal, a car, a building, etc. Two primary tasks in object detection are object localization and object classification [1]. Object localization aims to locate an object's position, and object classification seeks to predict the class of an object. Different application areas where object detection can be used are face detection, pedestrian detection, anomaly detection, computer-aided diagnosis systems, remote sensing object detection, event detection, etc. [2]. The detected objects in these areas can belong to generic type objects such as person, car, pen, etc., or some specific category like Albert Einstein's face, Ferrari, Fountain Pen, etc. An object detector is a model which can be used for the detection of objects. The first object detector, the Viola-Jones Object detector [3], was introduced in 2001, which was the use case for facial detection [4]. It uses Haar-like features and a sliding window to go through the whole image and detect faces in the image [5]. After the Viola-Jones object detector, the Histogram of Oriented Gradients (HOG) detector [6] was introduced in 2005 by N. Dalal and B. Triggs to balance the invariance in the features [5]. With the extension in the HOG detector, the Deformable Part-based Model [7] was introduced in 2008. In traditional methods, Object detection stages can be categorized into three parts (i) region proposals, (ii) feature extraction and (iii) classification [3]. In the first stage, the objective is to find the locations which may contain an object, also known as Region of Interest (RoI). A feature vector can be obtained by using a sliding window on each location of the image to get the information of that region. In the end, the classifiers are trained to categorize the object present in the region [8].

The hand-crafted features used in conventional methods lack the ability of generalization and robustness to illumination change [9]. For this reason, various deep learning methods are proposed to overcome these problems. Deep learning detectors are divided into two families: one-stage detectors and two-stage detectors [1]. OverFeat [10] is the first one-stage deep learning detector used for object detection [8]. With the proposal of Regions with CNN, the first two-stage detector, R-CNN [11], was introduced in 2014 by R. Girshick et al. Since then, deep learning object detection models have evolved rapidly. The two steps used for detection in two-stage detectors are region proposals and region classification. Various extensions of R-CNN, such as Fast R-CNN [12], Faster R-CNN [13], Mask R-CNN [14], etc., were introduced after the popularity of R-CNN and come up with better accuracy and speed. Two-stage detectors have high accuracy but high computation time because of the two stages. For real-time applications, speed is a crucial factor. One stage detectors were proposed to increase the speed of models, such as YOLO [15], SSD [16], DSSD [17], RetinaNet [18], etc. One stage detector combines the region proposal step and classification step to predict the class probabilities [4]. These deep learning models show excellent performance in competitions like VOC PASCAL...
Two-stage detectors divide the object detection task into two parts: (i) Region proposals (ii) class prediction. In the first phase, the detector tries to determine the regions which may contain the object. In the second phase, the object detector classifies the object present in the proposed region into different categories [8]. Some two-stage detectors in deep learning are R-CNN, SPP-Net [24], Fast RCNN, Faster RCNN, RFCN [25], etc.

2.1. R-CNN

Girshick et al. proposed the R-CNN [11] method to enhance the quality of candidate bounding boxes as region proposals [26]. R-CNN uses selective search to generate region proposals. It obtained 53.3% mean average precision, 30% more than the previous best results on PASCAL VOC 2012. Previous models detect objects based on traditional approaches, which are Haar-like features, HOG, etc. R-CNN-based object detection consists of three stages: (i) Proposal Generation, (ii) feature extraction, and (iii) classification [8]. Selective search generates about 2000 region proposals for each image and rejects regions identified as background regions. Then these region proposals are resized into fixed-sized regions and converted into 4096-dimensional feature vectors by the deep convolutional neural network. These extracted features are used to teach bounding box regressors to produce final proposals that contain the object. R-CNN adopts weights of network pre-trained on the ImageNet dataset. The last fully-connected layer was initialized again for the detection part.

To decrease false positives and enhance learning speed, R-CNN rejects easy negatives [8]. The threshold of 0.5 is defined for Intersection over Union (IOU) to set the examples as positive or negative. The region proposals having a threshold greater than 0.5 are considered positive, while below 0.5 are considered negative. Despite various advantages, R-CNN has few disadvantages also:

- Heavy duplicate computation is performed because the proposal features are extracted separately.
- R-CNN performs object detection in three stages which are very time-consuming and takes up ample space as features are extracted from different region proposals.
2.1.2 SPP-Net

A fully connected layer works on fixed-size input. Because of this, the region proposals are cropped to a fixed size. However, the region of interest may be of any ratio, and the information may lose while cropping the image. CNN architecture named SPP-Net [24] is introduced to overcome this problem, which uses Spatial Pyramid Matching (SPM).

Instead of cropping the image, the feature map is calculated for the whole image. The fixed-size feature vector is extracted from the feature map using the spatial pyramid pooling (SPP) layer. The feature map is divided into N x N grid, and the feature vector is extracted by performing pooling on each grid [8]. Then SVM classifier and the Bounding Box Regressor take the extracted feature vectors as input. There is no information loss in SPP-Net, and it has better efficiency than R-CNN because it can work on different image scales.

2.1.3 Fast R-CNN

SPP-net achieves better results than R-CNN; however, it still has some cons. SPP-Net uses the same pipeline as R-CNN to detect the objects. It requires a large amount of space. Fast R-CNN [12] uses the Region of Interest (RoI) Pooling layer to extract the region features. Feature maps are produced from the whole image, and then from each region proposal, a fixed-length feature vector is extracted with the RoI pooling layer.

Further, feature vectors are fed to two output layers: the classification layer and the other one is the regressor layer [8]. Unlike R-CNN, all three steps are optimized end to end without taking extra space to store the features in Fast R-CNN. Fast R-CNN has a better inference speed and can perform detection with greater accuracy than SPP-Net and R-CNN. It obtained 66.9% mean average precision (mAP), greater than R-CNN with 66.0% mAP on the PASCAL VOC 2007 dataset.

2.1.4 Faster R-CNN

Detectors till Fast R-CNN shows improvement in accuracy, but they all use traditional approaches for proposal generation such as selective search or slow edge boxes. To overcome this, Faster R-CNN was proposed after three months of Fast R-CNN proposal [2]. Faster R-CNN [13] generates region proposals using Region Proposal Network (RPN), which efficiently predicts region proposals with different scales and various aspect ratios. RPN takes the arbitrary size of an image, and object proposals are generated on each position of the feature map by sliding over the feature map using the N x N grid. The feature vector is then fed into two layers, the same as in previous detectors [8]. RPN uses anchor boxes to generate region proposals of different aspect ratios. Faster R-CNN achieves high precision and detection efficiency. It achieved 69.9% mAP on PASCAL VOC 2007 with a frame rate of 5 frames per second compared to Fast R-CNN with 66.9% mAP [2]. The different regions share feature extraction computation among them. However, the classification step does not share this computation in which each feature vector goes through the sequence of the fully connected layer. It increases the computation cost as one image can have hundreds of region proposals.

2.1.5 R-FCN

The problem of non-sharing of computation among classification steps is solved by proposing a Region-based Fully Convolutional Network. R-FCN [25] encode relative position information of different classes using Position sensitive RoI Pooling layer (PSROI pooling layer) [8]. It results in achieving better results than Faster R-CNN. Faster R-CNN has difficulty in detecting objects at different scales because it uses a single deep layer feature map. Another method is proposed to address this problem, especially for scale invariance, called Feature Pyramid Network (FPN) [27]. It makes the computation time and memory increase rapidly.

2.1.6 Mask R-CNN

Mask R-CNN [14] is proposed for object detection and instance segmentation by replacing Faster R-CNN RoI with RoIAlign [1]. RoIAlign is designed to preserve pixel-level spatial information [28] and extract a small feature map from each RoI. The quantization operation performed in traditional RoI pooling operation cause misalignment between extracted features and RoI. To overcome this problem, Mask R-CNN uses the RoIAlign layer, which is quantization-free and simple. This change brings a significant improvement in the model accuracy. Mask R-CNN used ResNet101-FPN as the backbone network and achieved the highest results for the COCO dataset for object detection and instance segmentation.

2.2 One-stage detectors

Instead of using two stages like two-stage models, one-stage model detects the object in one step. It does not have a separate stage for region proposals, but it considers every position as a potential object and then classifies it as a target object or background. Two-stage detectors are not very suitable for real-time detection because of the more time taken for detection. One-stage detectors take less amount of time, which is why they are used for real-time detection. So the detectors like YOLO, SDD are widely used nowadays. These methods adopt the idea of regression in which the input image is divided into cells, and each cell determines the class probability [9].

2.2.1 YOLO

Yolo [15], You Only Look Once, is the one-stage object detector that can perform object detection by applying a single neural network to the whole image [5]. YOLO was proposed to use a
whole feature map to predict confidences for multiple classes. The image is divided into an S x S grid, and the object centered in the grid is detected in each grid, and the confidence score is calculated. How likely an object is present in the grid is formally known as the confidence score. The conditional probability p(class|object) is calculated for each bounding box, which is the probability of an object belonging to a particular class among multiple classes. YOLO is faster than the previous two-stage models and runs at 45 FPS in real-time. YOLO is less likely to detect false positives from the image. YOLO sometimes fails to detect small objects in the image because only one object can reside in one grid cell. It predicts the five values in the image: image x, y coordinates, width and height of bounding box, and objectness [4]. YOLO gives 63.4% average precision on the VOC2007 benchmark. Despite high speed than previous two-stage models, it has various disadvantages: (i) it is difficult to detect small objects because, at a time, only up to two objects can be detected in a given location. (ii) Prediction of objects uses only the last feature map that cannot detect objects of various aspect ratios [8].

YOLOv2 [29] is the improved version of YOLO with enhanced speed and precision. YOLO uses a fully connected layer to predict the coordinates of the predicted boxes, but YOLOv3 [30] replaces this fully connected layer with anchor boxes, increasing the recall by 7% [2]. It also comes with various advancements such as batch normalization, dimension cluster, and multiscale training [26].

To perform classification on overlapping labels in various datasets, YOLOv3 uses multilabel classification [2]. YOLOv3 uses a Darknet-53 feature extractor inspired by ResNet [31]. After version 3, two more versions were introduced in 2020, YOLOv4 [32] and YOLOv5, with increased accuracy.

2.2.2. SSD

Single Shot Multibox Detector (SSD) [16] came in 2015 to predict the bounding boxes and offsets using anchor boxes for default bounding boxes at different scales and different aspect ratios [23]. The performance improved with multiscale feature maps and default boxes. The SSD detector also detects small objects that were difficult to detect in YOLO. Authors also use hard negative mining and data augmentation to improve the precision of the model [2]. It achieved 74.3% mAP on VOC2007 at 59 FPS, more significant than the two-step model RCNN and YOLO [28].

The prediction module and deconvolution module are added to SSD, making a different model called Deconvolutional SSD (DSSD) [17]. Resolution of feature map increases with the deconvolution module. Both modules predict objects with different sizes, enhancing mAP with 2.2% on PASCAL VOC 2007 test dataset [2].

2.2.3. RetinaNet

Despite various advancements in one-stage detectors, they had less accuracy than two-stage detectors. With the introduction of focal loss in RetinaNet [18], the model’s accuracy was improved while maintaining high detection speed [4]. Focal loss is the proposal of the loss function, which outweighs background examples and focuses on foreground examples [4]. Class imbalance and inequality between the types of training examples was the reason behind the lesser accuracy of one-stage models. It occurs because one-stage detectors focus on the whole image, resulting in more samples of the background region. RetinaNet works at around 60 FPS [4].

3. Evaluation Metrics

Different evaluation metrics are used in object detection tasks to evaluate models. One of the important metrics is Average Precision (AP). The basics which are used to calculate AP are:

- True Positive (TP) - Correct detection of a ground truth bounding box
- False Positive (FP) - false detection of a nonexistent object
- False Negative (FN) - ground truth bounding box that is undetected

There can be numerous bounding boxes that need not be detected, so True Negative (TN) is not considered to evaluate performance [33]. Based on the predicted box (O_i), the prediction can be correct or incorrect, which is decided using Intersection over Union (IOU) by comparing it with the ground truth bounding box (B_j).

$$IOU(B_i, O_i) = \frac{\text{area}(B_i \cap O_i)}{\text{area}(B_i \cup O_i)}$$ (1)

If this value is greater than the predefined threshold α set as 0.5, then the prediction is correct otherwise incorrect.

3.1. Precision and recall

Precision and recall can be calculated as

$$P = \frac{\text{Correct positives}}{\text{All detections}} = \frac{TP}{TP + FP}$$ (2)

$$R = \frac{\text{Correct positives}}{\text{All ground Truthes}} = \frac{TP}{TP + FN}$$ (3)

Where precision is correct detections for all detections by model and recall is correct detections from all ground truth boxes. The tradeoff between precision and recall is generally present in the model. When the FP is low, its precision will be high, but as a result of this, the FN will be high, resulting in a low recall. A detector can be good if it has high precision and high recall [33].

3.2. Average Precision

The top 11 predictions of a model are used to calculate AP. For a known TP, precision and recall are calculated in each one of the 11 predictions. An IOU value greater than 0.5 is considered a correct prediction. Maximum precision is calculated for recall.
values ranges from 0 to 1. Averaging these precision values give Average precision for a detector [4].

\[ AP_r(x) = \max(P_y) \]

\[ AP = \left( \frac{1}{|T|} \sum_{r \in T} AP_r(x) \right) \]  

(4)  

\[ mAP = \frac{1}{C} \sum_{x=1}^{C} AP_x \]  

(5)  

3.3. Mean Average Precision

Mean Average Precision (mAP) is the average of all calculated average precisions over different classes. MAP is the most popular used metrics for performance evaluation, and it can be calculated as

\[ mAP = \frac{1}{C} \sum_{x=1}^{C} AP_x \]  

AP\(_x\) is the average precision of a particular class \( x \), and \( C \) is the total number of classes [33].

4. Benchmarks

In the past years, several datasets and benchmarks are used to detect different kinds of objects. The most popular benchmarks are MS COCO, PASCAL VOC, ImageNet, and Open Images. PASCAL Visual Object Classes (VOC) [19] dataset has 20 categories: dog, person, cat, etc., and 11000 images with annotations. These images are further classified into different generic categories like animals, persons, household objects, and vehicles [1].

MS COCO [20] comes into competition in 2015 with 91 categories and 328000+ images divided into the training set, validation set, and testing set with 118287, 5000, and 40670 images, respectively [8]. ImageNet [21] has 20 categories with 450000 training images, 20000 validation images, and 40000 images for testing set [2]. Open Images [22] is another benchmark for detection, having 9.2M images with 600 object categories and 15440000 bounding boxes [5]. It is also used to detect the visual relationship between paired objects having some relation between them. The performance evaluation on different benchmarks with different models is given in Table 1 [1, 28].

Table 1: Performance evaluation of object detection models on different datasets

| Method          | Backbone    | Input size | Fps  | VOC 2007 | VOC 2012 | MS COCO |
|-----------------|-------------|------------|------|----------|----------|---------|
| R-CNN           | AlexNet     | 227 x 227  | <0.1 | 58.5     | 53.3     | -       |
| SPP Net         | V5          | 1000 x 600 | <1   | 59.2     | -        | -       |
| Fast RCNN       | VGG-16      | 1000 x 600 | <0.5 | 70.0     | 68.4     | -       |
| Faster RCNN     | VGG-16 (ResNet-101) | 1000 x 600 | 7    | 78.8     | 70.4     | 21.9    |
| RFCN            | ResNet-101  | 1000 x 600 | 9    | 80.5     | 77.6     | 29.9    |

5. Applications

Object detection finds its applications in many disciplines like transportation, medical, security, military, etc. Some of the applications are presented below:

5.1. Pedestrian Detection

It is used to detect pedestrians as an object and further used in surveillance and for autonomous driving. Several datasets are built especially for pedestrian detection, like the Caltech pedestrian dataset [34], UCSD anomaly detection, and Penn-Fudan Database [35]. Pedestrian detection can be difficult because of the occlusion, small pedestrians, and some objects similar to pedestrians [26]. These problems can be tackled with the use of deep learning to a great extent.

5.2. Face Detection

Face detection is used to detect people’s faces in an image, and it can further be used as preprocessing step for recognition and detection of facial expression. The first face detector was Viola and Jones [3], which used traditional features such as Haar-like features [26]. Face detection also has many challenges in detecting faces as there are many variations in people’s faces with different poses and different expressions. Deep learning is currently being used for face detection as it is more accurate and less time-consuming than traditional features.

5.3. Traffic Sign Recognition

With the advancements in technology, autonomous driving can become a reality. Driving without a human needs different techniques to drive on the streets. Various detection modules are used in autonomous driving. The important among them are traffic sign recognition and traffic light detection. The difficulties in detecting traffic signs are (i) bright light (ii) change in weather and (iii) blurriness in the camera.

5.4. Autonomous Driving

Autonomous driving considers various aspects to drive without a human being. Person detection, traffic light detection, traffic sign
5.5. Text Detection

Text detection determines whether there is text present in the image or not. If it is present, then the location of the text is found, and then the recognition process takes place [26]. There are various challenges in text detection; for example, text may belong to any of the many languages used. Moreover, text can have different fonts, which introduces challenges in performing detection. Some text may have some missing characters, and some may have been written very densely, which is further challenging.

6. Discussion and Conclusion

With the increase in technology, previous models are evolving into better and powerful models to fulfill the growing needs. Over the past two decades, object detection achieves huge success in producing accurate and precise models. From two-stage detectors to one-stage detectors, both have their advantages and disadvantages. If time is of prime importance in the task, one-stage detectors are a better option and can be used for real-time detection. The execution speed of these detectors is higher than two-stage detectors. However, if the requirement is high accuracy and computational cost is not an issue, a two-stage detector can be used because of its higher accuracy [28]. There is no superior model; based on the requirements, the detectors can be chosen to balance the tradeoff between speed and accuracy.

Future work can be initiated in various aspects, such as detecting small objects, which is still an issue with current models. Object detection and activity recognition can be performed considering the correlation between video frames. Generative Adversarial Network is an emerging area and can be explored for object detection and activity recognition. This paper has presented the different models used for object detection, including two-stage and one-stage detectors and various performance evaluation metrics.

REFERENCES

[1] Hao Zhang and Xianguang Hong. 2019. Recent progresses on object detection: a brief review. Multimed Tools Appl 78, 19 (October 2019), 27809–27847. DOI:https://doi.org/10.1007/s11042-019-07898-8

[2] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhidi Feng, and Rong Qu. 2019. A Survey of Deep Learning-Based Object Detection. IEEE Access 7, (2019), 128837–128868. DOI:https://doi.org/10.1109/ACCESS.2019.2939201

[3] P. Viola and M. Jones. 2001. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, IEEE Comput. Soc, Kauai, HI, USA, 1-511-1-518. DOI:https://doi.org/10.1109/CVPR.2001.990517

[4] Karanbir Singh Chahal and Kantal Dey. 2018. A Survey of Modern Object Detection Literature using Deep Learning. arXiv:1808.07256 [cs] (August 2018). Retrieved March 2, 2021 from http://arxiv.org/abs/1808.07256

[5] Zhengyou Zou, Zhenwei Shi, Yuhong Guo, and Jiping Ye. 2019. Object Detection in 20 Years: A Survey. arXiv:1905.05055 [cs] (May 2019). Retrieved March 2, 2021 from http://arxiv.org/abs/1905.05055

[6] N. Dalal and B. Triggs. 2005. Histograms of Oriented Gradients for Human Detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), IEEE, San Diego, CA, USA, 886–893. DOI:https://doi.org/10.1109/CVPR.2005.177

[7] P. F Felzenszwalb, R B Girshick, D McAllester, and D Ramalan. 2010. Object Detection with Discriminatively Trained Part-Based Models. IEEE Trans. Pattern Anal. Mach. Intell. 32, 9 (September 2010), 1627–1645. DOI:https://doi.org/10.1109/TPAMI.2009.167

[8] Xiongwei Wu, Dohenhahao, and Steven H. Choi. 2020. Recent advances in deep learning for object detection. Neurocomputing 396, (July 2020), 39–64. DOI:https://doi.org/10.1016/j.neucom.2020.01.085

[9] Wang Zhiqiang and Liu J. A Review of Object Detection Based on Convolutional Neural Networks.

[10] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. 2014. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. arXiv:1312.6229 [cs] (February 2014). Retrieved July 21, 2021 from http://arxiv.org/abs/1312.6229

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, OH, USA, 580–587. DOI:https://doi.org/10.1109/CVPR.2014.81

[12] Ross Girshick. 2015. Fast R-CNN. In 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, Santiago, Chile, 1440–1448. DOI:https://doi.org/10.1109/ICCV.2015.169

[13] Shaqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 6 (June 2017), 1137–1149. DOI:https://doi.org/10.1109/TPAMI.2016.2577031

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2018. Mask R-CNN. arXiv:1706.01568 [cs] (January 2018). Retrieved July 21, 2021 from http://arxiv.org/abs/1706.01568

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1606.02630 [cs] (May 2016). Retrieved June 21, 2021 from http://arxiv.org/abs/1506.02630

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. arXiv:1512.02325 [cs] (2016), 9905, (2016), 21–37. DOI:https://doi.org/10.1109/7978-3-319-46448-0_2

[17] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. 2017. DSSD: Deconvolutional Single Shot Detector. arXiv:1701.06659 [cs] (January 2017). Retrieved July 21, 2021 from http://arxiv.org/abs/1701.06659

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal Loss for Dense Object Detection.arXiv:1708.02002 [cs] (February 2018). Retrieved July 21, 2021 from http://arxiv.org/abs/1708.02002

[19] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. Int J Comput Vis 88, 2 (June 2010), 303–338. DOI:https://doi.org/10.1109/ACCESS.2010.20025

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft COCO: Common Objects in Context.arXiv:1405.0312 [cs] (February 2015). Retrieved July 21, 2021 from http://arxiv.org/abs/1405.0312

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis 115, 3 (December 2015), 211–252. DOI:https://doi.org/10.1007/s11263-015-0816-y

[22] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malolci, Alexander Kolesnikov, Tom Darig, and Vittorio Ferrari. 2020. The Open Images Dataset
V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. Int J Comput Vis 128, 7 (July 2020), 1956–1981. DOI: https://doi.org/10.1007/s11263-020-01316-z

[23] Junwei Han, Dingwen Zhang, Gong Cheng, Nian Liu, and Dong Xu. 2018. Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey. IEEE Signal Process. Mag. 35, 1 (January 2018), 84–100. DOI: https://doi.org/10.1109/MSP.2017.2749125

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2014. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In Computer Vision – ECCV 2014, David Fleet, Tomas Pajdla, Bernt Schiele and Tine Tuytelaars (eds.). Springer International Publishing, Cham, 346–361. DOI: https://doi.org/10.1007/978-3-319-10578-9_23

[25] Efeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. arXiv:1605.06409 [cs] (June 2016). Retrieved July 21, 2021 from http://arxiv.org/abs/1605.06409

[26] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. 2019. Object Detection With Deep Learning: A Review. IEEE Trans. Neural Netw. Learning Syst. 30, 11 (November 2019), 3212–3232. DOI: https://doi.org/10.1109/TNNLS.2018.2876865

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature Pyramid Networks for Object Detection. arXiv:1612.03144 [cs] (April 2017). Retrieved July 21, 2021 from http://arxiv.org/abs/1612.03144

[28] Li Liu, Wanli Ouyang, Xiangang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. 2020. Deep Learning for Generic Object Detection: A Survey. Int J Comput Vis 128, 2 (February 2020), 261–318. DOI: https://doi.org/10.1007/s11263-019-01247-4

[29] Joseph Redmon and Ali Farhadi. 2016. YOLO9000: Better, Faster, Stronger. arXiv:1612.03385 [cs] (December 2015). Retrieved July 21, 2021 from http://arxiv.org/abs/1612.03385

[30] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. arXiv:1804.02767 [cs] (April 2018). Retrieved July 21, 2021 from http://arxiv.org/abs/1804.02767

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs] (December 2015). Retrieved July 21, 2021 from http://arxiv.org/abs/1512.03385

[32] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934 [cs, eess] (April 2020). Retrieved July 21, 2021 from http://arxiv.org/abs/2004.10934

[33] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. da Silva. 2020. A Survey on Performance Metrics for Object-Detection Algorithms. In 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), IEEE, Niterói, Brazil, 237–242. DOI: https://doi.org/10.1109/IWSSIP48289.2020.9145130

[34] P. Dollar, C. Wojek, B. Schiele, and P. Perona. 2012. Pedestrian Detection: An Evaluation of the State of the Art. IEEE Trans. Pattern Anal. Mach. Intell. 34, 4 (April 2012), 743–761. DOI: https://doi.org/10.1109/TPAMI.2011.155

[35] Liming Wang, Jianbo Shi, Gang Song, and I-fan Shen. 2007. Object Detection Combining Recognition and Segmentation. In Computer Vision – ACCV 2007, Yasushi Yagi, Sing Bing Kang, In So Kweon and Hongbin Zha (eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 189–199. DOI:https://doi.org/10.1007/978-3-540-76386-4_17