From classical techniques to convolution-based models: A review of object detection algorithms

1st FNU Neha
Dept. of Computer Science
Kent State University
Kent, OH, USA
neha@kent.edu

1st Deepshikha Bhati
Dept. of Computer Science
Kent State University
Kent, OH, USA
dbhati@kent.edu

2nd Deepak Kumar Shukla
Rutgers Business School
Rutgers University
Newark, New Jersey, USA
ds1640@scarletmail.rutgers.edu

3rd Md Amiruzzaman
Dept. of Computer Science
West Chester University
West Chester, PA, USA
mamiruzzaman@wcupa.edu

***Abstract*—Object detection is a fundamental task in computer vision and image understanding, with the goal of identifying and localizing objects of interest within an image while assigning them corresponding class labels. Traditional methods, which relied on handcrafted features and shallow models, struggled with complex visual data and showed limited performance. These methods combined low-level features with contextual information and lacked the ability to capture high-level semantics. Deep learning, especially Convolutional Neural Networks (CNNs), addressed these limitations by automatically learning rich, hierarchical features directly from data. These features include both semantic and high-level representations essential for accurate object detection. This paper reviews object detection frameworks, starting with classical computer vision methods. We categorize object detection approaches into two groups: (1) classical computer vision techniques and (2) CNN-based detectors. We compare major CNN models, discussing their strengths and limitations. In conclusion, this review highlights the significant advancements in object detection through deep learning and identifies key areas for further research to improve performance.**

***Index Terms*—Object Detection, CNN, Deep Learning, Image Processing, Computer Vision**

I. INTRODUCTION

Deep learning (DL) has advanced image analysis, especially in object classification, localization, and detection tasks. In classification, the aim is to assign an image or object within it to one of several categories [1]. However, classification does not provide the object's location. Localization improves on this by identifying both the object's category and position, typically with a bounding box [2], though the precision of these boxes can vary. Object detection further extends classification and localization by detecting and classifying multiple objects in an image, providing bounding boxes for each [2]. The bounding box's top-left corner is represented by (X_{min}, Y_{min}), and the bottom-right by (X_{max}, Y_{max}), along with a label indicating the object's class as shown in Fig 1.

Object detection has applications across fields such as medical imaging, logo detection, facial recognition, pedestrian detection, and industrial automation. However, challenges arise from image transformations like changes in scale, orientation, and lighting. While classical computer vision techniques provided a foundation, advancements in deep learning (DL), especially CNNs, have significantly improved detection performance. Modern methods use hierarchical representations, enabling object detection in complex environments with occlusions and varying scales.

The first author contributed the most to this paper. Corresponding author: mamiruzzaman@wcupa.edu

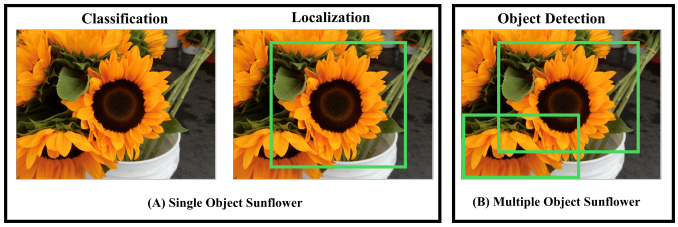

Fig. 1: (A) Single-object sunflower: A single bounding box localizes and classifies the central sunflower bloom. (B) Multiple-object sunflower: Multiple bounding boxes highlight and classify overlapping sunflowers and leaves, illustrating multi-scale object detection and localization within a complex scene.

Although many studies have reviewed specific deep learning models or object detection applications, few provide a comprehensive overview of both classical computer vision techniques and CNN-based approaches. This paper addresses this gap by offering an analysis of both. Key contributions include:

1) A review of classical computer vision techniques for object detection.
2) An analysis of general region proposal generation techniques.
3) A detailed review of convolution-based models for object detection, including two-stage and one-stage detectors.

The paper is organized as follows: Section 2 covers classical computer vision techniques for object detection, Section 3 discusses region proposal generation, Section 4 explores CNN-based detection architectures, Section 5 reviews applications, Section 6 lists popular datasets, Section 7 covers evaluation metrics, and Section 8 concludes with future directions.

II. CLASSICAL COMPUTER VISION TECHNIQUES FOR OBJECT DETECTION

Earlier computer vision techniques for image processing, particularly image similarity, relied on feature-based methods

[3]–[8]. These methods focused on extracting distinctive image features to reduce computational costs while enabling robust image matching despite transformations like scaling or rotation [3]. The Scale-Invariant Feature Transform (SIFT) algorithm overcame the challenge of scaling by extracting features invariant to scale, rotation, brightness, and contrast [4]. Other feature extractors, like the Canny Edge Detector, contributed to tasks like image comparison and panoramic stitching by providing resilience to transformations and occlusions [5]. The Histogram of Oriented Gradients (HOG) technique enabled efficient image analysis by measuring gradient magnitudes and directions, creating descriptive feature vectors [6].

Traditional object detection involves three stages:

1) **Proposal Generation:** Scanning the image at various positions and scales to generate candidate bounding boxes, often using methods like sliding windows or selective search algorithms.
2) **Feature Extraction:** Extracting features from the identified regions to capture relevant visual patterns.
3) **Classification:** Classifying the extracted features using machine learning algorithms, such as support vector machine (SVM).

In 2001, Viola et al. introduced a real-time (webcam based) facial detection classifier [7]. In 2005, Dalal et al. introduced an object detector using HOG features and an SVM classifier, effective across scales but limited by pose variations [6]. In 2009, Felzenszwalb et al. improved this with the Deformable Part Model (DPM), allowing flexible parts to handle poses, though it struggled with overlapping parts in multi-person images [8].

Studies from 2008 to 2012 on popular object detection datasets (see Section 5) showed key limitations in traditional methods. For instance, sliding windows require substantial computational resources and can generate redundant detections. Additionally, the performance of the classifier greatly impacts the results, necessitating more robust approaches.

III. GENERIC REGION PROPOSAL GENERATION TECHNIQUES

Object detection models integrate a bounding box regressor within the classification network to accurately locate objects [9]. Traditionally, this involves feeding cropped images to the localization network, resulting in excessive inputs. An OverFeat model enhances efficiency by using a sliding window detector within convolution layers, scanning images with a large filter and stride [10]. However, indiscriminate scanning of background regions necessitates predicting potential object locations. Methods such as interest point detection, multiscale saliency, color contrast, edge detection, and super-pixel clustering are employed for this purpose [11]–[14].

For instance, multiscale saliency leverages the Fast Fourier Transform to analyze features at multiple scales [11]; color contrast relies on color intensity differences [12]; edge detection identifies edges, followed by density analysis [13]; and super-pixel clustering groups similar pixels for detailed analysis [14].

Each method has specific limitations: multiscale saliency struggles with low-contrast objects, color contrast is ineffective with minimal contrast, edge detection may produce false positives or negatives, and super-pixel clustering requires refinement. Consequently, hybrid models are often developed to improve region proposal accuracy.

IV. CONVOLUTION BASED OBJECT DETECTION MODELS

Object detection initially relied on manual feature design, focusing on patterns and edges. With CNN advancements, networks such as Visual Geometry Group Network (VGGNet) [15] and AlexNet [16] now autonomously extract features through convolution and pooling layers, with fully connected (FC) layers followed by a SoftMax layer for classification. For localization, the final FC layer outputs bounding box coordinates, unlike classical methods which use filters and machine learning-based models (e.g., SVMs).

Training CNN-based models involve adjusting weights via backpropagation to align predictions with ground truth bounding boxes. Detection models fall into two categories: (1) Two-stage detectors, which generate region proposals before classification, including R-CNN [17], Fast R-CNN [18], Faster R-CNN [19], and Mask R-CNN [20]; and (2) One-stage detectors, treating detection as direct regression or classification tasks, like YOLO [21] and SSD [22]. Table I summarizes their strengths and limitations.

TABLE I: Comparison of CNN-Based Object Detection Architectures

Model	Strengths	Limitations
R-CNN (2013)	Simple, foundational; applies CNNs for classification.	High computation for 2000 region classifications; slow (47 sec/image); no end-to-end training.
SPPNet (2015)	Faster than R-CNN; supports multi-scale input via spatial pyramid pooling.	Does not update conv. layers before SPP layer during fine-tuning.
Fast R-CNN (2015)	Faster than SPPNet; introduces ROI pooling to handle varied input sizes.	Relies on selective search for region proposals, not learned during training.
Faster R-CNN (2015)	Uses RPN for fast region proposals; improves efficiency.	Limited in detecting small objects due to single feature map.
Mask R-CNN (2017)	Adds instance segmentation, detecting objects and masks simultaneously.	High computational demand; struggles with motion blur at low resolution.
YOLO (2015)	Real-time detection at 45 fps; single forward pass.	Poor detection of small objects; produces coarse features.
SSD (2016)	Handles various resolutions; uses multi-scale feature maps for detection.	Default boxes may not match all shapes; possible overlapping detections.

A. Region-based Convolutional Neural Network (R-CNN)

In 2014, Girshick et al. introduced R-CNN, a two-stage network that combines classical techniques like selective search with CNNs for object detection [17] (see Fig. 2). R-CNN's training involves three steps:

- Fine-tune a pre-trained network (e.g., AlexNet) on region proposals generated by selective search.

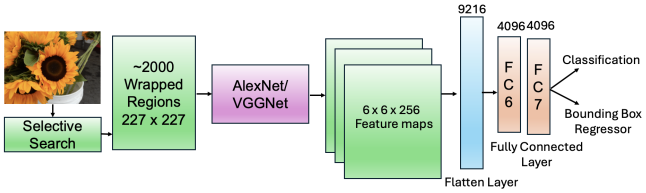

Fig. 2: R-CNN Architecture

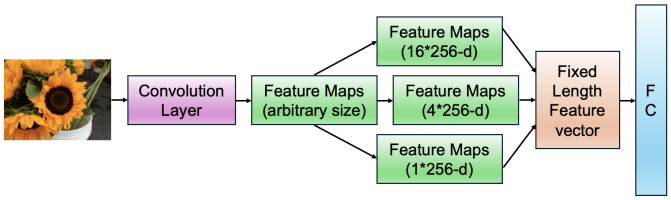

Fig. 3: SPP-Net Architecture

- Train an SVM classifier for object classification.
- Use a bounding box regressor to improve localization accuracy.

Selective search generates around 2000 region proposals, each resized to 227x227 pixels for CNN input, reducing the computational cost of exhaustive sliding windows.

Initially, R-CNN achieved 44% accuracy, improving to 54% after fine-tuning on warped images. Adding a bounding box regressor boosted accuracy to 58%, and using VGGNet further increased it to 66%. While nine times slower than OverFeat, R-CNN's focus on region proposals reduces false positives, improving accuracy by 10%.

However, R-CNN has some limitations:

- Feature extraction is performed independently for each proposal, resulting in high computational costs.
- The separate stages of proposal generation, feature extraction, and classification prevent end-to-end optimization.
- Selective search relies on low-level visual features, struggles with complex scenes, and does not benefit from GPU acceleration.
- Despite higher accuracy compared to methods like OverFeat, R-CNN is slower due to these inefficiencies.

B. Spatial Pyramid Pooling-Net (SPP-Net)

In 2015, He et al. introduced SPP-Net to improve detection speed and feature learning over R-CNN [23]. Unlike R-CNN, which processes each cropped proposal individually, SPP-Net computes the feature map for the entire image and then applies a Spatial Pyramid Pooling (SPP) layer to extract fixed-length feature vectors (See Fig. 3). The SPP layer divides the feature map into grids of varying sizes (N $\times$ N), enabling pooling at multiple scales and concatenation of the resulting feature vectors.

SPP-Net allows multi-scale and varied aspect ratio handling without resizing, preserving image details and improving both accuracy and inference speed over R-CNN. However, its multi-stage training hinders end-to-end optimization and requires extra memory for feature storage. Additionally, the SPP layer does not back-propagate to earlier layers, keeping parameters fixed before the SPP layer and limiting deeper learning.

C. Fast Region-based Convolutional Neural Network (Fast R-CNN)

In 2015, Girshick et al. introduced Fast R-CNN, a two-stage detector designed to improve on SPP-Net's limitations [18]. Fast R-CNN computes a feature map for the entire image

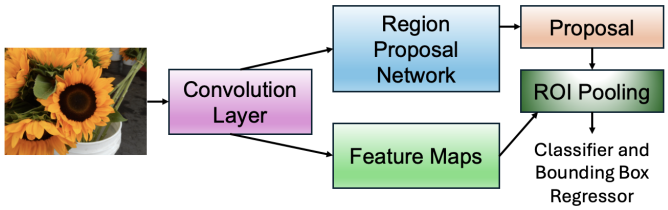

Fig. 4: Faster R-CNN Architecture

and uses a Region of Interest (ROI) Pooling layer to extract fixed-length features from each region, dividing proposals into a fixed N $\times$ N grid. Unlike SPP, ROI Pooling backpropagates error signals, enabling end-to-end optimization.

After feature extraction, features pass through FC layers, outputting (1) SoftMax probabilities for C+1 classes (including background) and (2) four bounding box regression parameters. Fast R-CNN achieved better accuracy than R-CNN and SPP-Net but still relied on traditional proposal methods.

D. Faster Region-based Convolutional Neural Network (Faster R-CNN)

In 2015, Girshick et al. introduced Faster R-CNN, which utilizes the Region Proposal Network (RPN) to generate object proposals at each feature map position using a sliding window approach (Fig. 4) [18]. This method shares feature extraction across regions, enhancing efficiency and achieving state-of-the-art results. However, the separate computation for region classification can be inefficient with many proposals, and reliance on a single deep feature map makes detecting objects of varying scales difficult, as deep features are semantically strong but spatially weak, while shallow features are spatially strong but semantically weak.

E. Mask R-CNN

In 2017, He et al. introduced Mask R-CNN, an extension of Faster R-CNN that performs pixel-level instance segmentation [20]. It adds a new branch for binary mask prediction to the two-stage pipeline, alongside class and box predictions. This branch uses a fully convolutional network (FCN) atop the CNN feature map. Mask R-CNN also replaces RoIPool with RoIAlign to better preserve spatial accuracy, enhancing mask precision. However, it struggles to detect objects with motion blur in low-resolution images.

F. You Only Look Once (YOLO)

To increase speed, one-stage models like YOLO (You Only Look Once) were developed, bypassing region proposals. Introduced in 2015 by Redmon et al., YOLO treats detection as a regression task [21]. Dividing the image into an S × S grid, YOLO predicts class probabilities, bounding boxes, and confidence scores per cell. This captures context well, reducing false positives, but the grid structure can cause localization errors and struggles with small objects.

YOLO has undergone several iterations, enhancing its performance:

- **YOLOv2/YOLO9000 (2017):** Introduced batch normalization and anchor boxes for improved speed and accuracy [24].
- **YOLOv3 (2018):** Added multi-scale predictions and residual connections for better detection across various sizes [25].
- **YOLOv4 (2020):** Enhanced with the CSPDarknet backbone and advanced training techniques, achieving higher precision [26].
- **YOLOv5 (2021):** Focused on usability, scalability, and deployment flexibility with various model sizes [27].
- **YOLOv6 (2022):** Optimized for edge devices with improved backbone and attention mechanisms [28].
- **YOLOv7 (2023):** Employed AutoML techniques for dynamic model optimization, enhancing adaptability [29].
- **YOLOv8 (2023):** Incorporated a transformer-based backbone for better detection in dense scenes [30].
- **YOLOv9 (2024):** Utilized adversarial training to improve robustness against variations [31].
- **YOLOv10 (2024):** Implemented real-time feedback loops for dynamic adjustments, boosting accuracy [32].

These enhancements have established YOLO as a versatile and powerful option for real-time object detection.

G. Single Shot MultiBox Detector (SSD)

The Single Shot MultiBox Detector (SSD), introduced by Liu et al. in 2016, is a one-stage model that improves on YOLO by using anchors with multiple scales and aspect ratios within each grid cell [22]. Each anchor is refined by regressors and assigned probabilities across categories, with object detection predicted on multiple feature maps for different scales. SSD trains end-to-end with a weighted localization and classification loss, integrating results across maps. Using hard negative mining and extensive data augmentation, SSD matches Faster R-CNN's accuracy while allowing real-time inference.

V. APPLICATIONS

Object detection, powered by CNN, has diverse applications, spanning from targeted advertising to self-driving cars and beyond. It is utilized for handwritten digit recognition, Optical Character Recognition (OCR), face detection, medical image analysis, sports analytics, and more.

- **Optical Character Recognition (OCR):** OCR converts images of text into machine-encoded text, facilitating tasks such as document digitization, automated data entry, and cognitive computing.
- **Self-Driving Cars:** Object detection is essential for autonomous vehicles to detect and classify objects such as cars, pedestrians, traffic lights, and road signs.
- **Object Tracking:** Used in tracking objects in videos, object detection has applications in surveillance, traffic monitoring, and sports analytics.
- **Face Detection and Recognition:** Widely employed in computer vision, object detection is used for social media image tagging and biometric security systems.
- **Object Extraction from Images or Videos:** Facilitates segmentation and meaningful representation of images, potentially enabling applications like video object extraction.
- **Digital Watermarking:** Embed markers into digital signals for copyright protection and authentication purposes.
- **Medical Imaging:** Assists clinicians in diagnosis and therapy planning, particularly in tracking anatomical objects.

  Object detection technology continues to evolve, promising further advancements and expanding its applications across various industries.

VI. POPULAR DATASET

Key datasets in object detection include Pascal VOC [33], COCO [34], ImageNet [35], and Open Images [36]. Pascal VOC (Visual Object Classes) offers a manageable size, balancing complexity and computational efficiency, making it ideal for testing. COCO (Common Objects in Context) provides extensive annotations with multiple objects per image, including segmentation and key points. ImageNet, primarily used for classification, also includes object detection annotations. Open Images, with over 600 labeled categories, stands out for its large scale, offering both bounding box annotations and segmentation masks. Table II summarizes the key attributes of each dataset, emphasizing their unique features and primary usage. Table III provides a comparison of the performance of RCNN, Fast RCNN, Faster RCNN, Mask RCNN, YOLO, and SSD on these datasets in terms of mAP, inference speed (measured in Frames Per Second, or FPS), and model size.

VII. EVALUATION METRICS

Object detection models are assessed using several key metrics: Intersection over Union (IoU), Mean Average Precision (mAP), Precision, Recall, Confidence Score (CS), F1 Score, and Non-Maximum Suppression (NMS). Table IV summarizes these metrics, highlighting their limitations and potential biases.

A. Intersection over Union (IoU)

IoU measures the overlap between the predicted and ground truth bounding boxes, calculated as the ratio of the intersection area to the union area:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

TABLE II: Popular Object Detection Datasets

Dataset	Number of Images	Number of Classes	Usage
Pascal VOC	0.01 million	20	Initial model testing
COCO	0.33 million	80	Object detection
ImageNet	1.5 million	1,000	Object localization and detection
Open Images	9.2 million	600	Object localization

TABLE III: Quantitative Performance Comparison of Object Detection Models on different Dataset

Model	Pascal VOC (mAP)	COCO (mAP)	ImageNet (mAP)	Open Images (mAP)	Inference Speed (FPS)	Model Size (MB)
RCNN	66%	54%	60%	55%	∼5 FPS	200
Fast RCNN	70%	59%	63%	58%	∼7 FPS	150
Faster RCNN	75%	65%	68%	63%	∼10 FPS	180
Mask RCNN	76%	66%	69%	64%	∼8 FPS	230
YOLO	72.5%	58.5%	61.5%	57.5%	∼45–60 FPS	145
SSD	75%	63.5%	66.5%	61.5%	∼19–46 FPS	145

B. Mean Average Precision (mAP)

mAP evaluates model performance by averaging the precision across all classes. The Average Precision (AP) is computed as:

$$\text{AP} = \frac{\sum_{k=1}^{n} \left(P(k) \times \text{Precision at Recall}(k)\right)}{n}$$

where $P(k)$ is the change in recall from the previous highest recall, and precision at recall k is the maximum precision observed at any recall level j where $j \geq k$.

C. Precision and Recall

Precision is the ratio of true positives to all positive predictions, while *Recall* is the ratio of true positives to all ground truth positives.

D. Confidence Score (CS)

The Confidence Score reflects the model's certainty that a predicted bounding box contains the correct object. Higher scores indicate greater accuracy and help set thresholds for accepting or rejecting detections.

E. Non-Maximum Suppression (NMS)

Non-Maximum Suppression refines bounding box predictions by sorting them by confidence scores and selecting the highest one while suppressing overlapping boxes. This process ensures each object is detected once, improving accuracy and efficiency.

VIII. Discussion and Future Directions

This review examined prominent object detection models, classifying them into classical computer vision techniques and CNN-based methods. While recent CNN architectures have significantly improved accuracy to below 5%, they also increase complexity and resource demands. Traditional models like Deformable Part Models (DPMs) are shallower and more lightweight, making them better suited for edge deployment compared to modern deep learning architectures like AlexNet and VGGNet.

Key future directions for object detection include:

- Speed-Accuracy Trade-off: Enhancing both accuracy and speed for real-time, low-power applications.
- Tiny Object Detection: Improving the detection of small objects in areas such as wildlife monitoring and medical imaging.
- 3D Object Detection: Leveraging 3D sensors for applications in augmented reality and robotics.
- Multi-modal Detection: Integrating visual and textual sources for better accuracy in complex scenarios.
- Few-shot Learning: Developing models that can effectively detect objects from limited examples, particularly in low-resource settings.

This review aims to foster interest in advancing object detection models and to inspire innovation to address current limitations, including minimizing environmental impacts.

Acknowledgment

This study was partly supported by the West Chester University faculty development fund.

References

[1] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing*, vol. 13, no. 22, p. 4712, 2021.

[2] C. B. Murthy, M. F. Hashmi, N. D. Bokde, and Z. W. Geem, "Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—a comprehensive review," *Applied sciences*, vol. 10, no. 9, p. 3280, 2020.

[3] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image matching from handcrafted to deep features: A survey," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 23–79, 2021.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[5] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[7] 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. CVPR 2001*, vol. 1. Ieee, 2001, pp. I–I.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE conference on computer vision and pattern recognition*. Ieee, 2008, pp. 1–8.

TABLE IV: Evaluation Metrics: Limitations and Potential Biases of Object Detection Models

Model	Metrics Used	Limitations	Potential Biases
RCNN	IoU, mAP, Precision, Recall, F1 Score	- Separate region proposal step slows inference. - High memory usage due to multiple stages.	- Favors larger objects due to reliance on selective search. - Struggles with scale variations and densely packed objects.
Fast RCNN	IoU, mAP, Precision, Recall, F1 Score	- Dependent on external region proposals. - Not optimized for real-time applications.	- Similar biases as RCNN: prefers larger and well-separated objects. - Performance drops in high-density scenes.
Faster RCNN	IoU, mAP, Precision, Recall, F1 Score	- More complex architecture with integrated Region Proposal Network (RPN). - Requires careful hyperparameter tuning.	- Favors objects with distinct features detectable by RPN. - Limited accuracy on small or thin objects compared to single-shot models.
Mask RCNN	IoU, mAP, Precision, Recall, F1 Score	- Increased computational overhead from mask prediction. - Longer training times.	- Bias towards classes with abundant and detailed segmentation data. - Misses small or occluded objects in segmentation masks.
YOLO	IoU, mAP, Precision, Recall, Confidence Score	- Lower detection accuracy on small objects. - Struggles with overlapping objects and crowded scenes.	- Prioritizes objects at the center of the image. - Predefined grid may miss objects at image edges.
SSD	IoU, mAP, Precision, Recall, Confidence Score	- Performance degrades on very small objects. - Limited by predefined anchor box scales and aspect ratios.	- Bias towards predefined anchor boxes, affecting generalization for unseen scales. - Struggles with variable object shapes and sizes not covered by anchor boxes.

[9] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, and H. Bischof, "Accurate object detection with joint classification-regression random forests," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 923–930.

[10] P. Sermanet, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[11] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5455–5463.

[12] K. Fu, C. Gong, J. Yang, and Y. Zhou, "Salient object detection via color contrast and color distribution," in *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11*. Springer, 2013, pp. 111–122.

[13] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 391–405.

[14] K. Fu, C. Gong, J. Yang, Y. Zhou, and I. Y.-H. Gu, "Superpixel based color contrast and color distribution driven salient object detection," *Signal Processing: Image Communication*, vol. 28, no. 10, pp. 1448–1463, 2013.

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

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[17] 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.

[18] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083*, 2015.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[20] 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.

[21] J. Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[24] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[25] A. Farhadi and J. Redmon, "Yolov3: An incremental improvement," in *Computer vision and pattern recognition*, vol. 1804. Springer Berlin/Heidelberg, Germany, 2018, pp. 1–6.

[26] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[27] G. Jocher, A. Stoken, A. Chaurasia, J. Borovec, Y. Kwon, K. Michael, L. Changyu, J. Fang, P. Skalski, A. Hogan *et al.*, "ultralytics/yolov5: v6. 0-yolov5n'nano'models, roboflow integration, tensorflow export, opencv dnn support," *Zenodo*, 2021.

[28] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[29] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.

[30] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," https://github.com/ultralytics/ultralytics, 2023, aGPL-3.0 License.

[31] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," *arXiv preprint arXiv:2402.13616*, 2024.

[32] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.

[33] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[35] 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*. Ieee, 2009, pp. 248–255.

[36] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International journal of computer vision*, vol. 128, no. 7, pp. 1956–1981, 2020.