Received: January 18, 2026. Published: June 30, 2026

Journal of
Advanced Applied Technology and Engineering (JAATE)

https://journals.univ-msila.dz/index.php/JAATE

Detection of Human-Object Interactions with Deep Learning

Lahouaoui Lalaoui [1]* Ahd Lakhnech[2]

[1]Mohamed Boudiaf University, Department Deprt of Electronic Laboratory of Electrical Engineering, Faculty of Technology,PO Box 166 Ichebilia, 28000 M'sila, Algeria
ORCID Link (https://orcid.org/0000-0003-0482-3293)
Email:lahouaoui.lalaoui@univ-msila.dz

[2]Mohamed Boudiaf University, Department Deprt of Electronic Laboratory of Electrical Engineering, Faculty of Technology, PO Box 166 Ichebilia, 28000 M'sila, Algeria
ORCID Link (https://orcid.org/0009-0006-4479-0819)
Email: ahd.lakhnech@univ-msila.dz

* Corresponding author's Email:lahouaoui.lalaoui@univ-msila.dz

Abstract: The In Human-object interaction (HOI) detection is a key area in computer vision, with applications in robotics, surveillance, autonomous vehicles, and human-computer interaction. The advent of deep learning, particularly the development of advanced architectures like R-CNN and YOLO, has significantly enhanced the accuracy and efficiency of object detection in complex environments. This paper explores the role of these models in improving HOI detection, focusing on their ability to identify and track objects in challenging scenarios. This research examines how deep learning models overcome challenges through advanced feature extraction, data augmentation, and multi-scale processing. R-CNN and YOLO, two prominent models, each offer distinct advantages. R-CNN excels in accuracy due to its region -based approach, while YOLO is known for its speed and real-time process sing capabilities, making it ideal for applications requiring fast, on-the-fly detection.

Keywords: Human-object interaction detection, deep learning; Faster R-CNN, and SSD, U-Net++.

1. Introduction

In recent years, the field of artificial intelligence has made significant strides, particularly in enabling machines to perceive, interpret, and interact with their environment. One of the most promising and challenging tasks in this domain is HOI detection, which focuses on identifying not only objects and people within a scene but also the specific interactions between them—such as holding, sitting on, pushing, or using an object. When HOI detection plays a crucial role in the development of intelligent systems across a variety of applications, including autonomous robotics, smart surveillance, augmented reality, and human-computer interaction. By understanding how humans interact with objects, machines can make more informed decisions and provide more context aware services. Toolbar The rapid advancement of deep learning techniques, particularly Convolutional Neural Networks (CNNs) [1], has significantly improved the performance of object detection and recognition tasks. Models such as YOLOv5, Faster RCNN, SSD, and U-Net++ have shown impressive results in detecting objects with high accuracy and speed. These models form the foundation for many HOI detection systems, enabling the recognition of both entities (humans and objects) and the actions linking them. Despite these technological advancements, HOI detection remains a complex and unresolved challenge. Realworld environments introduce numerous difficulties, such as: Occlusions, where objects or parts of the human body are partially hidden [2]-[6], variability in human poses, gestures, and object orientations [7]-[9],

Received: January 18, 2026. Published: June 30, 2026

ambiguity and context dependence of interactions [10]-[12], diversity of object categories and interaction types [13]-[15], and real-time processing constraints required for practical applications [16]- [18]. These factors make it difficult for existing systems to generalize well outside of controlled datasets or lab conditions. How can we develop a deep learning-based system capable of accurately and robustly detecting HOI in real-world, unconstrained environments? To answer this, the study explores the theoretical foundations and current methodologies of HOI detection, performs a comparative analysis of state-of-the-art object detection architectures, and conducts experiments using advanced convolutional neural networks. The ultimate goal is to identify best practices and propose improvements for the next generation of HOI detection systems. The remainder of this paper is organized as follows: Section 2 presents the different classification methods; Section 3 discusses the simulation results; and finally, the conclusions and future perspectives are provided in Section4.

2. Methods

In recent years, research on object detection models has attracted significant attention due to the rapid growth of the computer vision market. To interpret an image or a video, a computer must first detect the objects and accurately estimate their locations within the image or video before classifying them. Object detection involves several sub-tasks such as face detection, pedestrian detection, skeleton detection, and more. It also has common use cases in areas like surveillance systems and autonomous vehicles. In this article, we will review some of the different types of object detection algorithms that are popular today. There are various kinds of object detection algorithms—some based on traditional techniques and others built on modern, recently developed methods. These architectures differ from one another in terms of accuracy, speed, and hardware requirements [19].

2.1 Region-Based CNN

The R-CNN (Region-based Convolutional Neural Network) method is a powerful approach for object detection in images. It combines region proposal techniques with CNNs to identify and localize objects within an image. Its structure is based on a three-stage workflow. In the first stage, region proposals are generated using algorithms such as Selective Search, which identify areas of interest in an image without considering specific object categories. The second stage involves extracting features from each proposed region using a deep CNN. This network converts each region into a fixed length feature vector. Finally, in the third stage, these feature vectors are analyzed by dedicated SVM classifiers for each category, along with a bounding box regression module that precisely refines the position of the detected objects. This method successfully combines the strengths of deep neural networks with a modular detection approach, leveraging region proposals to reduce the overall complexity of the task.

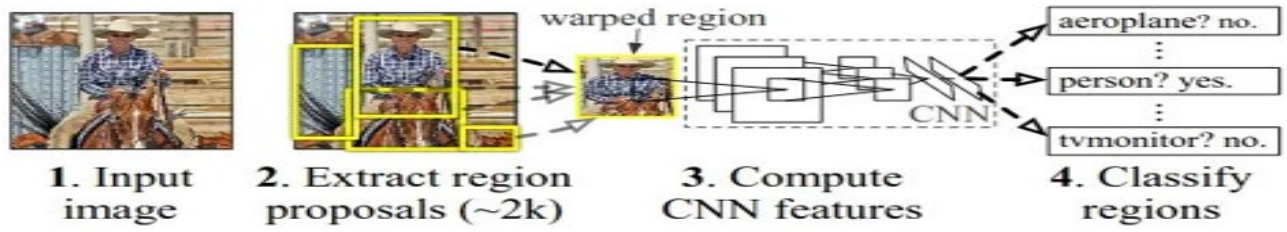


Fig. 1 Architecture of the R-CNN model

2.2 Fast R-CNN

Ross Girshick proposed a new learning algorithm that addresses the limitations of R-CNN and SPPNet while improving both speed and accuracy. This method is called Fast R-CNN because it is comparatively faster to train and test. The network takes as input a full image and a set of object proposals. First, the entire image is processed through multiple convolutional and max-pooling layers to generate a feature map [20]. For each object proposal, a Region of Interest (RoI) pooling layer then extracts a fixed-length feature vector from this map. Each feature vector is fed into a sequence of fully connected (FC) layers, which eventually split into two sibling output layers [21]:

- One layer produces softmax probability estimates over K object classes plus a “background” class.

Received: January 18, 2026. Published: June 30, 2026

• The other layer outputs four real-valued numbers for each of the K object classes. Each set of four values encodes the refined bounding box coordinates for one of the K classes [22].

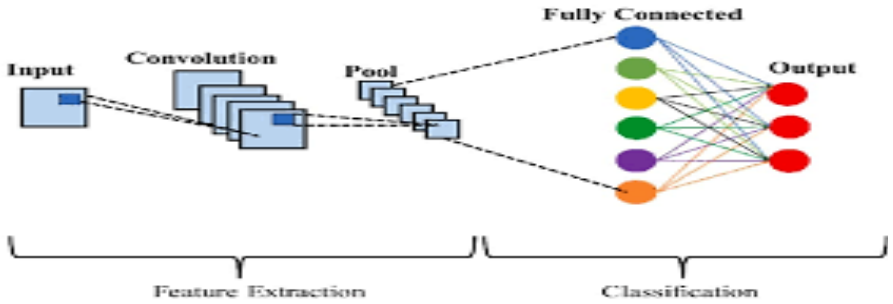


Fig. 2 Architecture of the Fast R-CNN model

2.3 Faster R-CNN

Faster R-CNN is a cutting-edge architecture for real-time object detection. This technique integrates a Region Proposal Network (RPN) that shares convolutional features with the detection network, enabling the generation of region proposals quickly and efficiently [23]. The RPN is trained end-to end to simultaneously predict object boundaries and objectness scores at each location [24]. By combining the RPN with the Fast R-CNN detector, the overall system achieves outstanding performance on benchmark datasets such as PASCAL VOC and MS COCO, while maintaining high processing speed [25].

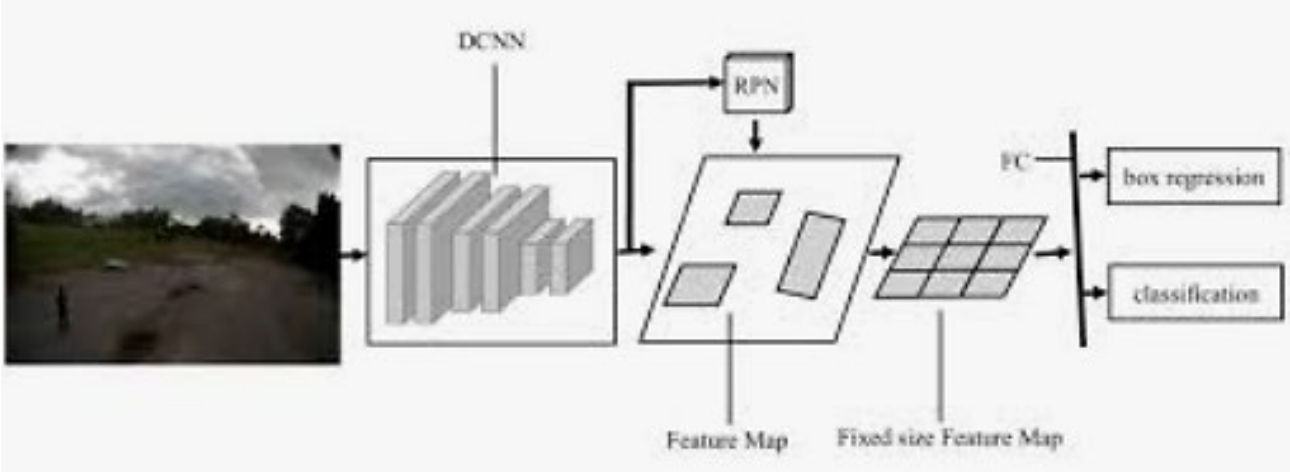


Fig. 3 The architecture of Faster R-CNN deep learning

2..4 Method of You Only Look Once

You Only Look Once (YOLO) is one of the most popular object detection algorithms used by researchers worldwide. It was first introduced in 2015 by Redmon et al. The network uses features from the entire image to predict bounding boxes and class probabilities simultaneously for all objects in the image. This global reasoning enables YOLO to process images in real time with high average precision. Its architecture supports end-to-end training, making it efficient and fast for real-time applications [22].

Received: January 18, 2026. Published: June 30, 2026

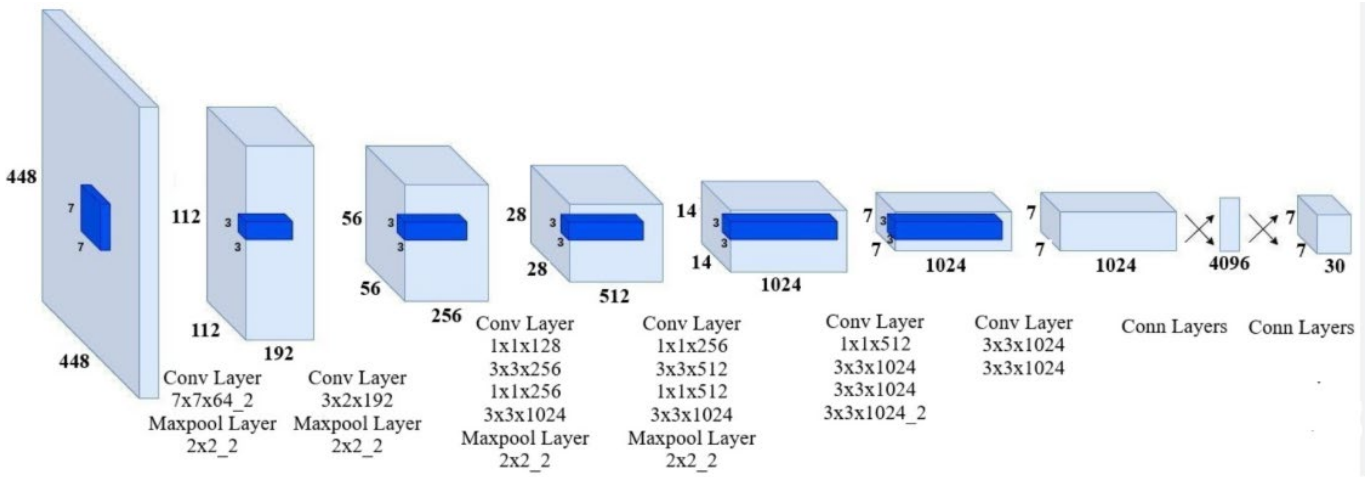


Fig. 4 Architecture with YOLO model

3. RESULTS AND SIMULATION

In this article, we will explore the various stages involved in the creation and execution of a system designed to identify different objects (such as cars, people, and motorcycles, etc.) in a series of images using an appropriate programming language (in our case, Python). We will also discuss the hardware and libraries used, the different tests conducted, and the results obtained. The image original, shown in Fig. 5, is a clear image with dimensions of 300 x 168 pixels. The file is of JPG format (.jpg) and has an actual size of 11.4 KB (11,706 bytes), while the space it occupies on the disk is 12.0 KB (12,288 bytes). This image serves as a clear visual reference, used as a comparative baseline to assess the effects of noise in the subsequent processing steps.

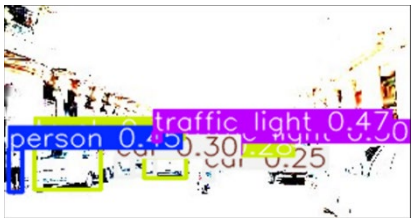
Fig. 5 Image original

The noisy image, depicts a degradation of the original image caused by the addition of Gaussian noise, which reduces its visual quality. This image maintains the same dimensions as the clear image, namely 300 x 168 pixels, and is also saved in JPG format (.jpg). Its actual size is 9.15 KB (9,370 bytes), while the space it occupies on the disk is 12.00 KB (12,288 bytes). The reduction in size compared to the noise-free image reflects the loss of information caused by the degradation introduced by the noise. The image with "salt" noise, shown in Fig. 6, results from the addition of white specks that degrade the sharpness of the original image. It retains a resolution of 300 x 168 pixels and is saved in JPG (.jpg) format. Its actual file size is 14.6 KB (15,011 bytes), and it occupies 16.0 KB (16,384 bytes) on disk. This increase in size compared to the noise-free image can be explained by the presence of highly contrasted pixels, which make JPEG compression more complex.

3.1 Algorithm Yolov5

Headings, YOLOv5 (You Only Look Once version 5) is a deep neural network model designed for real-time object detection in images or video sequences. Developed by Ultralytics using PyTorch, it stands out for its accuracy and ease of use in Fig. 6.

Received: January 18, 2026. Published: June 30, 2026

		
Fig. 6 (a) Clear image using the YOLOv5 method	Fig. 6 (b) Noisy image using the YOLOv5 method	Fig. 6 (c) Bright noisy image processed using the YOLOv5 method

3.2 Faster R-CNN

Faster R-CNN is an advanced object detection model that improves upon its predecessors (such as Fast R-CNN) by integrating a RPN directly into the architecture, making detection both faster and more accurate presented in Fig. 7.

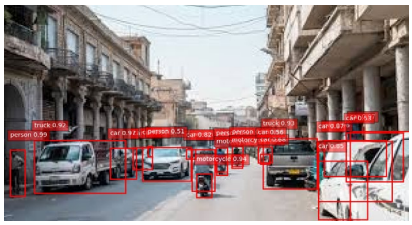
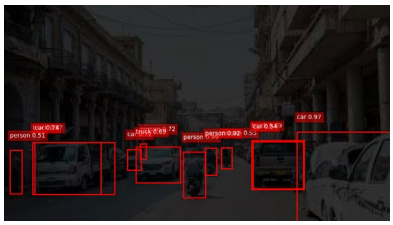
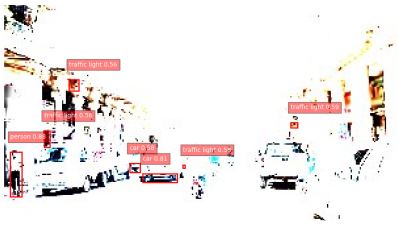
		

Fig. 7 (a) Clear image using the Faster method	Fig. 7 (b) Noisy image using the Faster method	Fig. 7 (c) Bright noisy image processed using the Faster method

3.3 SSD (Single Shot MultiBox Detector)

Single Shot MultiBox Detector (SSD) is a straightforward object detection model capable of accurately identifying elements within an image. In a single step, it locates the boundaries of each depicted object and determines their category. Unlike architectures such as R-CNN that require multiple processing stages, SSD achieves this detection performance in one pass thanks to the multi-level analysis of its neural network. Its bounding boxes and class predictions are directly generated from feature extraction performed at various scales, allowing it to accurately detect both the smallest details and the largest components of the visualized scene presented in Fig. 8.

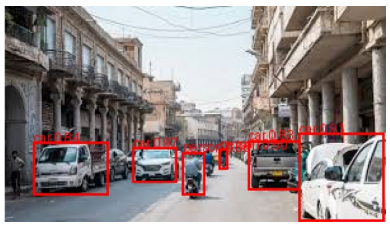
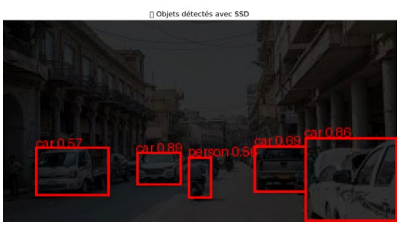
	

Fig. 8 (a) Clear image using the SSD method	Fig. 8 (b) Noisy image using the SSD method	Fig. 8 (c) Bright noisy image processed using the SSD method

3.4 U-Net++

U-Net++ is a CNN architecture based on the encoder-decoder model, featuring dense and nested skip connections between its various layers. It aims to improve segmentation performance by reducing the

Received: January 18, 2026. Published: June 30, 2026

semantic gap between features extracted at different depths of the network in Fig. 9.

Fig. 9 (a) Clear image using the U-Net++ method

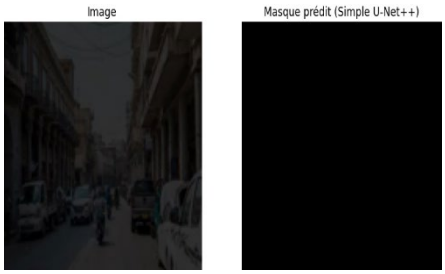


Fig. 9 (b) Noisy image using the U-Net++ method

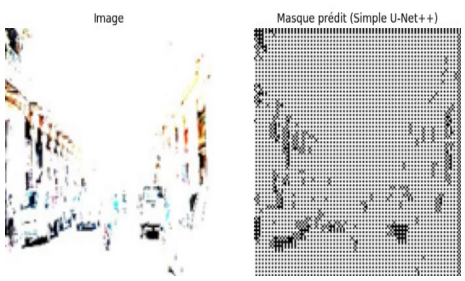


Fig. 9 (c) Bright noisy image processed using the U-Net++method

In the evaluation of classification and object detection models, several key metrics are used to assess performance. Accuracy measures the overall correctness of predictions by comparing the number of correct predictions to the total number of cases. However, in imbalanced datasets, accuracy alone may be misleading. Precision quantifies the proportion of true positive predictions among all predicted positives, indicating the model's ability to avoid false positives. Recall, on the other hand, measures the proportion of true positives detected among all actual positives, reflecting the model's ability to capture relevant instances. The F1-score provides a balanced harmonic mean of precision and recall, offering a single metric that accounts for both false positives and false negatives. Together, these criteria offer a comprehensive framework for evaluating the effectiveness and reliability of machine learning models.

Table I Classification Report for Different Methods on a Clear Image

Criteria / Methods	Precision	Recall	F1-score
YOLO	0.14	0.33	0.20
Faster R-CNN	0.25	0.50	0.33
SSD	0.50	0.50	0.50
U-net++	0.23	0.21	0.22

Table II Classification Report for Different Methods on a Noisy (Dark) Image

Criteria / Methods	Precision	Recall	F1-score
YOLO	0.14	0.33	0.20
Faster R-CNN	0.33	0,50	0,40
SSD	0.50	0.50	0.50
U-net++	0.18	1.00	0.31

Table III Classification Report for Different Methods on Image with Bright Noise

Criteria / Methods	Precision	Recall	F1-score
YOLO	0.14	0.33	0.20
Faster R-CNN	0.33	0.50	0.40
SSD	0	0	0
U-net++	0.19	0.75	0.31

In the case of images with "bright" noise, the performance of the methods varies significantly. Faster R-CNN achieves the best overall results (F1-score of 0.40), demonstrating good robustness to noise. U-Net++ detects the majority of objects (high recall of 0.75) but produces many false positives. YOLO shows low precision (0.14), while SSD fails completely. These results highlight the importance of selecting a method suited to noisy

conditions or incorporating a denoising preprocessing step. These methods are evaluated based on their accuracy, speed, robustness to noise, and suitability for various detection tasks. YOLO demonstrates high processing speed and real-time capabilities, making it suitable for time-sensitive applications, but may show reduced accuracy in detecting small or overlapping objects. Faster R-CNN offers superior detection accuracy and robustness across different conditions but is computationally more expensive. SSD balances speed and accuracy well, though it can be sensitive to certain types of noise. U-Net++, primarily designed for image segmentation, excels in tasks requiring precise boundary detection, particularly in medical or pixel-level applications. Experimental results highlight the trade-offs between precision, recall, F1-score, and inference time, providing valuable insights for selecting the most appropriate model depending on the application requirements.

Table IV Comparison of the Results of the Methods

Methods	Robustness to Noise	Strengths	Weaknesses
Faster R-CNN	Good robustness across all noise types	Consistent performance - Best F1-score under bright noise	Moderate precision - Slower inference compared to YOLO and SSD
SSD	Excellent on clean and dark noisy images; fails under bright noise		- Highly sensitive to bright noise - No detections under this condition
U-Net++	Variable: high recall but low precision	- Extremely sensitive (recall up to 1.00)	- Generates many false positives - Overall low precision
YOLOv5	Low robustness under all conditions	- Very fast inference time	- Lowest F1-score - Poor performance with and without noise

4. Conclusion

This article has highlighted the growing importance of HOI detection in the field of computer vision, emphasizing the substantial progress enabled by deep learning techniques. Through a comparative analysis of neural network architectures, we have demonstrated their ability to significantly enhance the accuracy, robustness, and scalability of detection systems. These architectures are particularly effective in addressing complex challenges such as occlusions, simultaneous multiple interactions, contextual variations, and unstructured environments. The experimental results confirm the potential of the studied approaches for a wide range of applications, including smart security, collaborative robotics, assistive systems, and human-machine interfaces based on gesture and behavior understanding. Several research directions emerge from this work: Improvement of contextual robustness: Integrating contextual attention modules or spatiotemporal modeling could enhance the models' ability to interpret complex scenes in dynamic environments. Reduction of annotation dependency: Exploring weakly supervised or self-supervised learning methods may allow leveraging large volumes of unlabeled data while maintaining competitive performance. Real-time optimization: Adapting architectures for deployment on embedded or resource-constrained devices (edge computing) is crucial for industrial and mobile applications. Finer semantic understanding: Expanding models to recognize not only interactions but also their intent or purpose could pave the way for systems capable of anticipating human actions in a predictive framework.

References

[1] Y.LeCun, Bengio, Y. and Hinton, G. "Deep learning, (2015), Nature, vol. 521, no. 7553, pp. 436–444, doi: 10.1038/nature14539.

[2] Y. Chao, Liu,Y. Liu, X. Zeng, H. and Deng, J. (2018), Learning to detect human-object interactions, in Proc. IEEE Winter Conf. on Applications of Computer Vision (WACV), pp. 381–389.

[3] D. Papadopoulos, P. Uijlings, R. Keller, F. and Ferrari, V. (2017), Training object class detectors with click supervision, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 5794–5803.

[4] B. Wan, Zhou, T. Liu, Y. Ye, Q. and Jiao, J. (2019), Pose-aware multi-level feature network for human-object interaction detection, in Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV), pp. 9469–9478.

[5] , Z. Cao, Hidalgo, G. Simon, T. Wei, S. E. and Sheikh, Y. (2021), OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields," IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 172–

186, Jan.
[6] G. Gkioxari, Girshick, Dollár, R. P. and He ,K. (2018), Detecting and recognizing human-object interactions, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 8359–8367.
[7] Y. Li, Zhou, S. Huang, Z. and Zhang, Y. (2021), HOI Transformer: Human-object interaction detection with transformers, arXiv preprint arXiv:2104.13682.
[8] Y.-W. Chao,.Y. Liu, X. Liu, H. Zeng, and J. Deng, (2018), Learning to detect human-object interactions, in 2018 ieee winter conference on applications of computer vision (wacv). IEEE, pp. 381–389.
[9] S. Qi, Wang, W. . Jia, Shen, BJ. and Zhu, S.-C. (2018), Learning human-object interactions by graph parsing neural networks, in Proceedings of the European conference on computer vision (ECCV), pp. 401–417.
[10] B. Wan, Zhou, D. Liu,Y. Li, R. and He, X. (2019), Pose-aware multi-level feature network for human object interaction detection, in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9469–9478.
[11] T. Gupta, Schwing, A. and Hoiem, D. (2019), No-frills human-object interac tion detection: Factorization, layout encodings, and training techniques," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9677–9685.
[12] O. Ulutan, Iftekhar, A. and Manjunath, S. B. (2020), Vsgnet: Spatial attention network for detecting human object interactions using graph convolu tions, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognitionpp. 13617–13626.
[13] Y. Liu, Chen, Q. and Zisserman, A. (2020), "Amplifying key cues for human object-interaction detection," in European Conference on Computer Vision. Springer, pp. 248–265.
[14] D.-J. Kim,. Sun, X. Choi, Lin, J. S. and Kweon, I. S (2020), Detecting human object interactions with action co-occurrence priors, in Proceedings of the European Conference on Computer Vision (ECCV). Springer, pp. 718–736.
[15] X. Zhong, Ding, C. Qu, X. and Tao, D. (2020), Polysemy deciphering network for human-object interaction detection," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, Proceedings, Part XX 16. pp. 69–85.
[16] T. He, L. Gao, J. Song, and Y.-F. Li, (2021), Exploiting scene graphs for human-object interaction detection, in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15984–15993.
[17] P. Zhou, , Hou, Q., and Cheng, M. (2022), Label decoupling framework for human-object interaction detection," IEEE Trans. Image Process., vol. 31, pp. 5389–5402,.
[18] H. Xu, Zhu, Y. Choy, C. B. and Fei-Fei, L. (2019), Learning to detect and classify human-object interactions," Int. J. Comput. Vis., vol. 127, no. 3, pp. 258–275.
[19] Q. Hou, P. Zhou., , & Cheng, M. M. (2021). HOI Analysis: Understanding human-object interaction by modeling object semantics. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 3027–3036). https://doi.org/10.1109/ICCV48922.2021.00302
[20] R. Jain, Gandhi, H. and Bhattacharya, P. (2022), Lightweight human-object interaction detection for edge computing," IEEE Access, vol. 10, pp. 92846–92859,.
[21] J. Redmon, and Farhadi, A. (2018), YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767,
[22] Z. Zhao, Zheng, Q. P. S. Xu, T., and Wu, X. (2019), Object detection with deep learning: A review," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–323.
[23] Girshick, S. (2015), Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440–1448, doi: 10.1109/ICCV. 169.
[24] S. Ren, He, K. Girshick, R. and Sun, J. (2015), Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Advances in Neural Information Processing Systems (NeurIPS 2015).
[25] J. Redmon, Divvala, S. Girshick, R. and Farhadi, A. (2016)"You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779–788, doi: 10.1109/CVPR.2016.91.

.