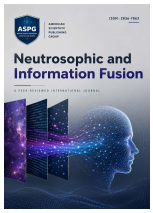
Neutrosophic and Information Fusion

♦ ISSN: 2836-7863 Vol: 03, No. 01, 2024 ♦ PP. 12–19

Article DOI: https://doi.org/10.54216/NIF.030103



FULL LENGTH ARTICLE

On The Analysis of Some Deep Learning Algorithms for Object Detection and Applications

Sandy Montajab Hazzouri[1,*]

[1] Faculty of Informatics Engineering, Albaath University, Syria

Email: Samonhaco1994@gmail.com

Received: April 29, 2023 Accepted: January 10, 2024 ⋆ Corresponding author

ABSTRACT

The traditional methods of discovering objects no longer meet the requirements of the times as a result of their reliance on non-dynamic methods and their slow performance in light of the world's dependence on a huge amount of multimedia and social media. The rapid development of deep learning provides more powerful tools capable of manipulating high-level and complex semantic features of objects. Several techniques have been developed to detect objects using deep learning algorithms. This research presents a comparative analysis of the most famous deep learning techniques for object detection, explaining their mechanisms and use cases and providing an experimental evaluation of their performance.

Keywords: Algorithm ▪ Deep learning ▪ Object detection ▪ Model

1. INTRODUCTION

CNN deep networks significantly improve image classifi-cation and increase the accuracy of object detection. The process of detecting objects is more difficult than classifying images, so it requires more complex methods. This com-plexity arises because the detection process requires accurate positioning of objects, while traditional methods of training models are usually slow and inaccurate. This creates two main challenges:

1. Several locations of candidate objects must be processed; these are often called "proposals."
2. Only an approximate location can initially be obtained, and it must then be adjusted to achieve a more accurate determination.

The solutions to these challenges often affect the speed, accu-racy, or simplicity of the design. In this research, we compare the most commonly used techniques for detecting objects.

2. RESEARCH OBJECTIVE

This research aims to provide a comparative analysis of the most popular object-detection technologies based on deep learning. It provides a simplified explanation of each technique, including its use cases, strengths, and mechanism of action, and then conducts experiments to measure and compare performance.

3. PREVIOUS STUDIES

- The research paper by Girshick et al. [1] discusses the possibility of using a pre-designed CNN network to classify images and adding the task of detecting objects to it. The paper concludes that it is possible to develop a simple system that improves mean average precision by more than 40%. The framework combines powerful computer-vision techniques for generating bottom-up region proposals with recent advances in CNN learning. The experiments provide insight into what the network learns to represent, revealing a rich hierarchy of discriminating and often meaningful features.

© 2024 The Author(s). Published by ASPG. This article is licensed under CC BY 4.0.

- Redmon et al. [2] present a different way to perform object-class detection using CNN networks that is much faster than contemporary alternatives. Many modern object-detection systems are based on object proposals. This paper instead divides the image into grid cells and studies both the probability that an object is present in each cell and the bounding-box regression for each object. Speed is the main advantage because the network processes the image only once. Accuracy is about 7% lower than that of other techniques, but inference is much faster at 45 frames per second.
- Wu et al. [3] provide a comprehensive survey of recent developments in visual-object detection using deep learning. They systematically analyze existing object-detection frameworks and organize the survey into three main parts: detection components, learning strategies, and applications and benchmarks. The survey covers detector architectures, feature learning, proposal generation, sampling strategies, and several future research directions.
- Jiao et al. [4] provide a comprehensive review of research related to video-object detection. The work systematizes recent research using representative models, explains the differences and connections between video-object detection and similar tasks, and presents evaluation measures for approximately 40 models on two datasets. It also discusses applications and challenges in video-object detection.

4. DEEP LEARNING

Before discussing object-detection technologies based on deep learning, this section reviews deep learning and introduces the structure and advantages of convolutional neural networks (CNNs).

4.1 Introduction to Deep Learning

Deep models can be described as neural networks with deep structures, meaning that they include a larger number of hidden layers. The history of neural networks can be traced back to the 1940s [5]. Deep learning became widely popular after 2006 [6]. Its prevalence can be attributed to the following factors:

- The appearance of large amounts of annotated training data, such as ImageNet [7].
- The development of high-performance computing systems, including GPU and TPU clusters.
- Significant developments in network structures and training strategies that make the training of deep neural networks effective.

Various network architectures, including AlexNet [8], OverFeat [9], GoogLeNet [10], VGG [11], and ResNet [12], have been extensively studied to improve performance.

4.2 CNN Structure and Features

A convolutional neural network, also known as ConvNet or CNN, is a type of feed-forward neural network used for tasks such as image classification, natural-language processing, and other complex pattern-recognition problems [13]. It can identify, detect, and interpret patterns in images and text. Convolutional neural networks examine one patch of an image at a time and progressively scan the image to extract complete information [8].

4.2.1 CNN Layers

The principal CNN layers are [13]:

- convolutional layer;
- rectified linear unit (ReLU) layer;
- pooling layer;
- normalization layer; and
- fully connected layer.

5. OBJECT-DETECTION TECHNIQUES USING DEEP LEARNING

Object detection is a computer-vision technology that identifies and locates objects in an image or video. It can be used to count objects in a scene, pinpoint their locations, and track them while assigning accurate labels. For example, in an image containing two cats and a person, object detection classifies the types of objects and determines where they occur in the image.

5.1 What Is Object Detection?

Object detection is a computer-vision technique for recognizing and locating objects within an image or video. Specifically, it draws bounding boxes around detected objects, allowing their positions, and potentially their movement, to be determined within a scene.

5.2 Difference Between Image Classification and Object Detection

Image classification and object detection are important computer-vision methods. These technologies help machines understand and identify objects and environments in real time using digital images as input.

5.2.1 Image Classification

Image classification is used to classify or predict the category of a particular object in an image. Its main goal is to identify image features accurately. In general, image-classification techniques can be categorized as parametric or non-parametric and as supervised or unsupervised. Supervised classification produces results according to learned decision boundaries that depend on the inputs and outputs provided during model training. In unsupervised classification, the result is instead based on analysis of the input dataset without directly supplied labels.

The principal steps in image classification are: determining a suitable classification system, extracting features, selecting good training samples, preprocessing images, choosing an appropriate classification method, applying post-classification processing, and evaluating overall accuracy. The input is usually an image of a specific object, and the output is the predicted category that identifies the object. CNNs are among the most common neural-network models used for image classification [14].

5.2.2 Object Detection

The object-detection problem requires determining both where objects are located in an image and the category to which each object belongs. Thus, object detection combines image classification with localization. It can search for categories such as cars, people, animals, and birds and has been

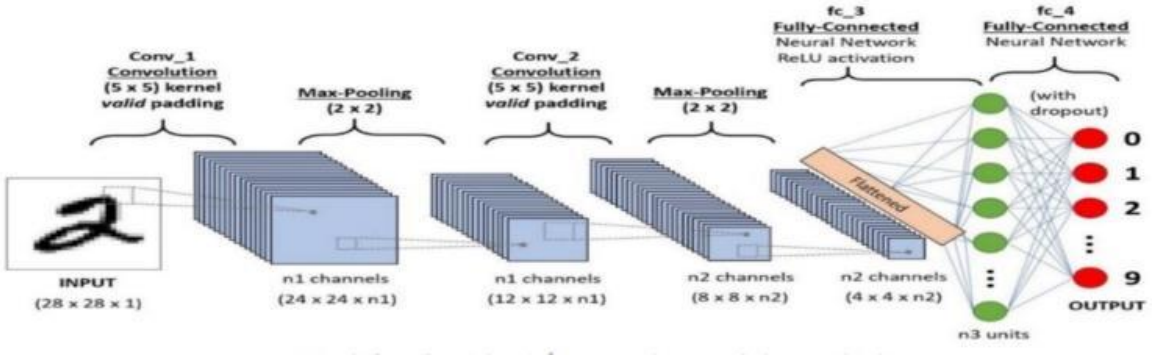


Figure 1. CNN sequence for classifying handwritten numbers [8].



Figure 2. An example of object detection.

successfully used in video-processing systems. Recent developments in this area have been enabled by deep learning.

Conventional object detection can be divided into three main stages: selection of an information-containing region, feature extraction, and classification.

5.3 Stages in the Development of Object Detection

Object detection has evolved over the past two decades. Its development is commonly divided into two historical periods: before and after the introduction of deep learning.

Before 2014, traditional methods included:

- the Viola–Jones detector (2001), a pioneering method that initiated the development of object-detection techniques [15];
- the HOG detector (2006), which introduced widely used features for object detection in computer vision and image processing [16]; and
- deformable part models (2008), which introduced bounding-box regression.

Important two-stage deep-learning algorithms developed after 2014 include:

- R-CNN and SPP-net (2014) [1];
- Fast R-CNN and Faster R-CNN (2015) [17, 18];
- Mask R-CNN (2017) [19];
- feature pyramid networks (FPN, 2017) [20]; and
- G-RCNN (2021) [21].

Important single-stage algorithms include:

- YOLO (2016) [2];
- SSD (2016) [22];
- RetinaNet (2017) [23];
- YOLOv3 (2018) [24];
- YOLOv4 (2020) [25]; and
- YOLOR (2021).

Modern object-detection methods can therefore be classified into two main types: single-stage and two-stage techniques. Deep-learning-based methods generally extract features from the input image or video frame and solve two consecutive tasks:

1. find a variable number of objects; and
2. classify each object and estimate its size and location with a bounding box.

These tasks can be separated into two stages, whereas single-stage methods combine both tasks in one step to obtain higher speed, often at the expense of accuracy.

5.4 Two-Stage Object Detection

Two-stage algorithms use one model to extract object regions and a second model to classify the objects and refine their localization. Such methods are relatively slow but powerful. Advances such as feature sharing have reduced their computational cost and brought it closer to that of single-stage models. These methods depend strongly on preceding work, so understanding the main two-stage algorithms is important.

5.4.1 R-CNN

The 2014 R-CNN paper [1] proposes a straightforward CNN-based two-stage object-detection algorithm that was improved and accelerated in subsequent work. As shown in Figure 3, its workflow consists of three stages:

1. generate category-independent region proposals in which objects are expected to occur;
2. use a convolutional neural network to calculate features from each candidate region; and
3. classify the regions using support-vector machines (SVMs).

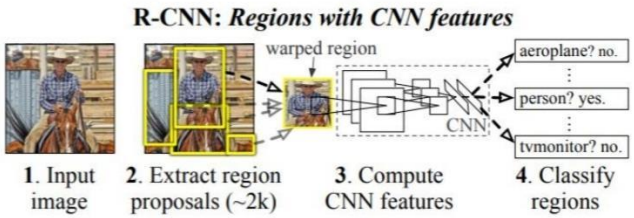


Figure 3. Overview of the stages of R-CNN.

Region proposals can be created using several methods; R-CNN uses selective search [26]. In selective search, a segmentation algorithm is applied to the image and region proposals are generated from the resulting segmentation map. The regions are repeatedly merged in a bottom-up process, gradually producing larger proposals, as shown in Figure 4.



Figure 4. Selective search.

Selective search proceeds as follows:

1. similarities in color, texture, area, and filling are used for object-independent segmentation, producing many small fragmented regions;
2. a bottom-up approach combines small fragmented regions into larger regions; and
3. several candidate bounding-box region proposals are generated.

After selective search, each bounding box is passed to the feature-discovery stage. A geometric correction is first required because the CNN accepts a fixed-size input, whereas selective search produces proposals of different sizes. All proposed regions are therefore warped to squares. The original R-CNN paper uses an AlexNet-style CNN in the second stage, although other CNN architectures can also be used.

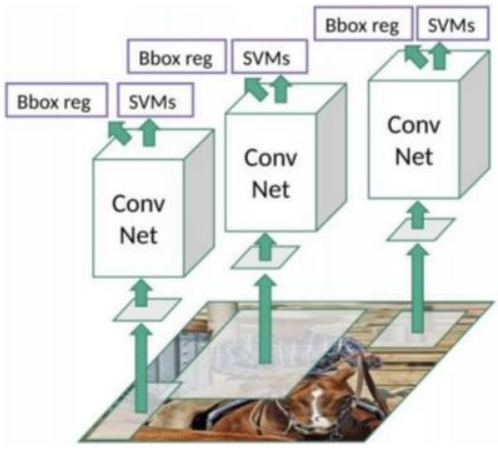


Figure 5. R-CNN operation.

The proposed regions are individually fed to the CNN convolutional layers to extract features. Fully connected layers then predict the object category. After prediction, bounding-box regression refines the object location. To reject overlapping proposals in which multiple boxes point to the same object, a proposal is suppressed if it has a high intersection over union (IoU) with another proposal that has a more confident prediction.

R-CNN outperforms earlier methods because it combines bottom-up selective search with CNN features to improve localization accuracy. However, R-CNN has three principal limitations:

- training is time-consuming because approximately 2,000 region proposals must be classified for each image;
- it cannot operate in real time, taking approximately 47 seconds per test image in the original implementation; and
- selective search is a fixed algorithm, so no learning occurs at that stage and poor candidate regions may be generated.

5.4.2 Fast R-CNN

Fast R-CNN [17] addresses several disadvantages of R-CNN. Rather than submitting each region proposal separately to the CNN, the input image is passed through the CNN once to create a feature map. Region proposals are then identified on the feature map and reshaped to a fixed size using an RoI pooling layer. The resulting RoI feature vector is fed to fully connected layers. A Softmax layer predicts the class of each proposed region, while a parallel regression output predicts bounding-box offsets.

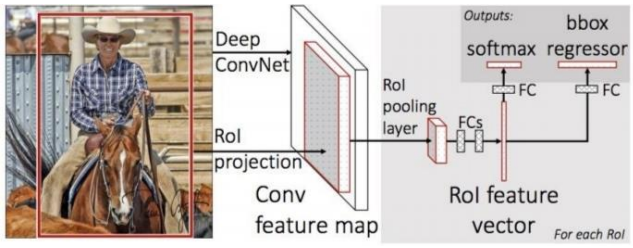


Figure 6. Fast R-CNN.

Fast R-CNN is faster than R-CNN because the convolutional operation is performed only once for each image. Its main contributions are:

1. an RoI pooling layer that extracts equal-length feature vectors from all regions of interest;
2. a single-stage training network, in contrast to the separate proposal, feature-extraction, and SVM-classification stages of R-CNN;
3. shared convolutional computations across all RoIs;
4. no need to cache extracted features, greatly reducing disk-storage requirements; and
5. improved accuracy relative to R-CNN.

The RoI pooling layer divides each region proposal into a grid of cells and applies max pooling to each cell. All cell values form the feature vector. For example, if the grid size is 2×2, the feature-vector length is four. This vector is passed through fully connected layers, after which the network divides into two branches: a Softmax layer for class prediction and a fully connected layer for bounding-box prediction.

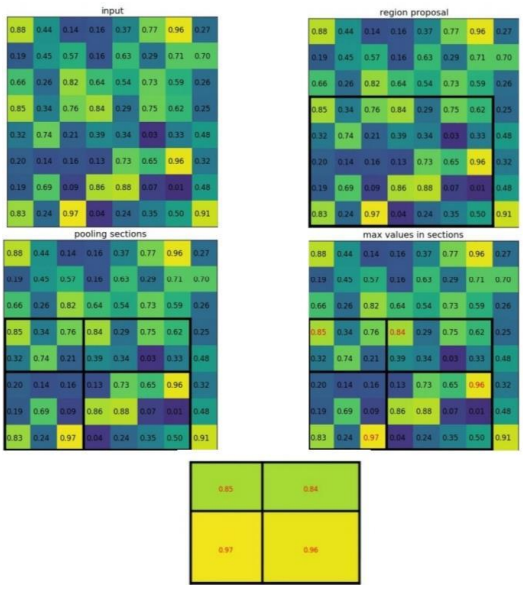


Figure 7. The process of determining the region of interest.

Despite its advantages, Fast R-CNN still relies on selective search, which is time-consuming and cannot be customized for a specific object-detection task. It may therefore fail to propose all target objects in a dataset.

5.4.3 Faster R-CNN

Faster R-CNN [18] extends Fast R-CNN and increases its speed by using a region proposal network (RPN). Its main contributions are:

1. an RPN, implemented as a fully convolutional network, which generates proposals at different scales and aspect ratios and directs Fast R-CNN to promising image regions; and
2. anchor boxes, which replace image or filter pyramids with reference boxes at several scales and aspect ratios, enabling objects of different sizes and shapes to be detected.

Convolutional computations are shared between the RPN and Fast R-CNN, reducing computational time. As shown in Figure 8, the architecture consists of the RPN for generating proposals and Fast R-CNN for detecting objects in the proposed regions.

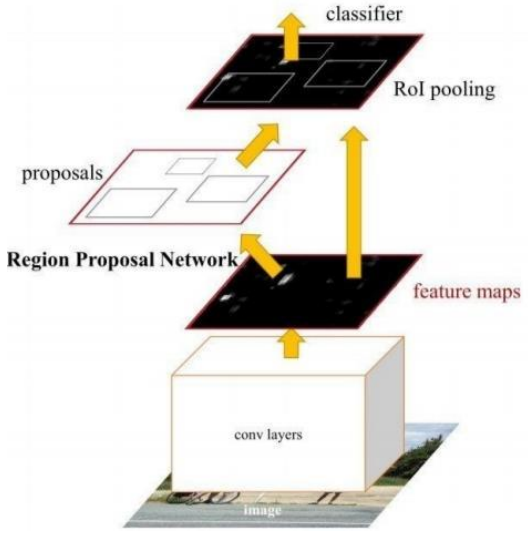


Figure 8. The mechanism of action of Faster R-CNN.

Faster R-CNN operates as follows:

1. the RPN creates region proposals;
2. a fixed-length feature vector is extracted from each proposal using the RoI layer;
3. the extracted features are classified using Fast R-CNN; and
4. the class scores and refined bounding boxes are returned for the detected objects.

5.5 Single-Stage Object Detection

Single-stage models directly predict object bounding boxes and classes from an image without an intermediate region-proposal task. This approach produces a simpler and faster model structure.

5.5.1 You Only Look Once (YOLO)

YOLO proposes a unified network that performs all detection operations at once and can be trained end to end [2].

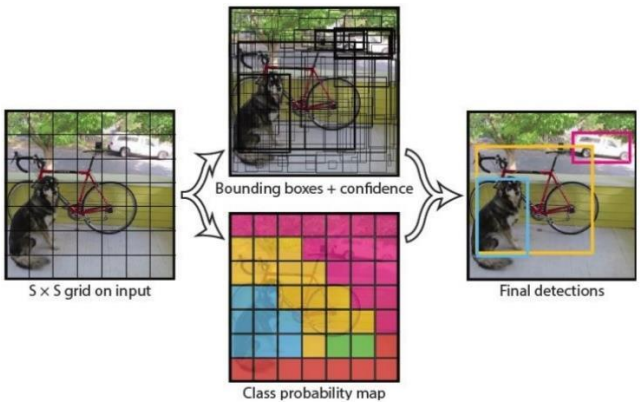


Figure 9. YOLO unified detection.

The input image is divided into an $S \times S$ grid, with $S = 7$. If the center of an object falls within a grid cell, that cell is responsible for detecting the object. Each grid cell predicts $B = 2$ bounding boxes and confidence scores. Each bounding box consists of five predictions: x, y, w, h, and confidence.

- Coordinates (x, y) represent the center of the box relative to the boundaries of the grid cell.
- Width w and height h are predicted relative to the entire image.
- Confidence represents the IoU between the predicted box and the ground-truth box.

Each grid cell also predicts conditional class probabilities, $P(\text{class} \mid \text{object})$, for 20 classes.

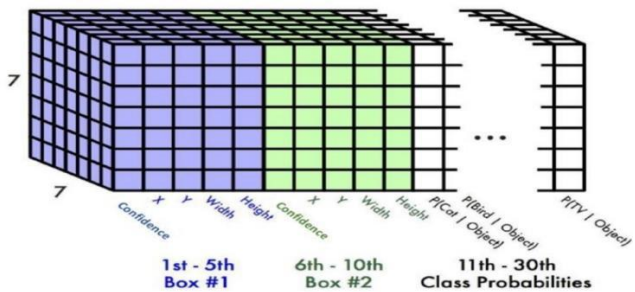


Figure 10. YOLO output.

The resulting output volume is

$$7 \times 7 \times (2 \times 5 + 20) = 1470. \quad (1)$$

YOLO network architecture. The model consists of 24 convolutional layers followed by two fully connected layers. Alternating 1×1 convolutional layers reduce the feature-space dimensionality of the preceding layers, as in GoogLeNet.

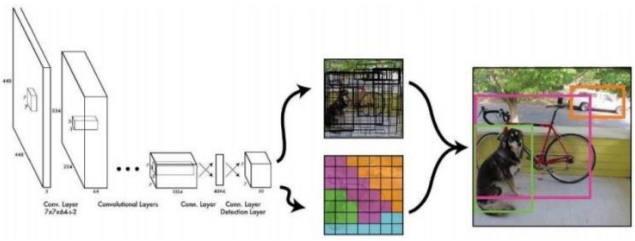


Figure 11. Stages of YOLO.

The input image passes through the network only once before objects are detected, enabling end-to-end learning. Except for the final layer, the network uses leaky ReLU activations. The first 20 convolutional layers were pretrained on ImageNet and achieved a top-5 accuracy of approximately 88%. The complete network was then trained for approximately 135 epochs on the Pascal VOC training and validation datasets.

5.5.2 Single Shot Detector (SSD)

SSD was published in 2016 [22]. It accelerates object detection by eliminating the region-proposal network. To compensate for the resulting decrease in resolution, SSD uses multi-scale feature maps and default boxes. These improvements enable SSD to match the accuracy of Faster R-CNN at a lower input resolution and higher speed.

An object is detected by SSD in two main steps:

- extract feature maps; and
- apply convolutional filters to detect objects.

Each prediction consists of a bounding box and 21 class scores, including one class for the absence of an object. The category with the highest score is selected. For example, the Conv4_3 layer produces $4 \times 38 \times 38$ predictions, or four predictions per cell regardless of the depth of the feature maps. Many of these predictions contain no object, so SSD uses class 0 to represent the background. Producing multiple predictions with bounding boxes and confidence scores is called MultiBox.

The number of predictions is much larger than the number of objects, producing many more negative matches than positive matches and creating an imbalance that can harm training. SSD nevertheless requires negative samples to learn what constitutes a poor prediction. Instead of using all negatives, it sorts them by confidence loss and selects the hardest examples while keeping the negative-to-positive ratio at no more than 3:1. This produces faster and more stable training.

6. COMPARISON OF OBJECT-DETECTION TECHNOLOGIES

The following tables compare the algorithms reviewed above. They cover the region-proposal technique, support for multi-scale inputs, learning method, loss function, Softmax layer, end-to-end training, platform, and implementation language.

Table 1. Comparison of proposal and learning methods.

Framework	Proposal	Multi-scale input	Learning method
R-CNN	Selective Search	No	SGD, BP
SPP-net	EdgeBoxes	Yes	SGD
Fast R-CNN	Selective Search	Yes	SGD
Faster R-CNN	RPN	Yes	SGD
YOLO	—	No	SGD
SSD	—	No	SGD
YOLOv2	—	No	SGD

Table 2. Comparison of loss functions.

Framework	Loss function
R-CNN	Hinge loss (classification) and bounding-box regression
Fast R-CNN	Class log loss and bounding-box regression
Faster R-CNN	Class log loss and bounding-box regression
YOLO	Class sum-squared error, bounding-box regression, object confidence, and background confidence
SSD	Class Softmax loss and bounding-box regression
YOLOv2	Class sum-squared error, bounding-box regression, object confidence, and background confidence

Table 3. Comparison of implementation characteristics.

Framework	Softmax	End-to-end	Platform	Language
R-CNN	Yes	No	Caffe	Matlab
Fast R-CNN	Yes	No	Caffe	Python
Faster R-CNN	Yes	Yes	Caffe	Python/Matlab
YOLO	Yes	Yes	Darknet	C
SSD	No	Yes	Caffe	C++
YOLOv2	Yes	Yes	Darknet	C

7. EXPERIMENTS AND RESULTS

The techniques described above are evaluated on standard datasets to examine their efficiency and relative advantages.

7.1 Training Datasets

The development of computer vision has increased the need for large-scale image datasets. Large amounts of annotated data are a principal reason for the success of deep-learning technologies in object detection. The Internet plays an important role in building comprehensive datasets with images that cover a broad and diverse set of objects. Five datasets are especially common in general object detection: Pascal VOC 2007, Pascal VOC 2012, ImageNet, Microsoft COCO, and Open Images.

7.2 Results

Tables 4–7 present the results obtained by the reviewed techniques on the Pascal VOC 2007 and Pascal VOC 2012 test sets.

8. DISCUSSION OF RESULTS

The results support the following observations:

Table 4. Results of the Pascal VOC 2007 test on object-detection techniques.

Methods	Trained on	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN (Alex)	07	68.1	72.8	56.8	43	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	68.6	58.5
R-CNN (VGG16)	07	73.4	77	63.4	45.4	44.6	75.1	78.1	79.8	40.5	73.7	62.2	79.4	78.1	73.1	64.2	35.6	66.8	67.2	70.4	71.1	66
Fast R-CNN	07+12	77	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82	76.6	69.9	31.8	70.1	74.8	80.4	70.4	70
Faster R-CNN	07	70	80.6	70.1	57.3	49.9	78.2	80.4	82	52.2	75.3	67.2	80.3	79.8	75	76.3	39.1	68.3	67.3	81.1	67.6	69.9
Faster R-CNN	07+12	76.5	79	70.9	65.5	52.1	83.1	84.7	86.4	52	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83	72.6	73.2
Faster R-CNN	07+12+COCO	84.3	82	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9	78.8
SSD300	07+12+COCO	80.9	86.3	79	76.2	57.6	87.3	88.2	88.6	60.5	85.4	76.7	87.1	89.2	84.5	81.4	55	81.9	81.5	85.9	78.9	79.6
SSD512	07+12+COCO	86.6	88.3	82.4	76	66.3	88.6	88.9	89.1	65.1	88.4	73.6	86.5	88.9	85.3	84.6	59.1	85	80.4	87.4	81.2	81.6

Table 5. Summary of Pascal VOC 2007 results.

Methods	Trained on	mAP
R-CNN (Alex)	07	58.5
R-CNN (VGG16)	07	66.0
Fast R-CNN	07+12	70.0
Faster R-CNN	07	69.9
Faster R-CNN	07+12	73.2
Faster R-CNN	07+12+COCO	78.8
SSD300	07+12+COCO	79.6
SSD512	07+12+COCO	81.6

Table 6. Results of the Pascal VOC 2012 test on object-detection techniques.

Methods	Trained on	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN (Alex)	12	71.8	65.8	52	34.1	32.6	59.6	60	69.8	27.6	52	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1	53.3
R-CNN (VGG16)	12	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82	74.8	76	65.2	35.6	65.4	54.2	67.4	60.3	62.4
Fast R-CNN	07++12	83.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73	55	87.5	80.5	80.8	72	35.1	68.3	65.7	80.4	64.2	68.4
Faster R-CNN	07++12	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5	70.4
YOLO	07++12	77	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8	57.9
YOLOv2	07++12+COCO	88.8	87	77.8	64.9	51.8	85.2	79.3	93.1	64.4	81.4	70.2	91.3	88.1	87.2	81	57.7	78.1	71	88.5	76.8	78.2
SSD300	07++12+COCO	91	86	78.1	65	55.4	84.9	84	93.4	62.1	83.6	67.3	91.3	88.9	88.6	85.6	54.7	83.8	77.3	88.3	76.5	79.3
SSD512	07++12+COCO	91.4	88.6	82.6	71.4	63.1	87.4	88.1	93.9	66.9	86.6	66.3	92	91.7	90.8	88.5	60.9	87	75.4	90.2	80.4	82.2

Table 7. Summary of Pascal VOC 2012 results.

Methods	Trained on	mAP
R-CNN (Alex)	12	53.3
R-CNN (VGG16)	12	62.4
Fast R-CNN	07++12	68.4
Faster R-CNN	07++12	70.4
YOLO	07++12	57.9
YOLOv2	07++12+COCO	78.2
SSD300	07++12+COCO	79.3
SSD512	07++12+COCO	82.2

- The principal disadvantage of two-stage object-detection technologies is their need for high computational power.
- The structure of two-stage technologies is more flexible and accurate than that of single-stage technologies.
- Single-stage object-detection techniques such as YOLO and SSD require less time because they use efficient CNN feature extraction, avoid preprocessing algorithms, and have no separate region-proposal stage.
- The performance of single-stage technologies is generally less reliable for small objects.
- When sufficient hardware is available to overcome processing-time constraints, two-stage technologies are often the best option.

9. CONCLUSION

The success of object detection depends primarily on the techniques used. Single-stage techniques are simpler and faster, whereas two-stage techniques are generally more accurate and flexible. The ultimate goal is to develop object-detection technology capable of accurately and efficiently identifying thousands of new object classes in open, real-world scenes. Larger datasets with more categories are therefore needed because current benchmark datasets cover only a few hundred object categories, far fewer than humans can recognize.

This paper reviewed the concept of object detection after a brief introduction to deep learning and convolutional neural networks. It then presented the most important practical papers that introduced deep-learning-based object-detection algorithms. The structures, motivations, and operating mechanisms of the best-known algorithms were studied. Finally, the techniques were evaluated and the results discussed to identify criteria for selecting an appropriate object-detection method.

REFERENCES

[1] R. Girshick, 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, pp. 580–587.

[2] J. Redmon, 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, pp. 779–788.

[3] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[4] L. Jiao, R. Zhang, F. Liu, S. Yang, B. Hou, L. Li, and X. Tang, "New generation deep learning for video object detection: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3195–3215, 2021.

[5] W. Pitts and W. S. McCulloch, "How we know universals: The perception of auditory and visual forms," *The Bulletin of Mathematical Biophysics*, vol. 9, no. 3, pp. 127–147, 1947.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[9] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[10] C. Szegedy, 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, pp. 1–9.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] K. He, 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, pp. 770–778.

[13] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[14] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[15] P. Viola and M. Jones, "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*, vol. 1, 2001, pp. I–I.

[16] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *European Conference on Computer Vision*, 2006, pp. 428–441.

[17] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[18] S. Ren, 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*, vol. 28, 2015.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[21] A. Pramanik, S. K. Pal, J. Maiti, and P. Mitra, "Granulated RCNN and multi-class deep SORT for multi-object detection and tracking," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 1, pp. 171–181, 2021.

[22] W. Liu, 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, pp. 21–37.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[24] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[25] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[26] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[27] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[28] L. Aziz, M. S. M. R. Salam, U. U. Sheikh, and S. Ayub, "Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review," *IEEE Access*, vol. 8, pp. 170 461–170 495, 2020.