Deep Learning in Object Detection: A Comprehensive Review, Methodology, and Experimental Analysis

Neha Aggarwal[1], Swati Nigam[1,2*]
[1]Department of Computer Science, Banasthali Vidyapith, Rajasthan, 304022, INDIA
[2]Centre for Artificial Intelligence, Banasthali Vidyapith, Rajasthan, 304022, INDIA
neha.aggarwal2807@gmail.com, swatinigam.au@gmail.com*

Abstract. This paper presents a thorough investigation into the application of deep learning techniques in the domain of object detection. The study begins with an extensive review of traditional methods and the evolution of deep learning architectures in addressing object detection tasks. Motivated by the limitations of conventional approaches, the research outlines a methodology that integrates advanced data preprocessing techniques with state-of-the-art deep learning models, including YOLOv7 and Vision Transformer (ViT). Through a comparative analysis across diverse datasets, such as Pascal VOC 2012, Hard hat workers, Number operands, and PKLot dataset, the performance of these algorithms is rigorously evaluated based on precision, recall, and F1 score metrics. YOLOv7 achieves the highest precision of 0.956 on the PKLot dataset, suggesting to a great choice when false positives are to avoided. ViT, on the other hand achieves the high recall of - 0.903 on the Hard Hat Workers dataset and offers a balanced trade-off between precision and recall on different datasets. Furthermore, the study discusses the strengths and limitations of each algorithm and identifies potential future research directions aimed at enhancing object detection capabilities. Overall, this research contributes valuable insights into the effectiveness of deep learning methodologies and provides guidance for further advancements in object detection systems.

Keywords: Deep Learning, Object Detection, YOLOv7, Vision Transformer, Experimental Analysis

1 Introduction

In digital images, detecting and localizing various objects is a crucial challenge in computer vision. The objective is to enable computer vision technology to identify examples of particular class categories, such as people, animals, or vehicles. The essence of the challenge is to address the queries, “What objects exist?” and “How many of each are present?” The primary measurable metrics of performance are processing time and the precision of classification and localization [1, 2].

Moreover, for many applications in computer vision such as image captioning, instance segmentation, and object tracking, the initial and fundamental building block is object detection. The most recent advances in deep learning and the rapid growth and interest of the global research community have led to a proliferation of interest in the

© The Author(s) 2026
A. Shastri et al. (eds.), *Proceedings of the Fourth Workshop on Applications of Computational Intelligence and Interactive Systems (WACIIS 2026)*, Advances in Intelligent Systems Research 217,
https://doi.org/10.2991/978-94-6239-832-0_15

field. Within many domains, including computer vision, object detection remains a foundational technology and is applied to a wide range of applications including video surveillance, robotics, and driver less cars [3].

1.1 Background and motivation

With respect to the field of computer science, computer vision especially self-driving cars, security camera systems, and similar technologies are extremely useful and popular thanks to their ability to detect objects. One of the most challenging functions of computer vision and to some of the most advanced and expensive functions. Conventionally, attempted detection of objects has utilized numerous advanced algorithms and designed features; some competitors provide better similar end results that are less expensive and far simpler to scale. Traditional detection of objects has utilized numerous advanced algorithms and painstakingly designed features. There has traditionally been a strong reliance and robustness on extensiveness provided by algorithms. There remain advanced algorithms and features designed for deep learning. From All of the extensively and painstakingly designed features that were robustly complex and were constructed, there still exists scalable results that are both simpler to procure and superior. The aim of this document is to provide detailed documentation on the detection of objects to provide a basis for advanced deep learning techniques to be used to stimulate discourse in this specific area and to positively propose further research surrounding optimized deep learning to provide better pathways for computer vision technologies to provide for the future.

1.2 Objectives

Goals of the research are as follows: -

1. An aim is to analyze the evolution of the techniques of object detection from the classical methods of detection using hand-crafted techniques to recent methodologies involving deep learning.
2. An aim is to examine the extent to which deep learning, especially end to end learning, has affected the overall performance of object detection.
3. Address the major techniques and architectures of deep learning used in object detection, and in particular, CNN, R-CNN variants, SSD and YOLO.
4. Explain how the existing systems for object detection work in detail as possible.

1.3 Overview of deep learning in object detection

We focus on the transition from feature extraction by hand to complete learning systems in deep learning-based object detection covering two primary models Convolution neural and recurrent neural networks and major breakthroughs of the future like RCNN variations, SSD, and YOLO which are responsible for attaining fast and accurate object detection. The aim is to explain how these models and procedures fostered the success of deep learning in object detection.

1.4 Contributions

The paper contributes the following:

1. Proposal of the unique object detection framework that incorporates hyperdata preprocessing methodologies and the most efficient architectures.
2. Extensive comparative studies of the most significant models of object detection based on deep learning that demonstrate their advantages, disadvantages, and areas of application.
3. Practical advices on training and fine-tuning of reasonably efficient object detection subsystems.

2 Literature Review

Innovations in object detection have seen a paradigm shift in the last two decades. There has been a switch from a model dependent on customized feature engineering to fully automated deep learning. This report summarizes the evolution of object detection systems focusing on the significant changes in paradigm and the impacts from the multitude of scientists working on the systems.

2.1 Handcrafted Feature-Based Object Detection

In the early years of innovation for object detection systems, there was great reliance on the engineering of systems within a framework and subsequent application of use of search techniques on the computational window. The first model for object detection was the Viola-Jones face detector. This detector achieved real-time face detection in 2001 using Haar-like Features, integral images and a cascade of classifiers [4, 5]. The face detection was real-time and computationally energizing due to the presence of a high-speed microprocessor, thus detection was achieved without the introduction of delays on subsequent processing tasks. The system did achieve real time detection of faces as objects, but due to an extensive slide window technique, the system was ill suited for more complex objects.

In 2005, Dalal and Triggs [6, 7] introduced the Histogram of Oriented Gradients (HOG), the first of which was specifically designed to detect pedestrians, capturing local gradient features that are insensitive to lighting and/or small geometric changes. The use of HOG features in a linear classifier allowed those models to generalize well and become the primary workhorses of the deformable part-based models (DPMs) [8–10]. However, those models, which combine HOG features with linear classifiers, are ultimately inefficient, leading to massive compute waste, and hence became inadequate, paving the way for a shift to more data driven approaches.

2.2 Transition to Deep Learning-Based Detection

2012 saw the arrival of Convolutional Neural Networks (CNN) [11], which became a game changer in the rekindling of the object detection challenge as they automated

feature engineering and learned hierarchical abstractions of data at different levels from the raw pixels. The arrival of CNNs allowed a dramatic increase in the accuracy of detection of objects across classes and within classes (intra-class) as well.

The inception of Region-Based Convolutional Neural Networks (R-CNN) (Girshick et al. 2014) [12] remains an equally important evolution in RCNN-based detection. R-CNN pioneered the use of region proposal in conjunction with feature extraction deepening to mAP advancements. The new target became the overlap in proposals with redundant feature computation. The drawback of the new target was the increased inference latency. R-CNN's efficiency was overshadowed in [3] Spatial Pyramid Pooling Networks (SPPNet) and [13] Fast R-CNN. The overshadowing was due to the introduction of shared convolutional computation and end-to-end training.

Eventually, [14] Faster R-CNN became the culmination of this two-tier detection system. The introduction of a Region Proposal Network (RPN) into the detection pipeline created a new balance of high detection accuracy and near real-time speed. The new balance became the foundation for many new works including [15] FPN and [16] RetinaNet with the added contribution of improved multi-scale object detection and focal loss for class imbalance.

2.3 One-Stage Detectors and Real-Time Object Detection

The emergence of single-stage detectors developed concurrently with two-stage detectors. The single-stage detectors have a greater emphasis on speed and simplicity for practical real-world applications. Therefore, it is a highly preferred alternative.
The algorithm described in [17] operates in a single stage termed YOLO, which redefines the detection task as a single regression problem. Thus, this achieves the capability of drawing inferences in real time. Earlier versions of YOLO had poor localization accuracy, a trade-off which later versions sought to improve in the YOLO architecture, training methods [18, 19] and tiered variants. One stage ennoblement in detection was first popularized by Single Shot Multi-Box Detector or SSD [20] which harnessed the power of multi scale feature maps and default anchor boxes to yield faster run times and admirable accuracy. SSD was a prominent contributor of small object detection advancement and feature alignment [21, 22] model which combined system robustness and accuracy in scenarios with multiple objects.

2.4 Summary and Research Gap

All in all, deep learning integrates detection systems with more dynamically flexibly learned features. While the systems with deep learning may be real time operationally and efficient, more issues arise in other areas when trying to scale in size, generalize in the matrix of environment variables and optimize in environment variables and optimize in computation. These still ongoing and unresolved issues in deep learning systems and object detection in deep learning mirrors ongoing applied explanations to different system architectures, and hybrid frameworks to meet accuracy and operational performance for continued efficient use of detection systems.

3 Methodology

This section discusses the methodologies employed in this research study focusing on the techniques of data preprocessing, deep learning model architectures, the training and-optimization methods, and the assessment metrics to evaluate the performance of object detection.

3.1 Data Preprocessing

Adequate data preprocessing is essential to enhance the model's generalization ability and detection performance.

All four datasets used in this study (Pascal VOC 2012, Hard Hat Workers, Number Operands, and PKLot) were obtained from Roboflow in an already pre-processed, ready-to-train form. Roboflow's dataset export pipeline had, prior to download, already applied image augmentation (rotation, translation, and horizontal flipping), pixel-level normalization/resizing, and a fixed train/validation/test split to each dataset. These augmentation and split operations were therefore inherited from the Roboflow export rather than performed by the authors. The study focuses on detecting objects in annotated images, not predicting tabular features. The bounding-box annotations supplied with each dataset were used as provided, without modification.

3.2 Deep Learning Model Architectures

The object of this research is to analyze two of the most popular deep learning architectures for object detection, YOLOv7 and the Vision Transformer (ViT).

YOLOv7 Architecture YOLOv7 (You Only Look Once version 7) is a pinnacle one-stage object detection model. It is designed to be highly efficient, and thus, is ideal for real-time detection. It processes input images by dividing them into a grid and calculating bounding box coordinates and determining the class associated at the corresponding pixel in realtime. All of this is accomplished via a single model, and because of the architecture, it can easily be trained and fine-tuned for different applications. This versatility makes it ideal for real-world applications.

Vision Transformer (ViT) Architecture: The Vision Transformer employs a transformer-based approach, modelling images as ordered sequences of a predetermined patch size. Each patch is embedded before processing through the self-attention mechanism of the transformer, which allows the model to retain long-range spatial information. ViT has shown remarkable effectiveness in a variety of visual recognition challenges, providing a valuable alternative view of models that rely on convolution for object detection.

Reason for model selection: YOLOv7 and Vision Transformer are selected for the evaluation as both are based on different architectures. YOLOv7, is a one-stage detector and uses Convolutional Neural Network (CNN) for object detection, whereas ViT follows a transformer-based architecture and processes images in patches and to create

such patch, ViT uses self-attention mechanism. The study tries to compare the two models under the same experimental conditions, under a consistent set up, using the same default parameters and same datasets.

Training Process: The complete training process combines the aspects of data preprocessing and model-specific detection pipelines. For the YOLOv7 model, the pre-processed image data moves through a series of convolutional layers for feature extraction, then moving onto the bounding box per prediction step, and finally through a process of non-max suppression to remove repeated detections.

For the ViT model, the images are separated into small patches which are then embedded, and run through the transformer layers which help to generate the set of closely related images. These are then changed into the detection head that predicts the classes of the objects, in addition to the bounding boxes, non-max suppression is also applied to remove redundant detections in the final set of detections.

The data flow pipelines for the YOLOv7 and for the ViT object detection frameworks are shown in Fig.1.

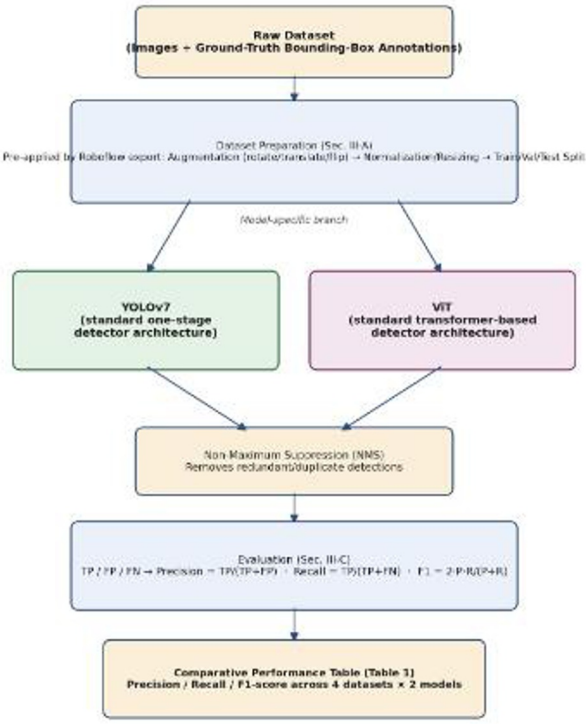

Fig. 1 Overall workflow of the proposed object detection methodology, illustrating data preprocessing, model selection, YOLOv7 and ViT pipelines, and performance evaluation

3.3 Evaluation Metrics

To quantitatively evaluate object detection performance, the following metrics are used. The Figure 2 presents Accuracy metrics assess the model's ability to properly identify the objects, recall measures the ability to detect all relevant objects, and the F1 score is a balanced measure that incorporates both recall and precision. All of these metrics provide a comprehensive measure of performance in the area of object detection.

3.4 Reproducibility and hyperparameter configuration

For reproducibility, Table 1 brings together the training settings shared by both frameworks and the architecture-specific default hyperparameters for YOLOv7 and ViT-Base/16. Every experiment in our study uses a fixed 50 epochs per dataset per model. This is the only training-schedule parameter explicitly reported for all runs.

Table 1: Reproducibility and hyperparameter configuration reference

Configuration item	YOLOv7	ViT	Status
Training epochs (all datasets)	50	50	Fixed – as reported in Table 1
Loss function	Classification + bounding-box regression (Eq. 1)	Classification + bounding-box regression (Eq. 1)	Fixed – identical formulation
Optimizer	SGD	AdamW	Standard default optimizer per architecture
Momentum/ Weight decay	Momentum 0.937, weight decay 0.0005	0.001	Architecture default
Batch size	16	16	
Regularization	Dropout, batch normalization, weight decay	Dropout (0.1), batch normalization, weight decay	Fixed
Learning rate / batch size	0.01 (lr0)	0.001 (typical fine-tuning rate)	Architecture Default
Input image size	640 * 640	224 *224 (patch size 16 *16)	
Post-processing	Non-Maximum Suppression (NMS)	Non-Maximum Suppression (NMS)	Fixed
Preprocessing pipeline	Dataset augmentation, normalization/resizing, and train/val/test split – all pre-applied by Roboflow at export	Dataset augmentation, normalization/resizing, and train/val/test split – all pre-applied by Roboflow at export	Fixed – identical for both models (Sec. III-A)

4 Experimental Setup

4.1 Description of datasets used

This research assesses performance object detection across four different datasets for different use cases that were all obtainable from Roboflow. These datasets contain labeled images of handwritten characters and figures [23] (Number Ops); images from industrial safety and safety violations (Hard Hat Workers (https://public.roboflow.com/object-detection/hard-hat-workers)); images from parking lot vehicle detection (PKLot (https://public.roboflow.com/objectdetection/pklot)); and images from medical scans where a brain tumor was being located (Brain Tumor M2PBP [24]). All of the datasets offered bounding-box annotations and were checked to have undergone some combination of augmentation, normalization, or resizing to strengthen performance for the purposes of generalization. Each dataset was split into train/validate/test subsets so that each of the proposed object detection techniques could be thoroughly and fairly assessed for performance across different datasets.

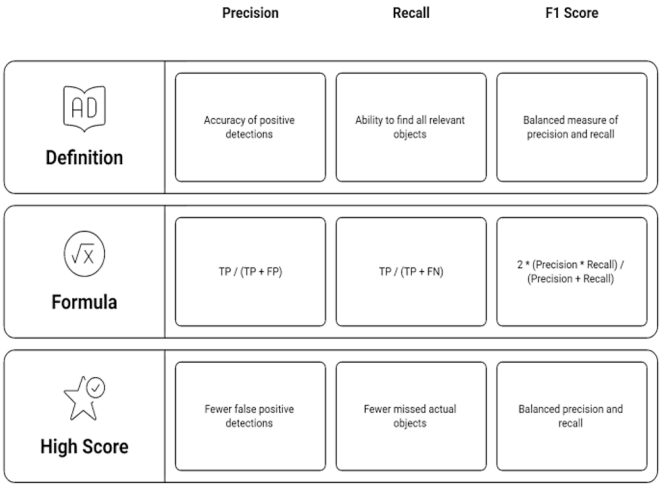

Fig. 2 Comparison of Precision, Recall, and F1-score for object detection evaluation

4.2 Hyperparameters and training details

For YOLOv7 and ViT frameworks, the default hyperparameters like learning rate, batch size, epoch count and other model architecture specific hyperparameters and for YOLOv7, network architecture, the default values of - anchor box sizes, aspect ratio, and momentum for learning rate are used. The table 1 consists of all the values corresponding to the hyperparameters. Each dataset will be divided into train, validation and test, and we will train the model for an optimized number of epochs applying forward and backward propagation. The loss function will be the sum of the losses from the object detector, bounding box regressor, and the confidence loss. The model will be

regularized with dropout, batch norm, and weight decay to avoid overfitting as we seek to train a model with more generalizable solutions.

$$L = L_{cls} + \lambda L_{reg} \quad (1)$$

where L_{cls} denotes the classification loss, L_{reg} represents the regression loss, and λ is a regularization parameter.

Table 2: Precision, Recall and F1 score comparison of all 4 datasets using Yolov7 and ViT.

Dataset	Images	Methodology	Epoch	Precision	Recall	F1 Score
Pascal Voc 2012	17112	Yolo v7	50	.703	.718	.7095
Hard hat workers	6020	Yolo v7	50	.831	.729	0.777
Number oper-ands	7126	Yolo v7	50	.781	.729	0.754
PKLot dataset	12416	Yolo v7	50	.956	.934	0.9448
Pascal Voc 2012	17112	ViT	50	.852	.783	.814
Hard hat workers	6020	ViT	50	.851	.903	.871
Number oper-ands	7126	ViT	50	.932	.884	.902
PKLot dataset	12416	ViT	50	.862	.791	.824

5 Results and Discussion

5.1 Analysis of experimental results

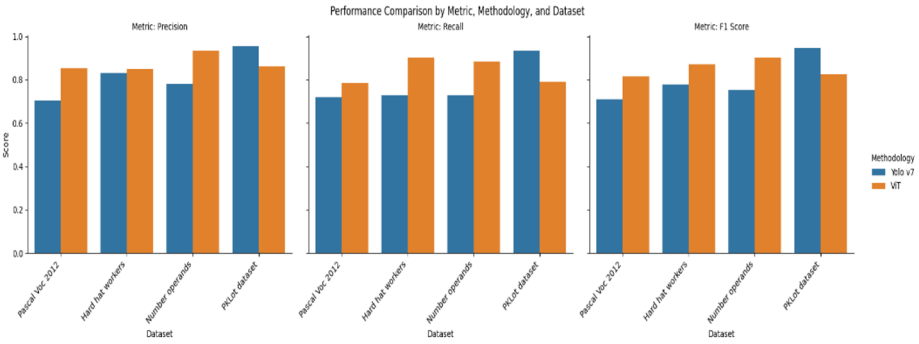

Fig. 3 Performance Comparison: Yolo v7 vs ViT

The Table 2 presents a comparative performance analysis of YOLOv7 and Vision Transformer (ViT) architectures across four datasets: Pascal VOC 2012, Hard Hat Workers, Number Operands, and PKLot, with all experiments conducted for a fixed 50 epochs. The three metrics – precision, recall and F1-score was considered for evaluation

of the models on four different datasets. Based on the results from Table 2, YOLOv7 is seen to have precision values ranging from ranging from 0.703 on PASCAL VOC to 0.956 on PKLot dataset, recall values ranging from 0.718 on PASCAL VOC to 0.934 on PKLot dataset, and F1-scores values ranging from 0.7095 on PASCAL VOC and 0.9448 on PKLot dataset, displaying strong results towards precision. In contrast, ViT demonstrates comparatively higher and more stable recall values ranging from 0.783 on PASCAL VOC to 0.903 on Hard Hat workers dataset, and precision values ranging from 0.851 on Hard hat Workers and 0.932 on Number operands datasets and F1scores ranging from 0.814 on PASCAL VOC to 0.902 on Number Operands dataset, reflecting better balance between precision and recall. The F1-score visualizations further go with the trends, showing that YOLOv7 shows the highest-precision, while ViT offers improved recall and balanced detection performance across datasets. Overall, the results in Table 2 shows that both the models are effective for object detection, with performance dependent on the various datasets, various parameters, the number of images, the number of classes etc.

5.2 Discussion on strengths and limitations

The results for all the datasets are represented in Fig.2 which are there to support the conclusions. YOLOv7 performs best on parking Lot dataset, achieving a precision of 0.956 and F1-score of 0.9448. The images in parking lot dataset are mostly having two conditions either fill or vacant and this limited visual variation may help in reducing the false positives.

The better choice depends on the task: YOLOv7 is preferred when precision is the main concern and visual constraints must be taken into account, whereas ViT suits recall-critical applications involving visually heterogeneous data.

Including the metrics, the three metrics- Precision, Recall and F1-score are already included in this study. We'll try to incorporate or consider Map score for the future papers.

6 Conclusion and Future Work

This paper analyzes the object detection capabilities of YOLOv7 and Vision Transformer (ViT) using various datasets and several different evaluation metrics. YOLOv7 is more applicable and precise in instances where false positives are extremely detrimental. YOLOv7 is also more user-friendly in justifying positive object detection, whereas the user has more options of the objects that the user might justify the detection of an object that is relevant. This is the difference in recall and the higher the F1 score suggests that for varying detection tasks the selection of detection model is not as critical as for one of the other metrics. There is also the opportunity to build on YOLOv7 and ViT the architecture from more advanced object detection systems to hyperparameter tuning, model architecture optimization, and pre-processing to transfer learning and ensemble techniques with streaming for robust model that adjusts with relative ease

for the task domain to focus on. All of which come together to provide a more effective model that is precise, adaptable, and rapid for the real-world challenges of object detection systems to be more widely deployed.

References

1. Hariharan, B., Arbelaez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: European Conference on Computer Vision, pp. 297–312 (2014). Springer
2. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: CVPR, pp. 3150–3158 (2016)
3. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV, pp. 346–361 (2014). Springer
4. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, vol. 1, p. (2001). IEEE
5. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision 57(2), 137–154 (2004)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005). IEEE
7. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV, pp. 89–96 (2011). IEEE
8. Felzenszwalb, P.F., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR, pp. 1–8 (2008). IEEE
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Cascade object detection with deformable part models. In: CVPR, pp. 2241–2248 (2010). IEEE
10. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence 32(9), 1627–1645 (2010)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587 (2014)
13. Girshick, R.: Fast r-cnn. In: ICCV, pp. 1440–1448 (2015). IEEE
14. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
15. Lin, T.-Y., Dollar, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR, vol. 1, p. 4 (2017)
16. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR, pp. 779–788 (2016)
18. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
19. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: Yolov7: Trainable bag-offreebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696 (2022)
20. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV, pp. 21–37 (2016). Springer

21. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
22. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 850–859 (2019)
23. 100, R.: number ops Dataset. Roboflow. visited on 2024-04-07 (2023). https:// universe.roboflow.com/roboflow-100/number-ops
24. Almeida, P., Oliveira, L.S., Silva Jr, E., Britto Jr, A., Koerich, A.: Pklot–a robust dataset for parking lot classification. Expert Systems with Applications 42(11), 4937–4949 (2015)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

